TESTIMONIALS

"Karl Wiegers has written a delightful and practical book that challenges how we think about designing products. Karl's past experiences and publications identify him as a true expert in how to define software efficiently and effectively. His eye for details and thinking about how users will use products naturally extends to physical products too. Karl is absolutely an expert in how to write good software requirements; it only makes sense that he would equally be able to teach us about good design."

"As with Karl's other books, *The Thoughtless Design of Everyday Things* is lighthearted, entertaining, and thought provoking. He offers many tips regarding how to do thoughtful design and design mistakes to avoid, with loads of relatable real-world examples. You'll learn something about future product design and probably be more annoyed by those who haven't."

—Joy Beatty, Vice President at Seilevel

"Karl Wiegers's gift is his ability to look at an important topic and write a comprehensive study that gives practical advice, illustrated by real-world examples. And he does it in such a compelling manner! The densely packed wisdom, mined from Karl's years of design experiences, is interesting and insightful. I found it hard to put this book down; I just wanted to turn the page and see what came next. Anyone who designs items for real-world use—software, machines, medical instruments, homes, appliances, and the like—will find this book immensely valuable."

"Now, before I say my design work is done, I will review my design decisions against Karl's 70 design lessons. Most likely, I will find something I want to rethink, to make my work just that little bit better."

—Norman L. Kerth, software engineer
and author of *Project Retrospectives*

"Have you ever been irritated by a product that's way harder to use than you expected? Or outraged when something you've bought turns out to be useless for your needs? Why are such travesties so common? Karl Wiegers, well known in the software world for his excellent work on system requirements, explains. In *The Thoughtless Design of Everyday Things*, he calls out many of the egregious design errors that we've all encountered, drawing on examples from the kitchen, the car, the computer, and many other sources. Then he takes it further, using these "thoughtless design" examples to illuminate a specific design principle that the product designer should have followed."

"I enjoyed this book on several levels. Like all of Karl's writing, it is clearly written and easy to read. It's an entertaining diatribe about products that have made our lives hell at some point in time. I found myself saying, "Yeah, that exact thing happened to me." But, at a deeper level, I learned *why* designers tend to make such crazy errors. As a customer, I learned how to clarify my exact needs when choosing among rival products. And as a designer, I learned some valuable rules and processes for *thoughtful* design, including understanding the requirements for—and the contexts of—usage."

—Meilir Page-Jones, author of *The Practical Guide to Structured Systems Design* and several other software-design books

"This might be the most depressing book you'll read if, like me, you appreciate it when the products you use are well designed. On the other hand, it just might be the most entertaining read when you see the examples of calamitously bad products. Karl has put together here the most alarming, yet amusing chamber of horrors of poorly, thoughtlessly designed products. But it is not all on the dark side. Karl also brings you wonderful insights on design, design principles, how good design comes about, and how it makes so much difference to everybody's equilibrium."

—James Robertson, requirements guru and author of *Mastering the Requirements Process*

THE THOUGHTLESS DESIGN OF EVERYDAY THINGS

Karl Wiegers, PhD

J.ROSS PUBLISHING

ISBN-13: 978-1-60427-178-2

Printed and bound in the U.S.A. Printed on acid-free paper.

10 9 8 7 6 5 4 3 2 1

Library of Congress Cataloging-in-Publication Data

Names: Wiegers, Karl Eugene, 1953- author.
Title: The thoughtless design of everyday things / Karl Wiegers, PhD.
Description: Plantation, FL : J. Ross Publishing, [2021] | Includes bibliographical references and index.
Identifiers: LCCN 2020052550 (print) | LCCN 2020052551 (ebook) | ISBN 9781604271782 (acid-free paper) | ISBN 9781604278286 (ebook)
Subjects: LCSH: Product design. | System failures (Engineering)
Classification: LCC TS170 .W54 2021 (print) | LCC TS170 (ebook) | DDC 658.5/752--dc23
LC record available at https://lccn.loc.gov/2020052550
LC ebook record available at https://lccn.loc.gov/2020052551

Direct all inquiries to J. Ross Publishing, Inc., 151 N. Nob Hill Rd., Suite 476, Plantation, FL 33324.

Phone: (954) 727-9333
Fax: (561) 892-0700
Web: www.jrosspub.com

For Chris, whose design is excellent

CONTENTS

ACKNOWLEDGMENTS

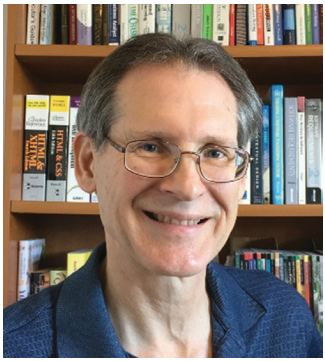
As with designs themselves, critical input from manuscript reviewers is, well, critical. I'm especially grateful for detailed input from Tony Higgins, Joyce Statz, and particularly James Shields. I also appreciate the helpful comments from Bob Beatty, Rodney Bell, Randy King, Eileen Lindsay, Adrian Marchis, Perry McLeod, Jeff Merrifield, Mark Mitchell, John Nelson, Don Norman, Terrie Novak, Kathy Reynolds, Linda Rising, Ben Ryan, Rik Smoody, and Chris Zambito. Discussions with Joy Beatty, Scott Meyers, and Meilir Page-Jones were particularly illuminating.

I'm grateful to the many people who shared with me their own examples of both thoughtlessly and thoughtfully designed products, software systems, and websites, including: Bob Beatty, Joy Beatty, Rodney Bell, Nancy Buset, Sal Cambareri, Will Compton, Christin Davis, Joan Davis, Bob Glass, Doris Grolbert, Payson Hall, Paul Hanes, Tony Higgins, Norm Kerth, Randy King, Terry Lewis, Terry Lund, Scott Meyers, Tony Miceli, Anne Millbrooke, Mark Mitchell, Geraldine Mongold, John Nelson, Terrie Novak, Meilir Page-Jones, Howard Podeswa, Linda Rising, Ben Ryan, Rik Smoody, Alice Tarachow, Bill Trosky, Bruce Wiegers, and Chris Zambito. Several of these people also generously provided photographs or screenshots of the products. Seeing so many examples of problems in such a wide range of products reinforced my perception that the world still has many opportunities for more thoughtful and considerate product designs.

I'm grateful to Steve Buda, Vice President of New Business Development at J. Ross Publishing, as well as to the J. Ross production staff for their fine editing and design work.

Thanks also to Gerd Altmann (Pixabay) and Mateus Campos Felipe (Unsplash) for providing freely downloadable photos that I incorporated.

ABOUT THE AUTHOR



Since 1997, Karl Wiegers has been Principal Consultant with Process Impact, a software development consulting and training company in Happy Valley, Oregon. Previously, he spent 18 years at Kodak, where he held positions as a photographic research scientist, software developer, software manager, and software process and quality improvement leader. Karl received a PhD in organic chemistry from the University of Illinois. During the past 50 years Karl has designed chemistry experiments, software applications and user interfaces, software development processes, books, songs, games, and training courses.

Karl is the author of the books *Successful Business Analysis Consulting*, *Software Requirements*, *More About Software Requirements*, *Practical Project Initiation*, *Peer Reviews in Software*, and *Creating a Software Engineering Culture*. He has written more than 230 articles on many aspects of software development and management, consulting, chemistry, military history, and self-help. Karl is also the author of a forensic mystery novel called *The Reconstruction* and a memoir of life lessons titled *Pearls from Sand*. Several of Karl's books have won awards, most recently the Society for Technical Communication's Award of Excellence for *Software Requirements, 3rd Edition* (co-authored with Joy Beatty). Karl has served on the Editorial Board for *IEEE Software* magazine and as a contributing editor for *Software Development* magazine.

When he's not at the keyboard, Karl enjoys volunteering at the public library, delivering Meals on Wheels, playing guitar, writing and recording songs, traveling, and wine tasting. You can reach him through www.processimpact.com or www.karlwiegers.com.

This book has free material available for download from the Web Added Value™ resource center at *www.jrosspub.com*

At J. Ross Publishing we are committed to providing today's professional with practical, hands-on tools that enhance the learning experience and give readers an opportunity to apply what they have learned. That is why we offer free ancillary materials available for download on this book and all participating Web Added Value™ publications. These online resources may include interactive versions of the material that appears in the book or supplemental templates, worksheets, models, plans, case studies, proposals, spreadsheets and assessment tools, among other things. Whenever you see the WAV™ symbol in any of our publications, it means bonus materials accompany the book and are available from the Web Added Value Download Resource Center at www.jrosspub.com.

Downloads for *The Thoughtless Design of Everyday Things* include a quick-reference summary of the nine design principles and 70 design lessons, as well as numerous additional examples of both thoughtless and thoughtful design that illustrate and reinforce the design principles.

CHAPTER 1

THOUGHTLESS DESIGN

In the early 1960s, my father was an officer in the United States Air Force, stationed in southern Italy. His job there was launching nuclear-armed Jupiter missiles at Russia. Fortunately, he didn't have much work.

Each missile site had both an American and an Italian launch control officer. The launch control panel was designed such that two well-separated people must insert their keys into locks and turn them simultaneously to launch a missile. This separation was to prevent a lone maniac from initiating Armageddon. It seems like an effective design.

One day, my father's Italian counterpart opened the launch control panel from the back and showed Dad how it was possible to circumvent the need for two keys to initiate the launch sequence. He didn't offer a full-cycle demonstration, fortunately. All it required was a short piece of wire with alligator clips on both ends to bypass one of the most critical safety mechanisms in the world at that time. Maybe that launch control panel design wasn't quite ready for prime time.

The launch control panel reflects an extreme design shortcoming—the inability to thwart a malicious actor with potentially devastating consequences—but we are all surrounded by countless products with design problems that annoy us and waste our time. Some of these design issues are minor irritations; others can render products useless or even harmful in certain situations. I've often encountered a product that appears to have been designed by someone who never used a product of that kind before. Otherwise, they surely would have made some different design choices.

THOUGHTFUL AND THOUGHTLESS DESIGN

Don Norman published his classic book *The Design of Everyday Things* in 1988, with a revised edition in 2013 (Norman, 2013). Norman took a

psychological approach to understanding design thinking and how people interact with products. *The Design of Everyday Things* explored why people make mistakes when using products and recommended ways to design better products, prevent errors, and help users get the most out of the product.

It appears that not all of today's designers have taken Norman's wisdom to heart. Too many products still exhibit silly design failings. And since *The Design of Everyday Things* first appeared, we now have a vast assortment of new design issues with software applications, websites, and mobile devices.

Designers do put thought into creating their products, of course, even if the results sometimes fall short of our expectations. A thoughtfully designed product is a joy to use. It lets you accomplish a task efficiently, without aggravation, and without having to think too hard about it. As Norman (2013) points out, "Good design is actually a lot harder to notice than poor design, in part because good designs fit our needs so well that the design is invisible." In my own experience, I do find it far easier to think of thoughtless design examples than to remember those products that are designed really well.

When I say *thoughtless design*, I'm referring to products that could have been designed better without a great deal of additional effort. These products—both physical items and software applications—range from being unusable for their intended purpose to being acceptable but with need for improvement. Thoughtless design can lead to products that:

- Are not easy or obvious to use;
- Do not work well in realistic usage scenarios;
- Do not function properly in certain environments;
- Make it too easy for users to make a mistake;
- Do not provide the user with meaningful feedback;
- Waste the user's time;
- Are not designed for the user's convenience;
- Fail to accommodate the range of human variation; or
- Place excessive mental burdens on users.

Chapters 3 through 11 address each of these nine categories of design deficiencies in turn, illustrating the problems with many real examples. The examples are all drawn from my personal experiences or those that others have shared with me. I'll describe why these product designs are less than optimal, how many of them could be improved, and what designers can learn from them.

You'll also find numerous examples of especially well-designed products in this book. You can probably think of others from your own experience. They stand out because they're so much better than too many of the products we routinely encounter.

But shouldn't good design be the norm? Aren't consumers entitled to products that are obvious and easy to use, respect our time, work properly, and don't aggravate us? Yes, we are. I would feel differently if it took considerably more effort to create superior designs or if the resulting products would cost a lot more, but often that's not the case.

Here's a simple example. Everyone wears clothes, so it's surprising to encounter an item of apparel that makes me wonder, "What were they thinking?" I have a nice medium-weight blue jacket. It's well made and comfortable, with a cozy zip-out lining, but that lining has a strange inside pocket. The pocket is five inches wide, a good size to hold a phone, a museum brochure, or sunglasses. For some reason, though, the manufacturer sewed a vertical seam down the middle of that pocket, which splits it into two pockets that are each just under 2.5 inches wide. You can't put much into a pocket that narrow, maybe a nutrition bar or a couple of pens.

This pocket subdivision epitomizes thoughtless design. It cripples a useful feature while adding no value. If the designer had asked for input from typical jacket-wearers, I can't imagine anyone would have said, "That five-inch pocket is way too big. How about if you split it in two?" It's hard to envision the thought process that led to that design choice.

You might not agree with all of the examples I consider to be thoughtlessly designed. Some properties I find puzzling or annoying might suit your preferences just fine. There are lots of consumers, with different characteristics, needs, expectations, and tolerances. Their needs often diverge or conflict, such that no single solution will delight everyone.

It's worth considering why a designer made the choices that resulted in a product that doesn't seem quite right to you or to me. Perhaps the nature of the product constrains the design such that there's only one possible solution, which might not be ideal. The designer could be working under constraints we consumers aren't aware of: regulatory restrictions, cost or competitive pressures, compatibility issues, and the like. Complex products are developed by multiple teams that may not communicate, collaborate, and compromise effectively, leading to parts of a product that are great and others that fall short. In too many cases, though, the product's design is simply flawed. Chapter 2 explores why

design is difficult; the rest of the book looks at designs that could be better and how to improve them.

ABOUT THIS BOOK

This book is for people who design or use anything: software systems, websites, automobiles, consumer products, bathrooms, appliances, user manuals—anything. Designers can apply the design principles and lessons I present to make a wide variety of products better suit customer needs. Consumers can learn to think more clearly about what they're really looking for before they buy a product. Managers will learn about the inescapable trade-off decisions they must make to reach the optimal balance of product attributes that will satisfy a diverse group of stakeholders.

There are many cases of large-scale design failures that did enormous damage, cost huge sums of money, and sometimes killed people (Shariat and Saucier, 2017). Television series like "Deadly Engineering" and "Engineering Disasters" tell the tragic tales of collapsing bridges and buildings, rupturing dams, and nuclear reactor accidents. I've watched episodes about new skyscrapers shedding windowpanes or tiles that plummeted to the streets below, endangering passersby. Airplanes have crashed because of hardware or software design flaws. Rockets and spacecraft have exploded on the launch pad or in flight. Huge ships have gone to the bottom on their maiden voyages. Engineers study such failures to learn how to avoid them in the future.

I don't discuss such large-scale catastrophes here. Nor do I focus on irritations with products or processes that don't specifically relate to design problems. Yes, there are way too many of those: poorly made products, buggy software with features that don't work right, processes that make no sense. A friend told me about a relative of his who could pass the vision test for renewing his driver's license only by wearing his reading glasses, even though he can't safely drive with those glasses. I find such things as ridiculous as you do, but that's not what this book is about.

My focus here is on perplexing design shortcomings in the consumer products we all encounter. Many products contain small defects or lack some functions you might like. However, I'm most concerned with silly issues that suggest the designer didn't consider real users, their usage scenarios, their environments, and their expectations.

I'll present dozens of examples of what I consider thoughtless design of both physical devices and software products. These examples violate the nine guiding design principles explored in Chapters 3 through 11. As we can gain powerful insights from problems, I call out 70 specific design lessons from the examples I describe. These lessons are collected in the Appendix for easy reference.

Many designers are well schooled in design principles and create useful, clever, and even brilliantly conceived products. I have also included examples of particularly thoughtful designs to show how well it can be done. Less experienced and less educated designers must accumulate expertise through trial and error, self-education, and learning from proficient mentors. Therefore, the final chapter describes numerous specific practices that can help any designer create pleasing products.

In most cases, I don't identify the manufacturer of the product I'm using to illustrate what I consider thoughtless design. I do identify the producer for some examples where it's particularly relevant or unavoidable, such as specific apps or devices. My purpose here is not to criticize particular companies, but rather to draw general lessons that can help designers do a better job on their next product. Software systems, websites, smartphones, and other products are modified over time, so some of my examples might look or function differently in the future. All of the examples presented are accurate as of the date of writing, though.

A THOUGHTLESS DESIGN CASE STUDY: THE SPONGE MOP

My wife once bought a new sponge mop. It seemed to have the necessary mop features and properties, including a handle for squeezing out excess water (Figure 1.1). We've found that the squeeze handle was useful on other mops we've owned.

However, we quickly discovered a design shortcoming in this squeeze handle. When you place the mop in the mopping position with the sponge on the floor, the squeeze handle also drops to the floor (Figure 1.2). There ought to be some way to clip the squeeze handle onto the mop handle when you aren't using it, as I've seen with other mops. We just assumed any reasonable designer would incorporate a clipping mechanism, because it's a nuisance to have the squeeze handle drag on the floor. I wondered if the designer ever used this mop before sending the specifications off to manufacturing.

Figure 1.1 This is an apparently reasonable sponge mop.

On closer examination, I discovered that the mop is indeed designed to clip the squeeze handle in place, but the clip doesn't work right. The latching mechanism is too weak to hold the squeeze handle—it pops right out again. The design is still flawed, but it's the latching mechanism itself that is poorly designed. That's a different problem than neglecting to include a latching mechanism at all. Someone who tested a prototype of this mop should have noticed immediately that the latching mechanism's current design was inadequate.

Beyond this specific design failing, there are three messages here that are worth exploring further:

1. The difference between quality and crap is often small.
2. Consumers don't always think carefully about their requirements before buying a product.
3. Product problems are not always design failures.

Figure 1.2 In reality, experience showed it to be a thoughtlessly designed sponge mop.

THE CRAP GAP

Hold your hand up with your thumb and index finger about one inch apart, like in Figure 1.3. In many situations, that short distance represents the difference between quality and crap (Wiegers, 2019c). Often, all it takes to close the crap gap is to do some more questioning, listening, thinking, measuring, designing, prototyping, or testing before you deliver a product or declare a job complete.

Bridging the crap gap takes little effort in many cases. It's a matter of measuring twice, checking one more time, and then making the cut. It's thinking about how people will use a product, not just its features, and using checklists to avoid overlooking something important. It's asking one more person for input, confirming a decision before proceeding, and validating your assumptions. It's testing a product under realistic conditions instead of merely assuming the design will work right. It's not cutting corners to make a bit more profit on an item that will quickly disappoint consumers and drive them to buy a replacement.

Many of the examples of thoughtless design that we'll encounter will exemplify the crap gap. You'll see that it wouldn't have been much harder

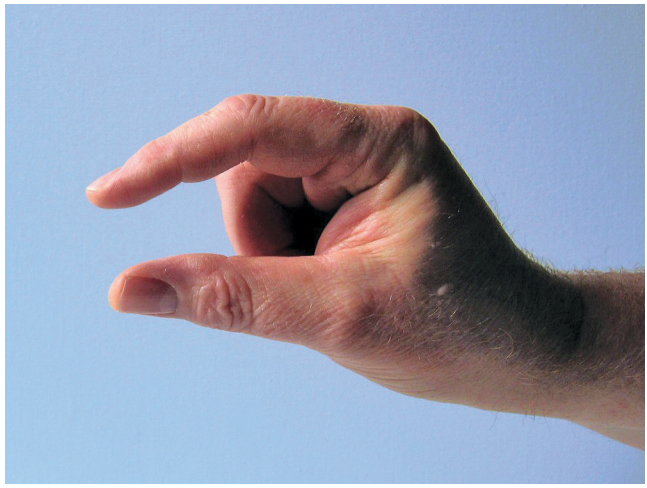


Figure 1.3 Beware the crap gap!

to design the product so as to avoid the problems that I describe. And you might wonder why a designer or manufacturer didn't get it right in the first place.

CONSUMERS AND THEIR REQUIREMENTS

Buying products with design deficiencies is partly our own fault. We consumers often do not think carefully about exactly what our requirements are before we go shopping. I need a sponge mop, so I go to a store like Target and look through their mop offerings. Ah, I like the kind with the squeeze handle, so I'll buy this one. I use the mop for the first time, perhaps a couple of weeks later. Only then do I discover the problem. "I can't believe they're selling a mop with a squeeze handle that drags on the floor!" I say, exasperated. By then, I've probably lost the receipt and can't return the annoying mop.

Designers, engineers, and developers base their work on specified product requirements. I've spent much of my career as a software engineer and consultant exploring better ways to define the requirements for software systems. The terms *assumed requirements* and *implied requirements* make my blood run cold. You cannot expect someone else to read your mind to learn just what capabilities and characteristics you expect in the product. As systems analyst Meilir Page-Jones says,

"Requirements are *always* made explicit eventually, preferably before the product is built, but too often afterwards. The difference in cost and frustration is enormous."

But we do assume certain requirements all the time with everyday products. We have a vague notion of what we're looking for, we acquire an item that seems reasonable, and then sometimes we're disappointed. Rarely do we carefully specify our requirements in advance and then assess each candidate product against them. We expect products both to be designed well and to work correctly when used as intended.

Perhaps you need a new kitchen spatula, so you simply buy one. Maybe you differentiate between a metal spatula and a plastic one that won't scratch your nonstick pans. You expect it to be pretty much like all the other spatulas you've ever used and that it will work just fine.

However, you might discover a problem the first time you make pancakes. Figure 1.4 shows a pair of spatulas that I own. I love the one in the foreground. It's comfortable and effective. The one in the background, however, is awkward to use. Since I'm not seven feet tall, that sharp angle between the handle and the spatula head forces my arm into an uncomfortable position when I slip the spatula beneath something in a frying pan. It makes me wonder whether anyone ever tried to flip a pancake with the second spatula before manufacturing thousands of them.

Figure 1.4 I prefer the comfortable and effective spatula (near) over the awkward spatula (far).

Perhaps this spatula wasn't intended for pancakes at all. Maybe it was designed to work well with certain kinds of steep-sided pans that I don't own, but it's still awkward to use. I could have thought more carefully about exactly what I needed before I headed to World of Spatulas.

As consumers, it's not our fault when basic items don't do a reasonable job, but we do have some responsibility to think through our requirements before pulling out the credit card. I've tried to train myself to be a better-prepared consumer so I know which product characteristics to look for and which to avoid. For large and infrequent purchases like a car or a house, I use detailed checklists and spreadsheets to help me assess the options and choose the right one. And as a consumer who cares about quality and products that last, I'm willing to pay a bit more if necessary, rather than settling for junk. However, even careful analysis won't help if a product—like the sponge mop—seems to possess the desired feature but fails in ordinary usage.

IS IT REALLY A DESIGN PROBLEM?

What might at first appear to be thoughtless design could have other explanations. There's a difference between conception and execution. Sometimes it's hard to tell if a product is poorly designed, is well-designed but poorly manufactured, is made with low-quality materials, or is simply defective. A friend of mine has had some problems with his dishwasher:

> *You would think both racks would glide in and out easily, a basic mechanical function you expect of a dishwasher. However, the top rack jams every time I push it in, and I have to lift or wiggle it back into place.*

This sounds like a design problem. But maybe there's some other root cause, like a poorly manufactured rack or even a missing part. Frankly, consumers don't care what the explanation is. They just want a new product to work right.

My Honda has a speech recognition system that lets the driver control navigation functions, climate control settings, and the audio system by voice. However, it responds to spoken commands s-l-o-w-l-y. Using the dashboard controls is much quicker than wading through the voice menus. The advantage is safety; I can make changes by voice with less distraction than if I have to take my eyes off the road to locate the correct

dashboard button. Unfortunately, this advantage is negated if I must glance at the display screen to remember the keywords to use for making adjustments. Also, the recognition accuracy isn't very good. Once when I told the speech recognition system I wanted to exit from the function, it turned on the air conditioner. Maybe "exit" sounded like "A/C" or something.

The mediocre speech recognition accuracy renders the system practically useless for navigation. It does get some things right, and it's a real convenience when it works. However, when I ask the system to "Find Place," it finds the location I'm seeking only about 20 percent of the time. It's vastly easier, faster, and more flexible to use the excellent speech recognition on my iPhone and display the map on the car's navigation screen via Apple CarPlay.

Now, is this speech recognition failure a design problem, a component problem, an implementation problem, or user error? Perhaps the microphone isn't sensitive enough or is too sensitive for accurate recognition, or maybe it's positioned poorly in the car. Maybe the software needs to be trained to better recognize each driver's voice, but the car provides no such option. Maybe I don't speak clearly or loudly enough, or I could be talking too loud or too fast. I don't know what the cause is. The result is that I rarely use the car's voice control system because it's so slow and error-prone, although it seemed like a cool feature when I bought the car.

Nontechnical considerations also can lead to subpar designs. An existing product's superior design might be protected by patents, constraining competing manufacturers either to license that better design or to invent an alternative approach. That's what happened when Kodak—where I worked for 18 years—entered the instant camera market in the 1970s. Because of certain features that Polaroid had patented previously, Kodak had to use completely different film technology and camera design approaches. The resulting camera was bulkier and more awkward to use than Polaroid's.

So, while there are many reasons why products can be deficient besides design flaws, thoughtless design is all around us. Designers can do better; consumers are entitled to it.

SOME TERMINOLOGY CONVENTIONS

One person might design both the software and the user interface for a small mobile app, but teams design complex products. In this book, I will

often use the term *designer* as a singular term, though I recognize that many people could have contributed to the overall product design.

The *customer* who acquires a product or specifies its requirements isn't always the ultimate, or end, user. A *user* directly interacts with the product, benefiting from the delights of thoughtful design and suffering the frustrations of thoughtless design. Some software systems also have indirect users who receive outputs from it but don't interact directly with the system themselves. A *consumer* could be an acquiring customer, a user, or both.

I should define some other terms that will recur throughout the book. A dictionary definition of *obvious* will suffice: easily seen, discovered, recognized, or understood. A product feature is obvious if its presence and usage are readily apparent without further explanation to anyone who engages with the product.

A product is *intuitive* if you can make a good guess at how to use it based on its similarity to other products from your experience or from other cultural or environmental cues. An intuitively designed product is one a user can understand and employ immediately, without consciously thinking about it or referring to a manual (Interaction Design Foundation, n.d.).

A feature or product is *easy to use* if the user can successfully perform a task with minimal effort and assistance. To make a product easy to use, a designer must understand who its users will be, their goals and expectations, their backgrounds, and the anticipated contexts of usage. Performing even a complex task with a software system could be regarded as easy by an expert who has done it many times before, whereas someone encountering the product for the first time could find it highly difficult to use.

A KEY ASSUMPTION

I make the assumption that product designers and engineers do the best job they can with the information and skills they have available at the time. Perhaps that's not always true, but I like to think it is. I'm not trying to be overly harsh on designers here, and I'm not accusing them of malice (except perhaps for a few cases described in Chapter 9). My examples are opinions about specific products that suffer from what some consumers consider to be thoughtless design. Sometimes things that are not obvious initially do turn out to be clear on a second look or to someone

else's eyes. We each interpret the world in the context of our own experiences and mental models. And we can't always put the whole onus on the designer. No matter how well a product is designed, someone somewhere will fail to make it work or will manage to break it.

A colleague of mine had previously worked in a call center providing technical support to software users. His group had an unofficial problem classification category of "user brain damage." We all make silly mistakes; we all misunderstand, overlook, and forget things. A woman I know related this experience:

> *One time a friend had me drive a hundred miles because her computer would not talk to her printer no matter what setting she selected. I got there. Using the power switch on the printer, I turned the printer on. Problem solved. We have all suffered brain blackouts.*

Many people like to say, "The customer is always right." That simply isn't true. Sometimes the customer is unreasonable, misinformed, or in a bad mood, which can lead to complaints that don't reflect design problems at all. Sometimes, though, a nugget of wisdom in the complaint can help us make the next design even more obvious, intuitive, and easy to use.

I used to carpool with a man who complained about the restricted view his new car offered through the rear window. I asked if he had noticed the visibility problem on his test drive. George sheepishly confessed that he hadn't taken the car on a test drive. If he had, he might have spotted the rear window problem before he bought it. Maybe the design really was deficient, but George should have explored the car more carefully first.

People sometimes use products incorrectly because they don't understand them well. Some hotel bathrooms have heat lamps in the ceiling light fixture. To save energy, many hotels have switched their incandescent light bulbs to more efficient LED bulbs. I've stayed in hotels that also had replaced their bathroom heat lamps with LEDs. LEDs make the bathroom brighter, but because they put out very little heat, they don't warm up the room. This confusion about the bulbs isn't a design inadequacy, but rather a matter of the hotel staff not understanding the product.

Consumers must take some responsibility for assessing their needs, evaluating candidate products, and becoming familiar with how to use them before they complain about product deficiencies. Even so, we're

still surrounded by products that disappoint and irritate us. Things should just work properly.

LOOKING AHEAD

In Chapter 2, I'll explore why it truly is hard to design products that perform in a way that makes consumers happy. I provide some background about requirements as the foundation for design, designing for usage rather than simply cramming in more product features, and the tensions that designers face as they attempt to satisfy multiple stakeholders who have conflicting expectations and constraints. If you don't need that technical background, feel free to skip ahead to Chapter 3.

Each of the subsequent nine chapters addresses a basic principle of effective design:

Chapter 3: Make the product easy and obvious to use
Chapter 4: Consider realistic usage scenarios
Chapter 5: Consider a wide range of usage environments
Chapter 6: Make it hard to make a mistake
Chapter 7: Provide meaningful feedback
Chapter 8: Don't waste the user's time
Chapter 9: Design for the user's convenience
Chapter 10: Accommodate the range of human variation
Chapter 11. Place the minimum mental burden on the user

Each chapter describes numerous examples of products that violate the design principle. Other cases of thoughtfully designed products exemplify the intent of each principle. We'll also see many lessons for designers to keep in mind as they strive to create better products. You can read these chapters in any sequence you like.

Finally, Chapter 12 recommends several techniques for creating more thoughtful designs. I'll discuss the importance of usage-centric design; ways to engage customers in the design process through product champions, personas, prototypes, and usability testing; and learning how to improve our future designs from project retrospectives and customer feedback. But first, let's see why creating good designs is so challenging.