USER FRIENDLY

How the hidden rules of design are changing the way we live, work, and play

Cliff Kuang with Robert Fabricant

For my wife and daughter

—CLIFF

To my family and friends
Hopefully this will explain once and for all
what I do every day, and why it matters

—ROBERT

MCD
Farrar, Straus and Giroux
120 Broadway, New York 10271

Copyright © 2019 by Cliff Kuang and Robert Fabricant
All rights reserved
Printed in the United States of America
First edition, 2019

Library of Congress Cataloging-in-Publication Data
Names: Kuang, Cliff, 1978– author. | Fabricant, Robert, 1966–
Title: User friendly : how the hidden rules of design are changing the way we live, work, and play / Cliff Kuang with Robert Fabricant.
Description: First edition. | New York : MCD/Farrar, Straus and Giroux, 2019. | Includes bibliographical references and index.
Identifiers: LCCN 2019022184 | ISBN 9780374279752 (hardcover)
Subjects: LCSH: Product design—Social aspects.
Classification: LCC TS171.4 .K83 2019 | DDC 658.5—dc23
LC record available at https://lccn.loc.gov/2019022184

Designed by Richard Oriolo
Illustrations by Chris Allen

Our books may be purchased in bulk for promotional, educational, or business use. Please contact your local bookseller or the Macmillan Corporate and Premium Sales Department at 1-800-221-7945, extension 5442, or by e-mail at MacmillanSpecialMarkets@macmillan.com.

www.mcdbooks.com · www.fsgbooks.com
Follow us on Twitter, Facebook, and Instagram at @mcdbooks

10 9 8 7 6 5 4 3 2 1

Contents

Introduction:
The Empire of User-Friendliness 3

Part I: Easy to Use

1. Confusion 15
2. Industry 49
3. Error 75
4. Trust 99
5. Metaphor 129

Part II: Easy to Want

6. Empathy 161
7. Humanity 187
8. Personalization 215
9. Peril 247
10. Promise 279

Afterword:
Seeing the World Through User-Friendly Eyes 301

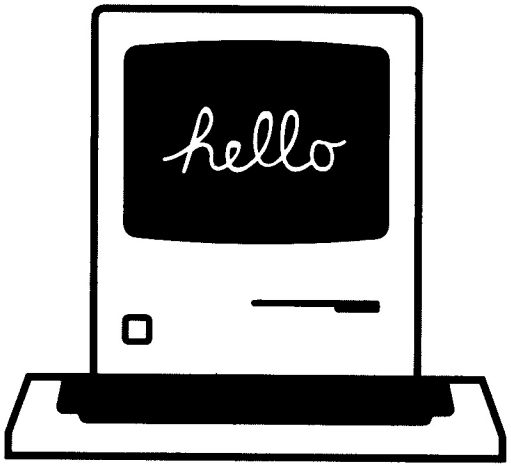
Appendix:
A Brief History of "User Friendly" 331

Notes 349

Bibliography 373

Acknowledgments 387

Index 393



Apple Macintosh (1984)

Introduction: The Empire of User-Friendliness

User Friendly

1. Computing. Of hardware or software: easy to use or understand, esp. by an inexperienced user; designed with the needs of a user in mind.
2. In extended use: easy to use; accessible, manageable.

At only four stories tall, the world's largest office building sits low to the ground but commands a footprint worthy of a UFO that could blot out the sun: a perfect doughnut shape, a mile around its edge. In the middle lies a grove meant to recall the time, just fifty years ago, when Silicon Valley wasn't Silicon Valley, but rather the Valley of Heart's Delight, covered in 10 million fruit trees—apricot, cherry, peach, pear, and apple. It took Apple, the computing giant, years to buy up all that land in secret, assembling some sixteen different plots across fifty acres in a $5 billion jigsaw. If the building looks like a spaceship, then it's one that lifted off directly from Steve Jobs's imagination to land in the heart of an otherwise sleepy suburb. It

was one of the last undertakings that the great man signed off on before he died.

Every morning during the construction of Jobs's last dream, Harlan Crowder woke up to the dull roar of heavy trucks on their way to the site, their alarms bleating as they nudged their loads into place. When we met, Crowder was seventy-three years old with three grown children. He wore a white goatee and a retiree's wardrobe of rumpled pants and floral shirts. In Crowder's neighborhood, the hubbub attending Apple Campus 2's arrival had unleashed a swarm of real estate agents trawling door-to-door and offering to make people rich.

These agents were mostly women buffed to a high gloss, and they came adorned in big brass jewelry that served as armor against holdouts like Crowder. "There was one, she said it just like that: 'I have ten people waiting to start a bidding war for your house,'" Crowder said in his Texas drawl as we sat talking on his back patio. Houses like his had originally been built in the 1960s for the population boom incited by the nascent transistor industry. When I visited, modest three-bedroom ranch houses like Crowder's were easily fetching $2.5 million. In a year, he assumed it could be 10 percent higher, maybe more. It was all some strange dream.

Crowder isn't a famous person—there are untold thousands like him, spread out in the Valley: people of technical ability who built the place but whose names are lost to history. But Crowder is one of the first people in the annals of history to use the term "user friendly" to refer to a computer. Every week or so, Crowder fends off the real estate agents and their offers of a multimillion-dollar payday. Apple fuels it all—Apple, the company that made "user friendly" into an idea that we live with every day.

Crowder looks on at the new Apple Park during his daily walks past the campus, the crown jewel in an empire built upon trillions

of dollars in iPods and iMacs and iPads and Apple Watches and iPhones—devices that, despite being some of the most advanced computers ever made, can still be operated by toddlers. Their power dwarfs that of the "supercomputers" Crowder once worked with at IBM. That he'd come to IBM at all seemed like another kind of dream. He'd failed eighth-grade algebra. After high school, he bummed around until enlisting in the Army, where he trained as a medic: a year and a half of doctor's training with all the theory stripped out, so that you simply knew the essentials required to save a life. The practicality and the pressure lit into him. The one-time layabout graduated first in his class. After service came college at East Texas State University, in Commerce. "It wasn't the edge of nowhere, but you could see the edge," he told me.

Crowder had seen an IBM recruitment flyer on a bulletin board at college, calling for science majors to enter a newfangled training program. He replied, and they called him back. He flew up to Yorktown, New York, without much idea of what to expect. IBM's research center was a crescent-shaped architectural landmark designed by the great Finnish American designer Eero Saarinen—the original corporate-campus-as-spaceship. Its facade was a gleaming curved-glass curtain wall, and the walls inside were hewn from New York's native granite. The building's design was a statement about the modern workplace from the world's smartest company, with as many Ph.D.s on staff as a university.

Crowder walked to his job interview in awe. The building resembled nothing so much as the spaceship from *2001*, a bright techno future that was big news in 1968. Here was a place where the water-cooler conversations were about scientific breakthroughs. "I'd have done anything to work there. I didn't care if I had to clean the toilets," Crowder said, his voice still shimmering with glee. IBM had run out of computer programmers, and the company wanted to

make more of them. Crowder got the job, and he took to the pragmatic nature of the work, using computers to solve real-world problems that you could measure by the ton, such as mapping shipping routes and calculating trucking loads. He found that he had a mind for visualizing the complex equations his job required.

The field where Crowder worked, operations research, began with World War II and the Marshall Plan. Rebuilding Europe meant shipping a mind-boggling amount of matériel across the Atlantic, but also shipping back a mind-boggling amount of matériel that had accumulated in dozens of countries during a war effort of unprecedented scale. Loading ships efficiently on the way over, then loading them up efficiently for the return home, was a math problem whose unruly complexity demanded computing power.

Crowder was working on these sorts of operational problems for IBM's clients in the 1960s. To create a computer program, he had to use a machine to punch intricate holes in cards the size of an airplane boarding pass. When he was done, he couldn't just walk up to the computer himself. The computer was a $5 million machine—about $35 million in today's money—patrolled by two security guards and a perky-eared German shepherd. Crowder would spend all day programming and then take his stack of cards to the computer attendant behind a window, who fed the cards into the machine. The computer would spend all night calculating, and the results would usually be ready for Crowder by the morning—if he hadn't made a mistake. This was a big "if." A mistake would be as simple as a misplaced character that gummed up the processing, or a poorly defined equation that divided by zero and sent the computer into looping infinities. (Shades of Apple again: The address of its former campus is One Infinite Loop.)

Faced with fitfully waiting overnight to see if all the days they'd spent programming had been wasted by a typo, a cultish group of

programmers at IBM found a way out. Working at a minicomputer tied to a mainframe down the hall and using a simplified programming language called APL—literally, A Programming Language—you could simply write a program and run it. You just typed up your program, saw whether the computer spat out meaningful results, and immediately knew whether your program was heading in the right direction. This was magic. Simply seeing the fruits of your work right away let you shape new ideas in your head just as soon as you had them. Years later, Steve Jobs would describe a computer as a bicycle for the mind—a fabulous machine that could turn the muscle power of a single person into the ability to traverse a mountain in a day. Crowder and his colleagues were among the very first to experience that ideal firsthand. The machine, once it was capable of *immediate feedback*, was actually augmenting what your mind could do. An insight might flash before you, and you could test that idea out right away. Seeing how well it worked would spur new ideas, and on and on. Thanks to these feedback loops, computer programming, which had once had the stately pace of chamber music, entered its own improvisational Jazz Age.

These "jazz musicians" traded their ideas in academic journals. The only thing was, it was awful trying to re-create the music someone had made on their machines. You couldn't tell how easy it would be to test another person's work or replicate it. The programs simply didn't consider what *someone else* might do with them. To Crowder, they weren't *user friendly*. And so Crowder proposed that a computer program be gauged not just on how well it solved a problem but on how easy it made the lives of the people trying to solve it. To be clear, he didn't actually invent the term. As far as he knows, it had been floating around in the air, and it was there when he needed it. Which tends to prove how powerful it was—how it encapsulated something that people had started to feel.

And yet IBM didn't go on to invent the user-friendly world—even though it hired some of the best designers in the world, such as Paul Rand, who created its logo; Eero Saarinen, who created its campus; and even Eliot Noyes, who designed its Selectric typewriter. Instead, that accomplishment is typically credited to Apple, which adapted ideas seeded at Xerox PARC to create the Macintosh. Just a decade after Crowder first wrote his paper describing user-friendly algorithms, Apple was making ads for user-friendly machines:

> In the olden days, before 1984, not very many people used computers—for a very good reason.
>
> Not very many people knew how.
>
> And not very many people wanted to learn . . .
>
> Then, on a particularly bright day in California, some particularly bright engineers had a brilliant idea: since computers are so smart, wouldn't it make sense to teach computers about people, instead of teaching people about computers?
>
> So it was that those very engineers worked long days and late nights—and a few legal holidays—teaching tiny silicon chips all about people. How they make mistakes and change their minds. How they label their file folders and save old phone numbers. How they labor for their livelihoods. And doodle in their spare time . . .
>
> And when the engineers were finally finished, they introduced us to a personal computer so personable it can practically shake hands.

There's a certain magic in how a few words can elide so many stories and so many ideas. This book is an attempt to paint a picture that's gone missing in plain sight.



It began as an idea first broached with me by Robert Fabricant, who at the time was vice president of creative at the firm Frog Design. We'd known each other for a couple of years, and Robert's pitch, which mirrored a decade of my own work as a writer and an editor, was simply that user-experience design, which had been the province of computer geeks and futurists, wasn't a niche anymore. In an era in which 2.5 billion people own smartphones, user experience now occupies the center of modern life, remaking not just our digital lives but also business, society, even philanthropy. It was Robert's idea to call this book *User Friendly*—a term whose very familiarity proves the thesis. And yet the history of the term, the meaning it conveys, the mechanics of how it works—these all remain, outside of a few professional circles, either untold or known only in pieces. What we first conceived over a period of a few months—and assumed would take six months to complete—eventually took me six years to write and report. That is the book you're reading now, one that tries to show how "user friendly" was invented, how it shapes our everyday rhythms, and where it will go next.

There is a certain generation of designers who quarrel with the term "user friendly." They disagree with the premise that gadgets should always presume a chipper familiarity to their users; they quibble about whether "friendliness" is the right relationship between a device and its user; they consider the idea condescending in how it implies that users must be treated like children. These critiques are sometimes reasonable. They also fail to undermine the term itself while missing a larger point.

Today, alluringly designed gadgetry has remade the texture of everyday life, from how we make money to how we make friends to how we make babies. We now expect that the tools we use to diagnose cancer or to identify a problem with an airplane engine will be as simple to use as *Angry Birds*. We aren't wrong to do so. Technology should become simpler over time. Then it should become simpler still, so that it disappears from notice. This has already happened with stunning speed, and that transformation is one of the greatest cultural achievements of the last fifty years. But even as the designed world shapes us, its inner logic remains almost totally absent from daily conversation. Instead, whether we're speaking to our kids or speaking to our grandparents, the only words we have are "user friendly." We hardly examine them at all—and yet they're the standard by which we judge the designed world.

"User friendly" rolls off the tongue unconsciously because we know what it means, or we think we know what it means. It means something like "Did this thing do what I want?" But even that simplified formulation raises a litany of questions: Why should some product defer to our desires? How does the person who created that object come to understand what I want to begin with? What can go wrong in translating our desires into artifacts? It took more than a century of progress and peril to answer those questions. This book is about how the idea of user-friendliness was born and how it works. We travel backward and forward in time, from the paradigm shifts that made user-friendliness into something people cared about at all to the present day, when user-friendliness has redefined nearly every minute of our waking lives.

Many of the ideas in this book will be familiar to user-experience designers—the people who observe our lives so that they might invent new products. Still, this story should be new. User-experience design, which has come to encompass everything from theme parks

to chatbots, simply hasn't had a narrative thread comprehensible to both laymen and experts. The great chain of ideas that spawned it typically hasn't been appreciated as a tapestry of personalities, happenstance, and ideological struggle. If user experience is foreign to you, I hope you'll come away from this book understanding how the world is being remade every day—the ideals, principles, and assumptions that lie behind the taps and swipes you take for granted. If you're a designer, I hope you'll better understand the origin of the ideas you swim among, so that you might better examine—and even challenge—the values you put into the things you make. At the very least, I hope you'll be able to offer this book to the people you know and say, "This is why user experience matters."