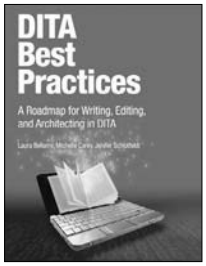
Developing Quality Technical Information

A Handbook for Writers and Editors

Third Edition

Michelle Carey, Moira McFadden Lanyi, Deirdre Longo,
Eric Radzinski, Shannon Rouiller, Elizabeth Wilde

Related Books of Interest



DITA Best Practices

A Roadmap for Writing, Editing, and Architecting in DITA

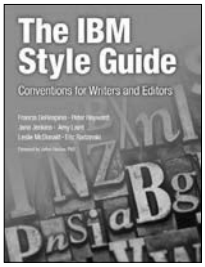
By Laura Bellamy, Michelle Carey, Jenifer Schlotfeldt

ISBN-13: 9780132480529

The Start-to-Finish, Best-Practice Guide to Implementing and Using DITA

Darwin Information Typing Architecture (DITA) is today's most powerful toolbox for constructing information. By implementing DITA, organizations can gain more value from their technical documentation than ever before. Now, three DITA pioneers offer the first complete roadmap for successful DITA adoption, implementation, and usage.

Drawing on years of experience helping large organizations adopt DITA, the authors answer crucial questions the "official" DITA documents ignore, including: *Where do you start? What should you know up front? What are the pitfalls in implementing DITA? How can you avoid those pitfalls?* If you're a writer, editor, information architect, manager, or consultant who evaluates, deploys, or uses DITA, this book will guide you all the way to success.



The IBM Style Guide

Conventions for Writers and Editors

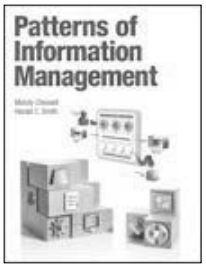
By Francis DeRespinis, Peter Hayward, Jana Jenkins, Amy Laird, Leslie McDonald, Eric Radzinski

ISBN-13: 9780132101301

The IBM Style Guide distills IBM wisdom for developing superior content: information that is consistent, clear, concise, and easy to translate. This expert guide contains practical guidance on topic-based writing, writing content for different media types, and writing for global audiences. The guidelines are especially valuable for businesses that have not previously adopted a corporate style guide, for anyone who writes or edits for IBM as an employee or outside contractor, and for anyone who uses modern approaches to information architecture.

Filled with many examples of correct and incorrect usage, *The IBM Style Guide* can help any organization or individual create and manage content more effectively.

Books of Interest



Patterns of Information Management

By Mandy Chessell, Harald Smith

ISBN: 9780133155501

Use Best Practice Patterns to Understand and Architect Manageable, Efficient Information Supply Chains That Help You Leverage All Your Data and Knowledge

In the era of "Big Data," information pervades every aspect of the organization. Therefore, architecting and managing it is a multi-disciplinary task. Now, two pioneering IBM® architects present proven architecture patterns that fully reflect this reality. Using their pattern language, you can accurately characterize the information issues associated with your own systems, and design solutions that succeed over both the short- and long-term.

Mobile Strategy

How Your Company Can Win by Embracing Mobile Technologies

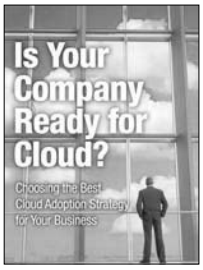
By Dirk Nicol

ISBN-13: 9780133094916

Navigate the Mobile Landscape with Confidence and Create a Mobile Strategy That Wins in the Market Place

Mobile Strategy gives IT leaders the ability to transform their business by offering all the guidance they need to navigate this complex landscape, leverage its opportunities, and protect their investments along the way. IBM's Dirk Nicol clearly explains key trends and issues across the entire mobile project lifecycle. He offers insights critical to evaluating mobile technologies, supporting BYOD, and integrating mobile, cloud, social, and big data. Throughout, you'll find proven best practices based on real-world case studies from his extensive experience with IBM's enterprise customers.

Books of Interest



Is Your Company Ready for Cloud?

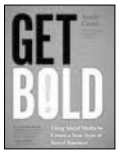
Choosing the Best Cloud Adoption Strategy for Your Business

By Pamela K. Isom, Kerrie Everall Holley

ISBN-13: 9780132599849

Make the Right Cloud Adoption and Deployment Decisions for Your Business

This is the first complete guide to cloud decision making for senior executives in both technology and non-technology roles. IBM® Global Business Services® Executive Architect Pamela K. Isom and IBM Fellow Kerrie Holley present practical business cases, vignettes, and techniques to help you understand when cloud investments make sense and when they don't. You'll find decision models that are anchored with practical experiences and lessons to guide your decision making, best practices for leveraging investments you've already made, and expert assistance with every aspect of the cloud transition.

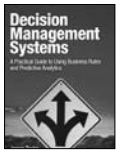


Get Bold

Using Social Media to Create a New Type of Social Business

By Sandy Carter

ISBN-13: 9780132618311

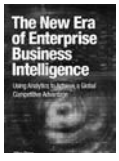


Decision Management Systems

A Practical Guide to Using Business Rules and Predictive Analytics

By James Taylor

ISBN-13: 9780132884389



The New Era of Enterprise Business Intelligence

Using Analytics to Achieve a Global Competitive Advantage

By Mike Biere

ISBN-13: 9780137075423

Opting In

Lessons in Social Business from a Fortune 500 Product Manager

By Ed Brill

ISBN-13: 9780133258936

The Greening of IT

How Companies Can Make a Difference for the Environment

By John Lamb

ISBN-13: 9780137150830

Sign up for the monthly IBM Press newsletter at ibmpressbooks.com/newsletters

This page intentionally left blank

Developing Quality Technical Information

This page intentionally left blank

IBM PRESS

Developing Quality Technical Information

A Handbook for Writers and Editors

Michelle Carey • Moira McFadden Lanyi • Deirdre Longo • Eric Radzinski • Shannon Rouiller • Elizabeth Wilde

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

IBM Press Program Managers: Steven M. Stansel, Ellice Uffer

Cover design: Michael Rouiller and Polly Hughes

Editor-in-Chief: Bernard Goodwin
Marketing Manager: Stephane Nakib
Acquisitions Editor: Bernard Goodwin
Publicist: Heather Fox
Managing Editor: Kristy Hart
Designer: Alan Clements
Project Editor: Andy Beaster
Full-service composition: codeMantra US LLC
Project Manager: Francesca Monaco, codeMantra US LLC
Manufacturing Buyer: Dan Uhrig

Published by Pearson plc

Publishing as IBM Press

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at corpsales@pearsoned.com or (800) 382-3419.

For government sales inquiries, please contact governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact international@pearsoned.com.

Library of Congress Control Number: 2014931735

ISBN-13: 978-0-13-311897-1

ISBN-10: 0-13-311897-5

Text printed in the United States on recycled paper at R.R. Donnelley in Crawfordsville, Indiana.
First printing: June 2014

This page intentionally left blank

Contents

This page intentionally left blank

Preface

About this book

Many books about technical writing tell you how to develop different elements of technical information, such as headings, lists, tables, and indexes. We took a different approach with this book; we organized it to show you how to apply quality characteristics that make technical information, including information embedded in user interfaces, easy to use, easy to understand, and easy to find. We hope you will find our approach useful and comprehensive—and we hope that you will find the information in this book easy to use, easy to understand, and easy to find!

Is this book for you?

If you are a writer, editor, information architect, or reviewer of technical information and user interfaces, then yes, this book is for you. If you work on software information, this book will be of particular interest to you because most of the examples in it come from the domain of software. However, the quality characteristics and guidelines apply to all technical information.

In general, this book assumes that you know the basics of good grammar, punctuation, and spelling as they apply to writing. It does not assume that you are familiar with what makes technical information effective or ineffective.

Changes in this edition

The organization of the book and the quality characteristics remain the same. However, within each quality characteristic, we made significant changes by replacing some guidelines with new ones, adding many new examples, and broadening the scope of the kinds of information that we discuss. If you

are familiar with previous editions, you'll find a great deal of new content in this edition. For example, the following guidelines are among those that we added:

- "Apply a pattern for disclosing information" in the chapter about completeness
- "Guide users through the information" in the chapter about retrievability
- "Put information where users expect it" in the chapter about organization
- "Make information interactive" in the chapter about visual effectiveness

These changes resulted from several developments in technical communication:

- Greater emphasis on the embedded assistance in user interfaces
- The need to plan for information access from mobile devices
- The pervasiveness of Google and other search engines as users' preferred method for looking for information
- Video as a delivery medium for technical information

As with earlier developments in this field during the many years that these quality characteristics have been in use, the characteristics remain relevant while the definition of technical information expands in scope. This quality framework continues to apply to the information that we provide today. In addition, we have found that the characteristics apply well to user interfaces, which benefit from application of the guidelines much as other content does.

We hope that you find this book useful in improving the quality of the information that you develop.

Acknowledgments

The predecessor of this book was an internal document called *Producing Quality Technical Information*. That document led to the first edition of *Developing Quality Technical Information*, which was published in 1998, followed by the second edition in 2004. And here we are 10 years later with the third edition.

After the second edition of *Developing Quality Technical Information* was published, its lead author and project manager, Gretchen Hargis, passed away.

Throughout the writing process for the first two editions of this book, Gretchen was vigorous in pushing the authors to do what was necessary to make the book as good as it could possibly be. We planned, we drafted, we edited, we haggled, we revised, we reedited, we proofread. Throughout the process, the concept of "good enough" never entered Gretchen's mind.

Sometimes, Gretchen's coauthors wished "good enough" had been just that, but in retrospect we are so glad that Gretchen persevered. Without Gretchen, neither the first nor second edition of the book would ever have been completed. Gretchen is sorely missed by all of her coauthors and colleagues.

We felt that she was with us every step of the way as we wrote the third edition, and we hope that this edition lives up to her standards.

Over the years, nearly a hundred talented people have in some way contributed to this latest edition. We thank all the people who helped with this book and its predecessors.

One of the biggest challenges to writing this book was providing the vast number of examples in each chapter. We were fortunate to get help identifying many excellent examples. Many thanks to Hassi Norlen, Richie Escarez, Ellen

Livengood, Ann Hernandez (author from the second edition), Beth Hettich, Erin Jerison, Marcia Carey (Michelle's mom), and Gary Rodrigues for helping with the nearly 400 examples provided in this book.

And thanks to the talented visual designers and writers who provided some of the examples in the "Visual effectiveness" chapter: Tina Adams, Daiv Barrios, Jessy Chung, Caroline Law, Adam Locke, Challen Pride-Thorne, Rene Rodriguez, Shannon Thompson, and Jacob Warren. Thanks especially to visual designers Clark Gussin and Sean Lanyi for always being available to advise and help.

Many of the clarity and style issues we discuss come out of trying to do what's best for translators, so we'd like to thank Sabine Lehmann, Ph.D., for her guidance about machine translation and linguistics and for her translations of French, German, and English examples.

We thank Michael Rouiller and Polly Hughes (second edition author) for their help with the cover graphic.

We'd also like to thank the following folks who helped us find the quotations that introduce each quality characteristic chapter: Christopher Clunas, Paula Cross, Fran DeRespinis, Jasna Krmpotic, Yvonne Ma, and Leslie McDonald.

For technical support, we'd never have finished the book if it hadn't been for Dan Dionne, Kevin Cheung, and Simcha Gralla. Many thanks to these gentlemen for their help.

Thanks to Andrea Ames for the hours and laughs we continue to share while defining and building education for embedded assistance and progressive disclosure of information within and outside of IBM. Thanks also to Jennifer Fell who gave us a wonderful metaphor that describes how users should be able to use technical information: "As a guided journey instead of a scavenger hunt."

Thanks to Lori Fisher who created a space over the course of many years for all of us to contribute to the craft of information development and to develop a framework for information quality. Thanks to Eileen Jones for sponsoring this edition and for fostering the profession of information development at IBM.

We thank our families, friends, cats, and dogs for their incredible patience and support throughout the writing process. We stole countless late nights, weekends, and decent meals from them, and we can never pay those back.

Lastly, we must thank our talented editor, Julian Cantella, for the many long hours he spent editing our manuscript. It's never easy editing a book that's written by a team of editors. Julian's thoughtful and meticulous work helped us add that extra polish to the book.

Michelle Carey
Moira McFadden Lanyi
Deirdre Longo
Eric Radzinski
Shannon Rouiller
Elizabeth Wilde

This page intentionally left blank

About the authors

The authors are all long-standing and respected members of the information development community at IBM. Although the authors have served in various roles throughout their careers, information quality has always been and continues to be their primary focus.

Michelle Carey is an information architect and technical editor at IBM and has taught technical communication at University of California Santa Cruz Extension. Michelle is the co-author of the book *DITA Best Practices: A Roadmap for Writing, Editing, and Architecting in DITA*. She is an expert on topic-based information systems, software product error messages, grammar, embedded assistance for user interfaces, and writing for international audiences. She also writes computational linguistic rules for a grammar, style, and terminology management tool. Michelle enjoys teaching, grammar, herding cats, and riding and driving anything with a lot of horsepower.

Moira McFadden Lanyi is an information architect and technical editor at IBM. She has experience with topic-based writing, DITA, embedded assistance, user interface design, and visual design. She created 99% of the artwork in this book. She is a co-author of the book *An Introduction to IMS*. Moira enjoys visiting San Francisco with her family as often as possible, cooking fresh, healthy meals, and watching her courageous son ride his unicycle and surf.

Deirdre Longo is an information architect and strategist at IBM. She has been a pioneer for embedded assistance in IBM: defining the scope of that term, developing standards for embedded assistance, and modeling how to work effectively in cross-disciplinary teams. She has taught webinars for the Society of Technical Communication (STC) and published articles on information architecture topics in STC's *Intercom*. She is an avid yoga practitioner.

Eric Radzinski is a technical editor and information architect for industry-leading mainframe database software at IBM. He is a co-author of *The IBM Style Guide: Conventions for Writers and Editors* and is well versed in topic-based writing, embedded assistance, DITA, and writing for a global audience. Eric makes his home in San Jose, California, with his wife and their three children.

Shannon Rouiller is an information architect and technical editor at IBM. She has experience with quality metrics, topic-based information systems, DITA, videos, embedded assistance, and user interface design. She is a co-author of the book *Designing Effective Wizards*. Shannon dabbles in sports photography and likes to solve puzzles.

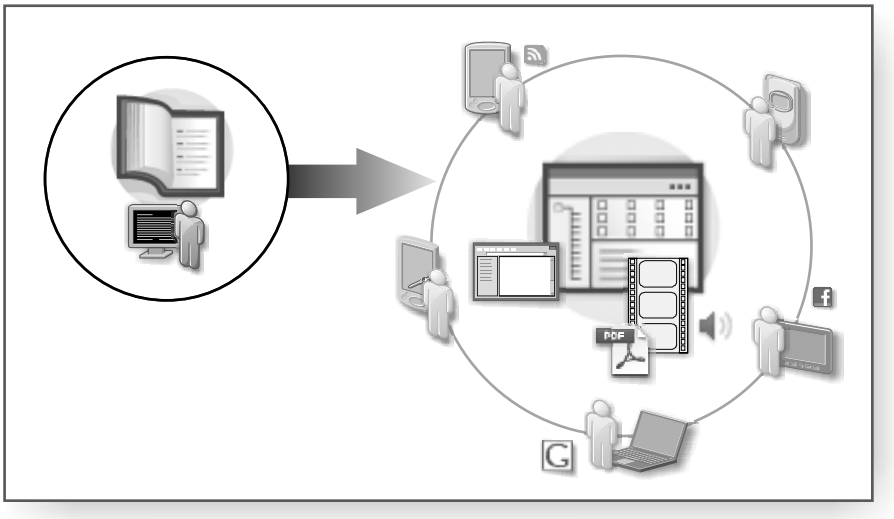
Elizabeth Wilde is an information quality strategist at IBM, developing strategies and education for developing high-quality content. She develops Acrolinx computational linguistic rules that enforce grammar, style, and DITA tagging rules. She teaches an extension course in technical writing at the University of California Santa Cruz. Her hobbies include growing cacti and succulents and collecting tattoos.

This page intentionally left blank

CHAPTER 1

Technical information continues to evolve

The nature of our work as technical communicators continues to change, more rapidly than ever. The authors of this book can see it even over the short course, relatively speaking, of our own careers in technical communication. Some of us began our careers delivering camera-ready copy for a shelf of physical books and then began producing context-sensitive online help that was installed with the product. With the advent of the web, we used our online help-writing skills to rework books into online topic-based documentation.



Today, writers sigh or laugh ruefully over the fact that users don't click help links. Testing with users validates this premise—that users don't want to ask for separate help—but that they *do* use all of the text they see in user interfaces to do their tasks. In surveys, users often say that their first response to trouble is to ask a colleague. In testing, when users were forced to seek additional assistance, a majority in our tests tried to search the Internet or visit a video site such as YouTube rather than reading the help. This finding is surprising at first, but on reflection, is the equivalent of asking a colleague.

One reason that users avoid help documentation is that we, as a profession, have taught them that, as one user told us, "There's nothing good there." For example, when we moved from command-oriented products to those with graphical interfaces, technical information was focused on helping users to understand how to manipulate the user interface. Although that focus made sense during the transition, many writers continue that focus today, 20 years after the transition. In spite of knowing better, we continue to produce huge amounts of help documentation.

As technical writers, we need to recognize this shift in our audience and move past it to address users where they are. A new generation of technology-savvy users is entering the workforce, existing workers are becoming more adept, and technology is becoming more sophisticated. Because of these changes, the emphasis is on more usable, intuitive, and appealing products. Now we need to expand our focus beyond topic-based information and onto the product user interfaces themselves, with input field labels, messages, and other embedded text, which collectively we refer to as *embedded assistance*.

We need to recognize that topics alone cannot address all needs. Topics work well in some contexts and for some types of documentation: planning, application programming, technical concepts, troubleshooting, and hardware diagrams. In many contexts, users expect to stay where they are and figure out how to do their tasks without reading separate documentation. But in many other contexts, especially mobile contexts, users want to watch a video introduction or a presentation by an expert. We need to write information for users where they are, focused on what they're trying to accomplish, instead of trying to make them read what they don't want to read.

Embedded assistance

Our profession has different definitions for embedded assistance.

Some groups refer to static, descriptive inline text in a user interface as embedded assistance and differentiate it from the interface labels and messages. Others use the term to refer to the mechanism that displays a pane of online

help text within the same window as the product. For the purposes of this book, we define embedded assistance as both of those and more—to define it more narrowly only reinforces the artificial separation between product and documentation that occurs because of the way most products and documentation are developed. When users buy or use a product, they don't differentiate between the interface, the documentation, and the functionality. To users, all of these are the product. We, with all members of our product development teams, must develop our products as a whole too.

Embedded assistance, therefore, encompasses all textual and graphical elements that users encounter in all types of products. In graphical user

interfaces, embedded assistance includes (numbers refer to Figure 1.1 below):

- Labels for user interface controls such as fields, radio buttons, check boxes, push buttons, menus, window titles, and so on (1)
- Input hints in fields (2)
- Descriptive inline user interface text such as introductory text in a window (3)
- Messages that appear on fields, in sections of the interface, or in dialogs
- Tooltips, which are one- to two-word names for tools that do not have labels in the interface
- Hover help, which are one to two sentences of description for fields, check boxes, radio buttons, and so on (6)
- Wizards for simplifying complex interactions
- Embedded help panes (8)

The following illustration shows some of these elements in a user interface:

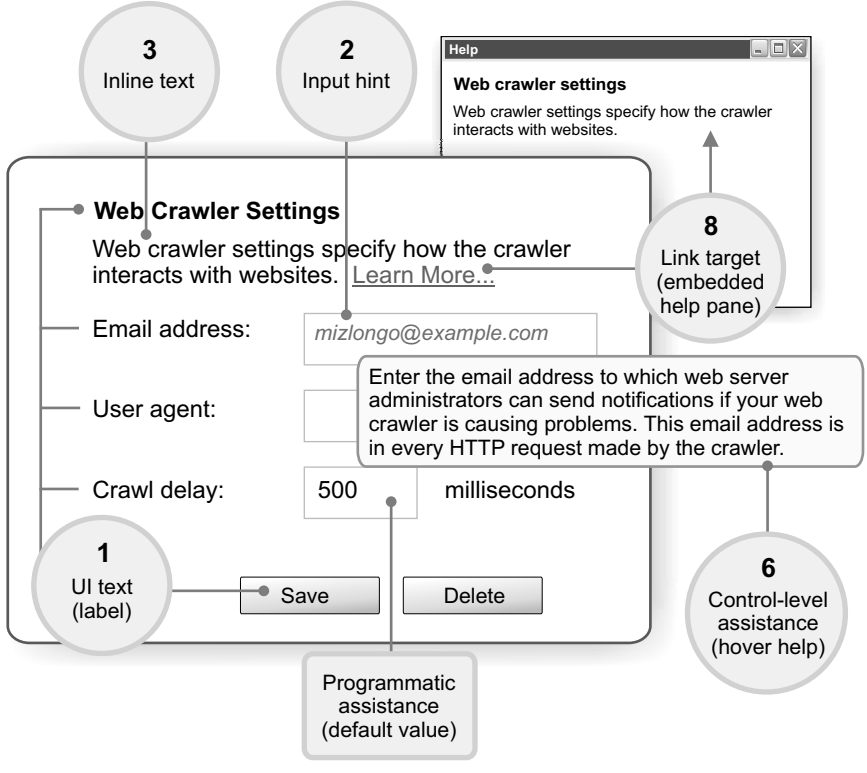
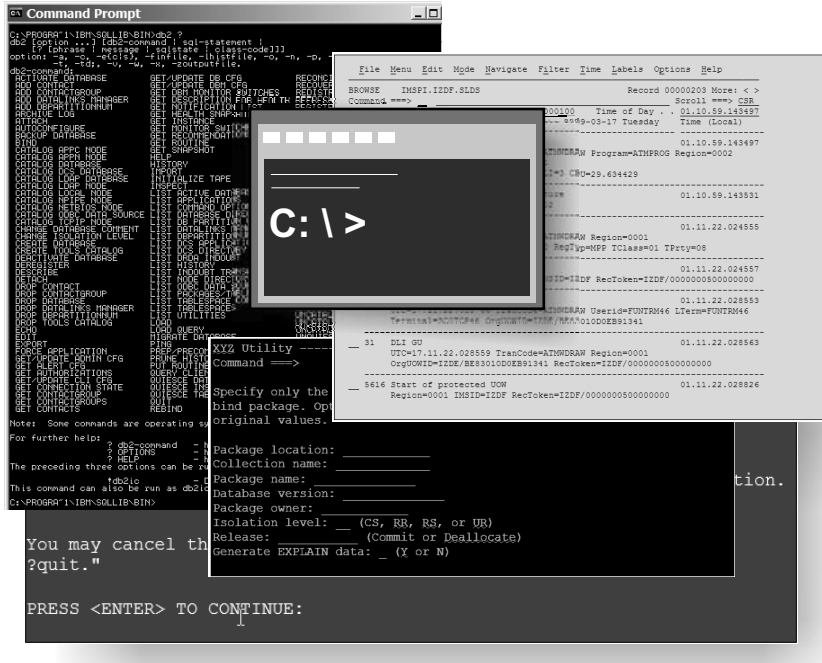


Figure 1.1 Embedded assistance elements in a user interface.

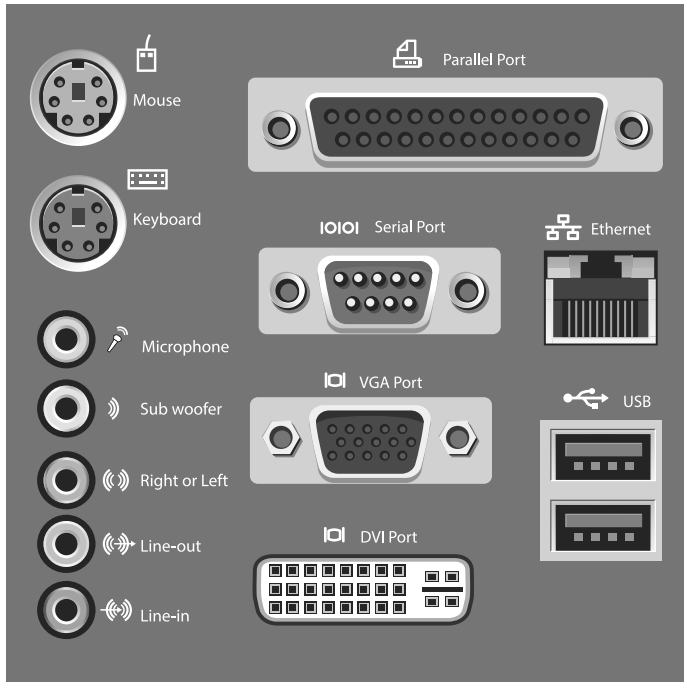
In nongraphical software contexts, such as ASCII-based interfaces, embedded assistance includes:

- Logged messages
- Command and parameter names
- Keyword names
- API names
- Utility or tool names



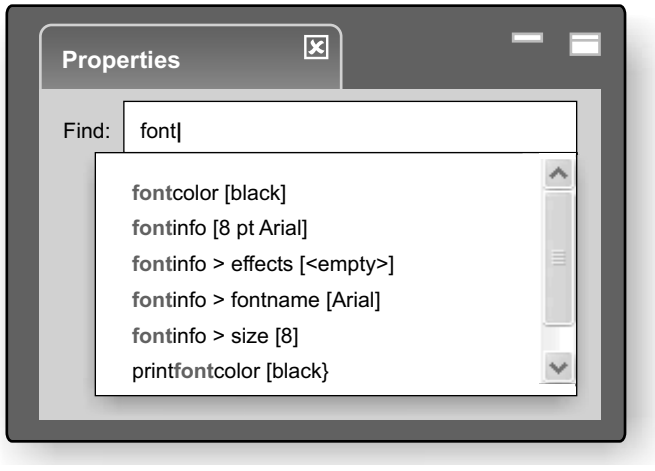
In hardware contexts, embedded assistance includes:

- Labels embedded on hardware wires, boards, or other equipment
- Labels attached on top of or around hardware, for example on an on/off switch
- Specifically sized slots for connectors
- Colors for wiring, for example, the color green indicates the ground wire in the US



Embedded assistance also includes *programmatic assistance* that does a step or task for a user. Examples of programmatic assistance include:

- Default values
- Detected values
- Autocompleted values, as shown in the following user interface:



Although we writers usually don't have the programming skills to develop programmatic assistance, we do need to understand these types of assistance well enough to advocate for them when they'll be helpful to users.

Our skills with writing embedded assistance and our fluency with words and graphics will be crucial as software development shifts to focus on mobile devices. Most users don't follow help links in desktop and web applications; they are even less likely to do so in mobile environments. The lack of hover capabilities in the mobile environment removes an element of embedded assistance in an already small user interface, a user interface that makes web and desktop interfaces seem enormous by comparison. Because of the small screen size in a mobile interface, the small amount of text that is persistent gets even more attention.

Progressive disclosure of information

Given the types of embedded assistance elements in the previous section, it's easy to see that writers can't work in isolation on each element, set of elements, or functional area of the user interface or piece of hardware. Instead, the entire set of embedded assistance, linked assistance, and separate documentation must tell a cohesive story.

The key to developing effective documentation is to apply and follow a pattern for progressively disclosing the information to the user. *Progressive disclosure* is not a new idea in the field of interaction design. Jakob Nielsen summarized it in 2006: "Progressive disclosure defers advanced or rarely used features to a secondary screen, making applications easier to learn and less error-prone." Applying such a progressive pattern to information ensures that you use available space in a user interface or on the hardware in the most effective way, consistently and without redundancy. Applying the pattern well also helps writing teams manage the complexity of information, providing clear paths to get to more complex or abstract information.

Information that is developed according to principles of progressive disclosure anticipates users' questions and provides a way for users to get additional contextual information when necessary. For example, in an installation wizard, a field might have the label **Application server version**, and a user might ask, "Is this the version I'm upgrading from or to?" Ideally, the label could be changed to clarify which version, or the field could be grouped under a heading **Upgrade from server**. If neither is possible, a hover help that explains which application server is being requested and how to find this information is helpful, but a hover help that says, "Enter the application server version" is not.

If you are used to writing books or help documentation, think about how you decide where (which book or help system) to deliver certain information today. Information that you put in an installation guide is not appropriate in an application development guide. A similar approach is true in user interfaces: different interface display mechanisms require different types of information.

Think of the available programmatic and textual assistance capabilities in a software or hardware interface as different delivery mechanisms. You can then use a pattern to map types of content to each mechanism. Because these delivery mechanisms are much smaller than a book or a web page, the pattern is also at a different scale. Instead of thinking about the type of content to deliver in a programming guide versus an installation guide, you think about the type of content to deliver in a field label versus hover help. Your pattern might look something like this:

Table 1.1 Sample pattern for progressively disclosing information in a web user interface

User interface element	Content
Labels (for fields, windows, buttons, group boxes, and so on)	Succinct nouns based on a short, well-managed list of product terms. Repeatedly review these labels to ensure consistency and coherence of the interface as it is developed.
Messages	Full sentences that describe the situation. For error messages, provide an action so that users can solve the problem.
Static descriptive text at the top of windows	• Describe the overall action that users accomplish on the window if it's not obvious. • Clarify anything users must do before completing this window. • Identify ramifications, if there are any, of the changes in this window.
Static descriptive text below fields	Examples for what to enter in a field.
Hover help	• Syntax for what to enter in a field. • Ramifications of the field change. • Descriptive information for what to enter in the field. • Links to additional information if needed.

The organization guideline "Separate contextual information into the appropriate type of embedded assistance" on page 332 describes the pattern in more depth, and the completeness guideline "Apply a pattern for disclosing information" on page 107 explains how to apply it to your information to ensure completeness.

When you become more adept at creating meaningful and effective embedded assistance and delivering it progressively, you create a better customer experience and become a more valuable member of your product team.

The technical writer's role today

Our roles as technical writers are evolving as quickly as the products that we write information for. Because we develop embedded assistance, the timing and ways that we work with our extended teams have changed. We are more involved with product design and user interface development, which means that we must be involved earlier than ever in the development cycle.

As discussed in "Embedded assistance" on page 4, the separation between product and documentation is an artificial one, in large part a result of the historic waterfall development processes. The *waterfall development process* is

made up of specific phases in which each participating team finishes its work and hands it to the next team. The problem with this process is that downstream teams have very little chance to change anything that happened upstream. Furthermore, because documentation is developed close to the end of the cycle, documentation often tries to describe poor design that can no longer be changed. Too often, technical writers who work in a waterfall development process must write comprehensive documentation that needs to atone for unwieldy design.

More and more development teams are using an *agile development process*, which depends on cross-functional teams working together throughout an iterative development cycle. Although members of these cross-functional teams all bring their own skills to the team from their unique disciplines, they are much more likely to look at and contribute to each others' deliverables so that products are a full team effort. Agile development, as the name implies, lends itself to making quick changes to product design when necessary.

In agile development, writers have a particularly effective role as the users' advocate. The Agile Manifesto (agilemanifesto.org) values "individuals and interactions over processes and tools" and "working software over comprehensive documentation." Writers who work on a project that follows the agile development process are critical members of the team throughout the entire process, from the earliest design phase, before a single line of code is written, to the final fit-and-finish stage. By participating in the design process in partnership with product developers, usability engineers, visual designers, and customers, writers can promote clear interaction and wise use of embedded assistance, thereby reducing or eliminating the likelihood of "papering the product" with unnecessary documentation.

The guidelines in this book describe the characteristics of quality technical information. However, your role in developing information and, indeed, in developing the product, is as important as any of the guidelines. Rather than trying to explain problems with the product design after the fact, focus on fixing real-world problems that users have.

When you develop quality technical information, you are responsible for:

- Knowing the user stories, which are the goals that users need to accomplish by using the product
- Being the users' advocate, ensuring that the product employs the necessary programmatic assistance and embedded assistance
- Owning the words, whether they are in labels in the user interface, error messages, or topics that are separate from the product

Redefining quality technical information

Quality is ultimately determined by users. When users have questions and quickly find the exact information they need, they perceive the product (and the information, though they don't distinguish between the two) as being of high quality. In fact, an overwhelming majority of customers report that information quality both affects their view of the product quality and their overall product satisfaction. Information quality also has a significant impact on customers' buying decisions.

Almost always, users seek answers to specific questions and don't want to read a book from beginning to end to find those answers. Quality information addresses users where they are, for example, in the user interface. That quality helps them accomplish real goals rather than forcing them to figure out how to accomplish their goals in the product.

Content that focuses on *domain expertise*, provided by experts in the field based on their experience and judgement, is the most highly valued content today. We can already see the beginning of another technical communication transition toward artificial intelligence, and our role in gathering real domain expertise for users becomes critical. Think of voice-driven assistance that provides real-world information about proximity to gas stations with the lowest prices or guidance for how to choose the right app from an online store. In these situations, the writer is the trusted colleague or the concierge, directing users to exactly what they need at that moment. Domain expertise is described in more detail in the concreteness guideline, "Consider the skill level and needs of users" on page 220.

Technical writers must be the users' advocate throughout the product development process. Ideally, writers have access to users throughout that process, but user engagement alone cannot ensure information quality. Writers must apply their own skills and expertise based on solid research and proven methods.

Quality characteristics for technical information must reflect what users expect and want from the information. Based on comments from users and

on experience in writing and editing technical information, the authors of this book have found that quality technical information has these characteristics:

Easy to use	
Task orientation	In the context of a product, a focus on helping users do tasks that support their goals
Accuracy	Freedom from mistake or error; adherence to fact or truth
Completeness	The inclusion of all necessary parts—and only those parts
Easy to understand	
Clarity	Freedom from ambiguity or obscurity; using language in such a way that users understand it the first time that they read it
Concreteness	The inclusion of appropriate examples, scenarios, similes, analogies, specific language, and graphics
Style	Correctness and appropriateness of writing conventions and of words and phrases
Easy to find	
Organization	A coherent arrangement of parts that makes sense to the user
Retrievability	The presentation of information in a way that enables users to find specific items quickly and easily
Visual effectiveness	Attractiveness and enhanced meaning of information through the use of layout, illustrations, color, typography, icons, and other graphical devices

You can apply the quality characteristics whether you're writing a book, a page, a paragraph, a sentence, or a single word in an interface. The quality technical information model of nine characteristics is flexible enough to support you as you develop ever smaller chunks of information to address the changing needs of users.

Index

A

B

C

D

E

J

K

L

M

N

O

P

Q

R

S

T

U

V

W