Related Books of Interest

**DITA Best Practices**  
A Roadmap for Writing, Editing, and Architecting in DITA  
By Laura Bellamy, Michelle Carey, Jenifer Schlotfeldt  

**The IBM Style Guide**  
Conventions for Writers and Editors  
By Francis DeRespinis, Peter Hayward, Jana Jenkins, Amy Laird, Leslie McDonald, Eric Radzinski  

*DITA Best Practices* is a 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* 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.

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

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

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

Decision Management Systems
A Practical Guide to Using Business Rules and Predictive Analytics
By James Taylor

The New Era of Enterprise Business Intelligence
Using Analytics to Achieve a Global Competitive Advantage
By Mike Biere

Opting In
Lessons in Social Business from a Fortune 500 Product Manager
By Ed Brill

The Greening of IT
How Companies Can Make a Difference for the Environment
By John Lamb

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
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preface</td>
<td>xvii</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>xix</td>
</tr>
<tr>
<td>About the authors</td>
<td>xxiii</td>
</tr>
<tr>
<td><strong>Part 1. Introduction</strong></td>
<td>1</td>
</tr>
<tr>
<td>Chapter 1. Technical information continues to evolve</td>
<td>3</td>
</tr>
<tr>
<td>Embedded assistance</td>
<td>4</td>
</tr>
<tr>
<td>Progressive disclosure of information</td>
<td>9</td>
</tr>
<tr>
<td>The technical writer’s role today</td>
<td>11</td>
</tr>
<tr>
<td>Redefining quality technical information</td>
<td>13</td>
</tr>
<tr>
<td>Chapter 2. Developing quality technical information</td>
<td>15</td>
</tr>
<tr>
<td>Preparing to write: understanding users, goals, and product tasks</td>
<td>16</td>
</tr>
<tr>
<td>Writing and rewriting</td>
<td>17</td>
</tr>
<tr>
<td>Reviewing, testing, and evaluating technical information</td>
<td>19</td>
</tr>
<tr>
<td><strong>Part 2. Easy to use</strong></td>
<td>21</td>
</tr>
<tr>
<td>Chapter 3. Task orientation</td>
<td>23</td>
</tr>
<tr>
<td>Write for the intended audience</td>
<td>25</td>
</tr>
<tr>
<td>Present information from the users’ point of view</td>
<td>27</td>
</tr>
<tr>
<td>Focus on users’ goals</td>
<td>32</td>
</tr>
<tr>
<td>Identify tasks that support users’ goals</td>
<td>33</td>
</tr>
<tr>
<td>Write user-oriented task topics, not function-oriented task topics</td>
<td>35</td>
</tr>
<tr>
<td>Avoid an unnecessary focus on product features</td>
<td>41</td>
</tr>
</tbody>
</table>
Contents

Indicate a practical reason for information 46
Provide clear, step-by-step instructions 49
  Make each step a clear action for users to take 51
  Group steps for usability 53
  Clearly identify steps that are optional or conditional 58
Task orientation checklist 64

Chapter 4. Accuracy 67
  Research before you write 69
  Verify information that you write 74
  Maintain information currency 79
    Keep up with technical changes 79
    Avoid writing information that will become outdated 82
  Maintain consistency in all information about a subject 86
    Reuse information when possible 87
    Avoid introducing inconsistencies 88
  Use tools that automate checking for accuracy 93
  Accuracy checklist 96

Chapter 5. Completeness 99
  Make user interfaces self-documenting 101
  Apply a pattern for disclosing information 107
  Cover all subjects that support users’ goals and only those subjects 115
    Create an outline or topic model 115
    Include only information based on user goals 118
    Make sure concepts and reference topics support the goals 121
  Cover each subject in only as much detail as users need 123
    Provide appropriate detail for your users and their experience level 123
    Include enough information 130
    Include only necessary information 136
  Repeat information only when users will benefit from it 141
  Completeness checklist 148

Part 3. Easy to understand 151

Chapter 6. Clarity 153
  Focus on the meaning 155
  Eliminate wordiness 161
Write coherently 174
Avoid ambiguity 180
   Use words as only one part of speech 180
   Avoid empty words 183
   Use words with a clear meaning 187
   Write positively 189
   Make the syntax of sentences clear 194
   Use pronouns correctly 199
   Place modifiers appropriately 201
Use technical terms consistently and appropriately 205
   Decide whether to use a term 205
   Use terms consistently 207
   Define each term that is new to the intended audience 210
Clarity checklist 212

**Chapter 7. Concreteness** 215
   Consider the skill level and needs of users 220
   Use concreteness elements that are appropriate for the information type 223
   Use focused, realistic, and up-to-date concreteness elements 240
   Use scenarios to illustrate tasks and to provide overviews 243
   Make code examples and samples easy to use 247
   Set the context for examples and scenarios 251
   Use similes and analogies to relate unfamiliar information to familiar information 253
   Use specific language 256
   Concreteness checklist 259

**Chapter 8. Style** 261
   Use active and passive voice appropriately 263
   Convey the right tone 267
   Avoid gender and cultural bias 273
   Spell terms consistently and correctly 276
   Use proper capitalization 280
   Use consistent and correct punctuation 284
   Apply consistent highlighting 296
   Make elements parallel 302
   Apply templates and reuse commonly used expressions 305
   Use consistent markup tagging 311
   Style checklist 314
Part 4. Easy to find

Chapter 9. Organization
Put information where users expect it
Separate contextual information from other types of information
Separate contextual information into the appropriate type of embedded assistance
Separate noncontextual information into discrete topics by type
Arrange elements to facilitate navigation
Organize elements sequentially
Organize elements consistently
Reveal how elements fit together
Emphasize main points; subordinate secondary points
Organization checklist

Chapter 10. Retrievability
Optimize for searching and browsing
Use clear, descriptive titles
Use keywords effectively
Optimize the table of contents for scanning
Guide users through the information
Link appropriately
Link to essential information
Avoid redundant links
Use effective wording for links
Provide helpful entry points
Retrievability checklist

Chapter 11. Visual effectiveness
Apply visual design practices to textual elements
Use graphics that are meaningful and appropriate
Illustrate significant tasks and concepts
Make information interactive
Use screen captures judiciously
Apply a consistent visual style
Use visual elements to help users find what they need
Ensure that visual elements are accessible to all users
Visual effectiveness checklist
Part 5. Putting it all together 485

Chapter 12. Applying more than one quality characteristic 487
   Applying quality characteristics to progressively disclosed information 488
   Applying quality characteristics to information for an international audience 494
   Applying quality characteristics to topic-based information 501

Chapter 13. Reviewing, testing, and evaluating technical information 515
   Reviewing technical information 516
   Testing information for usability 518
   Testing technical information 524
   Editing and evaluating technical information 527
   Reading and editing the information 531
   Reviewing the visual elements 536

Part 6. Appendixes 543

Appendix A. Quality checklist 545
Appendix B. Who checks which characteristics? 549

Glossary 555
Resources and references 565
Index 573
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.
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
Acknowledgments

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.
About the Authors

**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
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:

![Diagram showing embedded assistance elements in a user interface](image)
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:

![User Interface Example]

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

<table>
<thead>
<tr>
<th>User interface element</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Labels (for fields, windows, buttons, group boxes, and so on)</td>
<td>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.</td>
</tr>
<tr>
<td>Messages</td>
<td>Full sentences that describe the situation. For error messages, provide an action so that users can solve the problem.</td>
</tr>
<tr>
<td>Static descriptive text at the top of windows</td>
<td>• Describe the overall action that users accomplish on the window if it’s not obvious.</td>
</tr>
<tr>
<td></td>
<td>• Clarify anything users must do before completing this window.</td>
</tr>
<tr>
<td></td>
<td>• Identify ramifications, if there are any, of the changes in this window.</td>
</tr>
<tr>
<td>Static descriptive text below fields</td>
<td>Examples for what to enter in a field.</td>
</tr>
<tr>
<td>Hover help</td>
<td>• Syntax for what to enter in a field.</td>
</tr>
<tr>
<td></td>
<td>• Ramifications of the field change.</td>
</tr>
<tr>
<td></td>
<td>• Descriptive information for what to enter in the field.</td>
</tr>
<tr>
<td></td>
<td>• Links to additional information if needed.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Easy to use</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Task orientation</td>
<td>In the context of a product, a focus on helping users do tasks that support their goals</td>
</tr>
<tr>
<td>Accuracy</td>
<td>Freedom from mistake or error; adherence to fact or truth</td>
</tr>
<tr>
<td>Completeness</td>
<td>The inclusion of all necessary parts—and only those parts</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Easy to understand</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Clarity</td>
<td>Freedom from ambiguity or obscurity; using language in such a way that users understand it the first time that they read it</td>
</tr>
<tr>
<td>Concreteness</td>
<td>The inclusion of appropriate examples, scenarios, similes, analogies, specific language, and graphics</td>
</tr>
<tr>
<td>Style</td>
<td>Correctness and appropriateness of writing conventions and of words and phrases</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Easy to find</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization</td>
<td>A coherent arrangement of parts that makes sense to the user</td>
</tr>
<tr>
<td>Retrievability</td>
<td>The presentation of information in a way that enables users to find specific items quickly and easily</td>
</tr>
<tr>
<td>Visual effectiveness</td>
<td>Attractiveness and enhanced meaning of information through the use of layout, illustrations, color, typography, icons, and other graphical devices</td>
</tr>
</tbody>
</table>

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
abstract language 256
accessibility
    screen captures 458
    alternative text for graphics 478
color and contrast 481
tables 482
accuracy
    automated tools for checking 93
    checklist 96
definition 14
example of in task topics 501
grammar-checking tools 94
guidelines 68
information consistency
    types of inconsistencies 88
    information reuse 86
information currency
    examples and samples 84, 242
    messages 81
    increasing content lifespan 82
    technical changes 79
link-checking tools 95
overview 67
research
    direct observation 69
    hands-on experience 69
    information plans 71
    interviews 73
    outlines 71
    rough drafts 71
    topic models 71
spell-checking tools 93
technical reviews
    choosing reviewers 76
    exit criteria 77
    focus for 76
    interface testing 74
    quality control tests 78
    technical ownership 77
user interfaces 80
verifying
    hands-on testing 74
    quality control tests 78
    technical reviews 76
active voice 263
adverbial conjunctions 286
Index

agile development process 12
ambiguity
  coordination 204
  empty words 183
  fragments 198
  keywords as plain text 182
  long noun phrases 204
  modifiers
    dangling 201
    misplaced 202
    squinting 203
  negative expressions 191
  noun phrases 204
  overview 180
  positive writing 189
  pronouns 199
    that and which 201
  vague referents 199
  syntax 194
  translation problems 188
  word as more than one part of speech 180
analogies 218, 253
animation 443
audience
  See also users
    cultural backgrounds 222
    international 494
    mixed 129
    primary 123
    secondary 123
    skill levels 222
    writing for intended 25
autocomplied values 9, 104

C
callouts in illustrations 473
can vs. may 188
capitalization
  technical terms 280
  translation problems 280
  user interfaces 281
characteristics, quality 13
checklists
  accuracy 96
  all quality characteristics 545
  clarity 212
  completeness 148
  concreteness 259
  organization 366, 376
  retrievability 420
  style 314
  task orientation 64
  visual effectiveness 483
circular statements 166
clarity
  ambiguity
    coordination 204
    empty words 183
    fragments 198
    guidelines 180
    keywords as plain text 182
    modifiers 201
    negative expressions 191
    noun phrases 204
    positive writing 189
    pronoun use 199
    syntax 194
    that and which 201
    vague referents 199
    word used as more than one part of speech 180
checklist 212
coherence
  coordinating ideas 174
digression 174
  subordinating ideas 175
  transition words 174

Developing Quality Technical Information
Index

organization 320
precise language 219
programmatic assistance 217, 223
realistic 241
samples 218
scenarios 217, 243, 251
similes 218
visual 218
embedded assistance 223
error messages 225, 237
examples
code 247
highlighting in 250
level of detail 221
overview 218
realistic 241
guidelines 219
information currency 216
labels 224
nonnative users 494
overview 215
programmatic assistance 217, 223
reference information 233, 513
samples
code 247
highlighting in 250
overview 218
realistic 241
scenarios
business scenarios 243
overview 217
realistic 241
task scenarios 243
similes 218, 253
specific language 256
task information 227
testing 524
troubleshooting information 235
usability testing 518
conditional steps 60
conjunctions 286
consistency
common expressions 306
factual 532
information reuse 86, 306
organizational 354, 532
spelling 276
stylistic 533
visual
design guidelines 465
layout 463
style 460
contextual information
definition 323
embedded assistance 332
organization 352
pattern 334
separating 324
window-level assistance 352
contrast and color 481
control-level assistance
concreteness elements 226
definition 107
planning for 107
unnecessary repetition 145
coordination conjunctions 286
coordinated clauses 157
cross-references
See links
cueing graphics 475
cultural bias 273
currency of information
examples and samples 242
increasing content lifespan 82
technical changes 79
customized documentation 130

D

dangling modifiers 201
dashes
See em dashes
See also hyphens
default values
definition 104
programmatic assistance 9, 223
design review meetings 70
detected values
  definition 104
  programmatic assistance 9
digression, coherence 174
disabled UI control 105
DITA
  definition 305
  tagging style 312
domain expertise
  concreteness 220
  definition 13
  example 130
drafts, rough 71

E
editing
  conceptual information 530
  embedded assistance 529
  organization 530
  overview 528
  preparation 528
  reference information 530
  summarizing findings 534
  task information 529
elliptical style 194
embedded assistance
  API names 7
  capabilities 107
  color 8
  command names 7
  concreteness 223
  control-level 107, 226
  definition 4
  editing for 529
  hardware 8
  help pane 6, 109
  hover help 6
  icon assistance 107
  illustration 6
  inconsistent 358
  input hints 6
interface labels 6
keyword names 7
messages 6, 7, 108, 225
multiple quality characteristics applied to 493
navigation 395
parameter names 7
programmatic assistance 9, 104, 223
redundancy 145
retrievability 493
task orientation 43
text labels 224
tool names 7
tooltips 6, 107
types 332
user interface text, static 107
utility names 7
wizards 6
embedded help pane 6
em dashes
  guidelines 292
  using for emphasis 290
empty words 183
entry points
  definition 413
  highlighting 414
  tables 416, 420
  user interface 417
error messages
  appropriate wording 192
  common expressions 308
  concreteness 237
  troubleshooting information 237
  users’ perspective 30
evaluation tests 520
examples
  code 247
  level of detail 221
  overview 218
  realistic 241
experience levels
  experienced 128
  mixed 129
  novice users 126, 128
expertise, domain 13
expletive constructions 163

F
factual consistency 532
feature-focused content 41, 45
field observation 523
fonts 429
fragments
  ambiguity 198
  list introductions 300
front-loading 382
function-oriented tasks 35

G
gender bias 273
gerund phrases 156
goals, user
  See user goals
grammar-checking tools 94
graphics
  accessibility 478
  cueing 475
illustrations
  big-picture 438
  consistency 460
  creating 439
  guidelines 439
  interactive 441
  scenarios 434
  spatial relationships 437
  task flows and processes 431
  visual effectiveness 431
screen captures
  appropriate uses 449
  design tips 458
  drawbacks 458
  example 451
  unnecessary 453
videos
  appropriate uses 444
  common expressions 307
  definition 444
tours 445
tutorials 447
white space 426

H
hands-on experience 69
headings
  See titles
help
  See contextual information
help pane
  capability 109
  embedded 6
high-level steps 49
highlighting
  examples 250
  entry points 414
  text 296
  user interfaces 296
hints
  See input hints
hover help
  focus on user tasks 43
  illustration 6
  inconsistent 88
  navigation 396
  redundant 145
  sample content 11
hyphens
  ambiguous noun phrases 204
  guidelines 292
  spelling 293

I
icons
  legibility 481
  user assistance 107
illustrations
   big-picture 438
density 387
   consistency 460
location 385
   creating 439
names 7
guidelines 439
proximity 387
   interactive 441
retrievability 384
   scenarios 434
stuffing 387
spatial relationships 437
visual effectiveness 431

image maps 441
list-checking tools 95
imprecise verbs 169
link-checking tools 95
informal tone 268
links
information, unnecessary 136
in-sentence (inline) links 403
information verification
   interface testing 74
labels
   quality control tests 78
   review exit criteria 77
   technical ownership 77
   technical reviews 76
index verification
indexes 388
information, unnecessary 136
inline (in-sentence) links 403
information plans 71
input hints
   examples 258
   illustration 6
   structure 334
input hints
in-sentence (inline) links 403
in-sentence (inline) links 403
instructions 133
interactive illustrations 441
interviews 73
introductions, organization 366
items in series, commas 288
J
Jargon
   terminology 209
   tone 268
K
keywords
   clarity 182
   density 387
   location 385
   names 7
   proximity 387
   retrievalability 384
   stuffing 387
L
labels
   as plain text 182
   concreteness 224
   embedded assistance 224
   illustration 6
   inconsistent 88
   notes 374
   sample content 11
   layout, consistent style 463
   legal boilerplate 88
   link-checking tools 95
links
   appropriateness 399
   definition 399
   essential 400
   in-sentence (inline) 403
   navigation 394
   redundant 405
   search engine optimization 400
   strategy 396
   wording 409
lists
   bulleted 55
   instructions 55
   length 300
   nesting 300
   parallelism 302
   punctuation 300
   unordered 55
low-level steps 49
M
machine translation
   clarity 188
Index

syntactic cues 195
main points  
emphasizing 366  
websites 369
markup languages 311
messages  
accuracy 81
capability 108
concreteness 225  
embedded assistance 6
error 237
illustration 108
inconsistent 88
logged 7
reusing 88
sample content 11
meta tags 388
minimalism 136
misplaced modifiers 202
model, topic  
See topic models
modifiers  
dangling 201
misplaced 202
squinting 203
unnecessary 167
modifying clauses 159
nominalizations 170
noncontextual information  
definition 323
separating 324
topics 337
topic sets 343
topic types 339
nonnative users  
ambiguity 180
clarity 494
concreteness 494
multiple quality characteristics applied to 500
style 494
notes  
in examples 250
labels 374
misusing 374
noun phrases, ambiguous 204
novice users 126, 129
O
observation, direct 69
only, ambiguous use of 202
optional steps 58
organization  
challenges 327
checklist 376
consistency 354, 532
contextual information  
definition 323
embedded assistance 332, 352
definition 14, 319
editing 530
embedded assistance 332, 334
guidelines 321
information elements 320
information delivery mechanisms 322
main points 366
navigation 345
noncontextual information 323
notes 374
overview 319
progressive disclosure pattern 334

N
navigation  
breadcrumb trail 394
consistent 355
embedded assistance 395
hover help 396
links 394
optimizing 389
organization 345, 364
paths 394
retrievability 394
negative expressions 190
negative prefixes 192
negative space 467
See white space

Developing Quality Technical Information
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Redundancy</td>
<td>326</td>
</tr>
<tr>
<td>Relationship of elements</td>
<td>360</td>
</tr>
<tr>
<td>Retrievability</td>
<td>357</td>
</tr>
<tr>
<td>Secondary points</td>
<td>371</td>
</tr>
<tr>
<td>Separation of information</td>
<td>322</td>
</tr>
<tr>
<td>Sequential</td>
<td>350</td>
</tr>
<tr>
<td>Step-by-step instructions</td>
<td>353</td>
</tr>
<tr>
<td>Textual elements</td>
<td>424</td>
</tr>
<tr>
<td>Topics</td>
<td>337</td>
</tr>
<tr>
<td>Topic sets</td>
<td>343</td>
</tr>
<tr>
<td>Topic types</td>
<td>339</td>
</tr>
<tr>
<td>Outdated content</td>
<td>82</td>
</tr>
<tr>
<td>Outlines</td>
<td></td>
</tr>
<tr>
<td>Completeness</td>
<td>115</td>
</tr>
<tr>
<td>Research</td>
<td>71</td>
</tr>
<tr>
<td>Overcompleteness</td>
<td>136</td>
</tr>
<tr>
<td>Positive writing</td>
<td>189</td>
</tr>
<tr>
<td>Precise language</td>
<td>219</td>
</tr>
<tr>
<td>Prefixes, negative</td>
<td>192</td>
</tr>
<tr>
<td>Prepositional phrases</td>
<td>160</td>
</tr>
<tr>
<td>Pretentious tone</td>
<td>270</td>
</tr>
<tr>
<td>Primary audience, definition</td>
<td>123</td>
</tr>
<tr>
<td>Procedures</td>
<td>133</td>
</tr>
<tr>
<td>Process, writing</td>
<td>15</td>
</tr>
<tr>
<td>Products</td>
<td></td>
</tr>
<tr>
<td>Features, avoid focus on</td>
<td>41</td>
</tr>
<tr>
<td>Scenarios</td>
<td>33</td>
</tr>
<tr>
<td>Programmatic assistance</td>
<td></td>
</tr>
<tr>
<td>Autocompleted values</td>
<td>9</td>
</tr>
<tr>
<td>Concreteness</td>
<td>217, 223</td>
</tr>
<tr>
<td>Default values</td>
<td>9</td>
</tr>
<tr>
<td>Definition</td>
<td>9</td>
</tr>
<tr>
<td>Detected values</td>
<td>9</td>
</tr>
<tr>
<td>Embedded</td>
<td>223</td>
</tr>
<tr>
<td>Examples</td>
<td>104</td>
</tr>
<tr>
<td>Illustration</td>
<td>9</td>
</tr>
<tr>
<td>Types</td>
<td>104</td>
</tr>
<tr>
<td>Progressive disclosure</td>
<td></td>
</tr>
<tr>
<td>Applying a pattern</td>
<td>107</td>
</tr>
<tr>
<td>Definition</td>
<td>10</td>
</tr>
<tr>
<td>Embedded assistance capabilities</td>
<td>107</td>
</tr>
<tr>
<td>Hover help example</td>
<td>113</td>
</tr>
<tr>
<td>Messages example</td>
<td>111</td>
</tr>
<tr>
<td>Multiple quality characteristics applied</td>
<td>488</td>
</tr>
<tr>
<td>Navigation aids</td>
<td>394</td>
</tr>
<tr>
<td>Sample pattern</td>
<td>11, 109, 334</td>
</tr>
<tr>
<td>Pronouns</td>
<td></td>
</tr>
<tr>
<td>That and which</td>
<td>201</td>
</tr>
<tr>
<td>Vague referents</td>
<td>199</td>
</tr>
<tr>
<td>Prototypes</td>
<td>70</td>
</tr>
<tr>
<td>Punctuation</td>
<td></td>
</tr>
<tr>
<td>Colons</td>
<td></td>
</tr>
<tr>
<td>Explaining ideas</td>
<td>290</td>
</tr>
<tr>
<td>Lists</td>
<td>289</td>
</tr>
<tr>
<td>Combinations of</td>
<td>294</td>
</tr>
<tr>
<td>Commas</td>
<td></td>
</tr>
<tr>
<td>Comma splices</td>
<td>287</td>
</tr>
<tr>
<td>Essential phrases</td>
<td>284</td>
</tr>
<tr>
<td>Grammar tools</td>
<td>94</td>
</tr>
</tbody>
</table>
Index

items in series 288
nonessential phrases 284
run-on sentences 287
serial 288
with conjunctions 286
with quotation marks 294
correct 284
em dashes 292
hyphens
  ambiguous noun phrases 204
guidelines 293
compound words 293
lists 300
semicolons 291

Q
quality characteristics
  checklist 545
definition 13
determining 534
groupings 14
international audience 494
progressively disclosed information 488
reviews 516
topic-based information 501
usability testing 518
verification 549
writing cycle 17
quality
  control tests 78
  ratings 534

R
redundancy
  embedded assistance 145
  helpful 141
  unnecessary 145
  user interfaces 164
  words 163
reference information
  multiple quality characteristics applied 509
organization 339
reusing common expressions 309
samples in 233
titles 382
topic types 339
referents, vague 199
relationships of topics, revealing 360
relative clauses, long sentences 159
repetition
  helpful 141
  main points 366
  reasons for 141
  unnecessary 145
research
  observation 69
  hands-on 69, 74
  interviews 73
retrievability
  checklist 420
  conceptual information 508
definition 14
description 379
editing for reference information 530
embedded assistance 493
entry points
  definition 413
  highlighting 414
tables 416
  user interface 417, 421
guidelines 380
links
  cross-references 411
  essential 400
in-sentence (inline) 403
redundant 405
wording 409
navigation aids 394
optimizing
  browsing 381
  indexes 388
  keywords 384
  searching 381
tables of contents 389
topic titles 381

Developing Quality Technical Information
organization 357
reference information 513
task topics 506
testing 525
usability testing 519
user interfaces 397
reusing information 86
reviews
conferring with writer 535
design 70
exit criteria 77
guidelines 517
problem reporting 517
quality characteristics 516
technical 76
visual elements
conferring with editor and writer 541
description 538
guidelines 536
individual 540
preparation 537
summarizing findings 541
rewriting for clarity 153
roundabout expressions 161

S
samples
code 247
concreteness element 218
currency 242
realistic 241
scenario-based information 32, 217
scenarios
business
definition 243
effective 245
conceptual information 232
context 251
definition 32, 243
description 217
illustrations 434
integration 33
product design 33
realistic 241
tasks 243
screen captures
accessibility 458
appropriate uses 449
currency 458
design tips 458
example 452
translation 458
unnecessary 453
usability testing 450
search engine optimization
keywords 387
links 400
titles 381
secondary audience, definition 123
secondary points
de-emphasize 371
user interfaces 373
semicolons, guidelines 291
sentences
fragments 198
negative 190
passive 170
sequential organization 350
serial commas 288
sexism 273
similes
conceptual information 230
description 218, 253
example 253
purpose 253
single sourcing 86
slang, tone 268
spatial relationships, illustrations 437
spelling
consistency 276
correctness 276
hyphens 293
technical terms 276
tools for checking 93
squinting modifiers 203
static UI text
affordance 107
Index

inconsistent 88
sample content 11
step-by-step instructions
lists 55
organization 353
progression 49
reusing 88
steps
clear action 51
conditional 58, 60
grouping 53
step levels 49
optional 58
passive voice 264
style
active voice 263
capitalization 280
checklist 314
common expressions 305, 306
conceptual information 508
consistency 460, 533
cultural bias 273
definition 14
gender bias 273
guidelines 262
highlighting text 296
nonnative users 494
overview 261
passive voice 263
punctuation
colons 289
combinations of 294
commas 284
consistency 284
em dashes 292
hyphens 293
semicolons 291
sexism 273
spelling 276
tagging 311
task topics 506
templates 305
testing 524
tone 267
subject-verb proximity 155
subordinate
clauses 175
conjunctions 287
surveys 523
syntactic cues
ambiguity 194
that 196
syntax, inconsistent 88
T
table of contents, optimizing 389
tables
accessibility 482
entry points 416, 420
guidelines 473
styles 470
visual effectiveness 468
tagging
DITA 312
markup languages 311
style 311
task information
concreteness 227
editing 529
examples in 227
illustrated 431
multiple quality characteristics applied to 501
task orientation
checklist 64
definition 14, 23
embedded assistance 43
guidelines 24
intended audience 25
overview 23
product features 41
purpose of information 46
step-by-step instructions
clear actions 51
conditional steps 58, 60
grouping 53
levels of steps 49
optional steps 58
Developing Quality Technical Information
Index

users’ goals 32
users’ point of view 27
task scenarios 243
task topics
elements 35
multiple quality characteristics applied to 501
titles 382
topic type 339
user-oriented 35
technical ownership 77
technical reviews 76
technical terms
consistency 207
defining new terms 210
jargon 209
spelling 275
technical writers
conferring with 535
responsibilities 12
role 11
templates
DITA tagging 305
role in visual effectiveness 421
style 305
terminology management system 205
testing
guidelines 524
hands on on page 74
information in test scripts 78
laboratory 519
prototyping 519
style 524
task orientation 524
test cases 525
tools 525
usability
evaluation tests 520
outside laboratory 522
organization 519
quality characteristics 518
remote 523
retrievability 519
screen captures 450
validation tests 521
user interface 526
visual effectiveness 525
text, highlighting 296
text labels
concreteness 224
entry points 417
textual elements
fonts 429
highlighting 296
organization 425
user interface 426
that
essential clauses 201
syntactic cue 196
titles
inconsistent 88
for topic types 382
tone
colloquialisms 268
formal 268
humor 268
idioms 260
informal 268
neutral 271
pretentious 270
slang 268
tools
embedded assistance 7
grammar-checking 94
link-checking 95
spell-checking 93
testing 525
tooltips
embedded assistance 6
inconsistent 88
topic models
example 115, 120, 135
publishing 135
research 71
topics
definition 337
illustration 338
keywords for 385
multiple quality characteristics applied to 501
active 169
nominalizations 170
passive 170
phrasal 173
imprecise 169
with syntactic cues 196
weak 169
verb-subject proximity 155
videos
  appropriate uses 444
  commonly used expressions 307
design tips 447
tours 445
tutorials 447
viewlets 444
visual effectiveness
  accessibility
    alternative text for graphics 478
color 481
contrast 481
tables 482
checklist 483
description 421
elements
  callouts 473
cueing graphics 475
tables 468
white space 467
graphics
  illustrations 431
screen captures 448
videos 444
guidelines 423
impact on other quality characteristics 423
style consistency
  design guidelines 465
illustration 460
layout 463
textual elements
  fonts 429
organization 425
user interface 426
usability testing 519
visual elements
  appropriateness 533
cconcreteness 218
reviewing
  guidelines 536
overview 538
types 422

W
waterfall development process, definition 11
websites
  main points 369
  secondary points 373
which 201
white space 467
window-level assistance 352
wizards
  navigation 365
  self-documenting user interface 101
wordiness
  expletive constructions 162
imprecise verbs 169
redundancies 163
roundabout expressions 161
unnecessary modifiers 167
words
  clear 187
  empty 186
  transition 174
writer
  See technical writer