



The User Manual Manual

**How to Research, Write, Test, Edit and
Produce a Software Manual**



Also available in the UnTechnical Press Books for Writers Series



UnTechnical Writing

How to Write About Technical Subjects and
Products So Anyone Can Understand

The User Manual Manual

How to Research, Write, Test, Edit and
Produce a Software Manual

by Michael Bremer



UnTechnical Press, Concord, CA

The User Manual Manual

How to Research, Write, Test, Edit and Produce a Software Manual

by Michael Bremer

This book is dedicated to Linda.

It could not have existed without support and inspiration from
Barbara, Jeff, Wendy and the "other Michael."

Special thanks to Richard, Tom, Kevin, Debbie, Laura, Bob
and a few others who know who they are.

Published by:

UnTechnical Press

P.O. Box 272896 • Concord, CA 94527

Copyright 1999, by Michael Bremer. All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage-and-retrieval system without written permission from the author, except for the inclusion of brief quotations in a review.

Publisher's Cataloging-in-Publication (Provided by Quality Books, Inc.)

Bremer, Michael, 1955-

The user manual manual : how to research,
write, test, edit and produce a software manual /
by Michael Bremer. -- 1st ed.

p. cm. -- (Untechnical Press books for writers)

Includes bibliographical references and index.

LCCN: 99-90430

ISBN: 0-9669949-1-4

1. Electronic data processing--Authorship.
2. Computers--Handbooks, manuals, etc.
3. Technical writing. I. Title.

QA76.165.B74 1999

808'.066004

QBI99-1307

Bullfrog Productions Ltd., Dungeon Keeper, QuickHelp, E-Help, Adobe Acrobat, Macintosh, Windows, Microsoft Windows, Apple Guide, HyperCard, Director, Nintendo 64, Zip, Sony PlayStation, Amiga, Atari ST, Acorn and XRES are trademarks or registered trademarks of their respective companies.

Contents

Foreword	9
Part 1—	
The Software User Manual	11
Introduction	12
About This Book	12
About Manuals	14
About Software Products	15
Writing	17
Technical and UnTechnical Writing	17
Manual Grammar and Style	18
Other Topics	19
All About Manuals	27
The State of the Manual	27
Purposes of the Manual	28
How People Read and Use Manuals	28
The Software Manual Dilemma	30
Manuals and Product Design	31
Manuals and Interface Design	31
Parts of a Manual	31
Manuals for Kids' Products	47
Manuals and After-market Books	48
Beyond the Words	49
Beyond the Manual	50
Online and Onscreen Help and Documentation	53
Companies and Their Systems	59
The Organization	59
The Development Team	61
Beyond the Development Team	68
Design Documents	69
Software Development Lifecycles	71
The Tools	78
The Cost of Changes	82
Special Topics	83
Big Projects with Multiple Writers	83
Rush Techniques	84
Manuals for Multiple-Platform Products	85
Writing for Console Machines	86
Localization and Internationalization	87
The Future of Manuals	89
Part 2—	
The Sample Project	91
Project Overview	92
The Project's Purpose	92
The Sample Software	92
Project Organization	93
Projects and Different Companies	93

The Scenario	94
Before You Begin	94
Project Flowchart	95
Stage 1: Learning the System—Setup and Planning	96
Establish the System	96
Establish Project Goals	101
Establish the Reader	104
Your Tools	106
Problems and Solutions	107
Complications and Opportunities	109
Stage 2: Learning the Product—Research	111
Research	111
Problems and Solutions	115
Complications and Opportunities	117
Stage 3: Organizing the Information—	
The Outline and the Structural Edit	118
Thinking and Preparing	118
Writing the Outline	119
Pre-edit Feedback	119
The Structural Edit	119
The Finished Outline	120
Problems and Solutions	125
Complications and Opportunities	125
Stage 4: Writing the Manual—The First Draft	126
Time to Write	126
Format and Presentation	127
The First Draft	128
Problems and Solutions	164
Complications and Opportunities	165
Stage 5: Validating the Information—	
Content Editing and Testing	166
Preparing the Draft for Content Editing	166
Testing the Tutorial	168
Tutorial Test Results	169
Mini Design Review	170
After the Edit	171
Problems and Solutions	171
Complications and Opportunities	172
Stage 6: Rewriting and Retesting	173
Summary of Changes	173
Rewriting	175
Numbering the Graphics	175
Retesting	176
Final Draft	176
Problems and Solutions	210
Complications and Opportunities	210
Stage 7: Detailed Editing—The Line Edit	212
Preparing for the Line Edit	212
Selected Sample Edits	213

Problems and Solutions	216
Complications and Opportunities	216
Stage 8: Preparing the Manual for Layout	218
What You'll Deliver	218
Preparing the Final Draft and Style Sheet	218
Preparing Graphics	221
Problems and Solutions	225
Complications and Opportunities	225
Stage 9: Making It Look Good—Layout and Final Edits	226
The Graphic Artist	226
Writers' Contributions to Layout: Editing, Index, Table of Contents	227
Final Edit(s)	229
Problems and Solutions	229
Complications and Opportunities	230
Stage 10: Printing and Final Production	231
Final Preparation	231
Blueline—The Last Line of Defense	231
The Payoff	232
Problems and Solutions	232
Complications and Opportunities	232
Stage 11: Producing the Addendum or Quick-Start Guide	234
A Project in Miniature	234
Problems and Solutions	235
Complications and Opportunities	235
Stage 12: Producing Onscreen Help	236
A Little Background	236
Timing	237
Extra Reading	237
Writing for the Screen	237
Onscreen Help for The Personal Newspaper	238
Problems and Solutions	238
Complications and Opportunities	239
Stage 13: Writing the Readme	240
The Last Word	240
Writing the Readme	241
Problems and Solutions	243
Complications and Opportunities	243
Stage 14: Cleaning Up—The Project Backup	244
Backup	244
Problems and Solutions	245
Complications and Opportunities	245
Stage 15: Looking Back—The Postmortem: Lessons Learned	246
The Postmortem	246
Problems and Solutions	247
Complications and Opportunities	247
The End	248
Part 3—	
Exhibits	249
New Employee/Onsite Contractor Checklist	250

8 The User Manual Manual

Parts of the Manual Checklists	251
Quick-Start Guide (or Addendum) Checklist	253
Key to Proofreaders' Marks	254
Preparing a Draft for Layout Information Sheet	255
About the Sample Project Documentation	256
Sample Project Proposal	259
Sample Project Initial Design Document	265
The Finished Sample Project Manual	280
Part 4—Appendices and Index	305
Recommended Reading	306
Contact Information	309
Author Bio	309
About UnTechnical Press	309
Index	310

Foreword

After 10 years of writing, editing, managing and producing software manuals, I have a few bits of wisdom I'd like to pass on to everyone in the software world. The first one is:

A lot of manuals stink.

Manuals have a bad reputation, often for good reason. The good ones don't get the credit they deserve because the bad ones get all the attention. The worst part of it is that writers get the blame, even when they are given inadequate time and support to do a good job.

This book will help writers produce better manuals. It explains how, what and when to write, plus how to organize and plan to get more done in less time. More than that, it helps writers understand the business and politics behind the project, so they can beg, borrow or steal the support they need.

The next bit of wisdom is:

The manual is an integral part of the interface.

Most writers understand this, but many others, including engineers, producers and various species of managers, including financial types, still don't have a clue about what a good manual is, or why it is worth the time and money to do it right.

Software is still in the process of evolving from something only a technical person could love into a mass-market commodity. But it still has a long way to go before it is as universally accepted and understood as a TV or a toaster. And good manuals are part of the solution.

This book will help writers improve products and the customer experience with the products by creating manuals that are worthy of—or better than—the products they are written about.

Next bit of wisdom:

Writers can and should be more than just “manual writers.”

Writers, even technical writers, are creative people. Writers are skilled communicators. Writers are organizers and presenters of information. Writers are valuable members of a software development team, above and beyond writing the manual.

In addition, technical writers in the software industry spend years dissecting, analyzing and describing user interfaces. They spend weeks on each project trying and retrying each and every menu, button, function and control. They

10 The User Manual Manual

learn about user interfaces whether they want to or not, and if they put in some effort, they can become interface experts—again a valuable resource for a development team.

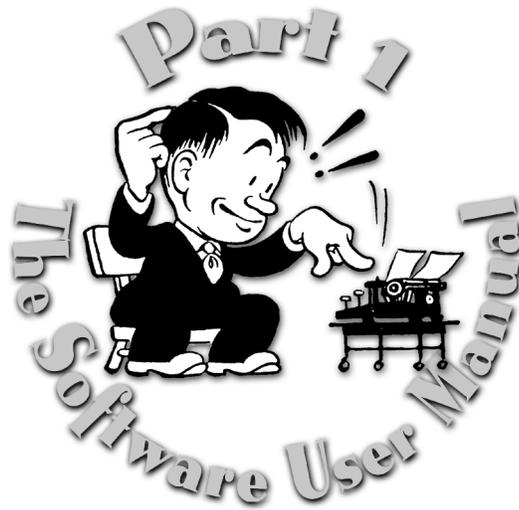
This book encourages writers to use their skills, gain more skills, ask the questions, help the project and expand their career options.

The last bit of wisdom is one that I recently had to swallow myself:

One book wasn't enough.

When I began writing the book *UnTechnical Writing*, I thought I could crystallize and preserve all my years of experience into a single book that would help all writers master all aspects of working in the technical world. But it wasn't enough.

That's why I wrote this book, one that focuses solely on the software manual, and that shows and explains every step of a sample project. Well, here it is. I hope it helps.



The chapters in this part of the book are:

- Introduction
- Writing
- All About Manuals
- Systems
- Miscellaneous Topics

The first part of this book gives a complete rundown on software user manuals, including processes, tools and techniques used in their creation, as well as their purpose, their composition and their readers. It also introduces software development teams and development methods, and how they affect you, your work and your manual.

Introduction

This short chapter quickly presents some basic background information that puts the rest of the book into perspective.

The sections are:

- About This Book
- About Manuals
- About Software Products

About This Book

Purpose

This book was written to:

- Teach new manual writers about the process of writing a software manual, the systems they'll be working within, the pitfalls they'll face and how to overcome them, and the possibilities and opportunities they can find if they know where to look.
- Give experienced manual writers the benefit of another writer's experience, and to offer well-tested processes and solutions to common problems, and suggestions for challenges.
- Get everything I learned from 10 years of writing manuals, editing manuals, managing and coaching writers and running a writing department within a software development company down on paper, and share it with all the writers I didn't have a chance to work with personally.

Organization

This book is organized into three parts:

Part 1—The Software User Manual explains the state, the art, and the state-of-the-art of manual creation. It covers everything from reader analysis to writing style and grammar to software development systems to online documentation.

Part 2—The Sample Project is a complete, step-by-step example of creating a software manual from day one through postmortem¹.

Part 3—Exhibits contains useful sample documents, checklists and worksheets,

¹A postmortem is a review session, verbal or written, that is held after a project is finished. It is a look back at the project's processes and results to summarize what worked and what didn't. The goal is to learn from the mistakes and successes of the past to make future projects better and smoother.

plus the sample project's proposal, design document and finished, laid-out manual.

Part 4—Appendices contains sources of useful information.

How to Use This Book

As an author who put a lot of work into this book, I'd love it if everyone who picks up this book reads every word, from the first to the last page. But realistically, I know that very few people read technical books that way. Everyone reads and absorbs information in their own way, so you're going to use this book any darn way you please.

But—just as a suggestion—if you're just starting to write manuals, I'd recommend that you read through Part 1 (The Software User Manual) and scan Part 2 (The Sample Project). When you're on a project of your own, read Part 2 carefully, a stage at a time, before you enter that stage.

If you're in a hurry to get started on a manual, focus on the subsection *Parts of the Manual* in Part I, then focus on the appropriate stages in Part 2 as you move through your project.

And feel free to use the exhibits as samples or templates whenever they can save you time and energy.

Assumptions and Methods

Some books on technical writing are actually courses in beginning writing. Some supply blank charts to fill in to help you write procedures. Some have whole chapters on active and passive writing.

If you need this kind of instruction, pick up a basic writing book—a few are listed at the end of this book in the Recommended Reading appendix.

This book isn't a beginning writing course. It assumes that you have basic writing skills. It touches briefly on style and grammar, but isn't filled with lessons on either subject.

This book teaches by supplying complete background information, extensive explanations and examples to analyze. More than that, it covers real-world subjects beyond the actual writing, including development systems, team dynamics and office politics, all of which have a major effect on your writing and your ability to get your job done.

The Sample Project

The project that this book follows all the way through is for a piece of software called *The Personal Newspaper*. It is a yet-to-be-developed product that was only halfway designed when this book was started.

It's perfect for this purpose because it is complex enough to demonstrate all the different aspects and stages of a manual's development, yet small enough that it's practical to include its multiple manual drafts in a book like this.

Design flaws and ambiguity have intentionally been introduced into the early stages of this project (that's right, I *meant* to do that), so the process of refining and finalizing a product can be shown in the sample project.

The early drafts of the sample project manual are based on the supplied initial design document. The journey from that to the final draft and final manual is documented step-by-step in the sample project.

A sample demonstration program that shows the screens and navigation of the program—but is not a functional model—is available for download from the publisher's website (www.untechnicalpress.com/downloads).

About Manuals

What's a User Manual?

User manuals are generally associated with software. Manuals for hardware products are more often called owner's manuals.

A manual of either type can consist of anything from a single diagram or paragraph on the product packaging to a one-page instruction sheet to a multi-hundred-page tome. It just has to explain what you can do with a product, and how to do it.

What Products Need User Manuals?

Unless a product is so completely clear and understandable that any proposed user can intuitively use it without more than a few seconds' pause to consider, it needs a manual of some sort.

There are very few software products today that are that clear.

And while many products don't really need a manual, some sort of documentation can still be useful. For instance, even something as simple as a toy ball could come with inflating and patching instructions, or contact information in case the ball is defective. And the legal department may want to include some disclaimers and protective warnings.

So just about every product could be accompanied by a manual of some sort, even if it's mostly legalese and incorporated into the packaging.

Manuals for Hardware Products

While the focus of this book is on manuals for software products, many of the processes and techniques presented here are useful for writing and producing manuals for hardware products. If you write about hardware, you'll get a lot out of this book.

But software, because of its fluid nature and constantly changing platforms, has a whole set of logistical problems, time constraints and complications for writers. Those problems and their solutions are integrated throughout this book, making this a far more valuable book for writers in the software world.

The Term "User"

I personally dislike using the term *user* to describe the person buying the product and reading the manual. After all, there's only one industry other than computers that refers to its customer base as users.

What options do we have? Operator? Too impersonal. Owner? While respectful enough, the term "owner" traditionally hasn't been used for software, probably because the people who buy software don't actually own it. They only buy the right to use it. It has to do with the nature of software and intellectual property. Read the license fine print on any package of software if you want to be further confused.

Whatever the future brings, the term *user* is universal enough today that I even used it in the title of this book.

About Software Products

What's a Product?

For the purposes of this book, products are what you write manuals about.

These products can be hardware or software (though this book concentrates on the special aspects of manuals about software products). Products can be for business or pleasure, utility or spectacle.

If there's no product, then there's no manual.

Product Complexity

Products can range from simple and straightforward to ridiculously complex. A well-designed product will seem far simpler on the outside than on the inside. Though it may be the most complicated piece of coding ever created by a team of programmers, it should appear clear and simple to the user.

As the writer of a user manual, you explain the interface—what the user sees. If it's a well-designed product, no matter how complex, writing the manual needn't be difficult or complex.

On the other hand, if a simple product isn't designed well, it can be a nightmare of confusion, inconsistencies and idiocy for you to explain and justify to the weary reader.

Open-Ended and Close-Ended Products

Something to be aware of when you write about software is whether the program is open- or close-ended.

A close-ended program is entirely predictable. It does a limited number of things. You can cover every possibility and contingency in the manual, and cover every operation in a tutorial. (You *can*, though you may not want or need to.)

An open-ended program is so flexible that you can't possibly cover every operation that the user might do with it.

A loan amortization program is close-ended. It has a limited number of operations. You can input certain things and you get certain outputs. There can be an unlimited number of different outputs, but there are a limited number of operations and processes.

A spreadsheet is open-ended. It can be set up and organized (and programmed) to serve an unlimited number of different purposes with unlimited variations.

A writer trying to explain how to use a spreadsheet can't possibly give every possible use or process, but you can:

- explain the overall premise of the program
- explain the basic interface functions (windows, menus, toolbars)
- define and explain each of the built-in functions and calculations, and
- give a few varying examples of how to use the program.

Similarly, in a chess program, you wouldn't be able to explain every possible strategy that can be used on a chessboard, but you can:

- explain the overall premise of the game
 - explain the interface
 - define how each of the pieces can be moved and the results of various moves, and
 - give a few sample strategies for different stages of a game.
-

Writing

This chapter gives a brief background in the basics of style and grammar you'll need to know when you write a manual. It also introduces the concept of *Un-Technical writing*.

The sections are:

- Technical and UnTechnical Writing
- Manual Grammar and Style
- Other Topics

Technical and UnTechnical Writing

All writing about technology isn't the same. The big difference is the audience: for whom it was written. Writing for the consumer audience—generally non-technical people living in our ever-more-technical world—is very different in process and practice from writing for highly-skilled technical people.

We at UnTechnical Press use the term *UnTechnical writing* to describe technical writing that is meant to be read by a nontechnical consumer audience. This kind of writing requires a little more diligence, a little more testing and a lot more patience, understanding and reader advocacy.

The sample project in this book is intended to be used by nontechnical customers, so many of the strategies and techniques of UnTechnical writing are incorporated throughout the book.

A few of the premises of UnTechnical writing are:

- Your job is to get the information into the reader's brain as quickly, easily and enjoyably as possible. The world won't end if someone actually enjoys reading your manual, as long as they gain all the necessary knowledge to use the product.
 - Your main function is to communicate, not to write a manual. If you can get the message across quicker and more easily using another method of communication, then use it. This could range from comics to wall charts to improved interface design.
 - Understand your audience. A lot of people are new to, uncomfortable with or afraid of technology. (You're writing about technology. It doesn't matter what the product or program is, if it's on a computer, it's technology.)
 - Write for people, not for professors. Don't make people feel like they're back in school. A lot of people didn't—and don't—like school. People bought
-

the product you're writing about to use the product, not to go back to school. If your manual makes them feel like they're back in the classroom, they won't read it.

- Be the reader's advocate.
- You are part of the user interface. You and your writing are a layer of the user interface. The more you know about user interface design, the better you'll do your job.
- Test your work, especially tutorials and instructions of any sort, with real, typical customers.

For much more on this subject and the necessary tools, skills, and processes needed to write about technology for normal people, see *UnTechnical Writing—How to Write About Technical Subjects and Products So Anyone Can Understand*,² available at better bookstores (physical and virtual) everywhere.

Manual Grammar and Style

For technical writing, and even more so for UnTechnical writing, grammar is important, but less important than communication. Writing can be correct without sounding academic or stuffy.

If you want people to read your manual (and you do), don't make them feel that they're in a classroom.

For instance:

- You can sound conversational.
- You can use *they* as a singular pronoun (sometimes) to avoid sexist language.
- You can use slang.
- You can end sentences with prepositions.
- You can even get away with starting sentences with *and* or *but*.

Of course, some things must be absolutely correct, including:

- Spelling,
- Proper word usage, and
- Punctuation.

There are few hard-and-fast rules any more. If you're working on an entertainment product, you have more leeway than if you're writing about project-management software, but even for strictly business products, it is better to have the manual read than collect dust on a shelf.

² Yes, a shameless plug.

What you can and can't get away with—or make use of—will depend on the company you're working for, your local editor and your boss.

Other Topics

Here are a few other helpful topics to know about when writing a manual.

Choosing Your Role

When you receive information for evaluation or absorption, you consider the source. When someone reads a manual, they also consider the source. Who is it that's talking (writing) to them? Is it someone who knows what they're talking about? Is it someone who's really trying to help them understand? Is it someone who "got stuck" with writing a manual, and only slightly prefers writing to getting a root canal? Is it a jerk who can't explain his way out of a paper bag?

When people read your writing and decide how seriously to take it and how much effort to put into understanding it, they take the source—you—into account. They don't know you, and most likely there's not a picture of the manual writer on the back of the book. But, whether you're aware of it or not, you are performing a role and are being defined—even if it's only in the reader's mind.

As long as you're being labeled, you might as well choose your own label. Choose your role.

It's a subtle thing. You usually shouldn't be obvious about it (unless it's a kid's game and you're writing from the point of view of a character in the program). But just knowing who you are (or want to be) to the reader can help flavor your writing. Choosing the role is something you do to help yourself; it's a tool for setting and controlling the document's tone, level of technology and rate of delivery.

As you think about how you'll tackle a project, try on these roles for size, and see what fits you and the project. If none of these roles works for you, find something that does. The idea here is to get into the right frame of mind to communicate with your audience at the right technical level.

- **The Tech Writer**—you are a technical professional explaining technology to other technical professionals. This writing is simple, to-the-point, unadorned and pretty much devoid of personality. This is the standard role that tech writers usually take. It is the best role to write from if you are writing for a highly technical audience. The rest of the roles in this list are better to take on if you are writing for a nontechnical, consumer audience.
- **Writer as Translator**—you're the link between a world that speaks Technese and people that don't know the lingo. Your job is to translate technical words,

concepts and processes into normal language.

- **Writer as Host**—you welcome your guests to a new, scary, complicated place. Make them feel at home. Treat them with respect. Talk to them in words that they understand and feel comfortable with. Not only the words, but also the format, feel and media contribute to the comfort factor as well.
- **Writer as Friend**—you’re trying to make your friends feel comfortable while helping them master this new experience.
- **Writer as Neighborhood Nerd**—you’re the guy who knows and loves technology, and you’re the one everyone on the block comes to when they have technical questions. You know the deeper inner workings of the technical universe, but know most of your neighbors don’t care about that, so you reassure them, then tell them what button to push to get the result they want (tutorial) and give them more information if and when they ask (reference).³
- **Writer as Teacher**—not the cranky teachers you hated, but your favorite teacher that lit that spark in you and made you want to learn.
- **Writer as Intrepid Explorer**—you’ve been down this dangerous road before, and are documenting your discovery process for the readers, allowing them to follow and see the sights, yet avoid the pitfalls. This is especially useful for explanations of very complex systems.
- **Writer as Tour Guide**—you’re guiding a group of tourists through a dangerous, foreign land. You’ve got to watch out for them, even protect them, while showing them a good time.

Non-sexist Language

Think about your audience. If it’s mostly men, then you can use the pronoun “he” freely and exclusively. If it’s mostly women, then you can use only “she.” If, as usual, it’s a mixed audience, then you have the potential to annoy them by choosing one gender or the other. While we proponents of UnTechnical writing may want to surprise or even shock a reader now and then, we don’t want to annoy.

When you find yourself in the pronoun dilemma, you have a number of choices:

- **Use only “he.”**

Using “he” as a pronoun to refer to an individual who may be either male or female (but you don’t know for sure) is the traditional English usage,

³ All those billion-dollar computer companies should set aside some sort of retirement fund for Neighborhood Nerds. Judging from the complexity of early computers and the incomprehensibility of many early computer manuals, if it hadn’t been for those noble Neighborhood Nerds simply explaining what button to push to get the desired result, computers would never have caught on.

dating from before political correctness. It isn't technically wrong, but it can make some readers think you're a jerk and pay less attention to you.

- **Use only “she.”**

On the opposite end of the scale, you can take the attitude that women have had to put up with being called “he” for a long time, so why not turn the tables and let the men see what it's like. While fair, and technically legal, it can still alienate a good portion of your audience, both male and female.

- **Alternate between “he” and “she.”**

This is the “fair” approach. Approximately every other time you use a pronoun to refer to a person whose sex you don't know, switch between “he” and “she.” While it's fair, legal, and generally acceptable, it is also noticeable. The reader is very aware that the writer is switching back and forth. This can be distracting; the reader stops thinking about the subject and thinks about the switch.

- **Use “he or she.”**

If a writer wants to cover all the bases without switching, he or she can use “he or she” each and every time he or she has to write “he” or “she.” Again, legal. But there's something ... wishy-washy about it. And it's longer and a bit more awkward. You can usually get away with it once in a while without it seeming awkward, but continual use makes it stand out too much, and readers once again notice the writing instead of concentrating on the subject.

- **Pluralize and use “they.”**

A very good way around the whole issue is to take advantage of the fact that in English, the plural pronoun “they” works for both sexes. Instead of writing something like:

If the reader likes the subject, then he or she should read the book.

You can write:

If readers like the subject, then they should read the book.

At least 80% of the time you can pluralize and use “they” to avoid sexist—and awkward—language without ruining your point. In those few cases where pluralizing just doesn't work, then you can fall back on one of the other options, or just use “they.”

- **Just use “they.”**

Many editors and grammar mavens are now allowing the use of “they” as an any-gender singular pronoun for technical, if not for academic, writing. A reader may find that they are referred to as “they” even though they are one person. Most readers won’t notice, and some editors are still sticklers. I use it over “he and she,” but not very often.

- **Get cute.**

A final possibility is to use some sort of combination of he and she, such as “he/she” or “s/he.”⁴ These work, but they seem like the writer is trying too hard. Maybe, if and when a combination like this comes into more common usage, it’ll lose its cuteness.

Consistency

There are a lot of ways to be consistent—and inconsistent.

- **Names of Things**

Don’t call it a doohickey in one chapter and a whatsit in another. Make sure the name you use in the manual matches the name on the actual product, and on the box.

- **Abbreviations**

When you use them, use them the same way throughout the whole document or series of documents. Consistently define them the first time they’re presented.

- **Level of Detail**

If you’re covering a number of similar subjects, they should all be covered in the same amount of detail—unless there is a reason that the reader needs to know more about a particular subject, in which case that should be made clear.

- **Look and Feel**

This includes the page-to-page and chapter-to-chapter look of a single document, including typography, tables and illustrations, and the book-to-book look of a series of related documents, including typography, overall design, and table of contents and index organization.

- **Presentation of Information**

Like information should be presented in a like manner. Always show the reader instructions within tutorials the same way. Always explain similar things in the same way—if you show a flowchart in one process, then show it for all processes.

⁴I’ve heard it said that even the combination word “s/he,” which means “she or he,” is showing undue prejudice against non-sexual entities, and the proper combination word should be “s/he/it.” Of course, it would never work in polite conversation.

- **Lists**

Use parallel style and construction. Note that the headings on this list of ways to be consistent consist of all nouns or noun phrases. If this item's heading was "Be consistent with lists," it would be inconsistent with the rest of the list.

- **Overall Quality**

Don't make half your manual wonderful and slap the rest together. And if you're doing a quick cheapie, don't make part of it wonderful—it'll make the rest seem worse than it is.

The Rule of Threes

The standard rule of threes is:

Tell them what you're going to say, say it, then tell them what you said.

For each chapter, topic, section or segment, you first explain what will be contained in that chapter, topic, section or segment. Next, you present the body of information. Then, you summarize or review it at the end of the chapter, topic, section or segment.

While many manual writers and editors adhere strictly to this rule, you may prefer to (and be allowed to) play fast and loose with it. You should almost always stand by this rule in a tutorial, but for introductions, reference sections and other parts of the manual, you may choose to use it or not depending on the technical difficulty of the subject. The more difficult the subject is for the reader to understand and master, the more you should stand by the rule—and the more often you should use it within a document, breaking the subject matter into smaller and smaller chunks of information presented in threes.

If the subject is light, and easily mastered by the reader, you may want to avoid the repetition and imposed format to either simplify and shorten that part of the document, or use the saved space to add flavor and entertain a little.

All in all, you really can't go *too* far wrong by adhering to the rule of threes. At worst, you might miss an opportunity to simplify and entertain.

Keeping the Reader Reading

If nobody reads the manual, we still get paid. But most writers—even technical writers—want to be read. And it's not just a matter of ego. It's wanting the work we do to be useful, to be worth the doing.

Most people don't read technical books and manuals for pleasure. They only do it because they have to. And they're looking for any excuse to stop, to take a break, to get a cup of coffee or to throw the book across the room in frustration. Anything we can do to keep readers reading is good. Anything we can do to help them get through it quickly and painlessly, we should.

Here are some helpful tools to keep readers reading:

• **Write in Bite-Sized Chunks**

Keep sentences, sections and paragraphs small—write in bite-sized chunks. This forces you to be clear, to keep your explanations simple, and stops you from rambling on too much. And it keeps the reader reading longer.

A small, simple-looking paragraph smiles at the reader and says, “I’m quick and easy. Come on and read me, you’ll be done before you know it.” A large, densely written paragraph says, “I dare you to read me, and to try to wrest meaning from the dark, twisty caverns of my many long sentences.”

It is especially important to keep the paragraphs at the beginnings of each chapter or section small and inviting. If they’re inviting enough, the reader will put off stopping until the end of the next chapter or section.

You should also present the writing in bite-sized chunks, even in early drafts. Make it as easy as possible for the people who have to read those drafts, and you’ll get better feedback.

• **Use Lots of Headings and Subheadings**

Using a lot of headings and subheadings serves four purposes:

1. It keeps the book divided into small, easy-to-absorb chunks.
2. It allows the reader to zero in on the exact section they want to read without fishing around.
3. It familiarizes readers with all the sections—as they look for the topic they want, they read all the other headings. When they need to look up something else, they’ll often remember that there was a heading about that subject.
4. It makes it easier for you to organize the book and explain each subject independently.

• **Advocate a High-Quality Layout**

The quality of the layout can also help keep the reader reading. Lots of white space on the page and a good space between paragraphs makes the page look inviting. A page crammed full of solid text looks scary. Do what you can to ensure that the layout is done by a good graphic artist.

• **A Little Humor Goes a Long Way**

By this I mean both that some humor is good and that too much is bad.

A little humor, spread thinly throughout a manual, is a good way to get readers to at least page through the whole book once. Too much—or inappropriate—humor and you lose the point of the whole book.

Humor diffuses a tense situation. If you can laugh at it, you're no longer afraid of it. Many readers are afraid of the technology you're writing about and afraid of breaking it and afraid of how it will change their lives and afraid of having to read your manual. They need a good laugh. If you can get them to laugh at something, they'll no longer fear it.

That said, here are the inevitable caveats...

Humor Caveat #1: Know when not to use it.

Don't force jokes about something that you personally can't comfortably joke about. Even more important is how the reader, and your client or employer feel about the jokes. Some people, including bosses, have no humor. Know when to keep your humor to yourself.

Humor Caveat #2: Test your jokes.

Try out your jokes on some trusted—and honest—friends and family members before putting them into a manual. If the general response is bad, kill the joke—even though you love it.

Humor Caveat #3: Ask for help.

If you don't have—I won't say good, but how about—a widely accepted sense of humor, get someone else to supply a few jokes or comics. There's bound to be someone around the house or office that can help.

"Men will confess to treason, murder, arson, false teeth, or a wig. How many of them will own up to a lack of humour?"

— Frank Moore Colby

Humor Caveat #4: Watch your mouth.

Unless the product you're working on is intended for adults with an adolescent sense of humor, stay away from insults and bathroom humor. Again, it's important to know your audience to know your limits, but when in doubt, go the tasteful route.

Humor Caveat #5: Humor is often a cultural thing.

Much of humor is culturally based, so what's funny to you may not be funny to someone in Japan or Germany or France. Some subjects that we casually joke about here may be seriously taboo elsewhere. If you know that what you're writing will be translated to other languages, be prepared for your humor to be chopped out. Make sure this chopping won't affect the usefulness of the document.

Your Duties—Above and Beyond Writing

Of course, your obvious duty is to write the manual. That's what you're being paid to do. But there's more to it. All of these, and more, can come into play, conflict, and compete for attention as you work on a project:

- Duty to self—your desire to do the best work you can, to write a masterpiece, to create art and to promote your career.
 - Duty to employer—your need to satisfy the boss for continued or future employment, *and* have a positive impact on the bottom line. A positive impact on the bottom line includes minimizing costs to the company by controlling the time and cost of writing, as well as the size and cost of documents and their related shipping costs. It also involves the cost of after-sales support (eliminating as many tech support calls as possible), and improving the opportunity for future sales of updated versions and other company products. (If a customer buys a product and doesn't understand it or can't use it, they won't buy another from the same company.)
 - Duty to team—whether you've been on the project since the beginning or just joined weeks before shipping, the design/development team has been working on the project for months or years. They deserve docs that “do right” by the product: show it in a good light, make it more useful and promote it. (And mention all their names—properly spelled—in the credits!)
 - Duty to product—often the product itself, whether spreadsheet, computer game, VCR or whatever, is such a good piece of work that it deserves to be given a good chance to succeed. And that means it should be well-documented, so it can be used and enjoyed as quickly and easily as possible.
 - Duty to customer—marketing will identify with and represent the customers up to the point where the product is sold, but it's up to you to be there for them once they get home with it. Beyond their absolute *needs*, what does the customer *deserve*? What would you demand, expect, or want if you just brought the product home or to the office and opened the box? What would satisfy you as a customer? What would satisfy your mother (assuming your mother isn't a technological wizard)?
-