

# **Who Should Write the Documentation?**

## **The Case for Scriptorium Publishing**

**Scriptorium Publishing Services, Inc.**

P.O. Box 12761, Research Triangle Park, NC 27709

919-481-2701 or sales@scriptorium.com

<http://www.scriptorium.com>



At Scriptorium Publishing, we make a living by providing technical documentation services. To new clients, especially small software start-up companies, this is often a tough sell. They ask, “Why should we hire you? The programmers here already know the product—they designed it. What value can you add to our documentation that they can’t?”

Our response to such questions is this: calculate your company's *total support cost*.

## **Total support cost**

Your total support cost is the cost of delivering information about your product to customers—including expenses for running a help desk, improving interface usability, and developing online help and documentation. It's important to consider all of these components of the product because their quality affects the total support cost.

When a software company delivers a product with a confusing interface, no user's guide, and sketchy online help, they realize an initial, short-term cost reduction because of the minimal expenses incurred to develop the interface and the supporting documentation. But the result is skyrocketing technical support costs. To get detailed information about the product, customers contact the company and expect to have their questions answered. If the problems are serious enough, they can lead to unflattering software reviews and reduced sales.

Another company might invest in improving the software's interface, delivering solid documentation and online help, and creating other support systems, such as wizards and interface labels.<sup>1</sup> All of this supporting information reduces users' confusion—and the company's technical support costs.

The choice is yours. You can develop documentation and deliver it to all your customers at once, or you can handle questions from your customers one-on-one. A typical documentation development effort will cost about as much as paying the salary of a single technical support employee for a year. If you can reduce your technical support staff requirements by just one person, your documentation effort will pay for itself.

---

1. Interface labels, sometimes called ToolTips, are small text explanations that pop up when you place the cursor over an item in a window.

## Specialized skills

We hope you're convinced that your product needs documentation of one form or another. Next, you might ask, "Who should write this documentation?" Small software companies are often tempted to add this task to their programmers' responsibilities.

We believe that this is a poor allocation of resources. Programmers are trained in their field—they design databases, write code, and create interfaces. Most of them are not trained as professional writers, who have honed some specialized skills, including the following:

- Document architecture
- Audience analysis
- Absorbing new software

### Document architecture

When a programmer begins a new project, it's usually by sketching out, or at least visualizing, the overall design of the project. The development team then focuses on each feature that the software needs and assembles the various piece of code into an integrated software package.

Similarly, professional technical writers begin by analyzing the software. "What does the end user need to know?" is the first question. Writers then break up the software into functional pieces and explain these pieces to their audience.

In a user's guide, readers prefer task-based descriptions that show them in a sequential manner how to perform particular tasks, and these explanations should be organized in a way that makes sense when someone is actually using the application.

In a reference guide, readers want information that is easily accessible. This means strict attention to organization (alphabetical?) and to information access methods (table of contents and index). Reference guides are also often good candidates for online delivery, which offers hyperlinking and powerful search features.

Good programmers know intuitively what will and won't work in their code. Similarly, technical writers have intuition about the best way to document a piece of software. This intuition comes from experience; it's unfair to expect programmers to have the same level of understanding about technical documentation as they do about software code.

Programmers are adept at writing code, designing databases, and creating interfaces. Writers are skilled in learning new software, designing the architecture of a document, and establishing a manual's audience.

Combining the skills of writers and programmers ensures high-quality documentation. Saddling programmers with the task of writing manuals does not.

Programmers often view software as a collection of windows or a series of menu choices because this is how the software is assembled. But end users rarely want a menu-by-menu listing of features. Instead, they want an explanation of how to accomplish a particular task.

## Audience analysis

A professional technical writer is adept at writing for a target audience. A document written for a software developer needs little or no explanation of basic operating system commands; a document written for a computer novice might need to explain what a mouse is.

Furthermore, some types of information are important for one audience but not another. For example, a user's guide probably includes minimal information about changing a password on a system; a system administrator's guide includes information about how users change passwords, how to control passwords globally, and perhaps some recommendations for password expiration policies. Putting the system administrator information into the user's guide would be inappropriate, but it does need to be delivered to the proper audience.

## Absorbing new software

Technical writers are professional learners. We spend most of our time with brand new, alpha-level software, learning how to make it work (and not work).

Scriptorium Publishing's writers and editors do nothing but learn and document software packages. Consequently, our team is skilled at approaching software as new users will approach it. Along the way, we've learned some tricks on how to break down the software into tasks. And our experience helps us ask the right questions about conceptual material and pre- and postprocessing requirements that might trip up new users. By using Scriptorium Publishing, the expertise of all parties is used to best advantage: we know how to ask the right questions and organize the material, and the software engineers serve as our technical experts on the software, thereby ensuring the accuracy of the documentation we produce.

*Without professional writers, you risk...  
Poorly organized documents... Problems with mechanics, such as grammar, cliches, and wandering sentences...  
Unattractive layout...*

## Bring in the pros

For these and other reasons, we strongly recommend that you hire professional writers. Your investment pays off in several ways:

- Documentation and online help are finished more quickly.
- It's easier for a professional writer to analyze the software objectively.

- Writers are trained to analyze audience requirements.
- High-quality online help and manuals reduce your other support costs.
- Programmers are free to concentrate on their code.

## Your advantage: Scriptorium Publishing

At Scriptorium Publishing, we hire writers who are highly adaptable. Our writers have broad experience with software; we've worked with databases, medical software, imaging and workflow, call centers, programming tools, and more. It's likely that we've seen something in your industry before.

Our writers are experts at determining the kind of information that is appropriate for a particular group of users. Relying on our years of experience, we establish the level of conceptual material necessary to give users enough context to optimize their use of an application. As we work with the software we are documenting, we determine when a task will be obvious to users and when it may require more explanation. And we provide information in an unintimidating, clear fashion, no matter how complex the software may be.

How can your software company ensure that your documentation is of the highest quality?

1. Let the experts write your documentation. Hire Scriptorium Publishing. We have the best writers and editors around. You wouldn't let just anyone write your code, would you?
2. Get us involved in your project as early as possible. We can create documentation quickly. But if you involve us earlier in your process, we can understand software more thoroughly and offer interface design evaluation if you want it.

## Contacting us

If you have any questions about Scriptorium Publishing Services, Inc., contact:

Scriptorium Publishing Services, Inc.  
P.O. Box 12761  
Research Triangle Park, NC 27709  
919-481-2701  
[sales@scriptorium.com](mailto:sales@scriptorium.com)  
<http://www.scriptorium.com>