

Working with a Professional Writer to Create Technical Documentation

Judy Bogart

You have a new software tool project, and you know the finished product is going to need good documentation. What does that mean, and what's it going to take? When should you get started working on the documentation, and how much of your team's time will be taken up by it?

This article gives you an idea of what to do and what to expect from your professional technical writer as you create and execute your project plan during the phases of your product development.

- *Part I: Designing documentation into your software project*

What to expect as you begin planning for a new project; how to incorporate a writer into your team and a documentation plan into your product design.

- *Part II: Developing excellent software documentation*

What to do and what to expect as the product and documentation begin to take shape; how to make sure your engineering team works with the writer to ensure that your product documentation is complete and accurate.



Part I: Designing Documentation into your Software Product

So you are developing a product, and realize you need some documentation to go with it. It's time to hire a professional writer to get things started. Here's how to work with a writer during the design and pilot implementation stage of software engineering.

When you decide to engage a professional software writer, you need to let your writer know what your product is—what it's supposed to do and who is supposed to use it. You should be far enough into the design cycle to give your writer a general picture of what the user needs to know.

A professional software writer is something like your intended user: a knowledgeable and experienced software professional, but one who isn't necessarily familiar with your particular area of expertise. Explaining the product to the writer can help you clarify your thinking on exactly who can use your product—that is, what an intended user should already be expected to know, and how your product fulfills that user's needs.

The Benefits of Working with a Professional Writer

You can see the benefits from working with a professional writer right from the start. In general, the sooner you get a writer involved, the better chance they have of helping you spot holes or fuzzy places in your design.

With your participation, a writer can provide a clear, high-level overview of the product and intended audience that helps you discuss it with your own management, and other people outside your group who don't understand your code words and specialized terms.

- The writer can create an up-to-date written version of the design. This enhances communication within the software team, and exposes design problems, while they're still relatively easy to fix.
- At this stage, the writer can also help you agree on fixed terminology. A glossary helps avoid confusion and further clarifies both internal and external communications.

Document the Design, not the Bugs

When you get your writer involved early in the project, keep in mind that your documentation is going to describe what

This article is part of a series of articles on technical writing that Expert Support hosts on its website, expertsupport.com. Expert Support is located in Los Altos, California and supplies contract technical writers to the computer and software industry.

the product is intended to do—not what it currently does. Although the documentation process helps you understand your own product better and clarifies the design and usage, it's not the same as QA. It isn't useful for the writer to figure out and carefully describe bugs and temporary workarounds as the code evolves—but it can be useful to create overviews and placeholders for features or systems that are not yet in place, to see how they fit in with those parts that are already implemented.

First, the Writer Needs to Understand the Product

In these early stages, the writer should meet with the project manager, and possibly with the designers who have a feel for the overall shape and direction of the product. The goal is to help the writer understand what you're doing and why. Your writer should have a good, solid background in software engineering in general, but won't be a specialist on your product or its technology. They need to get enough context to understand what your users already know, and what is new and needs to be explained.

Next, a Doc Plan

The next goal should be a documentation plan. A documentation plan describes the individual documents the product requires, and the specific purpose and delivery medium of each document. A documentation plan also specifies the resources you need to create and verify the documentation set.

Working with the writer, you will decide on a working style and delivery plan:

- **Deliverable format:** Will the docs be delivered as PDF? HTML? Online? Printable?
- **Tools:** What software will you use for development?
- **Sources:** Will the reference doc be generated from comments in source code? If your documentation requires editing source code, do you want the writer to have their own branch, or access to the main line?
- **Communication:** How will the writer and team share information?
- **Review:** How will the review cycle work?

Resources you Need to Provide

Working with a writer takes up your time and the time of the engineers. The enhanced communication and improved design should make up for that time with improved efficiency; however, as the team manager, you need to allocate engineering resources to the documentation task.

- Allocate time for meetings as the writer learns about your product and you iron out the details of the doc plan and development/delivery options.
- Arrange for your writer to have access in the source code control system to the relevant code and source documents.

This might include creating accounts and arranging server access as well as file access.

- If the writer has their own branch of source code, assign someone to merge it with the main line and decide how often to do this.
- Provide a place to store and archive both the document source files and the delivery files.
- Decide how often you want status reports, in what detail, and where you want them (email? wiki?).
- Make sure that you are prepared to inform your writer of important changes to the plans or schedules, such as a change to the ship date or feature set.



Real Benefits

By getting a professional software writer involved in your project from the beginning, you'll see a set of design documents that:

- Make your product better by clarifying the design and usage.
- Help you describe and explain your project to your own managers and business partners.
- Help your team communicate their actions and requirements with each other.

Your writer will have helped you develop a specific and realistic documentation plan, so you can confidently build documentation into your development schedule. And, perhaps most important, you will now have a writer who understands your product and has a relationship with your engineering team, who is ready to produce top-quality documentation that is appropriate to your particular project and release cycle.

Part II: Developing Excellent Software Documentation

OK, now your project is in the homestretch, and you want to work with your writer to finalize the documentation so it's ready to go out with the product.

You've hired a professional writer, oriented them to your project, and created a doc plan. You also have a set of design documents, and a writer who understands your product and has a relationship with your engineering team. It's time to start writing the final software and the product documentation.

What should you expect from the development process? What does your team have to do, and how much time should you allocate in your project schedule for documentation tasks?

The Doc Set

Most APIs require a Reference Manual (or web pages) with a detailed description of the API features and functions, and a User Guide, with usage guidelines and examples. There might be a Tutorial. You need a README, created just before release, that provides information on bugs—"known problems." The doc set might also include Installation Instructions. After the first release, you need Release Notes to describe what is changed or added.

In creating a doc plan, you decide whether all of these are needed, and which ones need to be separate documents. For a small project or an early release, you might want to pack everything into one document (usually the information is still divided up along these lines). For a second release (or an initial release after a beta or pre-release), you might expand the material into additional documents. As the project progresses, you and your writer refine details of the doc plan, such as exactly what goes where and how everything fits together and is presented.

When to Expect your New Documents

A writer doesn't churn out a fixed number of pages each day; documents take shape slowly over the course of the project. As a rule of thumb, the first 25% of writer time is spent producing 5% of the pages, the next 50% produces about 35% of the pages, and the last 25% produces the final 60% of the pages.

It takes a while for the writer to become familiar with the product and plan the document set before you see actual drafts. The first drafts show the evolving structure of a document, with preliminary rough content or placeholders for specific information. Later drafts have more and more specific information, which the writer and engineers refine through a review process. About halfway through your schedule, you probably start seeing drafts that are useful to your own team and to testers, partners, and pre-release customers.

Even before you have draft documentation to hand out, however, you'll see that the documentation process itself benefits your project and product quality. Your writer is

probably the first person outside your development team to look carefully at both the overall design and specific details of your product—with an eye to the end-user tasks and experience. This means that you're getting useful feedback about the usability, consistency, and completeness of your user interface and public API.

In some sense, your writer is like a really involved and articulate beta customer—and one you can depend on to tell you about their experience before it's too late!

Resources for the Initial Development Phase

Here's some idea of the resources you need to provide and the time you need to allocate during the first phases of doc development.

For the Reference

Typically, you start with the reference material, because that's what the engineers have been producing for their own use. Depending on the type of project, your writer can start with specs and header files, and other internal documents that you have been using for communication within the team and with QA and partners.

As the writer begins to turn this internal material into external documentation, they usually have questions; this means they need access to the engineers who are responsible for the product. Usually email access is enough, but they might need to schedule meetings, too. You must make sure you have room in the development schedule for the time your team spends in considering doc questions and providing answers.

For the User Guide

For the User Guide, your writer can typically start with use-case scenarios and sample code that the engineers have already produced, perhaps for the use of QA. This material inevitably brings up questions about intended usage and best practices. These questions might require engineers to spend time on the phone or in meetings, and more time developing better samples.

Given reasonably good samples, a professional software writer can abstract out the techniques and provide non-code explanations. Your writer can probably suggest which techniques need better illustration, and can usually help make sample code more readable and instructive. Ultimately, however, your team has to make all the decisions about the right way to use your product.

It might take some time to get the use cases right, but it's time well invested. If the documentation doesn't show the right way to use the software, users invent their own ways of using it. These unplanned usage styles can cause big headaches down the road for your support team, and ultimately have an impact on the ease with which you can upgrade your software.

The Review Process

As the docs progress, they need review by as many of the team members as possible. Usually, the writer produces a PDF that everyone can comment on. If you prefer (and if it is available), you can use a review tracker.

When you allocate resources for document review, remember that throughout the schedule, everyone should be able to spend a few hours a week looking at or discussing documentation, and they should understand that this is an important task.

As the docs near completion, several people need to read all of the docs carefully and thoughtfully to make sure they are correct and complete (that is, they address all of the issues that need to be explained and discussed). This is an iterative process; engineers make comments, the writer attempts to address them, the engineer checks the result and makes more comments, and so on. This means that you can't simply allot a week of doc review near release. Doc review needs to be a continuing task.

It's a good idea to set expectations within the team for an expected turnaround time for reviews and questions. It's generally more practical to set a short turnaround time and put aside other tasks for a day or two to get the review out. When you give people a long window of time, they tend to keep putting the review on the back burner, and often never get to it at all.

Releasing Docs

As you near release, you should plan for the docs to be fully reviewed and ready a week prior to document production tasks. What these tasks are, and how much time they take, depends on how docs are released, and such factors as whether it is a major or minor release.

Resources you Need to Provide and Allocate to Prepare for the Release

- If your document requires review by your legal department, you should set this up well in advance. We all know that getting things through legal can be a very slow and frustrating process!
- If your document requires artwork or other attention from the product management team or marketing department, make sure you know who to talk to and when to expect their input.

When the content is complete, the writer has production tasks to perform such as a final spell check, making sure that copyright usage is correct, and making sure the document looks exactly right when printed or built. While the main documents are in production, you can turn your attention to the ReadMe and Release Notes. Typically, the ReadMe is a simple text file that can be prepared at the very last minute, and edited by anyone if the need arises.

- One of the final tasks in the release cycle is for someone (often QA) to collect open bugs that are not showstoppers but need to be mentioned in the Release Notes. While

preparing the Release Notes, the writer edits the bug descriptions for clarity and appropriate language, and might have questions.

The Manpower Crunch

A few key members of the team frequently end up with the bulk of the documentation responsibility, either because they know the most about the product, or because they are the best communicators, or both. This can be a scheduling problem, because these are often the very people who are working hardest to resolve bugs and design issues as the project nears completion. However, the documentation itself is a powerful force multiplier for these key members. Even a preliminary document can do a lot to ease the load on key people by freeing them up from constantly having to explain what they are doing to other members of the team or to partners and testers.

Good Docs make a Big Difference

Have you ever encountered a really powerful software product that might do exactly what you need, if you could just make it work? Or one that is just frustrating to use, or has a reputation for a long learning curve? Have you had to deal with software that comes bundled with an expert? You can make sure your product doesn't go that route by providing really good documentation. Quality documentation gives software a real competitive advantage by cutting sales time, reducing support costs, and making customers successful.

Planning and preparation at the beginning of your project; a skilled writer; and the attention of key people in your development team: these are the ingredients that add up to clear, accurate, usable documentation. Quality documentation tells your users what they need to know, when they need to know it. It can mean the difference between usable software, and ... well, not.

You have invested in top-of-the-line software engineers, because you intend to build a really good product. Now protect that investment by adding a top-of-the-line professional software writer to make sure your product is actually used, and develops the reputation it deserves.

Judy Bogart is a Senior Technical Writer at Expert Support.