Effective User Manuals

Originally published in the April, 2000 issue of the Smart Access newsletter from Pinnacle Publishing (https://vb123.com/effective-user-manuals/).

Effective User Manuals

Peter Vogel

A user manual can reduce your maintenance costs and time on the phone—if it's a good one. Here's how to create a user manual that will actually get read.

Without a user manual, your users have to call you to find out how your application works. Without a user manual, your users will probably end up making errors that you will have to clean up. Yet, while a user manual should make your client's life easier, most manuals aren't read. The reason is that most user manuals aren't very good. Creating an effective manual isn't hard to do, though, provided you follow some simple rules. I'll begin by outlining the steps that you need to follow to create an effective user manual, and then drill down to the detail of how to lay out a page.

To my mind, if you're not willing to commit to following a defined process in creating your user manual, you're actually further ahead not to create a user manual at all. You can invest the time and money saved in other tasks. Before you consider dropping the manual, however, do bear one thing in mind: Your clients regard a user manual as a value-added item (even if it they never use it). When customers don't receive a user manual, they feel short changed. It also lowers the level of your professionalism and the quality of your work in their eyes. As a result, producing a user manual may be unavoidable. If you are going to invest the time and money, you might as well do it right and get some return on your investment.

The process

Creating a user manual does not consist of sitting down and starting to write. Just like developing an application that actually meets your customers' needs, there is a process that you can use to create a user manual that works. This process, among other benefits, defines a structure for early involvement by the end user. Since the final judge of the user manual is the end user, early involvement is essential for gaining user acceptance of the final product.

Like designing an application, the process for creating a user manual begins with a specification. The specification needs to answer these questions:

- Who is your manual aimed at? Your manual may be designed for new computer users or target sophisticated developers.

- What do you want your manual to accomplish? Is it to show how to use the basic functions of your application or will it provide in-depth understanding of the hidden features of the application?
- What is the manual's range? Is your manual designed just to explain how to use the application or how your application fits into the whole business process?
- Where does the manual fit? Is it designed as a standalone document or is it a part of a package that includes quick reference cards and a training program?
- What can you assume? Is the documentation aimed at an audience that is part of your organization and can be assumed to share its goals, mores, and a common set of reference points? Or is your documentation aimed at an external audience that will have very little in common with each other (and with you)?
- What is the organizational support for the manual? How will the manual be updated? How will its effectiveness be tested? What review process is in place?

When your specification is complete and you know what your documentation has to accomplish, you can move on to the next steps in the process. After a specification is drawn up, there are four phases that the documentation will go through: initial design, prototyping, construction, and delivery.

In the initial design phase, you decide how your manual is going to be laid out. Using the specification as your guide, you can decide what the topic headings will be, what graphics are required, and what style, tone, and vocabulary are required. At this point, you can start making time and cost estimates.

In the prototyping phase, you create mock-ups of the pages and sections of your manual. With mockups created you can let members of the manual's target audiences perform a "walk through" and tell you if the manual's organization matches their needs. Sample sections can be written out in detail and given to users as part of the application's testing process. The key question is always, "Given the manual, can the user perform the required task?" Walk-throughs consist of giving the user a task and seeing if they can find the relevant information in your manual.

The construction phase involves the hard work of grinding out the text. If the early work has been done well, the construction phase will produce a manual that meets your users' needs. Delivery includes the printing and distribution of the manual (which may, for instance, be given out as part of a training class).

Manual organization
The key to success in the design phase stage is to have a task orientation. Users pick up a manual expecting it to tell them how to accomplish whichever task they are performing right now. Your manual must meet that need.

A task oriented approach is not the one taken by the typical programmer developing a manual for a new application. The typical programmer will lay out the manual to match the organization of the program. Ideally, the structure of your application is hidden from your users, so a user manual organized around program structure will seem arbitrary to your users.

More enlightened programmers use a format similar to the reference manuals for the languages and tools that they are familiar with. However, the specification for a manual aimed at an Access developer will be very different than one aimed at sales clerks, with resulting differences in the manual design. Even a design that reflects the menu structure of an application may not be appropriate. Instead, it’s important to identify what tasks the end user will be using and develop the help manual around those tasks.

At the highest level, a task orientation breaks the manual into sections that reflect the main autonomous areas in the users work. Within a section, a user should find all the topics for a specific area of work and not have to refer to topics in other sections. Each topic should be organized around one idea, “How to perform this task,” and use verbs to identify each topic: “Creating an order,” “Updating an existing customer,” “Deleting a credit.”

In a user’s manual for an order entry system, I divided the user manual into sections on creating orders, changing orders, managing customers, reviewing status, and getting reports. This reflected the tasks performed by different groups in the company: Sales staff needed to be able to create orders, but changing orders was the responsibility of customer service reps, for instance. Both sales staff and customer service reps needed to be able to change customer information, while managers used the system’s reports.

Manual sections don’t have to be divided by job description. Often stages in the business process will define the tasks and the resulting sections in the manual. The manual for one system I worked on was used almost exclusively by the company’s purchasing staff. However, the manual still divided naturally into sections on creating orders, expediting orders, and reviewing status. While it was the same person performing all three tasks, users would first create an order and only days later need to expedite it. The review process was ongoing activity, independent of the other two processes.

At the highest level, a task orientation controls the sections and topics for your manual. At the next level of detail, a task orientation means that each paragraph in the manual should begin with the reason that the user wants to read it. Most paragraphs should begin with “To accomplish this….”: “To change the customer address…”, “To eliminate this message…”, “To have the system automatically process incoming mail…”. At the lowest level, a task orientation means that vocabulary should always reflect the terms used in performing the task. Rather than write about the “order entry form,” you can simply say “the order.”

When a task involves several steps, a number of techniques have been developed for handling sequences of instructions. My preference is for the PlayScript style. The PlayScript technique breaks instructions into a series of steps to be performed by the user. For each action, the PlayScript includes the feedback the user will receive. A sample PlayScript, describing how to create an order in an order entry system, would look like this:

1. Select File | New	Order Displays
2. Enter customer number	Message appears if customer not found
3. Enter order information	Message appears if products not found
4. Click on Date button	Calendar appears
5. Select delivery date	Calendar disappears
6. Click on Save button	Order is cleared

Page format

Ideally, each task will occupy one page. I say this because, if a topic fits on one page, the user doesn't have to turn pages to get all the information that they need. More than three pages on a topic is almost certainly too many.

Each topic should be accompanied by a graphic that develops the material for readers who learn best from pictures. Graphics should include screen shots that reflect what the actual screen will look like. By referring to the screenshots in the body of the manual, you provide a way for users to move from the word to the graphic to the action on the screen. Block graphics showing how individual steps fit together give users a sense of how all the parts fit together. If the topic fits on a single page, the facing page can include the relevant graphics.

What you leave out is as important as what you put in. It's not necessary to list every action to be taken as a series of mind-numbing simple steps. Instead, a "minimalist design" focuses on four key ideas:

- Secondary material is left out (overviews, introductions, etc.)
- The focus is on what users can use right now to be productive
- Exploration of the application is encouraged (details are left out; the manual reflects a "try this…" attitude)
- The relationship between the text and the computer screen is simple

In laying out the page, don't fill the page with text—leave room for the user to make notes. I leave a third of my page for the outside margin. The only things that I put in that space is subheadings that identify "hot spots" within a topic. Once the user has found the topic, they can scan down the margin to find the particular information that they need and write any comments right beside my text.

If you are minimizing the secondary material, including introductions and overviews, there is a real danger that your readers will get lost in your manual. As a result, you need to make the organization of the manual obvious. I use a "context box" in the upper left hand corner of the page at the start of each topic to show readers where they are and where they're going (if they're reading sequentially). The box shows the current topic in bold face along with the section the topic is part of and where the topic falls in the section. Readers can use the context block to find related items. If there's room, I include other section headings, so that readers can see the full range of the application. As the name implies, this block provides readers with a context that shows them where this particular topic fits in the scheme of things. Using this article as an example, a context block for this section might look like this:

Why User Manuals Are Important
The process
Organizing the manual
Laying out the page
    Using graphics
    What to put in
    Keeping readers from getting lost
Resources

All of the techniques that I've described in this article depend upon each other. For instance, in the PlayScript that I used above, a user may enter a bad customer number. The PlayScript simply says that an error message appears, since a minimalist approach to the user manual doesn't list everything. The PlayScript, for instance, doesn't mention that a "Create Customer" button is included on the error message's dialog. When the user clicks on that button, they will be taken to the Customer Update form. If the user was reading along with the PlayScript, a look at the context block might show "Adding a customer" as a topic heading or "Managing Customers" as a section in the manual that they can refer to. If the user wasn't using the manual but now needs to, they could pull the manual off the shelf and look for the "Adding a customer" topic in the table of contents.

Resources

You're not alone when you create your manual. In addition to the direction and feedback that you can get from your end users, there are a number of excellent books on creating user manuals available. The book that I've relied on for a number of years is "Writing Better Computer User Documentation" by R. John Brockmann from John Wiley & Sons. The book has a (at best) a workman like design, but the information is invaluable, and the structure is an excellent example of what a good user manual should be. The book describes a variety of styles and techniques and reports on a great deal of the available research.

Creating an effective user manual doesn't have to be a painful activity, nor is your manual doomed to be ignored by your users. If you clearly identify who your manual is aimed at and what you want to accomplish with it, if you build in early involvement from your users, if you maintain a task orientation, and organize your pages to present the information completely, then you will have a successful manual. And, when the phone calls drop off, you might even get some coding done.

Peter Vogel (MBA, MCSD) is a principal in PH&V Information Services. PH&V specializes in system design and development for COM/COM+ based systems. Peter has designed, built, and installed intranet and component-based systems for Bayer AG, Exxon, Christie Digital, and the Canadian Imperial Bank of Commerce. He is also the editor of the Smart Access and XML Developer newsletters, and wrote The Visual Basic Object and Component Handbook (Prentice Hall). Peter teaches for Learning Tree International, wrote their Web application development and Technical Writing courses ,and is the technical editor for their COM+ course. His technical articles have appeared in every major magazine devoted to VB based development and in the Microsoft Developer Network libraries. His business-oriented articles have appeared in Contract Professional and Datamation magazines. Peter also presents at conferences in North America and Europe. peter.vogel@phvis.com.