

Let's stop writing documentation and start working for the users

Nearly 20 years ago, the profession of technical communication began to focus on developing task-oriented documentation. Although task-oriented documentation has always been produced, particularly for consumer products, it was not the standard in the computer industry. More often, people writing about computer systems focused on the system rather than on the tasks people needed to perform. Systems-oriented documentation was the norm.

Systems-oriented documentation, originally written by the systems developers themselves, explained how the system was designed to operate. It began as information to be passed on to maintenance programmers who were responsible for making changes in the original program. It ensured that the original system design would be clearly understood by other technical experts. It even provided a means for the original programmers to remain familiar with the details of their own work.

Systems documentation of this sort serves an important role in computer product development. It is a necessary component of the development process. In fact, when most users were computer professionals, ensuring that mainframe programs were up and running, it was even appropriate for their use. Unfortunately, computer companies decided that this information was also sufficient for end users of their software. They often simply attached a new cover and shipped it off to unsuspecting computer novices. The novices, of course, were completely lost and quickly learned to rely on others in the workplace for information about what to do to use the software.

During the 70s and early 80s, as the number of end users of software began to increase through the introduction of terminals and standalone special-purpose machines, computer software companies hired technical writers, and we began to invent a new field of writing for the computer industry and for the new and growing number of end users. Unfortunately, the industry for which we worked had no tradition of writing for end users. Most of the initial user documents were modeled on the systems documentation produced during the software development life cycle. I remember many early projects in which we wrote documentation by directly reading the Cobol code to discover and communicate the details of data entry and reporting. Everyone agreed that this was what end users needed.

The shift to task-oriented documentation began about the time the PC was introduced. Those of us developing processes and teaching them to the rapidly growing group of technical communicators linked task-oriented information with the concept of "getting closer to the development life cycle." We urged new writers in the computer industry and in other fields to focus on users tasks at the same time they worked to become fully integrated into the development process. We wanted to ensure that the documentation was viewed as part of the product, rather than "a

necessary evil.” We were combating the point of view of developers and managers who couldn’t believe that anyone really needed instructions to use their obviously intuitive systems. We also began to hear feedback from customer service representatives that people were calling them in increasing numbers to understand how to use the systems and solve problems with the systems. These people often reported that they had not looked into the documentation. And those who did were often frustrated that they could either not understand what was being explained, or the information they were really looking for was not there. The more novice they were, the more problems they had.

Task-oriented documentation was touted as the solution. We could write procedures for end users who needed help performing standard tasks with the software (or other products). We could work closely with the programmers to understand all the functionality being included in the software, and we could explain to people “how to” use it. The end result has been the current state of documentation. Most of it explains how to perform all the tasks associated with the software in minute detail. The documentation often includes all the possible ways of performing all the tasks, even when the writers themselves cannot fathom why anyone would want to perform some of them.

For 20 years I have been urging technical communicators to get closer to the developers and to write task-oriented documentation. Today, looking at what we are actually producing and how poorly it seems to be serving the needs of our users, I’ve changed my mind. I believe we must stop writing documentation as we now know it, abandon the developers and their need to explain how everything works, and go to work for the users.

Going to work for the users means turning our product-oriented world up-side down. We need to get closer to the users, spending time in their workplaces, watching them struggle with everyday tasks, listening to how they describe their activities, and generally understanding their environment from their points of view. We need to seriously apply techniques such as contextual inquiry, ethnographic studies, workflow analyses, and others to ensure we are getting a clear understanding of the users’ world.

In addition to this level of user interaction, we need to participate in the design, aware of the users’ experience with the product. By understanding the users’ perspective, we can contribute significantly to the design of the software interface, adding elements that support effective user performance. We need to become skilled in developing performance support systems, including innovative interfaces, wizards and coaches, embedded self-learning opportunities, much improved help systems, and more. We need to engage in continuous communication with our user community, rather than being satisfied with putting online help into the product or a book into the box. We need to take advantage of what we learn from the users

about using our own products and maintain a dialog with them through the life of the product.

This is not to say that we do not need to understand the functionality that the products provide. We will always need to associate with the software development process. However, we must stop asking the programmers to tell us what the users need to know or how they need to know it.

Technical communication needs to move en masse to a new understanding of our roles and responsibilities. I invite you to challenge your own assumptions and those of others in your organization and begin to make the change today.

JoAnn Hackos, Ph.D., is director of the Center for Information-Development Management and President of Comtech Services, Inc., an information design and human productivity firm, headquartered in Denver, Colorado. Dr. Hackos is a leading expert in technical communication processes and management, addressing audiences worldwide about the need for improved interaction between users and products. She is a Fellow of the Society for Technical Communication, having served as STC President in 1992-93. Her published work includes three books from John Wiley & Sons: *Managing your Documentation Projects*, *Standards for Online Communication*, and *User and Task Analysis for Interface Design*.