TECHNICAL WRITING AS A SPECIALITY

by

Merri Boylan
Manager, Documentation Services
Computation Center
University of Chicago
5737 South University Avenue
Chicago, Illinois 60637
(312)753-8424

ABSTRACT

Good technical writing can enhance the quality of service to users by not only providing readable documentation, but by providing a different point of view in design and debugging of programs and systems that are being documented.

When I started in computing almost ten years ago, technical writing was a field that people fell into -- sometimes because they were interested in computers, but not all that interested, or because they were an English major and couldn't get a job, or because they were clerks at a computer center and someone needed to give them promotions. Training consisted largely of telling the novice writer to "go talk to the guy who programmed it." This haphazard means of making technical writers made for very odd technical writing.

We have slowly advanced from that stage, but careful attention to technical writing is too often considered an unnecessary burden to the computer installation. All that manpower just writing? Programmers can do it in their spare time. Sure. Why not?

I'll tell you why not. By the very nature of computing, its capacities and potential are opaque to those who do not know it intimately. Plain speaking about computing becomes a greater and greater necessity as computers become a part of the lives of a larger and larger percentage of the people. The first step to

that goal is professional communicators who know and are willing to divulge the "secrets" of the computer. In addition, when we look at documentation as the major intermediary between the computer industry and anyone who attempts and make use of it, our evaluation of the importance of good documentation may undergo a change.

Therefore, instead of this haphazard approach, we need to look at exactly what it is that you want a technical writer to be, and figure out exactly what it is you need to do to get that kind of person.

A technical writer as is a sort of computer general practitioner -- someone that understands the general system and languages and has actually done some programming. But this is a person who is more interested in communicating than in producing code.

A technical writer is not just someone who writes down what some programmer tells him or her that the program does. For a thorough technical writer, that is only the first step to preparing a technical document. The succeeding steps are that the technical writer sits down and tries the program, finds out if it works as the programmer described, and tries things users will probably try. Thus, the technical writer becomes not just a writer but a debugger and a user advocate. The technical writer should be able to say "this doesn't work right," "this works inconsistently with the rest of the program -- you should change it" and have a thorough enough understanding of the system and of the kinds of functions that the programmer is doing to be able to make this kind of statement.

On the other hand, the technical writer should have enough training to know that programs are finite. In order for a technical writer to able to respect what it is that a programmer does and the difficulty in that task, he or she has got to know something about the difficulties of programming: that it is not necessarily straightforward to make a change if that

change has technical implications for the rest of the program. There are times when an external change would look like a good thing, but it would mean a total redesign of the program. If we are going to talk about a technical writer as a debugger, and I think that that is an essential part of the technical writer's function, the writer must be aware of technical limitations as well as technical possibilities. Porblems in this area are sometimes more in attitude than in training. A technical writer who is a frustrated programmer may alienate the programming staff he or she is writing for by saying or implying that the programs being produced just are not good enough, and should be redesigned.

How do you get a person who has the potential to be a technical writer as I have defined it? And once you find that person, what do you do to make him or her the technical writer that you want? First of all I believe that the most important thing is for a person that wants to be a technical writer is to like to write and have an interest -- not necessarily a background, but a real interest -- in technical material. The writer's background should include something that is technical in some way. For example, although you don't normally think of psychology as a really technical field, there are enough intricacies in that field that the technical nature of computers will not be that foreign to a person trained in psychology. Another example: I had a interviewee that was a English major who serviced her Volkswagen herself. That's the sort of person that you are looking for -- someone that has a strong background in writing and editing but is interested enough in the technical kinds of information that they will pursue that information rather than expecting it all come to them by some magic means.

Within your own university or installation you may be able to find people that fit the description of the technical writer -- people that are in positions where you might not necessarily look, but who are looking for an advancement of some sort. For example, a secretary who has a certain amount of background in writing and who started using a text editing system and picked things up really quickly. That person might be a very good candidate for technical writing. Or a machine operator! What better candidate when it comes to background? I hired a machine operator who I noticed was always reading is his spare time. So I said, "Gee, he reads a lot, I wonder if he can write?" Sure enough, he could. And he's been working for me for 2 years now.

Once you find such people, what exactly is it that you want to do with them? What do they need to know? Let me use a somewhat odd analogy here.... What is it that someone needs to know to write about any-

thing? for example, a murder mystery writer? Does that murder mystery writer have to have murdered someone in order to write convincing murder mysteries? Well, I hope not. There are various posssibilities here. For example, there are murder mystery writers that have been detectives so they know it from that point of view. There are probably murder mysteries writers that fantasize about murders and know it from that point of view. I think the quality we are looking for in a tech writer is that of the detective. He or she does not have to know a program before starting to write about it. The writer has to learn to look for clues about whether something is working right or not. And in order to do so, the writer must have a firm understanding of the entire system starting with the operating system -- not to the detail of how control blocks interrelate, but concepts like how a batch job gets through the system, or how the online systems function. The writer needs enough of a background in the "computer mentality" (analogous to the "criminal mentality") to be able to put the pieces of a program together into a coherent document.

How do we go about training this would-be tech writer? Obviously, I'm making a very strong requirement for technical knowledge for a technical writer. How can you have this person start writing if you're not going to allow any technical naivete? Writing assignments that you would give to a technical writer who is just starting out are novice user type documents, things the writer can experiment and play around with. The document itself should be continuously monitored and edited by a technically knowledgable writer, or a programmer who is interested in documentation. Careful editing is the essential ingredient in this part of training.

As his first large assignment, one of my writers was asked to write an introductory document on partitioned data sets (PDS's). His major use of PDS's was in WYLBUR (text editor and RJE system), because that was the part of the system that he used the most. So he assumed that this was everyone else's major use. He didn't know a thing about load libraries or procedure libraries, where PDS's are essential on our IBM system. By careful editing of that document, the mistakes and misimpressions were corrected, and he was able to discover other parts of the system that he had not considered before. We ended up with a good document, too.

Every document needs to be edited no matter who writes it. But by taking special care in your editing, a beginning technical writer can be immediately productive for you, and at the same time be learning from his or her mistakes. A technical writer needs to possess personal

initiative and an interest in technical matters. So part of that special care is not to totally rewrite every part of the new person's text, but rather to point out the errors and tell him or her where to find the correct information.

This description of how you start a technical writer out is meant to demonstrate the general attitude that I believe fosters good technical writing. In such a short paper, I can give only a clue as to what I think the components are of a good technical writing staff. At the basis is my belief that technical writing as an activity can often show up problems in programs and systems, and that therefore technical writers who are technically trained can sometimes find flaws that go unnoticed by the programmers. Paying close attention to the quality of people you hire as technical writers, and encouraging those people to become technical experts in their own right so that they become really integrated members of your professional staff will add a quality to your service to users that only comes from thorough investigation of problems to be solved and thorough implementation of those solutions.

I believe that the industry needs to understand that technical writers can no longer be just glamorizers and grammarizers of documents -- that they need to be technical experts in their own right. That it is a different thing to be a systems analyst or a programmer, and it is yet another thing to be a technical writer. But a technical writer must be technical.