Audience identification for end user documentation

by JANIS G. RAYMOND

Michigan Wisconsin Pipe Line Company
Houston, Texas

ABSTRACT

Audience identification in writing is comparable to the analysis phase of systems development. Understanding who the users will be, what their business function is and how it relates to the computer system, and what the users need to know is the only way to write effective user documentation. Identifying the users determines the type and amount of detail to include as well as the format, tone, and level of the documentation. Audience identification is the planning phase of the documentation and as such controls the entire writing process. This paper presents guidelines for identifying end users and directing the documentation to their needs.
INTRODUCTION

Companies are just starting to recognize the importance of effective end user documentation. This recognition is based on two tangible benefits:

1. Good user documentation protects the investment in development costs by providing an understanding of what a system can do and how the system can be used.
2. Good user documentation makes the system more efficient by providing the information required for using the system.

With companies scrutinizing system development for a return on investment, it becomes imperative that each step in the development process be evaluated.

End user documentation is the final product, the last deliverable, in the system development process. Without documentation, a system is useless. That is a harsh statement, but true. Systems are developed to help users in their jobs. Without users there would be no system requirements and therefore no systems. Thus, the only way to make a system a profitable investment is to make the system usable. Effective end user documentation is the answer. And the first step in producing effective end user documentation is identifying the audience. Establishing who the users are and what their needs are controls the entire documentation process.

WHY IDENTIFYING THE USER IS IMPORTANT

Identifying the user before you begin writing ensures that the documentation meets the users' needs. Understanding who the users are, what business function the system performs, and what the users know and need to know allows you to write documentation that will help the users.

Typically, users are less knowledgeable about computers than the person writing the documentation. Since the purpose of user documentation is to describe the system functions in terms of the users' jobs, the writer must step away from computerese and present the information in terms that are relevant to the users. This may mean more work up front in the writing process, but ultimately it will save both the writer and the user time and frustration.

Users often see computers simply as a means to an end. They don't want to be burdened with endless computer jargon in trying to decide how to request a report. Nor do they want to wade through pages of paper looking for a simple instruction. Thus the language used and the organization of the documentation should be two primary considerations.

When daily use of a system seems more a problem than an aid, users lose interest in the system and become advocates of the "I can do it easier myself" school of thought. Documentation written from the users' perspective, in terms users understand, is the only way to make the documentation effective and the system worth the development time and costs.

WHO IS THE USER?

For such a simple question, the answer is seldom straightforward. It would be nice if the writer could quickly identify the user as Person X. Furthermore, if you knew the position Person X held in which department, and if you knew that Person X was responsible for Business Function A, you would have a relatively good understanding of the user.

Unfortunately, identifying the user is rarely that easy. Typically, the user is a department or a group of departments in the organization. To have a good understanding of the business functions involved, you must first find out how this department fits into the overall organization. Is there a central user department, but are there also decentralized departments in other areas of the company? If so, it is important to determine the needs of each of these departments to decide whether the same documentation will serve each group.

Once you understand what department or departments you are dealing with, you must become familiar with the internal structure of each group. You must determine who in the department will be using the system. Will only the clerical staff be preparing input data? Or will engineers and geologists as well as the management staff be involved?

It is also important to establish whether there is sensitive information that should not be included in the documentation. Additionally, find out whether authority is required before certain functions of the system can be performed. If either of these situations exist, you may need to have limited distribution of certain sections of the final documentation. Often the final user manual will be sets of documentation that when combined document the entire system.

Understanding these aspects is your first step in identifying the user. This knowledge gives you the broadest concept of who the user is and ensures that the documentation both includes and excludes the appropriate information. This knowledge also gives you the first clue to the organization of the overall user manual.

WHAT DOES THE USER KNOW?

For the documentation to address the specific needs of each user group, the second step is to determine what the users know. This knowledge will direct how much detail is included.

It is important during this step to keep in mind the purpose
of user documentation. Technically, or from a business point of view, you must assume that users know their jobs. A system's user documentation is not intended to be an overall on-the-job training manual. User documentation should explain how to use a system to aid the users in their jobs. The key word here is "aid." If accountants are trying to use a system to produce payroll checks, you have to assume they know the details involved, like gross pay, taxes, and insurance deductions. This does not mean that the documentation should be so brief that users can't decide how the system relates to their business functions. Therefore, a certain amount of overlap will be required to explain what business function is involved and how to use the computer to perform that function.

To do this, besides understanding the work the users do and how the system fits into their work flow, you must also know the educational levels of the users and the language or terminology they use. Certain words, like field, element, table, or key, that are common to computer personnel mean something entirely different in the user world. Therefore, to write the documentation in terms users understand, you must exercise caution in your choice of words. This is particularly true if the users are novices at the computer game. Too much computer jargon will not only confuse, but also alienate, the users.

Finding out what the users know requires interviewing and working with them. Writing the user manual from a computer person's perspective will not accomplish the purpose intended.

WHAT DOES THE USER NEED TO KNOW?

To use a system, users need to know answers to these questions:

1. What is the system designed to do?
2. How do they get data into the system?
3. What can they expect out of the system?

These, too, may sound like easy questions; but writing a user manual that answers these questions is not so simple.

Defining the System's Purpose

Telling the users what the system does may be the easiest part of the documentation to write. Assuming that the system does what it is supposed to do, you may be able to use the requirements document to help you write about the system's purpose.

If the system is multipurpose (like a database designed for use by several departments), your documentation will be set up in modules. In these cases an overview stating the purpose of the entire system is needed, as well as an overview for each module defining the system's purpose for each department.

A system overview does not have to be a lengthy explanation of every function performed, but it should provide enough information to allow users to determine whether the system is designed to perform their application. A purpose such as, "System X is a database system designed to provide the user with standardized reports" tells the user nothing. That statement could be the purpose of any system.

So, in the overview, give the users sufficient information to determine what business functions the system performs and to decide whether they can use the system to accomplish what they need to do. Write the overview in users' terms and from their perspective.

Getting Data into the System

Don't shortchange users when describing how to enter data. If the system is executed in batch mode, explain in detail how the data are entered. Do the users fill out forms which are submitted to data entry for keypunching or keying? If so, what forms are required (including batch control forms) and what is entered on each form?

If the system is executed in interactive mode, do the users know how to access screens, page forward, and correct errors as they occur?

Are data elements required, optional, dependent on the presence of other data? Are there minimum and maximum limits on data elements? For certain types of data, you need to give instructions on the units expected: Are volumes handled by the programs as MCF per day or MMCF per month? Are rates expected as cents or dollars? The users should be informed if data are converted from one unit into another within the system, since their output may be affected.

If the system is a database, the users will also need to know how to correct stored data or remove data. Are transaction reports stating acceptance or rejection of the data formatted so they are easily understood by the users? If not, the documentation should include a description of how to read the transaction report.

In general, while considering the system's input data, try to presuppose all questions a user might have and plan to answer those questions in the user manual. Additionally, plan a good format for describing the input. Write the input instructions in active voice and make the instructions easy to read.

Getting Data Out of the System

Users also need to know what reports to expect as output. Will the reports be generated automatically or must they be requested? If the reports are requested, what is the procedure for requesting them? Another form?

Think about what data are shown on the output reports. Some of the data will be a regurgitation of the data entered by the user; but what about the calculated items? Some users may want the calculations used within the system to be included in their documentation. A code listing will not suffice for this unless your users know how to read COBOL, FORTRAN, or whatever language has been used in the programming. Just as the calculations were translated into a programming language for the system, calculations presented in the user documentation should be retranslated into normal mathematical equations.

What about the error messages? "GETD FAILED
WHERE NO PARENT RG EXISTS” is not self-explanatory to users who are unfamiliar with databases. If it has not been done before, you should review error messages while you are writing the documentation. Even if the error message is straightforward, is it clear what the users should do to correct the problem? A section on error-handling procedures may be needed in the documentation.

Detailed information on a system’s reports is essential if a user is to understand what the system does. Knowing what functions the system performs and how to enter data will not help the users if they can’t understand what their reports indicate.

SUMMARY

Identifying the users is really just a series of questions—questions you ask yourself and questions you ask the users. Your goal is to make the documentation thorough, yet simple and easy to understand. Just like developing the system, writing the documentation requires planning, analysis, testing, and review. Months of work go into developing a system to make the users’ jobs easier and more efficient. Don’t throw those months away with poorly written user documentation. A system that is not understood cannot be used. Make your systems usable with well-written end user documentation.