Payoff
IS managers can assign technical communicators to help with many user-related tasks, such as user requirements analyses, the development of new information products, and the creation of customized guides and tutorials and computer-based instruction. The methods in this article represent improved ways to deliver training and documentation to users so they have the specific technical knowledge they need to operate software tools effectively.

Introduction
Users' needs have helped to create an expanded role for technical communicators; however, organizational managers of information systems have not yet fully recognized the need. Many owners, managers, and systems analysts of computer software businesses continue to depend on technical communicators solely to document software projects at the end of the development process. The technical communicator still has a one-dimensional image as someone qualified to write and edit. Yet these activities, coupled with the training provided by most academic programs, qualify the technical communicator for other user-related tasks, especially user requirements analyses.

In contrast to the stereotype, the technical communicator in an IS department can perform many roles: software and hardware documentation writer, trainer, computer-based instruction programmer, usability tester, hypertext developer, online help writer, quick reference guide writer, standards writer, front-end analyst, and instructional designer. The technical communicator can also fill the role of communication analyst by studying communication problems—and potential problems—and finding the best means to solve them. This diversity creates a varied work environment in which the technical communicator must be knowledgeable, flexible, and ingenious when facing many communication challenges.

The types of referential and instructional information created for users can be considered information products. These products include books, quick reference materials, video instruction, and online tutorials. To know what information products to create, the technical communicator needs to know both who the audience is and what the best course of action is to solve problems when they occur. Determining the appropriate product requires the ability to assess the needs of users.

This article explains the technical communicator's role in six areas: analyzing user requirements, clarifying the documentation process, developing information products, developing efficient documentation, providing adequate computer training, and anticipating future needs.

Analyzing User Requirements
At the beginning of a proposed training or documentation project, front-end analysis must occur before teaching or writing strategies are developed. When done properly, this analysis does not guess at or tell the users what they need; rather, it finds the real needs of
users through questionnaires, group meetings, and written and oral interviews. The analyst also recognizes that not all communication problems can be solved through training and writing. For example, if the problem is environmental or motivational, only a managerial decision can solve the problem. If the analysis shows that the problem can be solved by some form of training or the development of an information product, technical communicators must find the most effective way of solving the problem. The front-end analysis is conducted in two stages: a needs assessment and a detailed definition of the learning objective (see Exhibit 1).

**Front-End Analysis**

<table>
<thead>
<tr>
<th>METHOD</th>
<th>QUESTIONS TO BE ANSWERED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Needs Assessment</td>
<td>Who says there is a problem?</td>
</tr>
<tr>
<td></td>
<td>What should be happening?</td>
</tr>
<tr>
<td></td>
<td>What actually is happening?</td>
</tr>
<tr>
<td></td>
<td>What are the feelings of people with the problem?</td>
</tr>
<tr>
<td></td>
<td>What is causing the problem?</td>
</tr>
<tr>
<td></td>
<td>What are the costs if the problem is not solved?</td>
</tr>
<tr>
<td></td>
<td>Who or what can solve the problem?</td>
</tr>
<tr>
<td>Learning Objective</td>
<td>What should the user do after instruction?</td>
</tr>
</tbody>
</table>

**Needs Assessment**

Needs assessment is a systematic effort that gathers opinions and ideas from a variety of sources on performance problems or new systems and technologies. A thorough needs assessment is conducted through interviews, surveys, and small group interactions. This assessment identifies the people who say there is a problem, what should be happening versus what actually is happening, the cause of the problem, the feelings that affect the people encountering the problem, the cost of not solving the problem, and who or what should solve it.

**User Interface Specialists.**

For large projects, a technical communicator may need to work on project teams with various specialists. As one such specialist in an emerging field, the user interface professional—also known as a human factors-specialist—ensures that the design of both the input and output of any system takes users' needs into consideration. Working with the trainers and technical communicators, the user interface professional's goal is to design a product that is easy to use and as close to what a user needs as possible. If a project manager assigns a user interface professional to a project, the role of the technical communicator changes. For example, the user interface professional becomes responsible for preparing documents—the interviews, questionnaires, and formal needs assessments—that show the needs of the user. The technical communicator can then focus more closely on user manuals, online information, and other documentation that helps to support the product.

Regardless of whether the assessment comes from an individual or a team, the needs assessment should document in a written report the questions and opinions of the users. If users have problems finding proper equipment and tools or are not motivated to perform the required tasks, management must address these situations. When a needs assessment
finds that a communication problem can be solved by providing users with more or better information, the technical communicator must determine the best way to deliver the information. Some problems can be solved with a single form of written or online documentation, whereas others may require sets of documentation and classroom training.

**Defining the User's Learning Objective**

The next step of the front-end analysis is to specifically define what the user must learn. Defining the learning objective helps determine how information should be delivered. To meet the needs of a user or student, the instructional designer defines the objective of the lesson. Before instruction occurs, the designer must specify what the learner should be able to do after the instruction. To answer the question “What should the user do after instruction?” the Gagne-Briggs SLOAT method can be used as a guideline to define the Situation, the Learned capability, Object, Action, and the Tools (or constraints) that the user needs to learn. When these specific points have been defined, writers and trainers can clearly illustrate the most important instructional goals.

**Clarifying the Documentation Process**

Writers and trainers should also be aware of the design methods and standards that IS development teams use to create systems. Structured design strategies and department standards can affect the way a technical communicator provides information. Technical communicators can enhance the efficiency of a structured systems design and develop system standards that affect all aspects of information systems.

**Enhancing Efficiency in Structured Systems Design**

In theory, a structured design procedure sets guidelines that assist a systems design team in stating the functional requirements of a system in logical terms. Structured design is the process of deciding which components, interconnected in what way, will solve some well-specified problem; the design process uses specific techniques, such as a data flow diagrammer or a structure chart. With a structured design, the program can be planned correctly before any coding takes place. (How this methodology affects the technical communicator is discussed in further detail under “In-Department Documentation.”)

Although IS managers may acknowledge the advantages of a structured approach, they often jump at the opportunity to adopt time-saving short cuts. These short cuts usually involve slashing the first and last stages of the structured design approach. In the first stages, user meetings, diagrams, and charts of the problem are shelved for quick coding; and in the last stages, user testing, program documentation, user documentation, and training may be forgotten altogether. Slashing these stages of an approach results in apparent savings in the short run but often produces problems that increase costs during the life of the system.

Because one phase of traditional system design must be completed before the next, problems and errors in one phase seriously affect the later phases. To counteract this characteristic shortcoming of traditional structured designs, analysts have created alternative designs, such as Henson and Hughes's two-dimensional approach to system development. This system tries to foresee problems that may occur in future phases to avoid wasting
Implementing this kind of design requires developing new standards and editing documents that were created for previously used designs. As a result, documentation on the status of each phase will most likely be more elaborate in its detail compared with that produced using the one-dimensional approach.

For efficiency in producing software documentation, a writer needs basic knowledge of structured design. More important, a writer can assist in coordinating the entire project. Within a structured design team, the writer can help establish user meetings, facilitate communication between team members, and conduct usability testing of the program and documentation. The writer can also analyze the need for hypertext, online documentation, training guides, and user guides.

**Developing Documentation Standards**

In the utopian computer environment, an open system with total machine compatibility is the ideal of any organization. An open system is generally defined as the use of a stable, publicly recognized interface between systems and their components. Until there is a standard open system, organizations need to document their computer strategies for the benefit of management and vendors. Examples of such strategies involve software and hardware, relational data base management, electronic mail, and desktop publishing.

**Software and Hardware.**

When software and hardware are purchased, the IS department can set a standard for the entire organization. Although IS can set a standard without any input from affected divisions of the organization, ideally it should consult the users before setting the standards. With standards for software and hardware purchases, the entire organization can reduce vendor costs by allowing users to buy in bulk quantities. Such standards also save time and training costs. For example, if a single word processing package becomes a standard, training can focus on that single package. By surveying users to identify the real needs, the analyst can determine standard software packages. A technical communicator can help management and trainers by finding information on the following topics:

- What software is used the most.
- What software users would like to learn.
- What software users would like to use but do not have.
- How much microcomputers would be used in the future.
- What spreadsheet, data base, word processing, and operating system management packages they preferred.

**Relational Data Base Management Systems.**

When organizations decide to use a Relational Data Base Management System, they must establish standards that all analysts and programmers can recognize in

---

developing and maintaining forms. A technical communicator may write, edit, or help establish these standards. For example, the development of forms will require a set of standards. This documentation can be broken into four kinds:

- All forms.
- Query forms.
- Data entry forms.
- Data modification forms.

These documentation categories require internal standards for such routine operations as how to describe fields; where to find help; how to use function keys; where to validate data; how to organize formats for date, time, address, and name storage; and how to handle conditions that appear on more than one form.

**Electronic Mail.**

E-mail is an important aspect of efficient and quick communication within large organizations. The technical communicator should be aware of the problems that might arise for users with this type of communication. Although the design of some software programs removes the difficulties of E-mail systems, not all organizations have these programs. When the organization does not use a sophisticated software package, the technical communicator should be prepared to document standard ways for users to perform common tasks: enter and exit the system; read help messages; and perform such routine message tasks as listing, reading, saving, deleting, sending, and printing messages.

**Desktop Publishing.**

Another need for precise standards involves desktop publishing. Without standards, an organization can produce documents that are printed in too many different type styles and with mixed headings, a confusing use of visual cues (caps, bold, or underline), and an unwanted variety of letterheads. When publication becomes an important part of an organization, management may consider developing a Publishing Information Center to help centralize the work. Whether developing products through individuals or a publishing center, organizations should develop standards for logotype, color and typography of specific publications, design of the page grid, and the size, shape, and binding of publications.

**Developing Information Products**

User products are the heart of an IS department, so developing documents for these products becomes highly significant. User documents can be classified according to component type (hardware or software), mode of presentation (hard-copy or online), content (reference or tutorial), and the environment for use (internal and external).

For component type, software documentation may involve instructions on how a personnel program works from the user's view (e.g., how to log on to a secure system or how to print a spreadsheet graph). Hardware documentation may involve instructions on
how to operate and maintain machinery (e.g., how to test whether an asynchronous card is still operating or how to change a printer ribbon).

For the mode of presentation, textual user documentation is written, whereas online documentation appears on a screen. Online documentation should be context-sensitive information that allows a user quick, brief information that can be understood immediately. The content of user documentation is either referential or tutorial. The documentation will be used to refer either to a subject or to a tutorial in a specific aspect of knowledge. Exhibit 2 outlines the diverse approaches to user documentation.

### Types of User Documentation

<table>
<thead>
<tr>
<th>CLASS and SUBCLASS</th>
<th>EXAMPLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPONENT TYPE</td>
<td></td>
</tr>
<tr>
<td>Hardware</td>
<td>How to test an asynchronous card</td>
</tr>
<tr>
<td></td>
<td>How to change a laser cartridge</td>
</tr>
<tr>
<td>Software</td>
<td>How to log on to a secure system</td>
</tr>
<tr>
<td></td>
<td>How to create a spreadsheet</td>
</tr>
<tr>
<td>MODE OF PRESENTATION</td>
<td></td>
</tr>
<tr>
<td>Hard Copy</td>
<td>User manuals written on paper</td>
</tr>
<tr>
<td></td>
<td>Quick reference cards explaining DOS commands</td>
</tr>
<tr>
<td>Online</td>
<td>Help messages or help codes that appear on a computer screen</td>
</tr>
<tr>
<td></td>
<td>Hypertext</td>
</tr>
<tr>
<td></td>
<td>Computer-based instruction</td>
</tr>
<tr>
<td>CONTENT</td>
<td>User manual written on paper that lists the procedure to add a user to a network system</td>
</tr>
<tr>
<td>Tutorial</td>
<td>A step-by-step procedure on how to write and print a letter on a word processor</td>
</tr>
<tr>
<td>ENVIRONMENT</td>
<td>In-department procedures for backing up hard disks</td>
</tr>
<tr>
<td>Internal</td>
<td>In house newsletters</td>
</tr>
<tr>
<td>External</td>
<td>Flyers or brochures announcing software upgrades.</td>
</tr>
</tbody>
</table>

Perhaps a most significant category for IS personnel is the environment for use. For the purposes of this article, internal and external environments are expanded into three categories:

- In-department (those who work inside the IS department).
- In-house (those who work inside the organization).
- External (anyone outside the organization).

**In-Department Documentation**

Maintenance and update work require a programmer to use the history of the program from the developer's point of view. Such a history includes:

- Table of modifications by date.
This kind of program documentation can also include detailed user documentation. Because IS departments are notorious for neglecting detailed program documentation, programmers and analysts are usually eager to have someone help them write and organize the details of their system. Here again a technical communicator can help.

**Data Bases.**

Documenting data bases requires unique approaches that differ from non-data base systems. No matter what approach the documenter takes, the documentation package should be organized around the fundamental principles that guided the original design process. One approach, based on National Computing Center documentation standards, documents the multilevel structure description of a data base system according to standards set by the International Organization for Standardization. There are three main abstraction levels to document:

- **Conceptual.** A unique central description of the various information contents that may be in the data base, including knowledge about the subject matter and the concepts, conditions, constraints, and identifiers of the universe of discourse or the object.

- **Logical data base.** Units of logical data organization, such as records, tables or relations, fields, columns and attributes, and types of data values.

- **Physical data base.** Facilities of the computer system unique to the working data base, such as storage facilities and computer efficiency.\(^{110}\)

**In-House Documentation**

In-house user documentation is targeted for individuals throughout the entire organization. Although it comes in many forms, this documentation tells a user how to run the software or hardware. The documentation may range from the traditional written form of user manuals, quick reference guides, and training guides to the more sophisticated

---

means of hypertext, on-line help, and embedded training. Examples of systems to document include local area networks and optical imaging systems.

**Local Area Networks.**

A technical communicator should be able to assist the network administrator in documenting the LAN system. The user should be given documentation, ideally in quick reference cards, that covers these major topics about the system:

- Messages and codes for operating systems.
- Operator guides for equipment available to the user.
- Problem determination guides.
- Network configuration data.

**Optical Imaging Systems.**

With optical imaging systems, organizations can transfer paper-based documents to digital images. By running documents through sophisticated optical scanners, towers of paper can be reduced to manageable stacks of disks. Although organizations can have service bureaus do this work, they can save money by purchasing the systems to do it themselves. Because image conversion is so new, managers cannot always anticipate all the potential problems. Image conversion service bureaus warn potential users not to rely on verbal communication and strongly suggest that they document instructions, procedures, and amendments.

**External User Documentation**

The environment for external user documents involves anyone outside the organization who uses software or hardware. Designed with a broad audience in mind, such documentation can involve expensive productions that require attractive photography and graphics. Some users will have little or no knowledge of the subject and others will have profound knowledge. In addition to the documentation formats already discussed, all of which could be used in an external environment, comprehensive and selective documents can be designed to answer the problem of providing information for an audience with diverse levels of subject knowledge.

**Comprehensive Documents.**

Alphabetic software manuals are organized like dictionaries, with short explanations of key words listed in alphabetical order. They are recommended if the user is sophisticated, the user needs to know many different functions, there is online help, the user has reasonable keyword access, and the documentation will be updated often.

**Selective Documents.**

Selective documents come in various forms that are determined by their specific purpose. Newsletters are a quick, inexpensive way of keeping clients and users outside the
organization updated on important changes. Informational guides give a quick summary of information in the form of flyers and brochures. They provide the user a synopsis of information on new in-house systems and products for clients. For companies that are trying to sell software, demonstration disks can take a potential client through a guided tour of a program. The demo can simply present information or be interactive.

**Developing Efficient Documentation**

**Quick References**

When managers envision documentation, they often think instinctively of large volumes of text. However, a focus on the specific needs of specific users often indicates that a quick reference guide is needed. Quick reference guides are small and usually portable sets of information that help a user access needed information rapidly and conveniently. For example, quick reference guides may describe UNIX shell commands, MS-DOS commands, and word processing commands. Log-on cards attached to monitors are helpful for in-house and in-department users.

Within an IS department, an example of a quick reference guide is a syntax summary. This documentation gives a user a quick way of looking up the syntax of program commands (e.g., subroutine and field names) by listing the syntax in a logical manner, usually alphabetically. A quick training procedure guide can be a card that breaks down an important and frequently used procedure into short, easy steps. It can describe how to log on to a system, how to troubleshoot, or how to clear a paper jam in a copier. Its design should let the user find the information easily and use it. To ensure usefulness, the guide belongs in a pocket attached to the side of the machine.

**Providing Adequate Computer Training**

Computer training is not restricted to lectures and demonstrations in a classroom. Training occurs whenever a user accesses information to learn a skill associated with a computer-related job. The easier information is to access, the faster and better the user will learn. Determining who needs computer training and how it should be carried out is a complicated issue. Some research has challenged the widespread assumption that expert computer users make fewer errors than novice users. Other research suggests that trainers and software designers should emphasize error management—to know potential errors, be able to interpret errors, know strategies for recovering from errors, and develop strategies of error diagnosis. Error management should be a part of the design of any type of computer training.

Classroom training can be enhanced with online information, computer-based instruction, and customized training guides and tutorials. The writer should find the best way of presenting instructional and referential material for paper and computer screen.

---

Online Information Products

Putting information directly on the screen is the fastest way of providing instruction and reference. Online information appears on the screen in the form of codes, help messages, tutorials, and hypertext. Written instruction should give the computer user a sense of control by alleviating anxiety and promoting the feeling that the user can easily recover from mistakes.

Technical writers can assist instructional designers, classroom trainers, programmers, and analysts to write this kind of information and visually design its presentation. The designer of online documentation should understand visual design strategies that make an online document more readable. Along with traditional page layout issues (e.g., typography, white space, and line length), the designer should be concerned with color, reverse video, icon design, and other visuals that enhance the effectiveness of the electronic design.

Icons allow users to learn through images and to associate the image with prior knowledge rather than learn new information. When designing an icon, the writer should be aware of any standards that may be established by the organization. Writers using icons need to be sure that the images are easily understood and can be interpreted in only one way. Icons should be designed so that they can be understood as easily on screen as on paper.

Computer-Based Instruction

Students can receive efficient training if they learn on a machine similar to the one they are learning about. Computer-based instruction (CBI) can present information in many ways. Computer Based Instruction can solve a communication problem by providing interactive, tutorial training as part of an overall training scheme. CBI can also teach concepts in a deductive or inductive manner. A deductive tutorial presents examples and definitions followed by practice. An inductive tutorial presents examples and nonexamples to students and has them derive or induce the correct definition for the concept. Possibly the best way to use CBI is as part of an overall lesson plan in a lecture-style session that includes a variety of off-screen instructional or training activities (e.g., peer consultation, worksheets, practice, testing, and tutoring).

CBI should be developed by teachers and trainers, not by programmers. Teachers and trainers can use their expertise to develop tailor-made instruction for their students. Because many Computer Based Instruction software packages are simple enough, a teacher or trainer can develop a lesson without learning coding languages. When the software is too complicated, the Computer Based Instruction developer should still assume a designer’s role and pass the instruction to a programmer when the lesson is complete.

CBI’s strength is in its diversity (see Exhibit 3). It can be used within or outside an organization, as a reference or as a tutorial. It can deliver referential information through online text, diagrams, and video. CBI can be used to test prerequisite skills or find the most suitable learning style for a student before a training session. As a training method, it can be used as a standalone source that delivers interactive lessons that allow the students the benefits of self-study, and it can be used as part of a lecture-style training session.

Computer-Based Instruction
Because CBI appears on screen, effective text depends on the writer's command of such visual factors as color, screen fonts, spacing, and following organization standards. The writer must persuade users that the computer is a tool and they can become masters of the tool, not slaves to an unforgiving machine.

Part of lecture-style session

Customized Guides and Tutorials

Customizing written training guides and tutorials for users can make learning easier. Much current research focuses on how users respond to active, exploratory learning-while-doing instruction as opposed to hand-holding instructions. Users who like to learn new systems with a hands-on approach and who have an ability to learn quickly by doing (not reading) are active learners. Traditional tutorials have been designed around a reading-to-do model that presents the user with a variety of tasks in scenarios, then explains how the program works and the procedures following each task. By cutting text, a writer may be able to create a more efficient manual or tutorial. This approach, called minimalist or active learner design, gets a user actively involved by encouraging exploration of the system. Although research continues, some preliminary studies have shown that these approaches can have positive results.

Computer training requires technical communicators to provide information verbally, visually, and textually. Because training with computers can involve many media, writers and trainers need to be aware of the positive aspects of such innovative design strategies and design techniques as minimalist manuals, guided exploration, organizing and selecting content, conceptual information, procedural elaborations, and error resolutions. In addition, technical communicators should be familiar with commercial CBI software that might solve a training issue quicker than designing and coding tutorials from scratch.

Anticipating Future Needs

Among the most important advances affecting the workplace are those in desktop computing, networking, multimedia, telecommunications, optical storage, and artificial intelligence. As one example, the Engineering Documentation Management System (EDMS) is a relatively new part of the expanding optical imaging market. Developers of the EDMS are working to create an imaging environment that integrates Computer-Aided Design drawings, technical manuals, business records, and text into a single software package. The goal of the system is to capture engineering information electronically, then
manage it until the product or system is used. Ideally these systems can read, store, and retrieve huge amounts of drawings and texts; revise and enhance the information; and electronically deliver the images or documents to those who need the information. The systems have made their way into the microcomputer environment, but a fully functional system may start at more than $100,000.

Organizations like NASA and Ford Motor Co. are researching ways to improve their EDMS systems. Ultimately, corporations will be able to place products in the market faster by converting drawings, orders, bills, or other documents, integrating them with computer-aided design (CAD) image formats and standard raster-scan monitors, and distributing them quickly to the parts of the affected organization. As their organizational challenge, developers must solve the incompatibilities between computer-aided design (CAD) and paper formats and automate all technical documentation.

Multimedia is a means to bring together communication technologies into a single unit. Multimedia is the synthesis of communication media. By integrating sound, data, and images (video and photography) from several sources through a computer, multimedia can help solve a problem, educate, or present a subject in a more persuasive fashion. By linking images and text together in a nonlinear method, hypermedia allows users more control of what they want to experience.

Multimedia should have a profound impact on data processing, education, training, teleconferencing, and business. Computer-based instruction software, combined with a sophisticated multimedia system, has the potential to mimic the way the human mind links information and allow the user a faster way of learning and discovering new information. A Department of Defense study found that multimedia training was roughly 40% more effective than traditional training, with a retention rate that was 30% greater and a learning curve that was 30% less steep.\(^\text{112}\)

**Education Concerns**

Continuing education will give the technical communicator an edge. Graduate programs in technical communications, usually located in engineering or English departments, are preparing students for the field by offering courses in technical writing, editing, computer graphics, computer-based instruction, organizational communication, project management, interactive video, front-end analysis methods, research methodology, software documentation, rhetoric, and document design. The need for further education and training is agreed on by most in the technical communication profession, although the need for a credible certification program is debated.

Although graduate management program administrators within university business schools recognize that mastering the art of written and oral communication is extremely important, only 27% of those who responded to a survey on the subject indicated that they had a communication requirement in their curriculum. This survey showed a gap between desired outcome and curriculum commitment to excellent communication: only 26.5% required one course in written communication and 42.1% had one or two full-time faculty members dedicated to communication.\(^\text{113}\) Managers may not have the communication skills required to solve IS problems, which will require technical communicators to demonstrate their ability to identify and systematically solve communication problems.


Conclusion

Being an effective technical communicator within an IS department requires the communicator to produce more than written user documentation. A technical communicator can help management by analyzing, documenting, and solving communication problems, while also finding the best way to deliver information using many of the methods described in this article. When technical communicators demonstrate that they can provide more than just documentation, they will be perceived as not just one-dimensional, technical writers but valuable communication analysts and problem solvers. In this role, they can help not only computer users but IS management as well.

Bibliography


Author Biographies

John S. Craig

John S. Craig is director of training in the Office of Safety Information for the City and County of Denver. He has an MS in technical communication with an emphasis in instructional technology from the University of Colorado at Denver. He has written for *Technical Communication*, for the *Journal of the Society for Technical Communication*, and for *IEEE Transactions in Professional Communication*.

Charles E. Beck

Charles E. Beck is an associate professor and director of the master's program in technical communication at the University of Colorado at Denver. He also serves as a consultant in organizational communication and computer documentation. He has taught as a member of the full-time faculty of the US Air Force Academy, the US Naval Academy, and the US Air Force Institute of Technology, and also as an adjunct professor at Wright State University, George Washington University, and the University of Phoenix. He is an associate editor of *Technical Communication*. 