Technical Editing as Quality Assurance: Adding Value to Content

MICHELLE CORBIN, PAT MOELL, AND MIKE BOYD

TRENDS IN THE FIELD OF TECHNICAL EDITING

Technical editing is sometimes perceived to be simply a matter of grammar checking and proofreading. Perhaps fast-paced development environments, which often leave little time for editing functions, contribute to this perception—or, more precisely, this misperception. The levels-of-edit systems have helped technical editors manage the editorial functions in these hectic environments by providing a “framework within which editors can choose appropriate editorial tasks for a particular document” (Nadziejka 1995, p. 278). Recently though, technical editors are focusing even more on content editing, collaborating closely with technical writers on developing high-quality information (Nadziejka 1999; Bush 2000a; Bush 2000b; Bush 2000c). Taking this progression one step further, technical editing is beginning to be viewed as a quality assurance activity (Rude 2002; Hackos 1994; and Tarutz 1992).

Using levels-of-edit systems

To help manage their workloads, numerous technical editors adopted the Levels of Edit system developed by Van Buren and Buehler (1980) at the Jet Propulsion Laboratory. Levels of edit are varying combinations of nine types of editing tasks; these tasks range from substantive, language, and mechanical style edits to policy and coordination edits (Van Buren and Buehler 1980). In the last 20 years, many organizations have used these levels of edit to define their own editing processes.

Haugen (1991) criticizes levels of edit, admittedly a useful tool for editors, as a way to limit the technical editor’s involvement in the information development process:

Although the intent was not to define editing for the profession, in practice, the Levels of Edit has become a sort of standard, nonetheless. The unfortunate reality, however, is that Van Buren and Buehler’s levels rely heavily on rule-based tasks, so the levels of edit tend to narrow the conception of editing to largely rule-based activities. (p. 60)

Bush and Campbell (1995) suggest that levels-of-edit systems help only in determining the cost of editing, not in assigning the tasks of editing.

The original levels-of-edit system has not always been easy to apply to other environments, leading many organizations to modify the system for their needs. Prono and colleagues (1998) recently reported on a 2-year study, the result of which was a revision of their original levels of edit to three simple levels: proofreading edit, grammar edit, and full edit. Anderson and colleagues (1998) reported how they used a levels-of-edit system specifically designed for editing a Web site and discovered additional responsibilities for the editor, while clearly defining the work to be done at each stage of the project.

The most notable revision of the levels-of-edit system, however, is Nadziejka’s (1999), which addresses technical content at each of the three levels. He completely turns the focus away from grammar and style and toward the content. His levels are not sequential, but independent of one another. Even the lowest level of editing, which he calls the Rush Edit, remains completely focused on content.

Focusing on the content

Nadziejka (1999) presents his content-focused levels-of-edit system in a book published by the Council of Biological Editors; however, he clearly states that

For technical documents (by which is meant intellectual, scholarly, or highly complex documents in any field), the primary focus must be to help ensure that the technical content is complete, accurate, and understandable to the intended audience. (p. 5)

Grove (1994) agrees that technical editors need to move away from mechanical editing toward comprehensive editing focusing on “technical content, completeness,
and coherence” (p. 171). Since the 1980s, Bush (1981) has been encouraging technical editors to focus on content, to “improve the readability of the documents and the technical content of the information” (p. 16). In his Friendly Editor column that appeared first in *Technical communication* and later in *Intercom*, Bush has extolled the benefits and necessity of content editing and stated that, indeed, his career has focused on content editing (Bush 2000a; Bush 2000b; Bush 2000c). Bush reiterates that content editing goes beyond policing correctness; it focuses on clarifying content. Technical editing clearly is starting to be perceived more as content checking than as merely grammar checking. This content editing brings a “technical quality” to the information, instead of just an “editorial quality” (Nadziejka 1995, p. 283).

**Technical editing as quality assurance**

With more focus on content editing, technical editing can be seen as a quality assurance process. Mead (1998) discusses benchmarks for documentation costs, suggesting that one such benchmark is a “quality assurance ratio” that takes into account the amount of editing time in relation to writing time. In a software development environment, he also suggests that the quality of the documentation reflects on the customers’ perceptions of the quality of the software product itself. Technical editors play a large role in ensuring that quality. Hackos (1994) states as one of her guidelines for managing documentation projects, “Work hard to institute technical editing as the single most important quality assurance mechanism in the publications-development life cycle” (p. 376).

In a software development environment, quality assurance is part of the software testing environment. Software testers ensure the quality of the code that is written by computer programmers. Within this software development environment, comparisons have been made between technical writers and computer programmers. As Bresko (1991) states,

> Many similarities exist between computer programming and technical writing. For example, in the composition process for both programs and documents, programmers and technical writers gather all available, pertinent materials and begin writing in a logical order according to the rules of the language they are using. The difference is in the language and audience. Programmers write programming code for the computer to interpret and writers write words and sentences for their readers to interpret. (p. 218)

This comparison could be extended to assert that technical editing—and content editing in particular—provides the same quality assurance processes for technical information that software testing does for programming code. Just as a software product is not released without going through various software tests, product information (in the form of books, online help systems, Web sites, individual topics, and so on) should not be released without being subject to technical edits.

In this article, we review the typical software testing activities and then compare those activities to technical editing activities. This article shows that by providing quality assurance through content editing, technical editors add value to the information development process and help to give users the quality content that they deserve.

**OVERVIEW OF SOFTWARE TESTING**

Over the past 30 years, software testing has emerged as a critical part of the software development process. Software has become increasingly complex and has grown in size. To match that growth, software testing organizations, along with software testing methods and standards, have emerged (Kit 1995). The Institute of Electrical and Electronics Engineers (IEEE®) includes technical committees and standards boards that develop these standards in conjunction with software testing practitioners. These standards often are approved and adopted by the American National Standards Institute, better known to most by its acronym ANSI (Kit 1995).

Software testers are paired with software developers to create solid and usable software (Chillarege 1999). Software engineering best practices show that software developers work in tandem with software testers to produce the best quality product possible.

**Software testing** is the process of using the software to ensure that it meets established specifications and runs in various environments (Whittaker 2000). Kit describes the process of software testing as consisting of complementary processes of verification testing and validation testing (Kit 1995). These software testing processes must be “integrated at the most effective points in the development life cycle” (Kit 1995, p. 26).

**Verification testing** is the process of reviewing requirement specifications, design specifications, and the code. During verification testing, testers use formal inspections or informal walk-throughs to review both the specifications and the early versions of the software product so that they can determine whether the software product and specifi-
Simply put, the concept of quality applied to technical information means that the information meets the users’ needs by providing the right information at the right time. Specifications satisfy the customer requirements (Kit 1995).

Validation testing is the process of evaluating the software to determine whether it meets those specifications. During validation testing, testers use both low-level and high-level testing procedures. Low-level testing includes these types of tasks:

- **Unit testing**, in which the developers test an individual component of a software product to identify any discrepancies between the specifications and the actual behavior of the component (Kit 1995).
- **Integration testing**, in which developers and testers typically combine individual components of a software product to discover errors in the interfaces between the components (Kit 1995). For example, they determine how well the components work together or if one component causes a problem for other components.

High-level testing includes the following types of testing:

- **Usability testing**, in which testers or human factors engineers identify how the typical users interact with the user interfaces of the software product, including its documentation, for characteristics such as accessibility, responsiveness, efficiency, and comprehensibility (Kit 1995). For example, testers watch a person use the online help. The usability test fails if a person tries several different ways of locating information but still cannot find the required topic.
- **Function testing**, in which testers identify discrepancies between the functional specifications of a software product and its actual behavior (Kit 1995). For example, the specifications say that users can print a data table. The function test fails if the tester looks at the screen display and sees no Print button or if the tester presses Ctrl+P but does not get a response.
- **System testing**, in which testers take the perspective of the users and determine whether the software product meets the original specifications. System testing includes activities such as stress testing, security testing, performance testing, configuration testing, installation testing, recovery testing, serviceability testing, and reliability testing (Kit 1995).

Software testers add value to the software product by discovering errors and getting them on the table as early as possible; to save the developers from building products based on error-ridden sources, to ensure the marketing people can deliver what the customer wants, and to ensure management gets the bottom line on the quality and finance they are looking for. (Kit 1995, p. 23)

In summary, software testers add value by ensuring that the product meets the expectations of the users.

**TECHNICAL EDITORS AS ADVOCATES FOR QUALITY TECHNICAL INFORMATION**

Much as the software tester is responsible for the quality of the software product’s content, the technical editor is responsible for the quality of the information content in whatever format it appears. And what does quality in information mean? Hargis and colleagues (1998) identified several quality characteristics: accuracy, clarity, completeness, organization, retrievability, style, task orientation, and visual effectiveness. Simply put, the concept of quality applied to technical information means that the information meets the users’ needs by providing the right information at the right time. This goal of quality technical information can be accomplished only if the information is:

- Easily understood by the users
- Easily retrieved by the users
- Well-written, complete, and technically accurate

The technical editor ensures that these requirements are met. The technical editor is an advocate for the language, the company, the writers, and, most importantly, the users (Bush and Campbell 1995).

As an advocate for the language, the technical editor must have a strong foundation in grammar and usage. The technical editor copy edits the document at the sentence level. In addition, the technical editor checks for completeness, appropriateness of the language for the audience, logical structure, effective organizational flow, clarity, and conciseness.

As an advocate for the company, the technical editor checks for compliance to company style standards, legal correctness, safety and security (liability) issues, appropriate use of product names and trademarks, permissions to use borrowed data or text, and so on. The technical editor also ensures that copyrights are correct, that product information is not disclosed prematurely, and that appropriate disclaimers and copyright notices accompany different versions of the information.

As an advocate for the writer, the technical editor is part of the information development team. Technical editors get involved early in the process with writers and become collaborators with them in their work. In information development, the technical editor helps the writer with
templates, information models, style sheets, and repeated (or common) text. The technical editor uses language skills, writing skills, and information design skills to help bring clarity and cohesion to the writers’ work.

As an advocate for the users, the technical editor follows the instructions written by the technical writer and points out areas of difficulty. The technical editor is a good usability tester who reads the information, performs the tasks, and tries to find holes, errors, or ambiguities. The technical editor checks for appropriateness of the examples and graphics for international audiences. Also, the technical editor is a second check on the technical accuracy of the information. The technical editor helps make the information more concise for the users, paring the text to its essentials, eliminating technical jargon and unnecessary details, as well as eliminating marketing-related material when that type of information does not belong in the deliverables. Finally, the technical editor focuses on the index, the table of contents, and other access and navigational methods that can help the users to find content easily and to understand its context.

SOFTWARE TESTING AND TECHNICAL EDITING

Both software testers and technical editors focus on quality assurance activities. While technical editors perform both verification testing activities and validation testing activities, the most striking similarities are in the validation testing activities. These validation testing activities (unit testing, integration testing, usability testing, function testing, and system testing) are very similar to the following technical editing activities:

◆ Comprehensive editing, in which the editor reviews the content, organization, and design of the information to ensure that the reader can understand it easily and that the information meets its objectives (Rude 2002). The editor works closely with the writer to develop the information completely by adding, removing, and revising the information as necessary. Comprehensive editing is sometimes called developmental editing or substantive editing. While the terms are different, the activities performed during a comprehensive edit are nearly always the same.

◆ Usability editing, in which the editor becomes the first user of the information, reading and responding as a typical user might respond to the information (Tarutz 1992; Soderston 1985). The editor uses the table of contents, the index, the procedures, and the commands, and verifies that each entry is correct and that there are no misunderstandings or misrepresentations. Usability editing has its roots in usability testing and is particularly valuable when usability testing might not be available or viable for the information being reviewed. Some might consider usability editing part of comprehensive editing, while others might consider it an extension of copy editing, but it is clearly a distinct type of editing that a technical editor performs to improve the quality and usability of the information.

◆ Copy editing, in which the editor reviews the paragraphs, the sentences, and the words for spelling, punctuation, and grammatical errors. The editor uses the appropriate style guides to ensure that the document is accurate, complete, and consistent.

“When most people hear the word editing, they think of the activities that fall under copy editing” (Tarutz 1992, p. 87). However, technical editing is much more than just copy editing. At every stage of the information development process, the technical editor focuses on the content to ensure the quality of that information. Each of these editing activities (comprehensive editing, usability editing, and copy editing) includes content editing.

In the sections that follow, we outline specific content editing activities within these editing activities, showing how they are all quality assurance activities (see Table 1).

Comprehensive editing

As advocates for the user, technical editors focus their time, their talent, and their attention on the content of the information. They engage in the content, like a software tester performing system testing activities, thinking about how the information helps users complete their tasks. Early in the development cycle, technical editors perform comprehensive editing to improve the accuracy, clarity, and accessibility of the information. “The objective is to make the most extensive changes at the earliest stage possible, because the later the change, the higher the costs and the greater the risk to the schedule” (Tarutz 1992, p. 71). Also, Rude (1987) points out that comprehensive editing “increases the chances that the editing will be purposeful rather than reactive, and thoughtful rather than arbitrary” (p. RET-144).

As an advocate for the company (and the users), technical editors must carefully review the information specifications so that they can ensure that the appropriate information is available and that the information follows those specifications. Technical editors often help create the information specifications, serving as information architects and helping plan the product’s documentation sets (Tarutz
As early drafts of the information are developed, technical editors review the outlines and whatever portions of the information are available. Rosenquist (2001) reports in her surveys that editors can catch structural flaws early in the development cycle, thus improving quality and reducing costs, because additions and changes are not perpetuated within an already flawed structure.

At every draft, technical editors focus on the structure and organization. The process of comprehensive editing is, even more than a meticulous process, an intellectual process, requiring in-depth reading of the manuscript with all the technical knowledge the editor can bring to bear” (Nadziejka 1995, p. 280). This type of evaluation is similar to the integration testing that software testers perform. Henry (1998) describes this type of comprehensive editing as an “integration evaluation,” in which editors evaluate every information element as part of the integrated software system, confirming the information architecture and providing answers to many usability questions.

When technical editors perform comprehensive editing, they engage in many types of quality assurance activities. We will focus on the following content editing activities that are part of comprehensive editing:

- Ensuring technical accuracy
- Understanding and working toward the big picture
- Reducing the amount of information
- Re-using information
- Customizing information for different software solutions
- Enabling continuous improvement

Ensuring technical accuracy Perhaps the most important duty of an editor is “to make the language . . . conform to the actual facts” (Bush and Campbell 1995, p. 11). Bush encourages technical editors to move away from a focus on grammatical correctness toward content correctness.

As we all know, the main writing problem is not incorrectness, or even awkwardness. It is a massive blockage of information transfer, imposed by thoughtless organization, skewed structure, and misplaced emphasis, along with stultifying wordiness and gross imprecision in thought. (Bush 1981, p. 15)

While ensuring the technical accuracy of the information is not a new trend, it has become even more important for
these reasons:

- Our litigious society encourages companies and clients to sue suppliers that provide inaccurate information.
- Sales now are sometimes won or lost based on the quality of the information provided with the software.
- Software is returned because users cannot install or use it. If the product or its information is inaccurate, it is difficult for the user to even begin using the product.

The first line of defense in ensuring technical accuracy is the software development team: developers, testers, technical support personnel, marketing personnel, writers, and so on. The second line of defense is the technical editor, who checks for consistency and accuracy in terminology, user interfaces, product names, inputs producing the specified outputs, and so on. According to Tarutz (1992), part of the technical editor’s job is to do an accuracy edit, checking for contradictions and discrepancies in the information. Nadziejka (1999) also requires that technical editors verify that “statements are factually correct and logically sound” (p. 14).

To perform useful and appropriate edits, technical editors must be knowledgeable enough about the subject matter so as not to create inaccuracies in the information with their editorial comments. “The more you know about a subject, the better you’ll be able to ask intelligent questions. If you know the answers, that’s better still” (Tarutz 1992, p. 101). However, technical editors must constantly remind themselves about what the user knows and doesn’t know, so that the information meets the needs of the user and the user can understand the information that is provided.

Understanding and working toward the big picture

Whether technical editors work in a single-sourcing environment (where topics are re-used and delivered in multiple information deliverables) or in a more traditional information development environment (where books and online help systems are the norm), technical editors must have a clear vision of the “big picture.” As Grove (1994) states,

> General knowledge enables the editor to be detached enough to see the big picture, to understand how the parts fit together. Whereas analytical skills help the editor separate the parts, general knowledge helps the editor put them together. Having both these attributes, the editor is able to see both the forest and the trees. (p. 173)

Technical editors must know how each information deliverable fits with the others, and they must know the users and how those users work with the software. When technical editors make “changes in words or sentences, editors must be careful to consider the paragraph, or at times the entire page, to be sure the changes fit into the big picture” (Bush and Campbell 1995, p. 11). The largest part of a comprehensive edit, according to Tarutz (1992), is ensuring that the information is organized into a coherent whole so that the users can find the answers quickly.

Reducing the amount of unnecessary information

Often, technical writers tend to include too many details and too much repetition. Developers and subject matter experts frequently want to tell everything there is to know about a function, while users only want to know enough to get the job done. Technical writers filter out some of this information, but technical editors must finish the job. Hargis and colleagues (1998) recommend finding a balance between enough information and the necessary information. Technical editors consider the needs of the users and consider precisely what they need to know to get back on task. “In today’s usability designs, less is more, and an editor may be the ideal person to make sure that everything stated is required and oriented to the users’ needs” (Bush 2001, p. 43).

The minimalist writing approach, as outlined by Carroll, urged writers and designers to let users do more, read less, and experience fewer errors (Carroll 1990). Minimalism is not a simple theory of brevity or incomplete information systems, but rather uses research results to suggest ways of supporting users in a learning environment by providing action-oriented, task-oriented, user-oriented, and error-prevention information (Carroll and van der Meij 1996, p. 84).

Bush and Campbell (1995) dedicate an entire chapter in their book to cutting copy and reducing wordiness in a careful and thoughtful manner. Again, the focus remains on what users need to understand to be able to perform their tasks. An added benefit of reducing the amount of information is a reduced cost in producing the information.
Re-using information A company’s information set, both internal and external, is one of its most valuable assets. In today’s software development environment, information development teams must look to knowledge management solutions, which allow the teams to re-use information and to keep pace with the shortening software development cycles. Information must be written once and delivered multiple times, in multiple media. Instead of producing books, writers essentially are producing topics that are combined to create help systems, Web sites, troubleshooting guides, and sometimes books. This modular approach parallels the object-oriented approach to programming, where individual functions are created in modules of code that can be re-used in several different software programs.

When these topics are combined to create an information deliverable, technical editors step in as advocates for the users and help make that deliverable understandable and usable. For example, when a topic or set of topics is created for re-use, it is often stripped of its context (so that it can be used in multiple contexts). Technical editors must ensure that the topics remain usable in each new context. Rockley (2001) says,

Many organizations have reduced or eliminated the role of the editor. However, single-sourcing makes this role an important one to ensure that information can be re-used effectively. . . . It is particularly important that editors not just look at the words, but look at the use of information to ensure that it is effectively written to meet customer needs. (p. 193)

This new single-sourcing environment brings new challenges to the technical editor:

✦ Information must be usable and appropriate in both printed and online formats.
✦ Cross referenced information must make sense in both printed and online formats, such that cross-references with page numbers occur in printed information and hypertext links in appropriate phrases occur in online information.
✦ Topics must be self-contained and thus able to stand alone.
✦ Topics must be understood when they are linked together from different paths through the information.
✦ Transitions need to occur between the self-contained topics.

Much like software testers performing unit testing and integration testing, technical editors review the topics as soon as they are written and continue to review the groups of topics as they are being developed. At the beginning of the information development cycle, technical editors can assist the writers in creating templates along with a well-structured prototype (or model) to ensure that topics are indeed reusable. Technical editors also can determine whether information is repeated across multiple topics and can suggest additional ways to re-use the repeated (or common) information.

Customizing information for different software solutions As software is developed using object-oriented programming techniques, and as the resulting objects are combined to create new or customer-customized software solutions, the information for those software solutions must be created just as quickly and easily. As a result, sometimes information developers must take the existing set of topics and wrap industry-specific information around them. Such an approach is particularly needed when different industries use different terms to refer to the same concepts. Throughout the information development cycle, technical editors use the product specifications and their knowledge of the audiences for these customized solutions to

✦ Ensure that the appropriate terminology is used for the audience
✦ Test the navigational paths through the information to ensure it makes sense to the users
✦ Review the information for cohesion and clear transitions so that users are unaware that different products or modules are being combined
✦ Confirm that comprehension of each topic does not depend on the presence of a topic that has been deleted in the customization process

Enabling continuous improvement Part of comprehensive editing is performing the appropriate edits within the time allowed. Technical editors must have excellent project management skills to keep track of the information and the level of editing that has been performed on that information. In extremely tight schedules, technical editors perform policy edits, focusing on such issues as legal requirements, trademarks and copyrights, and so on. Then, as time
allows or when the next release of the information is planned, technical editors can schedule the appropriate content edits and copy edits for the information. As Nadziejka (1995) suggests, comprehensive editing takes precedence over copy editing because it has a greater impact on the quality of the information from the users’ perspective.

**Usability editing**

As an advocate for the users, technical editors not only ensure that the content is accurate and clear, but also that the content is usable. Spencer (1996) recommends an internal usability review as a systematic way to identify basic usability problems. To the extent possible, technical editors test the information and verify that the users will succeed. These “usability edits . . . give us a way of ensuring quality” (Soderston 1985, p. 17).

Technical editors cannot replace the usefulness of actual usability testing; however, they can stand in for the users by becoming the “first users” of the information. Technical editors use the table of contents and index of the information deliverable to try to retrieve various topics. They use the procedures and determine whether the wording and formatting are easy to follow and understand.

*Editing has close ties to usability. Usability implies worthwhile content, sensible organization, readable style, and effective design, all which are also prime goals of editors. Editors inherently judge logic and emphasis, root out redundancy and waste, and assess ideas and their effects on costs and benefits.* (Bush 2001, p. 39)

When technical editors perform usability editing, they engage in many types of quality assurance activities. We will focus on the following content editing activities that are part of usability editing:

- **Ensuring that the information can be retrieved easily**
- **Making the information accessible to all users**
- **Understanding the users well enough to make appropriate decisions about style and content**

**Ensuring that the information can be retrieved easily**

Writers often re-create text that already exists in a deliverable because they themselves cannot find it. And if writers cannot find the information, how can users? Technical editors can ensure the information can be retrieved easily by carefully reviewing the following:

- **Table of contents** Not all information deliverables require a table of contents. Technical editors must verify whether one is necessary. They also need to determine whether it is complete or requires additional topics or levels of topics.

- **Index** The index must be complete and must contain the appropriate entries, subentries, and synonyms for the target audience. To check the index, technical editors must try to locate the topics that they are editing within the information deliverable by looking in the index for appropriate and usable index entries. They also must review the index as a whole—as a separate deliverable—to ensure that it is complete, accurate, and appropriate. “A thoughtful index increases the value of any book and helps guarantee that the book will be used often because its content is accessible” (Sencindiver 1991, p. 3).

- **Topic titles** Topic titles often are forgotten as a navigational feature, but they provide context for the information. For example, topic titles for Web pages or for some online help systems are used in Favorites or Bookmark lists that users can build. Topic titles must be specific enough to show the context and yet remain succinct. For example, “Features” is much too vague, whereas “Spell Checker Features” is much more descriptive.

- **Metadata** Metadata (or data that describes other data) consists of keywords that contain information about the topic. Sometimes the users can use these keywords within a Web site or online help system to retrieve information in a search request. However, these keywords also are used in a content management system to help writers find topics. Technical editors can build and maintain a list of approved and appropriate keywords to use as metadata in either implementation.

Testing these navigational features—using them just as users would to follow instructions in the documentation or to find topics easily—is at the heart of most content usability testing.

**Making the information accessible to all users**

Document is accessible if it provides access to all users regardless of abilities or disabilities. While the content editing, usability editing, and copy editing help make the information accessible to most users, technical editors can become advocates for disabled users and can help make the information accessible to all users. In the U.S.,

For software documentation, this means providing alternative text for graphics, figures, and tables; making help available online in text format; documenting new ways to access the software such as command-line alternatives for graphical user interface-based products, and so on. Additionally, technical editors can assist writers in running accessibility checking tools, such as the Bobby tool provided by the Center for Applied Special Technology (http://www.cast.org/bobby/). Technical editors can advocate compliance with these standards and guidelines, along with all the other standards and guidelines, and ensure that the information is provided in accessible formats and structures.

**Understanding the users well enough to make appropriate decisions about style and content** Users expect information to be accurate, clear, and complete. Beyond expecting that the technical content will be understandable, users form perceptions about the information to be accurate, clear, and complete. Technical editors perform copy edits according to the company style guide.

"An editor, whose specialty is language and document design, can suggest ways to make the document easier for readers to understand and use. The editor knows how to use style, organization, and visual design to achieve specific goals." (Rude 2002, p. 13)

When technical editors copy edit, they perform many types of quality assurance activities. We will focus on the following content editing activities that are part of copy editing:

- Ensuring that the information can be easily understood (and can therefore be translated easily)
- Standardizing the information written by multiple writers
- Verifying each information deliverable

**Ensuring that the information can be easily understood (and can therefore be translated easily)** More and more companies today are selling their software internationally, expanding their customer base. Information provided with the product is often translated into multiple languages. The type of editing that improves the translatability of the information happens to improve the readability of that information for the native language, too. These editing techniques include the following.

- **Use terminology consistently and carefully** According to Batty, "Effective communication in any context is made easier by the use of a common language that both parties understand" (1998). Using standard terminology reduces the number of words that must be translated and ensures that concepts are translated the same way in each instance. Using the same word to represent the same concept throughout the information and not using synonyms is preferred in technical writing (a difference from our literary counterparts). Technical editors should maintain a master glossary to ensure the definitions of those concepts are all the same throughout the information set. Controlling the writers' vocabulary reduces the language load and machine translation costs, while increasing the readability and usability of the information (Thrush 2001).

- **Provide shorter, clearer text** Translation heightens the need for shorter, clearer text. Using fewer words not only reduces translation costs but also makes it easier for users to find the needed content (just as it does for users of non-translated information). Technical editors help provide concise text by eliminating redundancies, avoiding ambig-
skills, technical editors of multiwriter projects can make the tent look-and-feel to the information. Using their language prototypes, and company style guides to bring a consistency directly to knowledge bases or knowledge management systems. Technical editors use templates, style sheets, and include cross-cultural examples. Weight loss and weight gain, dollar amounts, types of currency, and graphics can have unknown or varied implications for different cultures. Examples must be judiciously chosen and implemented, and carefully reviewed, so that the information is technically accurate and does not offend someone from another culture.

**Standardizing the information written by multiple writers** Rarely is information for a software product written by a single writer. Soon—if not already—users, consultants, and other developers will contribute directly to knowledge bases or knowledge management systems. Technical editors use templates, style sheets, prototypes, and company style guides to bring a consistent look-and-feel to the information. Using their language skills, technical editors of multewriter projects can make the information seem as though only one writer has written the information.

**Verifying each information deliverable** Whenever there is time for proofreading, technical editors can ensure the quality of every information deliverable. As topics are combined to create an information deliverable, and as information is presented in multiple media, technical editors need to verify that the total package is accurate, usable, and readable. Verifying that the CD contents, the PDF files, the index and table of contents of an online help system, or the navigational links in a Web site are the most current and accurate are all critical steps in the process. Technical editors are responsible for performing these final checks as quality assurance of the information deliverables.

**TECHNICAL EDITING AS A VIABLE, SEPARATE FIELD** Developing quality content for information deliverables is clearly a team effort. Although writing and editing are similar tasks, they are distinct and separate ones. As noted in the main premise of this article, software programming and software testing are similar tasks, but they are also distinct and separate tasks. The value of team specialization is not always appreciated in the technical communication field, leaving information development teams to consider self-editing and peer reviewing as their technical editing functions. In describing the job of technical editors, Rude (2002) states,

> Technical suggests not only the subject matter but also the method of working with the subject matter—to analyze, explain, interpret, inform, or instruct. . . . The art and skill of editing require specialized knowledge of the use of language and methods of making sense of information. (p. 16)

The typical information development environment, much like the software development environment in which it exists, is fast-paced and fraught with potential errors. Technical editors are the advocates and supporters of a more careful process to ensure that the users receive the information that they need. To that end, just as software testing is a viable, separate field, so too should technical editing be a viable, separate field.

**Technical editing is different from technical writing** For some technical communication teams, having a separate technical editor is an ideal arrangement, but one that is not always achieved. As Taruts notes, “Many companies combine the writing and editing functions into a single position, and many others don’t use editors at all” (1992, p. 16).
fired, under the justification that you can’t be tough in industry. Editors are the last hired and the first must pay attention to detail and still maintain the big picture (1992, p. 15).

Tarutz also observes that many people “don’t distinguish between technical writing and technical editing” (1992, p. 13). To a degree, this lack of comprehension flies in the face of reality because the skills associated with these tasks are not identical. The writer and the editor do share skills—for example, “excellent writing skills” and “expertise in using publishing tools” (Tarutz 1992, p. 14). Tarutz also points out that certain skills are usually specific to, or more developed in, each group. For example, the writer needs interviewing skills to retrieve information from subject matter experts; the editor must pay attention to detail and still maintain the big picture (1992, p. 15).

Tarutz (1992) notes that editors “have always had it tough in industry. Editors are the last hired and the first fired, under the justification that you can’t edit a manuscript that hasn’t been written” (p. 363). David Dayton (2002) has reported his research on online editing techniques (from a survey of STC members in the writer-editor category). He discovered that approximately 43% of the writers said that they were writer-editors, 34% of the writers said they were peer-editing writers, and only 4% responded that they performed the traditional role of technical editors (p. 86). Dayton also reported that these numbers had declined from surveys he performed in previous years.

Although this ratio of editors to writers may be declining, we do not believe that technical editing is a field facing extinction. In fact, the specialized skills that technical editors provide will become even more critical as information is combined and recombined from a variety of sources into a variety of information deliverables. Technical editing will, of necessity, continue as a viable, separate field.

**Considering self-edit and peer review**

To save money, technical communication managers often rely on self-editing or peer reviewing, instead of having a dedicated technical editor as part of the team.

According to Hackos, “Self-editing, a popular label for a lack of external editing, is both expensive and likely to fail. Failures result because people run out of time to edit their own work and lack the perspective to do so effectively” (1994, p. 361). Perspective is gained when the writer can put the work aside and not go back to it for several days. And, this relates strongly to the time limitation. Besides the lack of time and perspective, Hart identifies a fundamental challenge when writers need to edit their own materials or information written by colleagues: “Editing and writing require entirely different mindsets, and it’s difficult to make the mental shift from creating to revising” (1998, p. 17). Writers are less likely to challenge the assumptions being made about users because they already understand what has been written.

Hackos (1994) writes that another failure of self-editing is that it is inadequate for ensuring product quality. “When you have been working on a text for hours, days, or weeks, you are rarely able to find evidence of inconsistencies, gaps, redundancies, or the other early warning signs of a decline in quality” (p. 362). Technical editors “can bring objectivity to the [information] that the writer may lose by knowing the subject too well” (Rude 2002, p. 13).

Where self-editing fails, others try peer reviewing, which has its own set of advantages and disadvantages. “The advantage of peer reviewing is that all team members get to look at one another’s work. . . . They profit from observing other writers’ skillful approaches and solutions to writing problems” (Hackos 1994, p. 367). However, to be successful, the peer reviewing process must be worked into the information development processes and schedules, or “the system breaks down under the strain of too much work and difficult deadlines” (Hackos 1994, p. 368).

Peer reviewers always put their own writing assignments before their peer reviewing responsibilities. According to Tarutz (1992), “peer editors don’t approach the task with the same attitude as a professional editor,” because it’s not their primary responsibility (p. 15). Another disadvantage of peer reviewing is that “the quality of the editing activities will vary dramatically” because each writer will have different editing abilities, with little clout to enforce standards (Hackos 1994, p. 368). Tarutz (1992) shares feedback she has received from writers who have had their materials edited by both full-time editors and peer editors:

> Writers who have worked with both professional editors and peer editors have told me that a technical editor’s skills and abilities are what set apart professional editors. (p. 15)
ADDING VALUE TO TECHNICAL CONTENT

Bush (2001) said it best: “Editing saves time, cost, and confusion. It’s magic” (p. 39). Perhaps it seems like magic because “good editing goes unnoticed by the reader” (Tarutz 1992, p. 25). It is this sleight of hand—this magic—that technical editors use to add value to the technical content—and to the product—throughout the product development process.

This article, which has compared the field of technical editing to software testing, shows the great impact that technical editors can have on the quality of a product. “Information is good quality only when it contributes to improved software usability” (Henry 1998, p. 215). When the information has been through extensive technical editing, just as the software has been through extensive testing, the users get quality information that supports a quality product in the following ways:

- Users gain access to information that is clearer, more concise, and more comprehensible. The information flows together well, unnecessary information is removed, and the sentences are clear and direct. Graphics, including lists and tables, help to present the information in a more concise and visually organized manner.
- Less information increases the document’s usability and also produces savings in production and translation costs.
- Users can retrieve the information easily from multiple navigational features (indexes and tables of contents) that have been tested and revised.
- The product information has fewer errors or areas of confusion for the typical users. This means that support costs—such as staffing the help desk—are likely to be reduced.

As Henry (1998) notes,

> Technical editing is a critical activity. It is not mere spell checking or ‘correcting a document’ as some believe. Good technical editing contributes significantly to critical design goals such as retrievability, readability, and clarity. Moreover, it brings consistency across all information elements. (p. 230)

As Tarutz (1992) reminds us,

> It’s not enough to mark up a manuscript. If your editing doesn’t produce improvements, if the writer ignores your great suggestions, if your ideas don’t increase customer satisfaction—then you’ve just wasted your time. All you’ve done is a meaningless exercise of putting marks on paper. (p. 366)

It is time for technical editors to answer this call to arms, to step up to being technical editors, or more importantly technical content editors, and to work within our companies, projects, and environments to show that technical editing is truly a quality assurance process. As champions of our profession, we need to educate writers, writing managers, engineers, engineering managers—and anybody else who influences [our jobs—about] the editing function. [We need] to break down that stereotypical image of the green-eyed editor hunched over galleys to proofread for missed commas. Shatter this image by showing you can contribute to customer satisfaction—through a better product and better documentation. (Tarutz 1992, p. 367)

Finally, just as software goes through a quality assurance process, with developers tracking and fixing defects before the software is shipped to the customer, so must the product information go through similar quality assurance processes, including technical editors editing the content though comprehensive editing, usability editing, and copy editing. For technical editors, the documentation is the product. Like software testers, technical editors add value by ensuring that the product (the documentation) meets the users’ expectations and helps them easily use the software.

ACKNOWLEDGMENTS

Several people reviewed the manuscript and offered their comments and feedback. The STC Carolina chapter’s Technical Editing Special Interest Group (SIG) was extremely supportive in kick-starting this article, listening to a presentation of its key concepts, and offering ideas. Anne Tice, the manager of that SIG, as well as John Kohl, of SAS Institute, reviewed the first draft of this article. We were also fortunate enough to have Judith Tarutz review the second draft of this article. Their encouragement, editorial expertise, and detailed content editing comments improved its quality immensely. Finally, the Technical communication anonymous peer reviewers provided outstanding detailed comments about our manuscript. Our heartfelt thanks to all of them!

TRADEMARKS

IBM and Tivoli are registered trademarks of IBM Corporation.

IEEE is a registered trademark of Institute of Electrical and Electronics Engineers.

REFERENCES


**ADDITIONAL READINGS**


**MICHELLE CORBIN** is a senior technical editor at IBM® working on Tivoli® software. She has 12 years experience as a technical writer and technical editor, focusing much of her energies on the design and implementation of online information systems. She has a BA in English and an MS in Technical Communication, both from North Carolina State University. She’s a senior member of STC and a past president of the Carolina chapter. Contact information: corbinm@us.ibm.com

**PAT MOELL** is a manager of the technical editing department at SAS Institute Inc. in Cary, NC. She has a BS in English and mathematics education from SUNY Plattsburgh and an MLS from the University at Albany (SUNY). She has also
completed course work for a doctorate in Information Studies from Syracuse University. She is a senior member of STC and has received a Distinguished Chapter Service Award from the Carolina chapter for her contributions to chapter leadership and for helping to establish the chapter’s Technical Editing SIG. Contact information: Pat.Moell@sas.com

MIKE BOYD is a self-employed editor and instructional designer in Cary, NC. He has a BS in radio and television from Indiana University, Bloomington, and an MA in educational technology from the University of Iowa, Iowa City. He is a senior member of STC. In addition to his editing and instructional design experience, he has worked as a technical writer. Contact information: blueboyd@bellsouth.net