1989

The Cat in the reaper : usability testing of software instruction manuals

Susan McLean Hebert
San Jose State University

Follow this and additional works at: https://scholarworks.sjsu.edu/etd_theses

Recommended Citation

Hebert, Susan McLean, "The Cat in the reaper : usability testing of software instruction manuals" (1989). *Master's Theses*. 3143.

DOI: <https://doi.org/10.31979/etd.t2zw-jq34>

https://scholarworks.sjsu.edu/etd_theses/3143

This Thesis is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Theses by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

INFORMATION TO USERS

The most advanced technology has been used to photograph and reproduce this manuscript from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book. These are also available as one exposure on a standard 35mm slide or as a 17" x 23" black and white photographic print for an additional charge.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

University Microfilms International
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313/761-4700 800/521-0600

Order Number 1338696

The cat in the reaper: Usability testing of software instruction manuals

Hebert, Susan McLean, M.S.

San Jose State University, 1989

Copyright ©1990 by Hebert, Susan McLean. All rights reserved.

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

THE CAT IN THE REAPER:
USABILITY TESTING OF SOFTWARE INSTRUCTION MANUALS

A Thesis
Presented to
The Faculty of the Department of Journalism
and Mass Communications
San Jose State University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

by
Susan McLean Hebert
August, 1989

APPROVED FOR THE DEPARTMENT OF JOURNALISM
AND MASS COMMUNICATIONS

Dr. Diana Stover, Professor of Journalism

Dr. David Grey, Professor of Journalism

Professor Lois Rew, Coordinator,
Career and Technical Writing Programs

APPROVED FOR THE UNIVERSITY

Serena H. Stanford

ACKNOWLEDGMENTS

The author is deeply indebted to Dr. Diana Stover, Dr. David Grey, and Professor Lois Rew for their steadfast encouragement and very knowledgeable guidance. John McLean selflessly donated many, many hours of childcare, without which this thesis would have been impossible. Opal McLean offered buoying encouragement at every rough moment, Joan McLean helped with childcare and vital errands, Alice Lum generously provided childcare in the crucial early stages of the study, and Diane Besterman provided the first set of industry contacts, which made the study appear "do-able."

I am also very grateful to the interviewees who so generously gave their time and insights, especially Dr. T. R. Girill, Russell Kahn, Palmer Van Dyke, Christine Browning, Stacy Lippert, Lori Asher, Sophie Schonka, Debbie Krysiak, Dortha Atwater, Patricia Averette, Gael Silvia, Carol Kaehler, and Shawna Pugmire.

Very special thanks are due to Sarah Cariwyl Hebert for being such a wonderful, well-behaved, and patient little girl. Most important, all this work would have been impossible without the support of my husband, Richard Hebert. I am deeply grateful to him for his encouragement, his help, and his toleration of the entire ball of wax.

ABSTRACT

In this study, the researcher examined the state of usability testing in a small number of high-technology organizations, four of which did usability testing and two of which did not. The study examined the developing philosophy of usability testing, including evaluators' preferences for different test methods and organizations' justifications for not determining the quality of their manuals. The list of interview questions was created and refined during six preliminary interviews with working technical writers. The study was based on eighteen interviews with writers, documentation managers, evaluators, and software developers from Apple Computer, Claris, International Business Machines Corporation, Ashton-Tate, Tandem, and the National Magnetic Fusion Energy Computer Center at Lawrence Livermore National Laboratories. Organizations' testing efforts were limited, but not thwarted, by time constraints and by the complexity of the software being documented. Of the non-testing organizations, one made extensive efforts to ensure document usability; the other made no such effort but concentrated instead on customer education to make its products usable.

TABLE OF CONTENTS

CHAPTER

I	INTRODUCTION	1
	Justification and Background	1
	Software Instruction Manuals as Mass Communication	2
	Statement of the Problem	7
II	REVIEW OF THE LITERATURE	11
	The Cat In the Reaper	11
	The Background of Evaluation	12
	The Need to Test Instruction Manuals	13
	Evaluation of School Texts	15
	Dick and Carey's Methods of Evaluation	19
	Evaluation's Evolution to Usability Testing	21
	The Basics of Software Instruction Manuals	24
	The Benefits of Testing	27
	Preparing for the Test	30
	Preparing the Laboratory	31
	Doing the Test	33
	Evaluating the Results	34
III	METHOD	36
IV	RESULTS	43
	Organizations That Evaluate	43
	Organizations That Do Not Evaluate	69
V	CONCLUSIONS	84

Organizations That Evaluate	84
Organizations That Do Not Evaluate	86
General Findings	88
Directions for Future Research	91
REFERENCES	95
Interviews	100
APPENDIX	102
Appendix A. Researcher's Questions for Interviewees	103

TABLE

Table 1.	Aspects of Mass Communication	
	Applied to Technical Writing	5

CHAPTER I

INTRODUCTION

Justification and Background

After centuries of slow progress, the technical writing profession is at last developing methods of determining whether information is being communicated to readers--methods that can provide objective, reproducible results. Specifically, the software publishing industry appeared to have disregarded until the 1980s the fundamental theories of communication, all of which include the concepts of the receiver understanding the message and channel noise interfering with comprehension. Because of the industry's developing interest in communicating well with users, this thesis is concerned with the usability testing of software instruction manuals, which are crucial to the users' understanding of the software. Usability testing (also occasionally referred to as user edit, user review, formative evaluation, and documentation evaluation) consists of having representative subjects use the manual under observation.

Any publication created to impart practical information should be evaluated for its effectiveness: If the publication is not conveying information as it should,

test subjects make mistakes that could have been avoided. valuations help the writer improve the publication, thereby reducing frustration, wasted time, and unnecessary expense. The quality of any publication is important to its readers, but the quality of a technical manual is much more important than that of, say, a work of fiction. Reading a poorly written novel may waste some of a person's leisure time, but struggling through a poorly written software instruction manual wastes valuable work time, endangers data, frustrates both business and home users, and discredits the software publisher.

Documenting technical products uses knowledge from many disciplines: English, journalism, instructional design, psychology, computer programming, and graphic design. Information applicable to the testing of manuals can be found in articles and books from each of these fields. However, even if writers heed every word of worthwhile advice, they can determine reliably the quality of a manual only by having people use it under controlled, realistic circumstances.

Software Instruction Manuals as Mass Communication

Technical publications are those that have to do with anything mechanical, industrial, practical, or scientific. Technical writing explains how to do things, how things work, or how things are constructed. Software instruction

manuals explain how to use a computer program.

Instructions do not have to be printed on paper; the words of on-line instructions are "printed" on the monitor.

Mass communication is only roughly defined by the number of people who receive a message. Any method that delivers the same message to a large number of people is an example of mass communication, including word of mouth, but in deciding whether a message is an example of print mass media, one becomes aware of the philosophical gray areas of the term. Some things are not examples of print mass media, such as printed business forms, interoffice memos, and unpublished reports, because either they contain very meager information or are not circulated to a large number of people. Some things are obvious examples of print mass media, such as The New York Times, the book Joy of Cooking, and Byte magazine. These reach millions of people, use organizations for reproduction and distribution, and meet other criteria that will be discussed later in this section. At the low end of the mass communication continuum are newsletters, technical articles, software instruction manuals, grade school textbooks, special-interest paperbacks, scholarly journals, vanity press paperbacks, specialized directories, military publications, brochures, bill enclosures, and package inserts, all of which fit Schramm's eight criteria for mass communication (listed

below). Many technical magazines claim circulations less than 1,000. For example, American Currents, the organ of the North American Native Fishes Association, had a circulation of 400 in 1987.

The point at which a publication stops being mass communication is debatable, but Schramm (1973, pp. 115-120) lists eight aspects of communication that are useful for determining whether a message is an example of mass communication:

1. Which senses are involved.
2. What the opportunity is for feedback.
3. Who controls the pace of the flow of information.
4. Who has what degree of power to multiply the message.
5. Who has what degree of power to preserve the message.
6. What message codes are used.
7. How attention is held ("the power to overcome selectivity").
8. To what extent the sender is able to meet individual needs.

To show how technical writing meets each of these criteria, the author has constructed Table 1, below.

TABLE 1
Aspects of Mass Communication
Applied to Technical Writing

	<u>Memo</u>	<u>Report</u>	<u>Tech. Article</u>	<u>Software Manual</u>
<u>Senses involved</u>	vision	vision	vision	vision, touch
<u>Level of feedback</u>	high	high	low	low
<u>Pace controlled by</u>	sender, receiver	sender	sender	sender
<u>Who multiplies the message</u>	sender, receiver	sender, receiver	sender, receiver	sender
<u>Who preserves the message</u>	sender, receiver	sender, receiver	sender, receiver	sender, receiver
<u>What codes are used</u>	verbal	verbal, graphic	verbal, graphic	verbal, graphic
<u>Who decides attention</u>	sender, receiver	receiver	receiver	receiver
<u>Service to indiv. needs</u>	high	average	average	average

Noting the ability of a sender to multiply and preserve the message is part of deciding whether a message is an example of mass communication, but the other aspects must be assessed also. What are the opportunities for feedback? Who is most likely to give feedback and how quickly? Is there generous feedback, or does only a small

proportion of the audience respond? Is it the sender or the receiver who sets the pace? Who decides when and how to change the pace? How much power does the receiver have in deciding to attend to a message? To what extent does the medium serve individuals' information needs?

Memoranda and interoffice technical reports reside in the "personal communication" end of the communication continuum, while technical articles and software instruction manuals are at the "mass communication" end because they are directed to all interested people, not just to co-workers; they are reproduced in large numbers; the pace is determined by the senders; feedback is limited and not instantaneous; the senders have very little power to hold the attention of the receivers; the material has limited application to individuals' information needs; and the target audience is anonymous to the senders. Technical articles and software instruction manuals have all of the characteristics of mass communication and have little similarity to personal communication.

Mass communication textbook authors appear to agree on three major differences between mass communication and interpersonal communication: audience size, source of the message, and means of transmission. Although the consensus is that mass communication audiences are large, Severin and Tankard have rounded out that aspect of the definition by including the low end: "A country newspaper

that reaches a few thousand people is also engaged in mass communication" (Severin & Tankard, 1979, p. 8).

Most of the authors agree also that the source of mass communication must be an organization, but again, Severin and Tankard have a more comprehensive view, and point out that "there have been some one-person communication efforts that have reached fairly large audiences" (Severin & Tankard, 1979, p. 8). I.F. Stone's Weekly comes to mind.

Statement of the Problem

Modern technical writing has historical roots that are amazingly remote. For example, the concept of writing a single document for multiple audiences goes back to Frontinus, about 97 A.D. The idea of defining terms at the outset of a piece goes back at least to Vitruvius in 27 B.C. And although the ancients wrote clearly, the idea of clarity lost credence in the Middle Ages and yielded to blustery, ornate writing. The Royal Society of London in the 1600s complained about overblown writing and demanded plain, accurate language when discussing natural history. The first Renaissance-era technical writer to "stuff his pride and his university degrees into his pocket and to interview the working technician" (Miller, 1961, p. 9) was Agricola, a German physician of the 1500s.

The ideal instruction manual is one that is

understood perfectly by all its users--one that causes no confusion at any point. Pragmatically, however, the ideal manual is one that can be written in a reasonable amount of time, without inordinate expense, and that is understood well by almost all its users. These ideals can be approached with usability testing: The evaluators learn to make the manual easily comprehensible by observing the users' reactions and suggestions. This appears to be a simple idea, but it only gained attention in the education field in the 1960s, and in the software field in the early 1980s. Many organizations are still reluctant to put any organized effort into refining the manuals, even though these manuals are usually the only source of information that the users have about the product. Manuals, especially poorly planned ones, are very expensive to produce because of salaries and publishing costs.

Usability testing techniques were first formalized in the 1960s to test the quality of grade school textbooks (Gagne & Briggs, 1974). The techniques have been borrowed and refined to accommodate software instruction manuals, so that now, technical writers and course developers can choose from a range of techniques, from the very elaborate and expensive to the expedient and inexpensive. Whether the expensive methods are more effective is unresolved. It appears doubtful that any significant advantage exists

in the elaborate methods, since the reported results in the literature are quite similar for simple and elaborate tests.

The purpose of this thesis is to determine the state of usability testing in a small sample of high-technology organizations. The study included two non-testing organizations and several organizations that do test. One may expect simplistic rationales from non-testing organizations, such as lack of time, lack of staff, lack of funding, or perhaps responses such as, "We don't have to, so why should we?" Interviewees from each of the non-testing organizations were asked whether they were aware of the concept of usability testing, its costs, and its benefits. Included in the study were questions that helped determine whether their attitudes toward their customers were different from those of people in organizations that do test their manuals, as well as questions about what do they do to ensure the usability of their manuals.

In addition, the thesis examines the developing philosophy of usability testing. What are the testers' reasons for preferring one method over another? How do organizations justify not determining the quality of their manuals? The usability (and therefore the quality) of manuals is difficult to prove without usability testing, and the results of such tests are very helpful in

improving the manuals.

CHAPTER II

REVIEW OF THE LITERATURE

The Cat in the Reaper

An illustration of the importance of evaluating informational materials comes in an example borrowed from folklore. The story is of a young farm-machinery salesman who had sold a reaper to a group of European peasant farmers. They had seen how well the machine could cut their wheat, but the salesman found that they would not take him seriously when he cautioned them about the hazards of the machine; in fact, they were letting their small children play on the machine while it was running. He knew he had to make his warnings forcefully clear, so he picked up an unfortunate barnyard cat that happened to be wandering by, and threw it into the churning reaper. The bloody event shocked the farmers into belief; thereafter they treated the machine with due caution and kept their children away from it.

The salesman's verbal cautions were an ineffective safety presentation: He was not communicating to his listeners. If his presentation had been tested with representative users, he would have known beforehand whether it was effective, and he would have known that he needed to develop a better presentation. It is hoped that after that episode he developed a presentation that didn't

require a supply of spare cats.

The parallels between this story and the current state of technical writing are clear to those involved in the profession. Technical writers are learning that untested materials give variable results, that poorly designed learning materials waste time and effort, that problems are found through simple observation, that problems can be remedied quickly and inexpensively, that effective materials can be developed and tested in a reasonable span of time, and that the burden imposed by poor documentation is borne by the user--the modern-day cat in the reaper.

The Background of Evaluation

The purpose of formative evaluation is to improve learning materials before distribution by having test subjects use the materials under observation. Formative evaluation contrasts with summative evaluation, which is done after distribution. The three kinds of formative evaluation are one-to-one (clinical), small-group, and field evaluation.

The most informative sources on evaluation are those that were designed for the original territory of evaluation, grade school instructional materials.

Instructional materials are:

film, television, radio, audio recordings,
photographs, projected visuals, printed materials and

the like [with a] deliberate arrangement of experiences to help a learner achieve a desirable change in performance. (Heinich, Molenda & Russell, 1982, pp. 8-9.)

The value of evaluating materials was recognized in the 1960s when the U.S. government was generously funding curriculum development projects. Taxpayers and politicians wanted to know how well the money was being spent, and were disappointed by the answer. To improve the situation, educational researchers suggested that materials be tested for effectiveness prior to printing and distribution. The object of this formative evaluation was, and is, to help writers revise intermediate drafts of learning materials to optimum effectiveness instead of distributing flawed materials to the field (Scriven, 1967). Even so, until the 1970s, grade school instructional materials almost always went straight from the author to the typesetter, with no proof of the materials' effectiveness (Dick & Carey, 1978). Many teachers and students would be righteously indignant to see Dick and Carey's remark that "too often instructors and students have been blamed for poor learning when in fact the materials were insufficient to support the instructional effort" (Dick & Carey, 1978, p. 172).

The Need to Test Instruction Manuals

The inadequacy of many manuals is made apparent by

the existence of articles such as "Losing the Fear of Manuals" (Rony, 1983), "Getting Help" (Levy, 1984), and "The Tyranny of Words" (Burgess, 1984); which are only three of many complaint pieces that tell of data loss, frustration, and wasted time. Each such article offers suggestions for improving manuals, usually accompanied by forcefully stated reasons. Another indication of the inadequacy of manuals is the existence of companies such as Datamost, Inc., and Nexus, which publish replacement manuals for software products (Rony, 1983). Also, several software publishers provide technical assistance by telephone, which they would not have to do if the manuals were adequate. The problem even extends to educational software, some of which is "classroom tested," most of which is not. Some grade school teachers, when handicapped by inadequate manuals, rewrite them to meet their students' needs (Drotter & Roth, 1984).

The quality of manuals appears to be improving, but there was still a good market for manual replacement books in 1989 according to Gael Silvia of Ashton-Tate and Dorteia Atwater of Claris. Whether manuals actually are better now than they once were is difficult to determine in the absence of quantitative and qualitative studies.

As the reader will see throughout this thesis, there are many variations on the basic usability testing idea, but the intent of each is to improve readers'

comprehension. For instance, Marshall (1979) suggested that materials be written to the seventh or eighth grade level. However, readability and comprehensibility are not identical. Marshall cites an example of a textbook that had a readability level of 8.5 and was more difficult to understand than a similar book at the 12.5 level. She analyzed the text and found that when connectives, pronouns, and vocabulary are reduced, "cohesion is lost . . . [and] comprehension becomes much more difficult because the reader must infer the missing information (Marshall, 1979, p. 542)." As Robert Gagne (1967, p. 112) stated succinctly: "What one really wants to know about a given curriculum is whether it works."

Evaluation of School Texts

Formative evaluation is the determination of the quality of "an intermediate stage in the development of the teaching instrument" (Scriven, 1967, p. 51). The purpose of formative evaluation is to improve learning materials before distribution by having test subjects use the materials under observation. Formative evaluation contrasts with summative evaluation, which is done after distribution. The three kinds of formative evaluation are one-to-one (clinical), small group, and field evaluation.

One-to-one evaluations can be inexpensive, easy to

do, and brief. Dick and Carey (1978), whose work helped form the foundation of usability testing, wrote an extensive description of formative evaluation for grade school materials, and recommended using more than one clinical evaluation because of the magnitude of intervening factors such as the test subjects' ability and the teaching ability of the evaluator. Wager (1983) suggests that the evaluator should use at least two one-to-one evaluations using one participant who is a slightly above-average subject and one below-average subject to achieve more reliable results.

Clinical evaluation is an efficient approach that is ideally suited to any situation with budget and time restraints, since reliable results can be gained by using very few test participants. This is why it has gained acceptance among the organizations that make the effort to improve their manuals in a verifiable manner.

Problems revealed in the one-to-one evaluations are mechanical errors such as typographical errors, missing pages, missing graphs, incorrect captions; and stylistic errors that interfere with the clarity of the instructions. The materials should be revised for the next subject so that the evaluator can use the time to best advantage, and so that the next subject has the opportunity to test the new changes and perhaps reveal other problems with the materials. The person evaluating

the materials should allow the subjects to put effort into absorbing the material; evaluators must also avoid being too solicitous, yet remain easily available.

After choosing test subjects who are typical of the anticipated users, Dick and Carey (1978) suggest that the evaluator sit with one subject at a time as the subject uses the materials. The evaluator encourages the subject to mark the materials and ask questions; meanwhile, the evaluator keeps precise notes on what changes the writers should make, including these elements: The rate at which information is presented in the text; the level of detail; changes in examples, illustrations, text, vocabulary, and phrasing; and the arrangement of the information.

Dick and Carey advise that small-group evaluations use a representative sample of subjects and that they be pretested and posttested. The idea of pretesting was developed for use with grade school children who were divided according to their performance on the pretest. In the absence of pre- and posttests, all the test-learner variables are unknown, and there is no support for the validity of the results. Both the pretest and posttest should be done "in their intended format" (Dick & Carey, 1978, p. 178). Pre- and posttesting were designed for use with students who were engaged in reading-to-learn activities (as opposed to reading-to-do). Reading material for learning concentrates on topics, while

reading material for doing specifies action (Redish, 1988). Persons who are reading to learn make a variety of efforts to help themselves retain the material, while those who are reading to do concentrate on the task instead of on retention. Software users, for example, know that they can refer to the material anytime, so posttests should be designed to accommodate that fact. Even among software users, some are reading to do and others are reading to learn, and the same people can be using either technique depending on what they want to accomplish. For instance, if a person wants to remember a frequently-used boot-up sequence, he or she may try to retain the information by taking notes or repeating the sequence several times, while a person who almost never boots up may simply follow the directions and make no effort to retain the information because it is so infrequently used. The test should evaluate the quality of the manual rather than the memory of the subject; for example, subjects working on word processing tasks can demonstrate how quickly they can find the information necessary to merge documents, or the test can determine whether users find certain passages confusing.

These issues inspired Redish's article, "Reading to learn to do" (1988) on materials "that help users both to use a program and to learn (p. 223)," such as tutorials. Redish stated that:

A tutorial should help users to: Gain a basic understanding of the concepts and structures of the program; become comfortable with the program so that they will want to continue to use it and be satisfied with it; be able to perform basic, relevant tasks with the program after using the tutorial; [and] transfer what they have learned from the examples in the tutorial to other situations that were not directly covered in the tutorial. (p. 226)

Dick and Carey's Methods of Evaluation

Although these methods were designed to evaluate grade-school textbooks, Dick and Carey greatly influenced the development of usability testing, as seen in the research method discussed later. However, their influence on the quality of manuals is limited, since not all organizations do usability testing. Small-group evaluations as described by Dick and Carey involve a representative sample of test subjects who should be pretested and posttested.

When the material is tested with a small group of users, Dick and Carey advocate that the evaluator helps only when a subject asks for assistance. Then the evaluator notes the area of difficulty and the way the problem was resolved. Since the purpose of small-group evaluation is to test the materials in a situation that closely resembles typical usage, schools use the regular classroom setting, and usability testing done by software companies with adult test subjects also uses settings that

resemble those of the typical user, including such aspects as location, number of people who might offer help, noise level, sample tasks, deadlines, and telephone assistance. The test subjects themselves are chosen to match the typical user profile in experience, education, and any other characteristics that may be important to the study. Users are encouraged to mark all the troublesome spots in the materials so that the evaluator may remedy them.

Small-group evaluations should be done with 10 to 20 test subjects to ensure adequate representation of population subgroups, yet avoiding the collection of superfluous data. Dick and Carey contend that "the data from additional students will not provide you with a great deal of additional information for careful evaluation and analysis in small group settings," (1978, p. 169). A small-group evaluation more closely resembles the target population's normal learning environment because the evaluator's teaching influence is almost absent, very much like the new owner of a software package sitting alone at home with only the manual and an 800 number for help.

Field evaluations are similar to small-group evaluations, with the sole exception that the group is larger (30 or more test subjects). Evaluators use subjects who are typical of the target audience and use pretests and posttests.

Dick and Carey (1978, p. 161) also suggest an

attitude questionnaire with typical questions:

1. Were the sections too long? Too short?
2. Was there any part that was too hard?
3. Were the diagrams and pictures helpful or confusing?
4. What would you change in the manual?
5. What would you change in the program?

These questions, asked after the posttest, are designed to help the researcher get feedback that may not come from the evaluation.

Evaluation's Evolution to Usability Testing

The procedures described for the evaluation of grade-school textbooks reflect the government funding used to pay for their implementation: They called for an enormous amount of staff time and large numbers of test subjects. Four course developers at Bell Laboratories (Bastian, Edward, Medsker, & Schimmel, 1983) wanted a method of evaluation that would be less time consuming than the Dick and Carey method, so they combined the small-group and field evaluations into a modified field trial. Both trials used pretests and posttests, and both used volunteer test subjects since the developers could not use random selection. The local trial used thirty subjects and the field trial used about four times that number. It is not surprising that these course developers found the whole process extremely time consuming, even with

compromises on the number of evaluation stages, the sample of subjects, the completeness of data, and the control of test-taking situations.

According to other researchers, however, organizations do not need to go to such lengths to improve their manuals. One study showed that, if the materials are clinically evaluated with a few people of various abilities, the final draft of the material imparts information quite efficiently (Wager, 1983). Kandaswamy, Stolovich, & Thiagarajan (1976); Carey (1978); and Wager agree that if the clinical evaluations are thorough, small-group evaluations "may not be necessary for the effective revision of instruction" (Wager, 1983, p. 7). Presumably they would also agree that field evaluations are unnecessary, too. This approach addresses the concern of diminishing returns: How much evaluating is necessary to have a good document? Companies would much rather get the document to the customer than spend ever greater amounts of time and money improving it. Indeed, by 1984, the ideas of small-group and field trials were apparently abandoned for intense, one-to-one evaluations. The vestige of field testing is the occasional inclusion of questionnaires in the manuals.

Evaluation methods for grade-school instructional materials were one way to verify that the manuals were effectively imparting their information. Another example

of formative evaluation is perhaps the first true usability test. Alan Neal and Roger Simons of IBM wanted to evaluate manuals without affecting the user or the software, so they used slaved terminals and videotape as the subjects used the software. Although their method is similar to Dick and Carey's clinical evaluation, it removes the evaluator from immediate contact with the subjects, and directs the evaluator to solicit test-learner feedback "only after [the subjects] have had some practical experience with a product" (Neal & Simons, 1984, p. 83), which unfortunately means that the subjects work out the manuals' problems away from the observation of the evaluator, rather than giving the evaluator the chance to find out where those problems are. In some instances, the videotape would be played back in the presence of the subjects so that evaluators could ask the subjects questions. Dick and Carey advocate that the evaluator stay with the subjects throughout the test to uncover every troublesome spot without reliance on memory. With modern, affordable equipment, however, the evaluator can record the subjects' every move, and has the advantage of replay. At any rate, subjects who have worked through the rough spots on their own are much less valuable test participants because they may forget which spots gave them trouble, forget many of the suggestions that occurred to them, or feel embarrassed to admit that they did not

understand things that appear obvious later.

Neal and Simons' article is valuable for its suggestions that the objectives and scope of the test should be clear, and that quantifiable items can help define problem areas. For example, the test can cover the whole product or part of it or can concentrate on command syntax, the accomplishment of discrete tasks, the use of menus and indices, the frequency of each error message, and the time a participant takes to learn the entire package or part of it. This underscores the message that the test should be task oriented; that is, it should test the instructions and not the subjects. If the subjects are not able to accomplish the assigned tasks, then the manual should be revised.

The Basics of Software Instruction Manuals

Depending on the users' intent, reading the manual is usually either reading to do, reading to learn, or as Redish (1988) points out, reading to do to learn. Of those who read to perform a specific task, three fourths would expect to read the material again the next time the task comes up (reading to do), while those who read to learn make an effort to help themselves retain the information (Seifert, 1979). Since the users decide whether to read the instructions once, occasionally, or every time a particular task comes up, the manual should

be designed to accommodate any style of learning. Manuals have both learning and retrieval functions, so the manuals should have easy-to-follow directions that are also easy to find; if not, the users avoid using the manual and rely instead on coworkers, managers (Mirel, 1987, p. 290), or experimentation. Mirel observed users for two hours a week over a three-month period, and never saw the users refer to the manual, even though they had many difficulties (1987, p. 290). The users consulted each other instead.

As users become more familiar with the program, those who use the manuals use them differently. Novices need tutorials and a quick reference section, while experienced users need a more detailed reference section (Maynard, 1979). A continuing challenge for manual writers is to write one manual to serve two or more audiences.

Testing the documentation can also help writers accommodate different learning styles. Redish (1988) found:

two types of learners: those who want to play with the program and who are happy to try out their own examples, and those who want to have their hands held and who are only comfortable when we give them specific examples to use. (p. 228)

Even so, the study by Carroll, Smith-Kerker, Ford, & Mazur-Rimetz (cited in Redish, 1988) experimented with a "minimal manual"--a set of cards, each of which told how

to start one task and what to do if something went wrong.

Redish (1988) noted:

In a comparison test with the original, commercial manual, users who learned with Carroll's cards spent half as much time in learning, committed half as many errors, recovered more often from their errors, and did much better in transferring the learning to other tasks. (p. 228)

Users prefer function-oriented manuals over product-oriented manuals (Maynard, 1979). (Product-oriented manuals are those that are divided into sections according to the structure of the product, such as file manager, linker, loader, and job controller. Function-oriented manuals derive their organization from tasks that users do, such as system start-up, creating files, printing, and so on [Maynard, 1979].) A product-oriented manual demands an experienced user; that is, one with enough computer background to figure out the manual and the product through inference. Duffy (1983) noted that even experienced technicians scored poorly when they had to use such a manual for trouble-shooting because they had to infer the steps to be done. However, when novice and experienced technicians used a function-oriented manual, both groups scored in the 95th percentile.

Foley described the effects of product orientation:

1) The manual's organization hinders the search for information, 2) narrative text forces the user to infer the procedural steps, and 3) trouble-shooting information

is often left out while irrelevant information is included (cited in Duffy, 1983). Foley also noted the lack of instructional design influence:

No explicit user-oriented design standards are provided for the level of detail to be included [in the manual], for illustrations required, or for the length, complexity, or vocabulary to be used in the description" (cited in Duffy, 1983, p. 159).

All conscientious writers construct manuals the way they believe most effective, but every individual sees problems in a different light. Intuition is a valuable guide to writers, but materials that are revised according to formative evaluations perform better than those that are intuitively revised (Baker, 1970).

The Benefits of Testing

Software users benefit most directly from usability testing since it reduces the tormenting frustration caused by poor directions. The main purpose of most usability testing is to make the software and manual easier to use, but many users would concur on the humanitarian aspect of producing comprehensible instructions. Mills and Dye (1985) note that:

some of the major computer companies now have usability testing as a requirement for all their computer documentation. This occurred because so many consumers complained about documentation quality. (p. 40)

Software organizations' writers also benefit from usability testing. Writers learn to make their manuals

easy to understand, technically accurate, complete, consistent, and well organized. The testing gives writers a more objective view of their work since it increases feedback in a low-feedback communication process. If the testing is done early, writers can change their methods before much money and effort is wasted on creating ineffective documents. Also, technical writers' time is used more efficiently since they can work on new projects instead of revising poorly written ones (Hernandez & Wilder, 1984).

Software publishers and their retailers benefit from usability testing by having a reduced number of consumer questions, complaints, and service calls (Mills & Dye, 1985). Gillihan and Herrin (1988) noted that post-distribution evaluation has three advantages: No market introduction delay for product improvement, customer feedback from actual users, and verification of the accuracy of the target audience definition. They also note, however, that post-distribution evaluation puts the company at risk for alienating customers. Mail surveys conducted with reader response cards included in the manuals have notoriously poor response rates and do not provide representative data. Post-distribution field research is at least as expensive as pre-distribution usability testing, and customer organizations may be reluctant to participate because it consumes their

employees' time (p. 171).

Evaluating manuals after they have already been distributed with the product has three main advantages: Companies can get a product on the market almost as soon as the product is ready; they do not have to delay the product introduction date because of manual revisions; writers can get input from customers who have used the manual with the product, rather from test subjects assumed to be typical of users; and writers can verify the accuracy of their audience analysis.

As evaluators develop a body of data about their manuals, they can develop ways of determining the quality of future drafts. Another little-recognized benefit is that product designers and software engineers can use the test as a life test of the equipment to find whether problems exist in the hardware or software (Cunningham, 1984).

Kleid presented a compelling argument for usability testing:

We've seen reasonable people destroy diskettes on which programs were shipped because they had never seen diskettes before and we didn't clearly explain how they were packaged. We've seen ingenious people create their own incorrect procedures because they couldn't see reference diagrams that had become obscured by movable parts of their machine. We've seen hard-working people become completely stopped by unnecessarily complicated procedures that could easily be redesigned. In every case, the information and product developers observing these problems have come away determined to make things easier for their customers. (1984, p. MPD-64)

On occasion, even the most elaborate test set-up will pay for itself with the improvements to one manual. Also, if the users of a product are found to be more experienced than expected in the user survey, a shorter manual written at a higher grade level will serve their needs and will be cheaper to produce (Kleid, 1984).

Preparing for the Test

The methods of usability testing have not been firmly decided on by technical writers in the field, so the starting point for test preparation in the product development cycle is yet unclear. Cunningham (1984) suggested that the test administrators start by writing test procedures and then going on to prepare the room in which the manuals will be tested. However, no one who was interviewed for this study manages test preparation so informally. Horton (1984) suggested extensive preparation before testing begins, including using the company's marketing survey (assuming it has done one) as the information base for a rudimentary telephone survey. In the absence of a marketing survey, he suggested sending out a questionnaire to the perceived target audience and culling telephone numbers from that. The advantages of talking to customers are already known to the mass communication researcher: The interviewer can get more of

can ask open-ended questions and expect coherent answers, and from their tone of voice the interviewer can interpret their level of interest, confusion, hostility, and so on. Horton goes on to suggest that the evaluator interview the respondents in their work environments to learn what they want from the product and from the manual, and how they will use both.

Hernandez and Wilder (1984) advised evaluators to start by defining and choosing the users' tasks for the test, since testing the entire manual is much too expensive and time consuming in most situations. One may wonder about the value of testing a miniscule portion of the manual: The purpose is, as Hernandez and Wilder noted, to do the testing early in the life of the manual so that the writers can see the genre of their errors. Observing users' reactions to each part of the instructions helped the writers anticipate the problems the users would have. In other words, the users taught them how to write.

Preparing the Laboratory

Since usability testing of manuals is often done with two videotape recorders, advance preparation of the laboratory may be necessary. In the case of software manuals, Kleid suggests (1984) that the room should have a table, a chair, electrical outlets, adequate lighting and

table, a chair, electrical outlets, adequate lighting and ventilation, a computer terminal, and perhaps a printer and other auxiliary devices. It should be a place where the subjects can work without unusual distraction, and where curious people cannot peer in and create a nuisance.

In addition to the multiple video cameras, Kleid also advocates using a room with a two-way mirror, four video cameras, two unseen evaluators, soundproofing, microphones, and audio recording equipment. One may question such extreme measures to document a usability test, but future research could resolve the question of how much is enough.

After the laboratory is equipped, the evaluator should do a practice run with a co-worker to make sure that all the equipment is working well. Evaluators do not agree on how many subjects to use. The absolute minimum number of subjects is two; others use up to fifteen. The average appears to be about five or six.

When the first subject arrives, the evaluator gives him or her the pretest to establish the person's ability and experience. The pretest may consist of a brief background questionnaire and simple commands to do a few basic tasks on the computer, such as turning it on, loading a program, creating a document. Depending on the product's target audience, it may be no cause for concern if the subject is unable to do anything with the computer.

The evaluator then explains that the manual is being tested, not the subject, that the subject can take breaks and is welcome to ask questions, that the cameras hum, and that he or she should begin at the evaluator's signal and not before (Cunningham, 1984). The evaluator avoids telling how long the test will take because that creates a time limit and a standard that the test subject may measure himself or herself against. The person may be afraid of being judged incompetent if the time limit approaches and the task is incomplete.

Doing the Test

Usability test procedures should follow the scientific methods used in any other test. For instance, the evaluators should not change test procedures after the test begins unless a flaw is discovered, and information given to test subjects about the test should be the same for each subject. The evaluator may record the time it takes each subject to complete each task or give up; the number of assists for each task; the reason for each assist; the amount of time the test learner spends reading; the amount of time the subject spends doing the task; the number of times the subject refers to the table of contents, the index, and the glossary; whether the subject does the practice exercises, and any comments the subject has (Cunningham, 1984).

Qualitative aspects of the test surface when the evaluator is not overly eager to help the subject. As the subjects resolve the problems their own ways, the evaluator observes the subjects' methods of resolving those problems, and can interpret them as suggestions for ways to remedy the problem in the manual. For instance, where do the subjects look for answers? Do they use a logical strategy such as using the index? Do the subjects have to flip back and forth several times before finding the answer to each question? Do they try to solve their problems by experimenting with the program or by using the manual (Horton, 1984)?

Tasks that might appear in a usability test of a software instruction manual include entering data, running programs, creating files, updating files, and printing documents. When the manual is designed to be used by programmers, one might test for tasks such as designing, coding, compiling, and debugging (Waite, 1984). The intent is to make the test include tasks that the user finds important. If the manual does not provide sufficient guidance for accomplishing those basic tasks, it is flawed.

Evaluating the Results

For the posttest, Cunningham (1984) advocates that the evaluator administer the same test used for the

pretest, and compares the results to get an indication of how effectively the manual helps the user. The evaluator then asks for the test subject's comments, such as likes, dislikes, what he or she found confusing, what was simple, and what specific problem areas there were, always using nondirective questioning techniques.

If there is more than one evaluator, all the evaluators meet to identify things that hindered the test, improved it, might improve it, or did improve it. They also talk about mistakes the staff made, mistakes the test subject made, test facility problems, new ideas that worked, and ideas for future testing (Cunningham, 1984). The evaluators then update the manual for the next subject, who uses the manual in its new version, and the evaluators go through the entire cycle again.

CHAPTER III

METHOD

The research for this thesis was accomplished in two parts: Six preliminary interviews conducted in 1986 and 1987 with working technical writers and managers, and eighteen interviews in 1989 with employees of four organizations which do usability testing and two organizations that do not. Of the preliminary interviews, all were done in person by the researcher, and all but two took place at the interviewees' workplaces. Of the eighteen later interviews, fourteen were extensive personal interviews done at the interviewees' workplaces, two were brief personal interviews (Darnell and Howd), one was a brief telephone interview (Grossman), and one was a long-distance telephone interview (Averette). The preliminary interviews helped determine whether the thesis question has value to technical writers; also, the list of interview questions was created and refined during the preliminary interviews. Some questions proved to be redundant or tangential and were dropped.

In the six preliminary interviews of people who worked for software companies, this researcher found only one person who had used any form of evaluation. He said:

What we used to do was, after we had finished a somewhat final run-through of the manual, we'd get

[the least computer-literate person] in the office . . . to try to use the program just by using the manual, because that way we knew for sure that anybody could understand it. (Ira Ellenbogen of Nanosecond Incorporated, personal interview, April 2, 1986.)

Yet Ellenbogen, now self-employed, said that his new company's software was intended for advanced users who would "understand what we meant even if we didn't say it clearly," which echoed the sentiments of several others. Another, whose product was intended for people with little or no computer experience, strongly resisted the idea of taking as much as two days to improve his company's mainstay product, explaining, "We're going to put out another manual six months later anyway (Scott J. Mansfield of Cable Data, personal interview, May 10, 1986)."

This study used personal interviews with technical writers, documentation managers, and evaluators (where applicable) to learn what their organizations were doing to ensure the usability of their software instruction manuals. The questions in Appendix A reflect what the author expected to learn, and provided starting points for further discussions.

In all cases, the interviewer used the guidelines for questionnaires described by Comstock, McCombs, and Christians in Research Methods in Mass Communication (Stempel & Westley, Eds., 1981); and Babbie's guidelines in Chapters 7 and 9 of Survey Research Methods (1973).

The questions were pretested with two extensive interviews to eliminate ineffective and superfluous questions.

The organizations participating in this study that test their documentation for usability were:

International Business Machines' Human Factors Center in San Jose, California; Claris in Mountain View, California; Apple Computer Corporation in Cupertino, California; and Ashton-Tate in San Jose and Torrance, California. The organizations were chosen for their test policies and their accessibility to the researcher. The most complete testing was done at IBM, where the employees had the most equipment and the most support from upper management, as detailed in the Results section. The other three test-using companies, when considered individually, were unable to provide the researcher with detailed information for this study. Since Ashton-Tate's test facilities are in Southern California, it was not possible for this researcher to do personal interviews there. Claris, as of June 1989, had not actually begun doing its own tests; it was hiring contractors to do occasional tests. Apple only tests its in-house training manuals and leaves the testing of user manuals to its subsidiary, Claris. However, considered together, Apple, Claris, and Ashton-Tate were the source of much useful information. The researcher hopes that this made the study more representative of industry practices.

Information on usability testing procedures at IBM's Human Factors Center is based on personal interviews with Sophie Schonka (April 20, 1989) and Debbie Krysiak (April 27, 1989), who are human factors engineers; Mike Darnell, software engineer (April 27, 1989); and Lori Asher (April 27, 1989), senior associate technical writer. The Human Factors Center is a 27-year-old facility in the Santa Teresa area of San Jose and was started by IBM's Richard Hirsch, who is regarded as one of the founders of the human factors field.

The information on Claris and Apple was obtained in personal interviews with Shawna Pugmire (February 16, 1989), training coordinator at Apple Computer (Claris' parent company); Carol Kaehler (March 2, 1989), software design specialist at Apple; Diane Besterman (February 16, 1989), instructional design contractor at Apple; and Dortehea Atwater (March 23, 1989), publications manager at Claris. Claris is a wholly owned subsidiary that creates and publishes Apple's software. Atwater came to Claris after several years of doing usability testing at Ashton-Tate, and as of this writing is establishing a full-time staff of five writers and one usability evaluator.

Information on Ashton-Tate was obtained in a personal interview with Gael Silvia (April 6, 1989), who is a documentation manager, and in a long-distance telephone interview with Patricia Averette (June 30, 1989), who is a

senior instructional technologist. Silvia recently transferred to Ashton-Tate's Northern California Product Center after several years of running Ashton-Tate's usability testing facility in Southern California. Silvia's background includes a master's degree in educational psychology and experience as an instructional designer, creating interactive tutorials. She also designs usability tests and decides the design and content of many of Ashton-Tate's software manuals. Averette has a master's degree in instructional technology and ran the usability test facility at Ashton-Tate's Torrance plant for two years.

The two organizations participating in this study that do not test their documentation for usability were the National Magnetic Fusion Energy Computer Center at Lawrence Livermore National Laboratories (called MFE by its employees) in Livermore, California, and Tandem of Cupertino, California.

The information on MFE's methods of documentation comes from personal interviews with Dr. T.R. Girill (March 23, 1989), who heads the documentation group at MFE and has done so since its inception in 1975; Russell Kahn (March 23, 1989), who worked at MFE for several years and is now a documentation specialist heading the Technical Information Department at the Open Computer Center at Lawrence Livermore National Laboratories; and Palmer Van

Dyke (February 22, 1989), who has been a technical editor at LLNL for 10 years. Kahn has an M.A. in journalism and an A.A. in computer science. Van Dyke has a degree in English and many years of experience as an engineering writer.

The information on Tandem's documentation process was obtained in personal interviews with Christine Browning, first-line manager of software technical publications (February 17, 1989); Stacy Lippert, technical writer, (March 3, 1989); and Mary-Leigh Burke, senior technical writer, (March 9, 1989). Information about Tandem was also obtained in a brief personal interview with Sheri Howd, documentation manager (March 9, 1989); and in a brief telephone interview with Ann Grossman, software developer (April 5, 1989). In February of 1989, Browning's staff had 25 to 30 manuals in progress, all of which were reference manuals or user guides for database management products for applications programmers, not end-users. Burke is a software technical writer who researches and writes manuals, uses the software, and writes sample tasks to help users learn the software. She has been at Tandem for five years, has a master's degree in Italian literature, and has taken several programming classes. At the time of the interview, Burke was documenting Nonstop SQL, a database management system and report writer. ("Reports" in this context refer to

spreadsheets or tabular information.) The users of this product are application programmers who use the product with other programs to write new programs for end-users. Lippert was a production editor for one year, then was a technical editor. As of this writing she is a technical writer developing the manual outline for a new product called X400, which is database management software.

CHAPTER IV

RESULTS

Organizations That Evaluate

The organizations that evaluate their software manuals for usability are IBM, Apple, Claris, and Ashton-Tate. The information gained from these companies provided information on a wide range of usability-testing practices. Some individuals used very formal, scientific methods; others used informal, common-sense methods, others used a combination of methods. Some tests were very expensive, some were inexpensive, some used five subjects, some used fifteen. Only one company was actually testing at the time of the study; the others were not testing locally or had not begun testing.

IBM's usability testing at the Human Factors Center.

IBM had a staff of psychologists who did extensive human factors testing. The psychologists had an enormous amount of equipment at their disposal, as well as access to a very large pool of test subjects. IBM's staff tested more of each manual than other companies in the study.

Information on IBM was obtained in personal interviews in April 1989 with Sophie Schonka, Debbie Krysiak, Lori Asher, and Mike Darnell.

Schonka, who has a master's degree in experimental psychology, tested products and their associated

documentation. The product she worked on most recently was Prompted Query, which was designed to teach people about query languages, especially people who know nothing about them. First, she learned how to use the product, and then set up usability testing tasks. She then had people not knowledgeable about query languages come into the testing lab and work through the Prompted Query training manual. She then gave them tasks to accomplish by using the manual to determine whether they could perform the tasks, which would tell whether the manual was good enough to teach people how to use the product. She did not videotape the tests, but rather used an improved version of Playback (see Neal & Simons, 1984), called the Software Usability Evaluator (SUE). This was a program that presented the tasks to the test subjects, recorded the keystrokes from subjects, copied the screen for the psychologists (also called human factors engineers or evaluators) to see in another room, and recorded the evaluators' typed comments. The SUE kept the two sets of data separate but could also combine data to show which events inspired which comments. These data could be printed out or replayed separately or together. Schonka explained that this feature was also useful for demonstrating bugs to software developers and for giving evaluators a second chance to add to their observations. Although the laboratory has video recorders, the

evaluators were not interested in subjects' facial expressions as some writers on this subject recommend; IBM's evaluators concentrated instead on seeing what the subjects were seeing and typing. This indicated that the evaluators relied on long pauses, mistakes, and oral comments from subjects to signal problems, and that they were satisfied with having a workable product, as opposed to having a very-easy-to-use product. The evaluators avoided assisting the subjects unless the subjects clearly needed help. When they did offer help, it was clearly recorded so that the evaluators knew what they had told the subjects and why.

The target audiences for products developed and/or refined at the Human Factors Center were defined by the software developers, since the products were designed to be used on mainframe computers and, therefore, to be used by database administrators, programmers, and system analysts. The software developers were already familiar with the characteristics of those target audiences and so did not consult marketing surveys. Schonka explained that there is a well defined group of people who are the target audience for each product, which gives the software developers, writers, and evaluators a good idea of the target audiences' capabilities.

Occasionally, the Human Factors Center has customers inform the staff of their problems with the product.

These meetings provide information that is used either to revise the product or to guide the development of new products. More formal sessions, called "Guide and Share" meetings, have the staff going to customer sites with similar goals. Some of their products include early customer support, which has customers sign non-disclosure agreements and then come to the Human Factors Center to comment on the product during the development phase.

IBM's commitment to usability testing has made it prominent in the industrial use of human factors testing, although even the interviewees conceded that their user manuals could be better. Indeed, a study that compared IBM's Manufacturing and Production Information Control System (MAPICS) to System Query Language (SQL, a standard language) found that "users of the menu system, IBM's MAPICS, complained about crowded screens, and found it hard to interpret some of the menu choices" even though it is "advertised as user-friendly . . . and is intended for installation and use by people having no special training. (Davis, 1989, p. 447)."

Prompted Query is a very small part of a much larger package, QMF (Query Management Facility). QMF is much too large to be tested completely, but Schonka was able to test all parts of Prompted Query. Whether a package was tested in its entirety depended not only on size, but also on what the software developers wanted to know about a

product's interface. A recent question was whether the manuals for DB2, a database management system, should be in five medium-sized manuals or one big binder. In that case, the evaluators and writers tested an enormous amount of the manuals because they needed to find out whether people preferred the big binder or the five separate manuals and which format allowed users to find information faster. For products that were not being completely tested, the staff chose those items that were new, and those items in the software or the documentation that had changed.

The usability test for DB2 was a "look up" test; that is, it consisted of having the subjects find information in the manuals to accomplish tasks with the software within a specified time, in this case, 30 minutes. Some tests ran four to six hours; others had the same subject coming back for three days. Some tests required the subject to spend several days reading about the product to be knowledgeable enough to use it.

Most tests at the Human Factors Center used six subjects, unless the subjects' responses were too varied, in which case more subjects were used so that a pattern of responses could be observed. Widely varied test results indicate a problem with the test design. Consistent results are needed so that the evaluators can learn where the problems are in the manual.

The evaluators started testing the interface as early in the development cycle as possible, which often meant testing fake software (prototypes) to discover where problems will be and to determine the best solutions. Each of the evaluators was assigned to a group of products and worked with the software developers so that they knew when the developers were starting to think about new functions for each product. The evaluators then created prototype screens to look like what the software developers were considering. The evaluators then used those screens, which had no underlying function, to learn whether the developers' ideas are valid.

Schonka stated that she and most of the other evaluators worked well with their software developers and had little difficulty with the technical-versus-nontechnical communication barrier that is often experienced by writers and evaluators. Some teams did have difficulty, which Schonka attributed to people not knowing each other well enough. Regardless of the quality of the relationship, the evaluators' job was to make the products usable, and if they saw something in the interface that was troublesome, they had the option of refusing to give Human Factors' approval to the product, which all products in the division must have. Schonka noted:

It can get kind of messy, but that's not to say that we don't have some give and take. Sometimes people

are up against deadlines and things fall through the cracks, so we have to compromise also, but at least it's documented so in the next release the problems are resolved.

The evaluators ran the tests on each other to discover whether the tests gather the targeted information, to verify that the procedures were correct and that the setup was working, and to estimate the amount of time each test would take. They then used one or two people as pilot subjects to determine how variable the results would be. By the time testing started, the evaluators felt they knew the product very well.

The test subjects must be representative of the product's target audience. At IBM, this was accomplished by using target-audience descriptions created by the software developers, and by dictating to purchasers the characteristics of the narrowly defined target audience. IBM's contracts, like those of some other large organizations that produce extremely technical software to use with very expensive computers, specify what kinds of personnel the customers must have to run the software. For instance, organizations that are using DB2 must have a system analyst on staff. According to Asher: "Otherwise, they don't get the software." This obviates the need for marketing surveys for this type of software, although marketing surveys are still needed for mass-market software.

The FrameNet product required people with good knowledge of English, two years of programming experience, and no knowledge of artificial language or the existing FrameNet product. IBM evaluators used in-house people or contractors from temporary employment agencies. If the test ran more than four hours, evaluators usually had to use the agencies because in-house people did not often have that much time to spare.

The evaluators stated that test-subject motivation presented no problems. Schonka said that those from the temporary agencies treated the testing as their source of employment, in-house employees saw the testing as a chance to influence the product, and outside users "may have an ax to grind, so there is no one who holds back" in letting evaluators know products' flaws. The evaluators told the test subjects during the orientations that, if at any point they did not want to continue the test, they were free to leave; however, Schonka said that only once has she had a test subject discontinue the test, so the degree of freedom perceived by the subject is debatable. Schonka felt that the person left because of frustration at being unable to do the assigned tasks and was, unfortunately, blaming herself instead of the inadequate product design, even though the evaluators took care to explain that it was the product and not the person that was being tested.

The Center had eight testing rooms and eight

observation rooms, the schedules for which were almost completely booked weeks in advance, which illustrates how much testing was done at the Center. Each testing room was equipped with tables and two chairs, computers and terminals, video cameras, a speaker and microphone so that test subjects and evaluators could talk to one another. The floors and walls were carpeted to dampen sound. One of the video cameras was in an upper corner to provide a view of all that is happening in the room; the other was in the ceiling over the subject so that observers could zoom in to see what page the subject was reading. It can zoom in close enough to read subjects' handwritten notes, if desired. Each observation room had tables, three chairs, a speaker, a microphone, a computer, a terminal that showed what the subject is seeing, a terminal that showed what the subject was typing, a terminal for the evaluator to type comments into, two video monitors, and a device for aiming and zooming the video recorder. Schonka estimated that outfitting these rooms cost at least \$500,000.

To stay well informed about developments in her field, Schonka reads Human Factors, IBM Systems Journal, Information Processing and Management, Journal of Systems Management, Computerworld, Ergonomics, Applied Psychology, and Experimental Psychology.

Asher, of the Information Development department at

IBM, was documenting the fifth release of DB2 and had participated with Schonka on the evaluation of DB2 manuals. Asher observed that the nature of the product had changed the characteristics of the user community: Prior to the development of relational databases (as opposed to hierarchical databases), there were no database administrators. Asher explained that the characteristics of the target audience for the product were well known to the staff, and the vocabulary was consistent between IBM and its customers: "We'll say this book is designed for a database administrator and they do have a database administrator there to use it."

The products were not supported with the telephone assistance that one thinks of for retail software users. Instead, customers were industrial users, and they would call the IBM support center, which would send someone to the customer site. There were also reader comment cards in the back of the manuals. The cards were answered by mail, and the Information Development department kept track of readers' comments. Customer comments on one of the products showed that many people thought that there were too many books for the product, and that the books' titles did not reliably indicate their content.

The books went through two reviews for accuracy; usability testing was done between reviews. These usability tests did not test content, but rather the

organization of the manual as one large book and as five smaller books. The test required ten weeks of preparation, three weeks to do the test, and two weeks to write the results.

As part of this study, the researcher observed a usability test being done by Krysiak and Darnell on FrameNet. The subject was an in-house engineer who fit the subject criteria except that he was a newcomer to English. He was given a very esoteric, printed description of the classification of sea animals and was asked to use the program to build a hierarchy. He stated worriedly that he didn't know anything about sea animals and appeared intimidated when he saw the vocabulary he would have to use. It appeared to Krysiak, Darnell, and this researcher that the subject thought he was supposed to know the words as part of his everyday vocabulary-- words such as cnidarian, spicule, hydroid, ostium, flagella, Demospongiae, and mesoglea. Darnell, who had chosen the text because it discussed a hierarchy, said that perhaps his choice was inappropriate because of the confusion it caused this subject.

Observing the flow of information during a usability test was like watching a baseball game: Long, uneventful periods punctuated by short bursts of events. The subject would do as expected most of the time, then make a mistake or get lost, or a software problem would erase his work.

Krysiak and Darnell collected their data via handwritten notes that described what functions the subject was using, how he made mistakes, what questions he asked, when and why they offered help. The notes were then typed and given to team members to facilitate discussion.

Claris and Apple Computer Corporation's partnership.

Usability testing at Claris was being done sporadically at the time of this study. Publications manager Dorteia Atwater planned to include usability testing in plans for online and print documentation. The usability evaluator to be hired would also be responsible for developing needs analyses for the target audiences, based on the software description and the target audience needs description from the marketing department. The Apple and Claris marketing departments used very detailed surveys to determine the characteristics of the intended users of the products. Their customers were non-technically-oriented people who need to accomplish technical tasks. Information on Apple and Claris was obtained in personal interviews in February and March 1989 with Shawna Pugmire, Diane Besterman, Carol Kaehler, and Dorteia Atwater.

Pugmire noted that people usually do not want to spend time with the manuals and sometimes do not open the box in which the manuals come. The ideal at Apple was that users should be able to use the product, at least in a simple manner, without the manual, and that if the user

needed the manual, then the product was too complicated.

The question of what to include and exclude from a manual was answered in part by students in the "Apple University" (in-house) training classes. Pugmire noted that many students expressed dislike of extensive "hand-holding" detail and preferred instead to rely on glossaries and "thin, straightforward" instructions. Students preferred to get a general idea of how each feature works and then find specific information quickly. It was understood at Apple that the easier a product is to use, the more likely it is that people will use it and recommend it to their friends. Also, the products were designed to accommodate incremental learning, meaning that people could use the products immediately and then build on what they had learned as they continued to use the products. Pugmire also found that after in-house users saw that the products were not time consuming to learn, they experimented with more products.

The organization of the training manuals was determined by the writers' experience in learning to use the product; that is, the writers organized the manual to reflect the way they learned the product, or in the sequence in which other manuals appeared to be most successful.

Pugmire acknowledged the problem of programmers' reluctance to communicate, citing the common problem of

programmers changing the product without telling the writers, then watching while the writers try to figure out why the program is giving them problems. However, one design specialist, Carol Kaehler, was highly regarded by several co-workers for her ability to establish rapport with programmers. Pugmire attributed this to Kaehler's writing ability, her familiarity with engineering terms and concepts, and her willingness to work the odd hours that the programmers work. As a result, noted Pugmire, Kaehler's suggestions for changes were more readily adopted by the programmers and incorporated into the products.

The writers began work on the manuals early in the product development cycle, which suggested that Apple's upper management realized that writers must have an intimate understanding of products to communicate well to the users. Instructional designers also influenced products early in the development cycle. Customer support kept a list of customer inquiries that was used to improve the subsequent iterations of each product; also, the training department kept "mental notes" of what items appeared to be causing the most problems among their students. Both departments would forward their comments to the writers for inclusion in revisions.

Atwater stated that one of the common problems of manuals is that they are based on the product's functional

specification. The software description is often developed from the functional specification, rather than taking it from the end-users' point of view, which should be the source of both the software description and the structure of the manual. Usability testing can be used to define the structure, content, and design of the manuals, instead of having that done by the functional specification, which is written by the programmers. If the functional specification is the pattern for the user manual, the manual will reflect the structure of the software rather than its use. Atwater noted that usability testing "makes the users the architects of the manual."

She used one contractor to conduct focus groups, which were small groups of users who gave the contractor detailed feedback about their experiences with the products. She used another contractor to evaluate the data from user-response cards that were included in the back of the manuals. Claris also used prototypes of software in a confidential environment to get user feedback on what looks difficult or inviting. They used that information to improve the design of products before the products had much development money invested in them.

For the summer of 1989, Atwater was planning consumer tests, which consist of bringing in a group of users who match the target audience description. They would be

seated at a computer with a shrink-wrapped box of manuals and observed learning the software. The evaluators would videotape the process and use the information to improve the learning process for future users.

Atwater noted that the writers and usability evaluators have had an influence on the design of the product as well as the manual:

The ideal thing is to have a video lab where you can show developers how end-users stumble on certain interface design issues or functions that they thought were intuitive, and the writers may not be able to influence them to change it because [the programmers think that the writers] don't know anything, but if they see a target audience end-user doing something, that is the ideal situation because it's almost like beta testing.

(Beta testing is product testing that is done at a customer site under special agreement prior to its general release. Alpha testing is that done at the originating company before beta testing.)

Atwater stated that at Apple and Claris, people in documentation and usability are perceived as user advocates, especially where the users' interaction with the program and manual is concerned.

Writers at Claris usually come onto projects after the functional specification has been written. Projects begin as marketing requirements documents, which are the first descriptions of products. These documents include descriptions of the supposed target audience and of the

product's features, as well as explaining how it fits into the marketplace, what its competition is, what it has to offer over its competition, and whether it is just an add-on to an existing product. For example, a computer-aided design (CAD) product may be directed to people in engineering-related jobs, but not small-business people. Within that category, it may be for people who are already familiar with similar products, or it may be for people who have always done the job with paper and pencil and who are now receptive to using a computer. The program must be designed with a complete understanding of its subject matter; for instance, if it is an accounting package, it must reflect a complete and accurate knowledge of accounting principles and vocabulary.

In Atwater's experience, product managers have been reluctant to leave anyone out of the target audience, which has compelled documentation writers to write for a very diverse target audience. The problem has been handled by dividing the manual into a "getting started" section, tutorials, advanced tutorials, and reference sections.

Many books on the market duplicate software companies' manuals. Atwater sees this as an indication of two things: That company manuals are somewhat inadequate and that publishers see a marketing niche among computer users. She saw this especially with dBase, an Ashton-Tate

product, and worried that the Ashton-Tate manual was not explaining dBase well enough to satisfy users, but found that an even greater worry was that "outside" books--those written by non-Apple/Claris writers--support software piracy. As a result, she is wary of supporting outside books that cover an entire product, but does support authors whose books deal with a specific application. "Support" in this context means allowing writers to come into the Claris plant to interview programmers and product managers.

When asked how writers and programmers deal with conflicts about how a product should be designed, Atwater stated:

It's not that often that you find a technical writer who is as 'up' technically as a programmer is, and I think programmers in general feel that writers don't really know that much, so one of our challenges in the publications department is to see that our writers are as much technical experts as they can possibly be so that they can communicate well with the programmer but also represent the product.

She has worked at companies where some programmers would avoid all contact with writers, but has found that the corporate culture at Apple and Claris supports cooperation. She noted that the tone has been set by top management and is enforced by the personnel department, which avoids hiring disdainful programmers.

Ashton-Tate's cost-saving options. Ashton-Tate's usability test of a simple product uses 5 to 10 subjects;

more complex tests use 10 to 15 subjects. The test may cover the entire product or parts of it, depending on the type of product, its size, and its place in the development cycle. Many products were tested more than once in the development cycle. Product stability influenced what could be tested, so prototypes were tested in the early months of the cycle since the products are usually expected to change dramatically. Information on Ashton-Tate was obtained in a personal interview with Gael Silvia and in a long-distance telephone interview with Patricia Averette in Torrance, California.

Usability testing started at Ashton-Tate in 1985 as the result of workers' efforts. Silvia explained, "A couple of people thought it should be done. Definitely not from on top . . . a team sort of level." At Ashton-Tate, the people who do the usability tests are not writers; rather, their entire job description is usability testing. They have bachelor's or master's degrees in education or educational psychology, and are able to construct a test that demonstrates understanding of reliability, validity, data collection, task analysis, and analysis of results. They test the interface as well as the documentation, and Silvia asserted that, in the interest of objectivity, it is better that the writers not test their own work, just as it is better to have Quality Assurance test the code than to have the programmers test

their own work.

Silvia stressed the importance of using test subjects who are typical of the target audience and noted that the main cause of failed tests has been the use of subjects with inappropriate backgrounds. Ashton-Tate used mostly non-in-house people as test subjects. They came from user groups and from university extension classes in dBase and other software programs. Silvia and Averette commented that these sources were excellent for finding novices who are motivated, yet who had the background that made them representative of the projected user population. Their desire to learn the software gave them the motivation typical of users who want to learn it and of users who must learn it for their jobs. Finding advanced users through the user groups was done by contacting the groups' presidents and asking them to put out Ashton-Tate's fliers telling of the test. When the people came for the test, they filled out questionnaires telling how much experience they had, what other products they had experience with, and so on. The subjects were paid \$10 an hour or, if the test was expected to take a long time, they received a copy of the software and documentation to keep since that was cheaper for the company than paying the people by the hour. A software copy may cost the company only \$35 or so, but may be worth several hundred dollars retail to the subjects, according to Silvia.

Detailed marketing surveys described the typical user and the target audience. Sometimes the target audience was defined as "anyone who will buy the product," which effectively gave the job of audience definition to the writers. Some products had a diverse audience, but the evaluators tested the manuals for them as if they were for a specific group, for example, novices only, who would need the most help from the manuals.

Ashton-Tate offered telephone assistance with all its products, but the transfer of information between the technical assistance department and the writing group was meager, according to Silvia, because the technical assistance group was geared to answering questions quickly instead of pinpointing the origin of the problem. For instance, they would not ask, "Did you look it up?" or "What was it about the manual that made it difficult for you to find what you need?" Questions to technical assistance were tracked, however, and as information developed it was given to the appropriate group, such as software development, marketing, or the writers for appropriate action. As of this writing, Ashton-Tate was in the process of devising a way to use the valuable information gathered by the customer service representatives who handle users' problems.

The amount of material tested depends on the

state of the software and of the manual. Averette stated:

[If the test is on] the software, the navigation, the user interface and those kinds of things, then there may be only portions of the manual that are required to do that. But if it's strictly to test the documentation, then we may test the entire manual, although we may not test all the manuals within a set, we may go through an entire learning manual.

The usability test administrators administered the tests to each other to estimate the amount of time the subjects would need to do the test, to verify the validity of the test procedures, and to verify that all the equipment was working. Some of Silvia's tests were done with videotape and some with audiotape, but most were done with human observation only, depending on the purpose of the test. When audiotape was used, it was most often for interface testing, with the subject telling the administrator what the subject was thinking at each step. Silvia stated that, if the test is well designed, the flow of information from the observations should not overwhelm the test administrator. This indicates that concise test goals may help keep costs down.

Averette's tests were all videotaped, and many used keystroke capturing (recording all keystrokes, their sequence, and context). Averette stated that the staff had 8 to 12 users participate in each one-on-one test. The consumer tests, which had users take the finished product home and use it for three or four months, included

up to 15 users. Data were collected in the clinical tests by the person who was administering the test, usually an instructional designer, who would sit with the test subject and take notes while the test was videotaped. Some tests used slaved terminals as well (terminals which copy what the subject is typing). The videotapes were used for review by the test administrators and were shown to other departments in the company, such as product development, marketing, and documentation.

Writers at Ashton-Tate were sometimes included early in the product development cycle, sometimes late, depending on the personalities involved and on the product origin. The influence of individual personalities on this aspect of documentation development was seen in every organization studied, and varied widely within each organization, a situation that appears to offer a tantalizing subject for research. "Product origin" refers to Ashton-Tate's arrangements with other software companies to improve its products; some software comes to them with only the rudiments, others are iterations of an existing product, others are confidential programs. Instructional designers are also included early or late for the same reasons.

Product and manual designs are not always changed to reflect the recommendations of the usability test results. Silvia noted: "Sometimes it's just ego--they still think

their design is best; sometimes it's just [lack of] time." When asked about the interaction between programmers and technical writers, Silvia responded with the same answer as from most of the other organizations: Programmer-writer interaction depends on the individuals, with the personalities of the programmers and the technical competence of the writers being the two determining factors in their rapport.

Silvia observed that the concept of early involvement of writers depends entirely on the software development managers at Ashton-Tate (as it does at all the other organizations that participated in this study):

We have about 20 on-going projects, and each one is composed of different people In some of them the writers are an integral part of the design team and take part in designing the interface, and in other teams, they're not. It depends on the experience level of the writers, whether they've done this before and they expect it and ask to get it and there's a project manager who wants [early writer involvement]; and there are other writers who are inexperienced and who maybe don't know that [early involvement] is an option, or the project manager is just not interested.

Silvia stated that usability testing has helped some products get good reviews, but noted that it is impossible to determine if the testing has resulted in better sales. Some of this company's products have had extensive usability testing and good reviews, yet did poorly in the marketplace, while others were not tested very much and did not receive good reviews and yet sold well, a

situation that may point to factors unrelated to this study, such as market targeting. Silvia usually regards other companies' manuals as examples "of what not to do," but she acknowledges gaining some useful ideas from other companies' quick reference guides.

The idea that usability testing is very expensive is a misconception, especially in view of what can be gained from it, Silvia stated. The main expense is staff time, and the other items needed for the test, such as a terminal, printed instructions, and subjects, are either already there or are inexpensive. Silvia stated that the most expensive test her department has run involved 10 to 15 people, cost about \$2,000 (not including staff salaries), and took about two months. "In our time frame at Ashton-Tate, if it took six months, you might as well not do it," Silvia said. Averette could not offer an estimate of testing costs.

Ashton-Tate's usability testing department put out an in-house quarterly report that detailed all that has been learned from the tests, including results across the product line. One recent compilation of data illustrated new guidelines for creating effective icons, based on problems seen when icons were too small or were not mnemonic enough. They also discovered that if the program demands more than five steps to do any one task, the users could not retain the sequence, and they forgot what their

objective was. She pointed out that even if the steps were clearly written, the users became distracted from the context of their efforts.

Silvia stated that looking at the larger trends of user activity was much more instructive for the evaluators than looking at hundreds of individual pages. The usability tests were specific to each product, yet were very useful in the design of products to follow, which was the main reason Silvia could not release a copy of this document to the author of this study. The report was logically organized and included the test procedure, objectives, subject profiles, test results, questionnaire results, and recommendations.

One practical example of what Ashton-Tate's usability testing uncovered was a test of the label-making procedure of a particular software package. The report stated that this was a surprisingly difficult task for the subjects because they couldn't correct their mistakes. The label procedure began by having the subjects "find line one," but no lines were shown. As the subjects began to add fields to the labels, they found that the fields were not distinguished from each other. The subjects could not determine which fields had already been chosen; instead, all they could see were blank boxes that did not disclose the field name, so they often chose the same field name twice. When they became aware of this, they did not know

how to delete the field name they had chosen. The software was reworked because of these test results.

The "meat" of the usability test is the observation of the subjects, and not the post-test questionnaire, Silvia stated. She has seen people struggle through difficult tests and then say, "That was pretty easy." This response may have been given to compensate for the subjects' perception that they should have found it easy, or that other subjects found it easy, or that they should face the challenge vigorously. Also, they may have been comparing it to other times when the struggle was much greater, or basing their responses on the idea that now that they know how to do it, it's not so hard after all.

Organizations That Do Not Evaluate

Some people at the non-evaluating organizations cannot offer persuasive reasons for avoiding usability testing. Others recognize the value of it but have not thought of a way to apply it to their products. Still others stated that usability testing would be nearly impossible for their materials, yet have devised methods to assure the quality of their documents.

National Magnetic Fusion Energy Computer Center at Lawrence Livermore National Laboratories. The users of the supercomputer system at the National Magnetic Fusion Energy Computer Center (called "MFE" by its employees) are

energy researchers located all over the world, mostly professors and graduate students at universities, which means that there are several hundred user sites, each with a small number of isolated users. MFE has about 26,000 pages online and changes four to five thousand pages each year. Information on this organization was obtained in personal interviews in February and March 1989 with T. R. Girill, Russell Kahn, and Palmer Van Dyke.

MFE is federally funded, so it is reviewed annually by the Department of Energy. In the confidential one-page letter that summarizes the Center's activities for the Secretary of Energy, the online documentation is usually mentioned as a helpful feature for users. That "honorable mention" is a source of great pride for the documentation director, T. R. Girill. Other centers have drawn on MFE's 14 years of experience: When the National Science Foundation built five supercomputer centers, documentation directors from all five came to MFE to see how supercomputer documentation was organized. Two adopted MFE's scheme. All the centers face the same problem of a widely dispersed user base using a central computing facility with a large amount of rapidly changing software.

The first way the Center addressed quality assurance of documentation was by having rapid turnover of online documents, and by making those documents easy to find. The group published all its documentation online, so

unlike creating a regular manual, which involves printing great numbers of copies and distributing them, the most current material was provided to the users instantaneously. The users got their manuals by running a program called DOCUMENT, which could be used to view specific instructions or print them on demand. They could make copies of the passages that interested them, and pay for them out of their computer budget. The writers often put out a draft manual for a new piece of software and then changed it week by week to reflect bug fixes and improvements, instead of accumulating enough changed material to justify a major revision.

Each of the manuals had its own index of keywords so that the users could scan the list and select the most appropriate passage. The global index and the FIND program allowed users to find what they needed when they did not know which manual to read. The manuals also had subject catalogs, alphabetical catalogs, and revision-date catalogs that could be accessed by the FIND program. These methods provided database-format and reference-book format interfaces so that people could use the method that suited their thinking styles.

The second way the Center manages quality assurance is by designing the manual text to be as usable as possible. The philosophy is that, according to recent studies, certain techniques are known to be very helpful,

while others are known to be more like obstacles, so if the writers use techniques that have been tested and proven to be helpful, then the manual will be much more useful, even if the manual itself is not tested.

The style manual that the writers use includes a paper by psychologist Richard Mayer that summarizes 20 years of psychological studies on good and bad text design to address the question: "Can we increase problem-solving performance with text design?" The techniques on which Mayer found agreement emphasized functional relationships in the text, giving concrete support for abstract rules and concepts, and promoting active thinking by the reader.

The third way the Center addressed quality assurance was usage monitoring. Everything that people type in and that the writers put into the online documentation was recorded so that the Center had a complete record of all use sessions. The monitoring system was being used to discover documentation that needed changing. For instance, after the PRINT command was the prompt, "MEDIUM." An unusually large number of users gave inappropriate responses to the prompt, which indicated that they didn't know to what the word "medium" was referring. The writers changed "MEDIUM" to "OUTPUT DEVICE," which made more sense to the users.

If referring to the documentation did not solve the users' problems, they could type "who," to learn whom they

could call for assistance. These helpers, called consultants, were MFE's equivalent of customer service representatives in that they answer users' questions about the system. They differed from customer service representatives in that they interacted much more extensively with the writers and thus gave the writers a much better idea of how well the user community was responding to the documentation.

Each summer the Center has had a university intern study the monitoring record. One discovered that about two-thirds of sessions were error-free, and most of those errors were typing nonexistent keywords and misspelling words. These were problems in retrieving particular passages, not in the running of the program. This finding inspired the documentation group to set up fuzzy matching (which has the system recognize keywords even if the user spells it only 75% correctly), and automatic truncation (which allows the user to enter part of a keyword to get a displayed list of all keywords that include that fragment).

The system was also monitoring the frequency of use for each document, including how often people read it on screen, how long they spent reading it, and how often it was printed out. Those statistics helped the writers decide which documents were improved first.

The founders of the Center did not need to be

convinced of the value of quality assurance; according to interviewees Girill and Kahn, the top management of the Center has been the most enthusiastic advocates of quality assurance for documentation: When the Center was created, the five originators made a list of the flaws that had annoyed them about the computer centers where they had worked before, and one of those flaws was the documentation. As of this writing, the Center was spending about \$500,000 a year on documentation, including a staff of four writer-editors and a technician who serve a user community of about 4,500 people.

The system includes 210 documents, which amounts to about one million lines of online documentation. Girill noted:

When other computer centers look at this online documentation scheme, they say, "Well, we could never afford the disk space for a million lines of documentation," but the Center here has always taken the opposite position, that that's a great economy The cost in users' time wasted if you have just a few people running around scratching their heads, it wastes lots of time and money. If you spend the money once on a way to prevent that easily, then you've saved them time and all that grief of not having the latest manual, not knowing how to read it. It's an investment.

Like technical writers everywhere, writers at Lawrence Livermore National Laboratories (LLNL) were under pressure to get documents written, in this case because a team of programmers was constantly producing more software. Kahn noted, "Our biggest concern is the

software we have today; otherwise, they might as well not be writing their codes." The first priority was to get the documents out correctly; second priority was usability. The introductory documents have to be easily understood, while the reference material, which is the bulk of the documentation, only has to be correct. Kahn asserted that even with all the extensive measures taken to improve quality, the manuals can still be very difficult for users to understand without the writers being aware of that problem, since the users must learn the material to do their jobs regardless of how difficult the manual is to understand.

Editors at LLNL have extensive technical backgrounds, which helps them communicate with the programmers. If a topic is exceptionally difficult, Kahn will "just write down what they say," and use that transcript as the starting point for the document. He observed that programmers are not inordinately ego-involved with the documentation and are usually quite approachable, especially since the writers have a good understanding of the computer system. He noted that most of their job applicants have a degree in technical writing.

Another aspect of the writing environment at LLNL is that most of the users are familiar with the system and prefer that the documentation not have too much detail. Of course, the writers cannot quantify how much is too

much to please the entire user community; also, the Center does serve those who are not yet familiar with the system, and the documents must have enough detail to support their efforts as well.

When writers create a new document, they use the organization of existing documents as a guide. When a document is created without an appropriate model to follow, the writer imposes a logical arrangement on the material, such as easy to difficult, or chronological order.

Kahn finds useful information from the Conference on Computing Machinery; the Society for Technical Communication; the newsletter of the National Center for Supercomputer Applications; and Vector, the newsletter of the Air Force Weapons Lab. Kahn finds these last two much more useful because they are written especially for people who work with supercomputers, which are very different from minicomputers and microcomputers in their memory, time-sharing, security, and applications.

Kahn observed that writers do not receive much feedback directly from users; instead, as mentioned before, writers rely on the consultants. "It's definitely a big issue that we don't get enough feedback," stated Kahn, who wants more qualitative feedback than that provided by the monitoring system. Similar opinions were expressed by writers at all the other organizations

involved in this study.

Tandem's alternate strategy. Since Tandem's users have extensive computer backgrounds, managers stated that usability testing was not necessary, and writers cited the products' enormous complexity as a reason for avoiding usability testing. The users learn the product with onsite training rather than relying on the manual alone, which is the clue to understanding Tandem's disinterest in usability testing. Usability testing of software manuals is almost inevitable: It will be done either by employees or by customers, so Tandem ensures that its customers understand the product by guiding them through it in intensive customer-education classes rather than by improving the usability of the manual and the software.

The technical writers at Tandem had English or journalism backgrounds; many also had computer science degrees. Filling job requisitions with this combination has been very difficult, since people with those qualifications are hard to find. Also, people with computer science degrees can often earn more as programmers than as writers.

Instructional designers were not included in the design of the product or the manual; instead, programmers design the product and writers design the manuals. Publication design consultants were hired when managers felt they were needed.

Manual drafts were sent to Tandem's software educators, but they rarely actually reviewed them, according to the writers interviewed. Occasionally a technical writer would take classes that were intended for customers; it was then that he or she received extensive, concrete feedback from both the customers and the software educators. Feedback from the software educators was otherwise very haphazard. Attending the user classes was a form of usability testing but was much too infrequent to be regarded as a part of an evaluation process at Tandem. Course developers have almost no contact with the writers in Mary-Leigh Burke's group, and so have no influence on the manuals. They only see the manuals after publication, when they use the manuals to design the customer education courses. The software developers review the manuals for technical accuracy but have no role in making the manual easy to use.

Christine Browning noted that her company has a very wide non-Western, English-reading usership, so the instructions must be written to serve those with limited English. The instructions must also accommodate different learning styles. She has observed that some people learn deductively, others inductively, which may indicate that some users find the introductions and overviews useless, while others find that those sections help organize their thoughts.

According to Stacy Lippert, writers in her group had no influence on the development of products, and, in fact, it was unusual for writers to be included early in the product development cycle, which was why she strove to be included on the X400 (database management software) project. Usually the writers have been brought into a project after the specification has been written (after the basic design of a product has been created), but there has been a company-wide push to get writers involved earlier in the product development cycle. With early writer involvement, management was hoping that the writers would be more knowledgeable about the product and more able to make useful suggestions. This policy will also cause the writers to do some incidental alpha (in-house, pre-release) testing as they use the product to learn how it works.

Tandem's vice president of quality, Eileen Burkwood, was the impetus behind the idea of early writer involvement. Burkwood worked for many years at Hewlett Packard, which is known for its emphasis on quality assurance. Burkwood's charter was to train employees at all levels to work with quality in mind. "This opens the development phase to many more people and trains the writers earlier so they can be technically apt by the time they start writing the manuals," observed Lippert. According to Lippert, Tandem did not have anyone

interviewing users to learn about their problems and preferences, but she observed interest developing in user feedback because of the new interest in quality assurance at her company.

In contrast to Browning's view that Tandem's users are sophisticated enough to do well without usability-tested manuals, Lippert stated that the product on which she was working was directed to programmers and system managers whom she believed did not necessarily have the experience to make up for inadequacies in the manual. Even though these users have had experience with computers, Lippert noted that X400 would be unlike anything on the market, which means that the users would have to rely on the manual, telephone helplines, and site analysts.

The organization of the manual was determined by its content and the product specifications. The specifications give the writers the information to be included; the writers decide what represents sequential order from the users' standpoint. The content of the manual is also derived from discussions with the developers on what the users have to do, how they would do it, what information they need to do it, and what information is unnecessary to those requirements.

When deciding what to include and exclude from the manual, writers had to determine what aspects of the

software are internal and external. (Internal features are those not seen by the users; external items are those the user will see.) Some parts of the Nonstop SQL are of concern only to the Tandem employees who are writing code that interacts with other Tandem software; customers do not need to know about background tasking and so are not burdened with unnecessary details. "The difference between internals and externals is a thorny problem," Burke noted, explaining that the writers mark on the first pass what they believe are the externals. They then write a publications plan with an outline that is read by software development reviewers. The reviewers ascertain that the writers' perception is correct. Documenting for users who are programmers makes determining the difference between internal and external features difficult since these users must be able to make their code interface with items that may be regarded as internal by other users. Documenting for end-users would be much easier, Burke said, because the division between internals and externals is relatively simple; that is, if they can see it or use it, it is external, and if they cannot, it is internal.

The writers' interaction with software developers depends very much on each developer's personality, according to Browning, Burke, Howd, and Lippert. Some are so difficult that writers avoid contact with them and go through their managers or co-workers, or use memos instead

of oral communication. Others are very easy to talk to, especially after the writers have gained experience with the software. Each writer agreed that approachable programmers outnumbered the unapproachable ones. When Burke deals with a programmer who finds communication difficult, she makes a printout of the section of code in question, takes it to the programmer and asks him why it does not work. Burke noted: "That puts them right in their element. Then we sit down and debug it together and everything is fine." She reiterates that it is important for writers to get a good technical education, "otherwise they don't really respect you."

The user manuals were organized around functions rather than structure, and depended heavily on a good index, which Burke believes "takes a knack to write," since each individual has a different way of organizing the material. Indexing, though, was regarded as "the last onerous task" in the manual-writing process. She noted that cross-indexing command names with their functions expressed as gerunds (words ending in -ing which act both as verbs and nouns) was useful since the user may not remember the name of a command but will know what he or she wants to do, and can therefore find information by looking under "compiling," "filing," "updating," etc.

Writers received a small amount of user feedback via Tandem's Product Reporting System, which is inadequate,

according to Burke, Lippert, and Howd. When users find problems, they inform Tandem's system analysts, who then send their comments through a computer network to the appropriate node of the electronic mail system.

"Unfortunately, messages often wind up in the wrong node, where they may languish for months before someone redirects them," noted Burke.

When Tandem customers find a problem, they can call the helpline, and if the question is not covered by the manual or is explained incorrectly, the customer service representative files a technical problem report that is passed on to the writers, who correct it in the next revision of the manual. Personal or telephone contact between writers and customer service representatives is rare. The manuals also have reader comment cards, but the information gained from these is meager. Burke noted that change pages are sent out to customers as needed with change bars in the margins, and that if the manual is changing more than 60%, the entire manual is revised.

CHAPTER V

CONCLUSIONS

Usability testing--having representative subjects use the manual under observation--was regarded by most of the interviewees as a worthwhile endeavor. Organizations that do usability testing spent a large amount of time and money on the testing because it requires extensive planning and careful administration for useful results. Evaluators and upper management regarded usability testing as an investment in the quality of their products. Even the most extensive testing efforts were limited, but not thwarted, by lack of time and by the complexity of the software. Of the organizations that do not do usability testing, the National Magnetic Fusion Energy Computer Center at Lawrence Livermore National Laboratories (called "MFE") made extensive efforts to ensure that the documentation was easily understood; Tandem made no discernible effort to improve the usability of its manuals. Tandem did review its manuals for accuracy, but made its products usable with the help of customer education and on-site assistance.

Organizations That Evaluate

At organizations that do usability testing,

evaluators had many opportunities to exercise judgment about usability testing methods, so the researcher witnessed significantly varied techniques even within organizations. Some evaluators used videotape only for some tests, some used it for all tests, some did not use it at all, some conducted tremendously expensive tests, some did very inexpensive tests, some tested as much of each manual as possible, others concentrated on a few small areas. All evaluators recognized the importance of using the scientific method to do their tests, including using representative subjects, getting reproducible results, and being as objective as possible in their observations.

Policies were divided about which staff members did the testing. The policy at Ashton-Tate and IBM was that having writers evaluate their own work created a conflict of interest; but Claris, which according to publications manager Dortha Atwater did not provide funding for a large writing staff, was reluctant to spend even more money on usability testing if its products were doing passably well in the marketplace, as determined by management.

Organizations that allotted more money to usability testing tested more of each manual and program than Claris. The decision on what to test was made mostly by the evaluators, who tested those items that were new,

those that have presented problems in the past, and those that appeared especially confusing. Writers and software developers occasionally asked that particular sections be tested. The typical test cycle took a month or two to prepare, two to eight weeks to execute, and a month or so to evaluate. This is only a rough estimation because of the tremendous variation among products. Most tested products were subjected to only one or two usability test cycles prior to release. No one in the study knew how long their companies had been doing usability testing, or what methods had been used and discarded.

Most test data were collected in the form of handwritten notes during the test, even though evaluators used audio and video recordings. The consensus among evaluators was that reviewing tapes of information that they have already witnessed was very boring and enormously time consuming. In choosing one method over another, the evaluators appeared to be searching for a balance between thoroughness and convenience.

Organizations That Do Not Evaluate

After being introduced to the concept of usability testing, one may believe that all software documentation should be tested out of deference for customers and for the company's health in the market. However, some organizations have compelling reasons for not doing

usability testing. At the National Magnetic Fusion Energy Computer Center at Lawrence Livermore National Laboratories (called MFE), the turnover of software and documentation is so rapid that testing would be meaningless since a large portion of the documentation has a very brief lifespan. Document lifespan in some organizations is seen as a reason to avoid usability testing (see Cable Data, mentioned in the Method section), but MFE, and particularly T. R. Girill and his staff, regard the brief lifespan of their documents as a benefit: Turnover is so rapid that corrections and improvements can be done almost instantly and as often as needed. They have made other efforts to ensure the usability of the documentation, such as making documents easily accessible, giving writers well researched guidelines, and by monitoring usage of the system to find documentation flaws.

Both MFE and Tandem relied on the combination of well trained technical writers and knowledgeable users to achieve communication, and user feedback was an important issue to Girill and Kahn. Tandem, however, did not use any formal measures to ensure the usability of their manuals. According to the interviewees, user feedback to Tandem writers through any channel was almost nil and inter-departmental communication on usability of manuals was also virtually nonexistent.

The philosophy at Tandem was that the product was far too complex to accommodate usability testing. However, based on the results of this study, well designed usability testing probably would help make their products seem less complex; that is, easier to understand. Tandem had software educators who could offer suggestions to improve the interface, but they rarely reviewed the manuals.

Tandem had a large budget for customer education. It would be interesting to see whether that budget could be reduced if the manuals were improved.

General Findings

All the participants in the study manage manual organization similarly. All rely first on the established order for the sake of consistency. When the established order does not serve, writers find a logical order to use: Chronological, easy to difficult, alphabetical, general to specific or vice versa. One person (from Apple University) said that her group takes a more organic approach and designs training manuals according to whatever sequence they think is best, based on their own experience in learning the product.

What Mary-Leigh Burke called the "techie/nontechie barrier" was a source of aggravation that most of the interviewed writers appeared to be dealing with quite

well. There were a few programmers with whom the writers found it difficult to communicate, either because of the programmers' disdain for nontechnical people or because of their general unease with conversation, especially with women, according to several interviewees. (The majority of programmers and engineers still are men; roughly half of technical writers are women, although most of the writers included in this study were women.) The problem was blamed uniformly on programmers' personalities versus writers' lack of technical knowledge. The underlying assumption of the writers and evaluators appeared to be that writers have to be pleasant because they are imposing on the programmers' time; therefore, the problem cannot be with the writers' personalities. Meanwhile, the view from the programmers' side, as expressed by several writers and evaluators, was that writers would not have to use programmers' valuable time if the writers knew about programming. All the writers in the study have taken the time to educate themselves about computers. Several expressed the opinion that a technical writer without a computer background would have a very difficult time finding employment in the current market.

All eleven writers and all five evaluators knew the characteristics of their target audiences, but there was some disagreement between the writers and their co-workers in the marketing departments about whom to include in

those target audiences. This problem appeared only in the companies that sell end-user software, not in those that sell to applications programmers, presumably because end-user products are marketed to the widest audience possible. Marketing surveys were used by those selling to end-users; IBM, Tandem, and MFE relied on their software developers to define target audience characteristics and included only target-audience users in the distribution of the software.

Telephone assistance was not offered as much as expected, and writers at all the organizations have very little contact with the customer service representatives who answer telephone assistance inquiries. Customer service representatives kept track of questions and complaints, but the flow of information to the writers from customer service representatives was not abundant at any company in the study. This was unfortunate, since the helplines were a rich source of information about the problems customers were having. Writers could gain much by answering questions on the helplines occasionally, especially if they were allowed to talk at length with the customers.

Only recently have organizations seen the value of including writers and instructional designers early in the product development cycle. This has enabled more of the staff to know more about the product, which gives them the

ability to make useful suggestions, which, if incorporated into the product, would improve its usability.

Deciding what to include and exclude from the instructions was relatively straightforward for most of those studied; however, Tandem writers were faced with the often-confounding task of differentiating between what the users would and would not see because of the nature of the products. This "internal-versus-external" issue was much clearer with the other organizations' software.

No one said they were influenced strongly by other organizations' manuals, but all acknowledged that if they saw something they liked they would use it. Writers also brought ideas from their previous employers and from the professional journals they read. IBM, however, also conducted competitive evaluations; that is, they formally compared competitors' manuals to their own.

Directions for Future Research

This report on the present and future states of usability testing can be the foundation for emphasis on usability testing in technical writing courses. The topic of usability testing touches an enormous number of potential research topics, including:

1. Helplines. How helpful are they? How accessible are they? How accurate is the information they provide? Who uses them? How extensively do helpline users

refer to the manual?

2. Tutorials. What percentage of users use the tutorials? Do the users benefit from the tutorials? What makes a tutorial most useful? What are the characteristics of people who use the tutorials? Are users learning what they expect from the tutorials?
3. Manual format. Do users prefer their manual set to be in one binder or separate binders? Do they prefer detailed instructions, brief instructions, or both?
4. Management. What specific arguments were used to persuade management to fund usability testing? What are the characteristics of the corporate cultures that support usability testing? That do not support it?
5. Education. To what extent is usability testing discussed in technical writing courses? Of those technical writers who were taught about usability testing, how many make use of it? How many try to persuade their managers to make use of it? How well do they design and administer tests?
6. Workplace. What improvements in efficiency can be demonstrated when the manual for, say, a point-of-sale terminal is improved? What improvements in worker satisfaction can be shown?
7. Software industry. What percentage of a large sample of software publishers do usability testing? What are their procedures? Why do some not test? What do the

non-testers do to ensure the quality of their manuals? Do their customers complain more than the customers of testers? About what? Do test-using organizations have lower expenses for customer support than non-testing organizations?

8. Programmers versus writers. What is the nature of the communication that takes place between writers and programmers? Why is it troublesome to so many technical writers? Why do so many programmers make changes in the software without telling those who are documenting it?
9. Audience analysis. What are the most commonly used methods of determining the target audience of a manual? How consistently are these methods used? Have the organizations determined that these analyses yield accurate descriptions of the actual users?
10. In-house versus outside manuals. Which offer better instructions? Why? How are the authors of in-house manuals influenced by the design and content of outside manuals, and how are the authors of outside manuals influenced by the design and content of in-house manuals?
11. Communication about software. Other than by user groups, how do users learn more about their software? Are computer magazines more reliable sources of information than newspaper articles? What mode of

communication do different types of users prefer and why?

This study is one of the first to examine actual industry practices, which makes it valuable to technical writers for comparison to their own work situations, and to students for what they can expect when they become employed in the field. Students should be made aware of usability testing so that, first, they will not be surprised by it when they begin employment; second, they will be able to design and administer usability tests; third, they will be able to help persuade management that usability testing is a worthwhile endeavor that can improve the company's product; fourth, they will be able to help reduce the frustration experienced by users. Some technical writers' knowledge of usability testing is limited to their own organizations' practices; they can broaden their exposure by reading about the topic in journals, but journal articles are usually concerned with research or with describing practices at organizations with near-ideal test environments. It is the hope of this researcher that she also offered readers a view of the less-than-ideal situations found in the workplace.

REFERENCES

- Babbie, E. R. (1973). Survey research methods. Belmont, CA: Wadsworth Publishing Company.
- Baker, E. L., & Alkin, M. C. (1973). Formative evaluation of instructional development. AV Communication Review, 21, 389-418.
- Bastian, R., Edward, K., Medsker, K., & Schimmel, B. (1983). Formative evaluation in industry, Performance and Instruction Journal, 22, 20-22.
- Burgess, P. (1984, April 16). The tyranny of words. Computerworld, pp. SI-1-9.
- Carroll, J. M., Smith-Kerker, P. L., Ford, J. R., & Mazur-Rimet, S. A. (1987-1988). The minimal manual. Human-Computer Interaction, 3, 123-155.
- Christians, C. G. & Carey, J. W. (1981). The logic and aims of qualitative research. In G. H. Stempel III & B. H. Westley (Eds.), Research methods in mass communications (pp. 342-362). Englewood Cliffs, NJ: Prentice-Hall, Inc.
- Cipolla, C. M., & Birdsall, D. (1979). The technology of man. New York: Holt, Rinehart & Winston.
- Comstock, G. & McCombs, M. E. (1981). Survey research. In G. H. Stempel III & B. H. Westley (Eds.), Research methods in mass communications (pp. 144-166). Englewood Cliffs, NJ: Prentice-Hall, Inc.
- Cunningham, W. R. (1984). Testing task-oriented publications in a laboratory environment. Proceedings, 31st International Technical Communication Conference (pp. WE-144 - WE-147). Washington, DC: Society for Technical Communication.
- Daumas, M. (1969). A history of technology and innovation, 1. New York: Crown Publishers.
- Davis, J. S. (1989). Usability of SQL and menus for database query. International Journal of Man-Machine Studies, 30, 447-455.
- Dick, W., & Carey, L. (1978). The systematic design of instruction. Glenview, IL: Scott Foresman & Co.

- Drotter, M. T., & Roth, G. L. (1984, December). Revising poor software documentation: Guidelines for modification in the classroom. Educational Technology, 47-49.
- Duffy, T. M., Curran, T. E., & Sass, D. (1983). Document design for technical job tasks: An evaluation. Human Factors, 25, No. 2, 143-160.
- Foley, J. P. Jr. (1978). Impact of advanced maintenance data and task-oriented training technologies on maintenance, personnel, and training systems. Wright-Patterson AFB, OH: Air Force Human Resources Research Laboratory, Technical Report AFHRL TR-78-25, September, 1978. (AD A063-277).
- Fortner, R. S., & Christians, C. G. (1981). Separating wheat from chaff in qualitative studies. In G. H. Stempel III & B. H. Westley (Eds.), Research methods in mass communications (pp. 363-374). Englewood Cliffs, NJ: Prentice-Hall, Inc.
- Gagne, R. M., & Briggs, L. J. (1974). Principles of instructional design. San Francisco: Holt, Rinehart & Winston, Inc.
- Gillihan, D., & Herrin, J. (1988). Evaluating product manuals for increased usability. Technical Communication, 35, 168-171.
- Heinich, R., Molenda, M., & Russell, J. D. (1982). Instructional media. New York: John Wiley & Sons.
- Hernandez, M., & Wilder, L. B. (1984). Usability testing: A tool for the technical writer. Proceedings, 31st International Technical Communication Conference (pp. WE-141 - WE-143). Washington, DC: Society for Technical Communication.
- Horton, W. (1984). Look me in the eye when you write that: How to put the user in user's guides. Proceedings, 31st International Technical Communication Conference (pp. WE-196 - WE-199). Washington, DC: Society for Technical Communication.
- Kandaswamy, S., Stolovich, H., & Thiagarajan, S. (1976). Learner verification and revision: An experimental comparison of two methods. AV Communication Review, 24, 316-328.
- Kleid, N. A. (1984). IBM's information quality measurement program. Proceedings, 31st International Technical

- Communication Conference (pp. MPD-62 - MPD-65).
Washington, DC: Society for Technical Communication.
- Levy, S. (1984, July). Getting help. Popular Computing, pp. 70, 75-76, 78.
- Marshall, N. (1979). Readability and comprehensibility. Journal of Reading, 22, 542-544.
- Maynard, J. (1982). A user-driven approach to better user manuals. IEEE Transactions on Professional Communication, PC-25, 16-19.
- Mills, C. B. & Dye, K. L. (1985). Usability testing: User reviews. Technical Communication, 32, 40-44.
- Mirel, B. (1987). Actual-use testing for in-house computer manuals. Technical Communication, 34, 289-291.
- Neal, A. S., & Simons, R. M. (1984). Playback: A method for evaluating the usability of software and its documentation. IBM Systems Journal, 23, 43-58.
- Redish, J. C. (1988). Reading to learn to do. The Technical Writing Teacher, 15, 223-233.
- Rony, E. (1983, December 25). Losing the fear of manuals. San Jose Mercury News, pp. 9F, 12F.
- Schramm, W. (1973). Men, messages, and media. San Francisco: Harper & Row.
- Seifert, M. (1979). Reading on the job. Journal of Reading, 22, 360-362.
- Severin, W. J., & Tankard, J. W. Jr. (1979). Communication theories. New York: Hastings House.
- Southard, S. (1988). Usable instructions based on research and theory: Part 2. Technical Communication, 35, 167-168.
- Tyler, R. W., Gagne, R. M., & Scriven, M. (1967). Perspectives of curriculum evaluation. Chicago: Rand McNally & Co.
- Wager, J. C. (1983). One-to-one and small group formative evaluation. Performance and Instruction Journal, 22, 5-7.
- Waite, R. G. (1984). Organizing computer manuals on the basis of user tasks. Proceedings, 31st International Technical Communication Conference (pp. WE-38 - WE-40).

Washington, DC: Society for Technical Communication.

INTERVIEWS

Preliminary interviews:

Duarte, Wilfrid; technical writing contractor, Atkinson, Inc. Personal communication, February 10, 1987.

Ellenbogen, Ira; president, Nanosecond Inc. Personal interview, April 2, 1986.

Goldstein, Deborah; technical writing contractor, Computer Sciences Corporation. Personal communication, January 5, 1987.

Mansfield, Scott J.; senior documentation specialist, Cable Data. Personal interview, May 10, 1986.

Rau, Taffy; technical associate, OnLine Incorporated. Personal interview, June 17, 1987.

Todhunter, Vivian; technical writer, Computer Sciences Corporation. Personal communication, January 5, 1987.

Interviews for the study:

Asher, Lori; senior associate technical writer, International Business Machines Corporation. Personal interview, April 27, 1989.

Atwater, Dortehea; publications manager, Claris. Personal interview, March 23, 1989.

Averette, Patricia; senior instructional technologist, Ashton-Tate. Long-distance telephone interview, Torrance, CA, June 30, 1989.

Besterman, Diane; instructional design contractor, Apple Computer. Personal interview, February 16, 1989.

Browning, Christine; first-line manager of software technical publications, Tandem. Personal interview, February 17, 1989.

Burke, Mary-Leigh; senior technical writer, Tandem. Personal interview, March 9, 1989.

Darnell, Mike; software engineer, IBM. Personal interview, April 27, 1989.

Girill, T. R.; director of documentation, National Magnetic Fusion Energy Computer Center. Personal interview, March

23, 1989.

Grossman, Ann; software developer, Tandem. Telephone interview, April 5, 1989.

Howd, Sheri; documentation manager, Tandem. Personal interview, March 9, 1989.

Kaehler, Carol; software design specialist, Apple Computers. Personal interview, March 2, 1989.

Kahn, Russell; documentation specialist, Open Computer Center at Lawrence Livermore National Laboratories. Personal interview, March 23, 1989.

Krysiak, Debbie; human factors engineer, International Business Machines Corporation. Personal interview, April 27, 1989.

Lippert, Stacy; technical writer, Tandem. Personal interview, March 3, 1989.

Pugmire, Shawna; training coordinator, Apple Computers. Personal interview, February 16, 1989.

Schonka, Sophie; human factors engineer, International Business Machines Corporation. Personal interview, April 20, 1989.

Silvia, Gael; documentation manager, Ashton-Tate. Personal interview, April 6, 1989.

Van Dyke, Palmer; technical writer, Lawrence Livermore National Laboratories. Personal interview, February 22, 1989.

APPENDIX

Appendix A

Researcher's Questions

Questions for technical writers who worked at organizations that tested their manuals included:

1. What is your main project right now?
2. Who are the intended users of this product?
3. What is their occupation, education, and age?
4. Has your company used marketing surveys to determine these factors, or was the product's target audience defined by another means?
5. Does the software package include telephone assistance?
6. Do the customer service representatives keep track of questions and complaints? How closely do they keep in touch with you about the questions and complaints?
7. Who does the usability testing?
8. How much of each manual do you test?
9. How do you decide what to test?
10. How much time does it take to test that much of a manual?
11. What clues do you have to help you estimate testing time?
12. At what point in the product development cycle do you test the manual?
13. Do you test it more than once?
14. How do you collect data?

15. Who do you use for your test subjects? How are they chosen?
16. At what point in the product development cycle do writers start working on the manual?
17. Are instructional designers included in the design of the manual? If so, at what point in the manual's development?
18. How was the organization of the manual worked out?
19. When did your company start evaluating manuals?
20. What methods were used?
21. How often does the staff run a usability test?
22. What convinced management that usability testing is worthwhile?
23. What influence have other companies' manuals had on your manuals?
24. What publications and professional organizations do you find to be most helpful in making your manuals more usable?
25. How much would you estimate that your company spends on usability testing? Do you see any cheaper alternatives being developed?

Questions for technical writers who worked at organizations that did not test their documentation included:

1. What is your main project right now?

2. Who are the intended users? What is their age, occupation, and computer background?
3. Was this information developed through a marketing survey?
4. Does the software package include telephone assistance?
5. If so, how do the customer service representatives communicate with the writers?
6. Does someone keep track of customers' questions?
7. How do you decide what to include and exclude from the instructions?
8. How was the organization of the manual worked out?
9. Are instructional designers included in the product development cycle? In the design of the manual?
10. At what point in the product development cycle are writers brought in?
11. What influence have other companies' manuals had on your manuals?
12. Do you have some indication of how well the users understand the manuals?
13. Have professional organizations and publications been helpful to you?