

A RESEARCH EXPERIMENT
TO EVALUATE THE ACCEPTABILITY OF
MICROCOMPUTER SOFTWARE DOCUMENTATION

by

William C. Coursey, Jr.

thesis submitted to the Graduate Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE
in
Industrial Engineering and Operations Research

APPROVED:

Marvin H. Agee, Chairman

T. J. Greene

C. J. Mullins

December, 1985
Blacksburg, Virginia

A RESEARCH EXPERIMENT
TO EVALUATE THE ACCEPTABILITY OF
MICROCOMPUTER SOFTWARE DOCUMENTATION

by

William C. Coursey, Jr.

ABSTRACT

Microcomputer software users require varying degrees of instructional assistance to effectively operate the software they purchase. Chapter I recognizes that this demand for quality documentation places a burden upon software suppliers to expend additional time, energy, and money to satisfy users. This research recommends a set of procedural guidelines for microcomputer software suppliers to follow as a means of supplementing basic documentation techniques.

Literature regarding microcomputer software documentation is an item of increasing demand in today's technical marketplace. The literature review, Chapter II, reveals that the most significant improvement in the documentation process has been the development of two specific reference standards, physical layout and instructional components.

Chapter III describes the research experiment used in obtaining information regarding the documentation associated with two current microcomputer word processing programs.

Four university students provided background information regarding the personal characteristics, attributes, associated with a given user population.

The research experiment evolved from a comprehensive documentation review to a structured data collection process. Chapter IV indicates that the discrepancy between actual and expected research gains justifies improving data collection techniques and recommending specific procedural guidelines for future documentation reviews.

The final chapter provides a detailed analysis of the research experiment and conclusions related to the documentation's effectiveness. Additionally, it proposes procedural guidelines designed to improve the experiment's data collection techniques. These guidelines can help future documentation writers more accurately gauge user capabilities and limitations.

ACKNOWLEDGEMENTS

I give special thanks to my committee chairman, Dr. Marvin H. Agee, for his technical support and guidance during the research and preparation of this thesis. During the past two years, Dr. Agee has been not only a competent advisor and dedicated educator, but also an extremely close friend. I will be forever indebted to him for the emotional support he generously gave me toward completion of my graduate degree requirements.

I thank Dr. Timothy J. Greene and Prof. Carolyn J. Mullins for serving on my committee and recommending changes that greatly improved the thesis. Dr. Greene helped me to develop a problem-solving approach in structuring the thesis, and he "motivated" me to finish it without further delay. Prof. Mullins constantly encouraged me to take full advantage of her technical writing expertise and knowledge of software documentation.

I thank my family and close friends for their urging me to pursue a professional career. Also, I especially thank _____ for her Irish spirit and word processing talents. Finally, I owe everything to my precious wife, _____, whose love, patience, and dedication to purpose have inspired me to accept this challenge and the ones that lie ahead.

CONTENTS

	<u>PAGE</u>
ABSTRACT	ii
ACKNOWLEDGEMENTS	iv
Table of Figures	x
CHAPTER I - INTRODUCTION	1
1.1 Nature and Scope of Research	1
1.2 Definitions	2
1.3 Problem Statement	4
1.4 Statement of Objectives	6
1.5 Assumptions	8
1.6 Method for Meeting Objectives	10
1.7 Demonstration of Research Method	13
CHAPTER II - LITERATURE REVIEW	14
2.1 Introduction	14
2.2 New Documentation Reference Standards	15
2.2.1 Physical Layout	15
2.2.2 Instructional Components	16
2.3 Current Software Documentation	18
2.3.1 Documentation Practices	18
2.3.2 Problems Associated with Documentation	19
2.3.3 Periodical Literature	21

2.4	Suggested Improvements of Documentation	25
2.4.1	Macro Design	25
2.4.2	Micro Design	28
2.4.3	Documentation Modules	31
2.5	Conducting Surveys	38
2.5.1	Personal Interviews	38
2.5.2	Advantages of Personal Interviews	39
2.5.3	Disadvantages of Personal Interviews ...	42
2.6	Preparation of Questionnaires	43
2.6.1	Questionnaire Format	44
2.6.2	Open-End Questions	44
CHAPTER III - RESEARCH METHOD		46
3.1	Introduction	46
3.2	Description of Research Experiment	46
3.2.1	Selection of Software/Hardware	47
3.2.2	Selection of Research Location	47
3.2.3	Identification of User Population	48
3.2.4	Orientation of Users	49
3.2.5	Final Orientation Meeting	50
3.2.6	Periodic Supervision of Users	53
3.2.7	Personal Interviews	53
3.2.8	Review of User Information	54
3.2.9	Organization of User Information	54
3.3	Demonstration of Method	55

CHAPTER IV - EXPERIMENTAL RESULTS AND PRELIMINARY FINDINGS	57
4.1 Introduction	57
4.2 Evaluation of Results and Preliminary Findings	58
4.2.1 Organization of User Information	59
4.2.2 Classification of User Attributes	59
4.2.3 Categorization of Documentation Topics	60
4.2.4 Cross-matching of Collected Data	60
4.2.5 Reorganization of Documentation	61
4.2.6 Development and Design of Documentation Framework	61
4.2.7 Incorporation of Boilerplate Material	62
4.3 Data Collection Problems and Findings	62
4.3.1 Sample Population	62
4.3.2 Collected Data	63
4.3.3 Open-End Questions	63
4.3.4 Control Group	64
4.3.5 Time Constraints	64
4.3.6 Performance Rating System	65
4.3.7 Quick-reference Modules	65
4.3.8 Personal Interviews	66
4.3.9 Function/Command Sequence Modules	66
4.3.10 Documentation Prototype	67
4.3.11 User Interaction	67

4.4	Procedural Guidelines for Documentation	68
CHAPTER V - ANALYSIS AND CONCLUSIONS		69
5.1	Introduction	69
5.2	Personal Evaluation of Experiment	70
5.2.1	Research Design Problems	70
5.2.2	User Difficulties	73
5.3	Recommended Procedural Guidelines	74
5.4	Critique of Experimental Design	79
5.4.1	Loose-leaf, Hardbound Manual	80
5.4.2	Function Key Template	81
5.4.3	Quick-reference Booklet	82
5.4.4	Supplementary Disk-based Tutorial	83
5.4.5	Page Layout	84
5.5	Personal Evaluation vs. New Literature Critique	85
5.5.1	Advantages of Personal Evaluation	87
5.5.2	Advantages of New Literature Critique	88
5.6	Full-scale Experimentation	90
5.6.1	Experimentation Problems	90
5.6.2	Experimentation Costs	92
5.7	Conclusion	94
BIBLIOGRAPHY		97
APPENDIX A - User Profiles		99
APPENDIX B - Personal Data Sheets/Resumes		104
APPENDIX C - Software Review Logbooks		109

APPENDIX D - User Recommendations	139
APPENDIX E - Matrix Summaries	153
VITA	160

Table of Figures

	<u>PAGE</u>
Figure 1 - DOCUMENTATION PROJECT SYNOPSIS FORM	26
Figure 2 - RESEARCH ACTION PLAN	30
Figure 3 - MASTER PLAN	33
Figure 4 - DOCUMENTATION FLOWCHART	34
Figure 5 - USER/TOPIC ANALYSIS	35

CHAPTER I
INTRODUCTION

1.1 Nature and Scope of Research

The microcomputer software industry is experiencing considerable growth as potential users continue to enter the marketplace. These users require varying degrees of instructional assistance to effectively operate the software they purchase. This demand for quality documentation places a burden upon software suppliers to expend additional time, energy, and money to satisfy users.

An examination of the documentation process reveals the complex nature of concisely and accurately documenting the steps of a computer program. Microcomputer users must reference information contained in the software package to answer their operating questions. Documentation that does not fulfill this primary obligation renders a program obsolete.

This research does not suggest complete replacement of current documentation practices. It does, however, recommend a set of procedural guidelines for microcomputer software suppliers to follow as a means of supplementing basic documentation techniques. For the purposes of this research, these guidelines pertain only to documentation manuals. The emphasis on manuals recognizes the continued benefit of maintaining hardcopy records of a program's

instructional material. These instructions are easily transported from one place to another, as opposed to other forms of documentation currently being considered.

This research was initiated two years ago. During this period, software suppliers have made significant improvements in their user manuals. These improvements are primarily the result of market pressures related to ineffective documentation. Since this research was initiated "early" in the microcomputer revolution, it reflects opinions and attitudes prevailing at that time.

1.2 Definitions

Microcomputer software documentation contains terms that are commonly used by software suppliers. Except for certain ambiguous terms noted in the literature review (Chapter II), most of the specific definitions used in this thesis are consistent with industry standards. The terms below are unique to this research and require definitions:

1. Documentation: The instruction manual associated with a software program, including any peripheral attachments such as reference cards, summary sheets, and hardware operating booklets.
2. Linearity: A term referring to the logical progression of instructional material from the beginning of a manual to the end.

3. Module: An independent instructional unit of documentation providing concise user assistance. The contents of modules include, for example, program commands, functions, and error messages, in addition to learning exercises and screen display illustrations.
4. Readability: A characteristic that indicates the user's overall impression of the software manual and the basic ease of use.
5. User Attribute: A characteristic such as microcomputer proficiency, educational background, and occupational role that distinguishes one user from others in the marketplace.
6. User Population: A group of microcomputer software users exhibiting similar attributes that distinguish them from other populations using the same, or different, software packages.
7. User-specific Documentation: "Personalized" instructional material suitable to the specific needs of a particular user population and resulting from the use of the recommended procedural guidelines.

1.3 Problem Statement

This research addresses one basic problem microcomputer software suppliers must face each time a program is developed. No single set of commonly used procedural guidelines appears to exist for writing documentation that provides different user populations with personalized operating instructions for the programs they buy. Associated with this single problem are the following:

1. User attributes may differ considerably among the user populations.
2. Information obtained from large-scale surveys of potential users may focus on a specific user population, but often provides no basis for distinguishing it from other potential user populations.
3. The documentation is usually written after the microcomputer program has been completed. As a result, documentation writers have the difficult task of accounting for subtle steps in a programmer's reasoning process.
4. No single documentation format (e.g., commonly used terms and manual sections) is available to assist users in locating similar material in different manuals rapidly.

5. Software suppliers have no simplified method of classifying user attributes or categorizing documentation topics.
6. Current documentation often does not provide easy access to a program's more advanced applications.
7. The quantity of instructions accompanying most manuals prohibits rapid referencing of specific items.
8. Documentation writers with considerable programming knowledge still may not fully appreciate the software's complexities; therefore, inexperienced users may receive less instruction than they need in order to use the program successfully.
9. Users may not fully recognize their limitations, capabilities, and specific documentation requirements when purchasing software.
10. Documentation research for microcomputer software is relatively new and does not possess a rigid set of criteria for determining the statistical validity of user surveys.
11. The proficiency and programming knowledge of advanced users may increase dissatisfaction with lengthy documentation, or even decrease sharply the operating efficiency of these users.

12. The dynamic nature of microcomputer software programming (i.e., more sophisticated versions of earlier programs) may create deficiencies in the initial documentation.

Each problem above reflects the dynamic nature of the microcomputer software industry and its relationship to a constantly increasing number of user populations. Problems will continue to arise as associated documentation is used and evaluated. The problems noted are commonly faced by software suppliers but do not, in fact, represent a comprehensive list of those difficulties attributed to selection, training, and retention of qualified technical writers.

1.4 Statement of Objectives

The primary objective of this research was to develop procedural guidelines for documenting microcomputer software. Accomplishing this objective required the identification of the following sub-objectives that address the principal components of the research experiment:

1. Review current literature on state-of-the-art microcomputer documentation practices. (Note: The reader is reminded that this research was initiated and primarily accomplished approximately 1.5 years prior to this writing.)

2. Identify and classify the individual attributes of the users participating in the research experiment.
3. Examine and categorize the documentation topics associated with the two word processing programs used in the research experiment.
4. Record and assess user responses to the word processing documentation provided by the software suppliers.
5. Discover the problems associated with using volunteer software users as aids in developing documentation.
6. Compare user needs and preferences with the format of sample word processing documentation.
7. Propose additional methods of obtaining information regarding the software's potential applications for particular user capabilities.
8. Demonstrate and qualitatively test the research experiment, including suggestions for improving research techniques.
9. Re-evaluate current documentation practices based on the research findings.
10. Establish criteria for educating users in how to choose software that offers suitable documentation.

The underlying motive for this research was a desire to supplement basic documentation practices with more user-specific procedures. Based on the results of the research experiment, Sub-objectives 3, 6, 9, and 10 were not completed. This was due primarily to a lack of sufficient detailed data regarding documentation topics (as discussed in Chapter IV), and research time constraints. Additionally, the exclusion of specific documentation reference standards (Chapters II and V) from the experiment reduced significantly the precision of user responses.

1.5 Assumptions

Consider the following assumptions related to the development of procedural guidelines in general, and the two word processing packages specifically:

1. Current literature is accurate in its assessment that state-of-the-art microcomputer software documentation requires a more innovative and personalized approach.
2. User attributes, software specifications, and program applications are dynamic in nature.
3. User attributes differ among various user populations.

4. The sampled volunteer users (i.e., industrial engineering undergraduate students) represent a single user population.
5. Each student user has a limited knowledge of documentation techniques.
6. The student users have little or no experience with the two word processing packages that were selected for review.
7. The word processing packages selected for the experiment are popular in the microcomputer software marketplace.
8. The user manuals associated with the two packages are good examples of current documentation.
9. Software suppliers do not usually develop documentation simultaneously with the program design.
10. Current documentation procedures do not involve the active participation of potential users.
11. Documentation improvements are the direct result of user feedback regarding software packages already in the marketplace.
12. Market pressures related to documentation shortcomings will determine which suppliers win or lose in the microcomputer software field.

1.6 Method for Meeting Objectives

The research method used to meet the objectives was an experiment that evaluated the acceptability of two common word processing manuals. This experiment's design originally included all of the steps listed below; however, the preliminary questionnaire used, for example, did not encourage specific categories of user responses. As a result, the student volunteers generated only limited quantities of significant data (as discussed in Chapter IV). Steps 8-13 require supplemental user information based on improved data collection techniques (presented in Section 5.2).

A basic summary of the original research method used is as follows:

1. Identification of volunteer student users who will review two word processing packages.
2. Orientation of these users regarding the research experiment's procedures, including a discussion of hardware operation and recordkeeping. The recordkeeping instructions, for example, request each user to record accurately the date and time of each review session.
3. Introduction of the associated user manuals, with a brief description of documentation topics.

These topics include, for example, command summaries, error messages, and tutorials.

4. Periodic supervision of users to assess initial responses to documentation irregularities and shortcomings. A quick inspection of logbook entries, for example, can indicate whether user responses are complete and accurate.
5. Personal interviews to supplement responses stimulated by the preliminary questionnaire. These interviews encourage spontaneous remarks by the users regarding serious operating problems.
6. Compilation of user information related to individual attributes and documentation preferences. This process notes, for example, each time a given documentation preference appears in the user logbooks.
7. Organization of user information into a matrix format for simplified referencing.
8. Classification of user attributes to establish a basic framework for developing a user population profile. The first classification level distinguishes, for example, between advanced users, intermediate users, and beginners.

9. Categorization of documentation topics. Principal categories, for example, are documentation format, content, and terminology.
10. Cross-matching of user attributes with documentation topics to determine suitable relationships. Advanced users, for example, may require only brief function command summaries to gain proficiency with basic program applications; whereas, beginners may require additional examples of correct inputs.
11. Reorganization of lengthy documentation text into independent modules (or chunks) of instructional material. The module containing, for example, a list of error messages may assist users in quickly overcoming mistakes.
12. Development and design of a framework that accommodates each instruction module without compromising its independence. This development and design process tests, for example, different module sequences and monitors the users' reactions to each.
13. Incorporation of boilerplate material into the above framework to soften the transition from one module to another. This material consists, for

example, of specific tables of contents highlighting important features of major sections.

1.7 Demonstration of Research Method

The demonstration of the research experiment used four independent-study students enrolled at Virginia Polytechnic Institute and State University. These student users judged the acceptability of two common word processing manuals. Data obtained from preliminary questionnaires and user log-book entries provided a basis for evaluating the overall effectiveness of this method.

CHAPTER II

LITERATURE REVIEW

2.1 Introduction

Literature regarding microcomputer software documentation is an item of increasing demand in today's technical marketplace. Numerous periodicals and software-oriented textbooks communicate microcomputer information between programmers, documentation writers, software suppliers, and users. These publications provide insight into the software program's specifications and the overall nature and effectiveness of associated documentation. Each of the following sections reviews current microcomputer software literature with particular interest directed toward documentation problems and possible solutions.

As noted earlier in Chapter I, the term "current" refers to literature available at the time the research was initiated. Since the inception of the research, a two-year period has elapsed, and a microcomputer software revolution has taken place. New literature is now available that notes significant improvements in software documentation quality made by software suppliers in response to market pressures. Demands of users will continue to determine who wins or loses in this competitive field.

2.2 New Documentation Reference Standards

As stated above, microcomputer software documentation has evolved in response to market pressures. Microcomputer software suppliers have responded to past complaints regarding the effectiveness of their user manuals. The most significant improvement in the documentation process has been the development of two specific reference standards, physical layout and instructional components as subsequently defined.

2.2.1 Physical Layout

Physical layout, narrowly defined, refers to the basic physical appearance of a single documentation page. A single page's overall appearance (or layout) is important to the manual's basic readability. This appearance communicates to the user nonverbal messages related to the documentation's clarity and effectiveness [26].

A broad definition of physical layout includes physical attributes of the manual. These physical attributes characterize the manual's packaging. Despite warnings against "judging a book by its cover," the basic appearance of a manual, quick-reference booklet, or keyboard template ultimately influences how well users receive the documentation [26].

The following criteria further define physical layout and provide a checklist for the documentation writer in developing a comprehensive user manual:

1. Page size
2. Paper weight
3. Type faces
4. Type sizes
5. Variation of type face (e.g., italics)
6. Number and types of print colors
7. Boldfaced headings for visual contrast
8. Number of instructional lines per section
9. "White" space (e.g., margin settings, paragraph indents, and spacing between lines)
10. Section dividers or tabs
11. Separate reference or learning booklets
12. Type of binding (e.g., three-ring)

2.2.2 Instructional Components

"Instructional components" refers to the specific instructional items presented by a manual and its associated user aids. Specifically, these items (or criteria) establish the basic format of the documentation. Each criterion below represents a minimum requirement for generating effective user manuals and supplemental aids [26]:

1. Format for section and paragraph headings
2. Format for page footers
3. Numbering system for pages, sections, illustrations, figures, tables, and instructions
4. Simplified terms based on common usage
5. Standardized word spellings
6. Descriptive abbreviations and acronyms
7. Accurate and concise wording
8. Active and imperative voices for writing style
9. Limited use of computer jargon
10. General table of contents
11. Specific table of contents at start of each major section
12. Basic hardware operating guidelines
13. Notes, precautions, and warnings
14. Function command/data entry modules (i.e., summaries)
15. Detailed command descriptions
16. Error messages
17. Description of potential program applications
18. Tutorials, exercises, and examples
19. Boilerplate information
20. Appendices
21. Glossary of important terms to clarify ambiguous meanings and eliminate misinterpretation

22. Index

2.3 Current Software Documentation

This section presents information related to current practices used by microcomputer software suppliers to document their programs. Specifically, it highlights problems associated with currently available documentation. A brief description of popular microcomputer-oriented publications provides a further understanding of how software suppliers communicate with users.

2.3.1 Documentation Practices

Software suppliers often document their programs at the conclusion of the programming process [23]. Large corporate suppliers use either in-house or contract technical writers to develop the necessary documentation. This documentation usually follows a manual format and contains standardized information despite differences in user requirements and software operating limitations [23].

The individual software suppliers are commonly the programming hobbyists or engineers who write documentation based on personal knowledge of a given subject. (Note: The reader is reminded that these early software suppliers were common about two years ago. Documentation now exhibits more

user perceptiveness.) The early years of the microcomputer industry produced technical writers who possessed computer knowledge, understood word processing fundamentals, and were motivated to record their programming ideas on paper [19].

2.3.2 Problems Associated with Documentation

Microcomputer software documentation exhibits various weaknesses due in part to a lack of universally accepted standards [22]. Technical writers, for example, do not have an established set of procedural guidelines to assist them in documenting programs for different users [12]. This deficiency results from the lack of suitable methods for re-searching the user/software relationship [19].

Manuals currently available may have one or more of the problems identified by software reviewers. The problems listed below provide a basis for recognizing the need for a single set of documentation guidelines [22]:

1. The manual has design weaknesses or "bugs" that decrease its effectiveness (e.g., logic errors, missing or inconsistent items, tangled structures, faulty syntax, and operations that exceed the capacity or abilities of the system or user) [24].
2. Insufficient instructional material forces users to learn by trial-and-error.

3. The manual is unprofessionally written, with misspelled words and grammatical errors.
4. Intended users find information written on an unsuitable level of understanding.
5. Instructional material is disorganized; therefore, users cannot locate pieces of information rapidly.
6. The manual lacks an index or has an incomplete one.
7. There is no table of contents to assist users in locating areas of instruction or in making an initial assessment of documentation quality before buying software [9].
8. Relevant error messages do not appear in the manual to explain incorrect entries by users.
9. The manual does not illustrate screen displays (with labels indicating the type and nature of information presented).
10. No "help" sections exist to aid the user with program troubleshooting.
11. The documentation is uninteresting and does not have a uniform and enjoyable style.
12. Manual style does not fit the subject or intended users.

13. Ambiguous terms and statements confuse or mislead the program's users, who may interpret them differently.
14. Advertising and promotional buzzwords in the manual misrepresent the program's potential applications by failing to specify the user proficiency required.
15. The documentation does not accurately describe the software's limitations.
16. The technical writer is a programmer and does not explain fully any assumptions made during the program's development.

2.3.3 Periodical Literature

Microcomputer software suppliers communicate their ideas to users by the manuals they produce. Periodical literature extends this documentation. The types of periodical literature are:

1. "Dedicated" literature: Dedicated literature describes publications which concentrate on a specific area of the microcomputer field. Many of these publications pertain to a particular brand of microcomputers and their associated software. For example, 80 Micro is a journal that features

articles and advertisements on the Tandy Corporation's microcomputers, peripherals, and compatible software.

Softalk, Nibble, and In Cider are examples of journals focusing on Apple personal computers. In addition to hardware systems-oriented information, this dedicated literature discusses software in general, specific classes of software (e.g., database management software), and potential applications of a particular software program. Dedicated literature provides the following additional information to increase user understanding of a program:

- i. Supplemental instructions regarding operating requirements of a program.
- ii. Summary of the documentation associated with a program.
- iii. Significant improvements of a specific software program over earlier versions and comparable products commercially available.
- iv. User assistance available from the software supplier, including toll-free numbers and company addresses, in addition to a brief description of services provided.

- v. Potential program applications and required user proficiency.
2. Professional publications: The periodical literature of professional organizations, technical societies, associations, and other "special-interest" groups features articles, technical notes, and advertisements relating to microcomputer systems (i.e., hardware, software, and peripheral equipment). Engineering conferences, for example, provide software suppliers with an open forum to exchange ideas and literature regarding program improvements. Potential users and suppliers receive additional input concerning documentation requirements and techniques [13].
 3. Publishing companies' periodicals and products: Major publishing companies have entered the microcomputer software field in various ways. A common method offers a book and related software as a package. This method has particular appeal to colleges and universities.

These companies also provide information regarding the general business side of the microcomputer hardware/software industry. Business-oriented periodicals expose new hardware and software ventures, and basic facts and figures to assist office automation. Consider the following:

- i. Business Week: This McGraw-Hill publication closely monitors the rapid expansion of the microcomputer hardware/software industry, including peripheral service industries. Specifically, it presents new hardware/software products becoming available and potential applications related to office automation. This periodical also addresses the impact of these products on office productivity, personnel requirements, and job scheduling.
- ii. McGraw-Hill endorsements: These endorsements provide vendors with additional outlets for their programs and guidance in writing the associated documentation.

2.4 Suggested Improvements of Documentation

Microcomputer software documentation is an important component of a software package. The diversity of available software requires a documentation method flexible enough to incorporate changes in software technology, presentation, content, and instructional approach. A project management method (described below in Sections 2.4.1 and 2.4.2) can provide basic techniques for documenting a program and maintaining its effectiveness [12].

2.4.1 Macro Design

The focus of the macro design is a "use" analysis that establishes the project scope. A user profile, project definition, preliminary documentation organization and parameters, and project management plan are additional elements of this design. These elements provide a framework for management, perform a quality control function, and ensure that micro design tasks are compatible with the overall documentation method. Figure 1 is a sample Documentation Project Synopsis Form [12].

"Use" analysis involves researching the specific situation in which a program is to be used. This research collects factual data by the following steps [12]:

**DOCUMENTATION
PROJECT SYNOPSIS
FORM**

Date:
Project Title:
Project Manager:
Use Analysis Summary:

User(s) Profile(s):

Project Definition (Goal):

Deliverables:

Documentation Outlines:

Action Plan Summaries:

In-house Personnel Requirements:
(Name, responsibilities,
phone number)

Outside Service Contract Requirements:
(Company, name, responsibilities,
phone number)

Budget:

Completion Date:

Figure 1. Documentation Project Synopsis Form

1. Identify what program is to be documented.
2. Determine why the program is being documented.
3. Establish who is going to use the program (i.e., "target" audiences).
4. Differentiate between various uses of the program by the target audiences (e.g., reference, learning data entry, and understanding output).
5. Isolate the exact needs of users regarding the documentation (e.g., a list of sequential steps to perform a particular function such as generating and reading a report printout).
6. Prioritize the needs isolated in Step 5.
7. Characterize the environment in which the program is to be used (e.g., computer room, office, or desktop).
8. Evaluate factors subject to change and the effects of changes upon the documentation's use.
9. Perform a user analysis and develop a precise profile incorporating, for example, computer proficiency and educational information.
10. Determine the documentation format based on the background data obtained in Step 9.
11. Define the nature of the project using both the user profile and the "use" analysis.

12. State the project goals clearly and concisely, observing their precise wording to ensure that these goals preserve the integrity of each decision/task used to complete the documentation.
13. Quantify resource parameters such as time, budget, and documentation personnel.
14. Organize the preliminary content of the documentation.
15. Develop a project management plan based on the information compiled in Steps 1-14, and store it in a word processor to allow changes to be made.
(Note: This plan can adopt many styles, including critical-path and matrix formats. It contains only essential information characterizing a particular situation. The project goals, budget, and schedule of task completion dates appear in the plan.)

2.4.2 Micro Design

The micro design consists of the research, writing, and editing elements of the documentation project. It involves the steps below:

1. Develop a brief action plan for each micro design element, based primarily on background data from

the document and project organization completed during the macro design. (Note: These action plans manage daily project operations and monitor the task completion schedule. Figure 2 is a sample Research Action Plan developed by identifying the information needed, the source of this information, and the required research techniques [12].)

2. Establish a schedule for completing the research.
3. Repeat Steps 1 and 2 for the writing and editing elements, and follow these procedures:
 - i. Select a technical writer with documentation training to write the manual. (Note: Programmers, systems analysts, and marketing departments do not perform the actual writing.)
 - ii. List each section and subsection of the manual to be written.
 - iii. Determine completion dates for each section and subsection, allocating time for typing, word processing, editing, revisions, proofreading, and review.

RESEARCH ACTION PLAN		
Manual Sections	Sources of Information	Completion Date
System Description	Marketing staff Specs Systems designer	
Keyboard Description	Specs Verify with hands-on experience	
Tutorial (writing a letter)	Hands-on experience	
Task Descriptions	Programs Specs System designer	
Error Messages	Programs Hands-on experience	
Commands	Programs Specs Systems designer	
Glossary	Notes from other sections	
Index	Final document	

Figure 2. Research Action Plan

2.4.3 Documentation Modules

The use of documentation modules or "docs" to develop a manual provides the software supplier with a basic set of information units. Each doc is an independent segment of the manual and consists of one or more pages of instructional material, depending on the type and quantity of documentation required. The steps below outline the method for combining these independent docs into an integrated user manual [24]:

1. Schedule conference. A documentation project team schedules a conference with the software's programmers, marketing specialists, and user consultants (i.e., sample users with real-world business experience). The user consultants (e.g., accountants, attorneys, or physicians) provide insight regarding their software requirements, personal capabilities, and operating constraints. Each group presents its recommendations and modifications for improving the documentation prototype (i.e., a rough version of the software's instructional material given in advance of the conference).

2. Develop master plan. The project team incorporates the recommendations and modifications into a master plan that structures the activities of each project team member in shaping the final documentation. Figure 3, a sample Master Plan, indicates the basic coordination of tasks and products associated with a structured documentation process [24]. The master plan includes:

- i. Recommendations and modifications proposed by each group.
- ii. Operating constraints imposed on each project team member to ensure overall compatibility of developed instructional material.
- iii. Requirements of each project team member to accomplish documentation objectives and avoid duplication.
- iv. Flowchart indicating the development of a manual, from an initial analysis of topics and users to the published document. (Note: Figure 4 is a sample Documentation Flowchart. Figure 5 is a sample User/Topic Analysis form for compiling data regarding the program's users and requested documentation topics [24].)

MASTER PLAN

TASKS	PRODUCTS
OUTLINE EACH "DOCUMENT"	◦ CONVENTIONAL OUTLINE ◦ STRUCTURED OUTLINE
DEVELOP STORYBOARD	◦ WRITE DOC SPECS FOR EACH HEADLINE ◦ ASSEMBLE "GALLERY"
REVIEW, REVISE STORYBOARD	◦ PLANNER REVIEW ◦ "AUTHOR" REVIEW ◦ "USER" REVIEW ◦ OFFICIAL REVIEW
ASSIGN DRAFTS	◦ WORK PLAN ◦ "FIRST DRAFTS" OF DOCS

Figure 3. Master Plan

DOCUMENTATION FLOWCHART

INPUT	PROCESS	OUTPUT
ANALYSIS OF TOPICS ANALYSIS OF USERS	TOPIC/USER MATRIX	DOCUMENTATION SET (DOCUMENT SPECS)
DOCUMENT SPEC	STORYBOARD ("WALKTHROUGH")	DOCUMENT DESIGN
DOCUMENT DESIGN	"FIRST DRAFT"	PRELIMINARY VERSION OF DOCUMENT
PRELIMINARY VERSION	TECHNICAL REVIEW STYLE REVIEW	PUBLISHED VERSION OF DOCUMENT

Figure 4. Documentation Flowchart

USER/TOPIC ANALYSIS						
USERS TOPICS	A	B	C	D	E	F
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						

Figure 5. User/Topic Analysis

3. Review project team organization. The project team reviews its organizational structure to ensure a suitable framework for optimizing the management and technical functions of the documentation process. Each project team member is responsible for writing a predetermined portion of the total documentation manual. The master plan provides the basis for reviewing the project team organization and assigning specific docs to each project team member (described below).
4. Assign functions (tasks). The assignment of functions provides a certain degree of specialization in the documentation process. A project team member develops and designs one or more docs depending on the degree of program complexity, the overall project schedule time constraints, and the number of project team members. For example, a word processing program may require the following functions which must be assigned to the project team members and put into modular form:
 - i. Creating/Formatting documents
 - ii. Editing documents
 - iii. Filing documents
 - iv. Deleting documents

v. Printing documents

5. Sequence independent docs. Each project team member submits a finalized version of the developed docs to the project team for further review and editing. The project team sequences the approved docs according to the master plan. This sequence establishes an integrated framework within the manual without compromising the independent nature of each doc.
6. Prepare final draft. The project team incorporates into the final draft any other recommendations and modifications considered significant to the sequenced docs. These finalized docs dictate the overall appearance of the manual and the readability of the documentation. Each item below supplements the final document and enhances the effectiveness of the docs:
 - i. Table of contents
 - ii. Description of program application
 - iii. Practice exercises and tutorials
 - iv. Precautions and warnings
 - v. Error messages
 - vi. Comprehensive glossary

vii. Index

(Note: Items ii, iv, and v are predictable patterns of organization and are repeated for each program task [26].)

2.5 Conducting Surveys

Information obtained directly from users surveyed is termed "primary" data. The personal interview method (discussed below) is useful to surveyors in acquiring primary data from sample users either on a one-to-one basis or in groups [6].

2.5.1 Personal Interviews

A significant characteristic of the personal interview method is the high percentage of acceptable returns obtained. Careful planning and proper organization can yield even higher percentages. For example, surveys of the general public can accurately predict cooperation from approximately 80% of the users sampled. The use of additional special procedures (e.g., questionnaires) can yield complete and accurate data for more than 95% of sampled users.

2.5.2 Advantages of Personal Interviews

The surveyor must consider the advantages and disadvantages of the personal interview method to assess its usefulness to the specific problem being addressed. This section lists the advantages realized by using personal interviews to obtain primary data. Section 2.5.3 focuses on the disadvantages of this method. Consider the following advantages [6]:

1. Personal interviews usually yield a high percentage of complete and accurate returns due to respondents being commonly receptive to these surveys.
2. Each participant in the survey can more effectively respond to, and be identified by, the interviewer. This yields a less biased sample of the population surveyed.
3. The interviewer can better maintain the integrity of information obtained by explaining the questions to the respondent, and dealing immediately with obvious inconsistencies in the answers.

Unintentional or deliberate falsifying of information can indicate to the trained interviewer a need for further clarification regarding a given question.

4. Personal interviews allow the interviewer to obtain supplemental data concerning the respondent's personal characteristics and attitudes. These are useful in interpreting and evaluating responses, as well as determining if the respondent is representative of the population studied.
5. Acting as an experimenter, the interviewer can introduce scoring and test devices into the personal interviews to measure the respondent's capabilities.
6. The interviewer can present visual material to which the person surveyed responds, such as equipment to be operated, instructional manuals, or diagrams and illustrations.
7. Additional interviews can assist the interviewer in completing each item on the questionnaire and correcting any mistakes identified. This eliminates extensive written correspondence between a surveyor and a respondent. Furthermore, these follow-up interviews guarantee a higher percentage of usable (i.e., complete and accurate) questionnaires.
8. The interviewer may catch the respondent off-guard and obtain more spontaneous reactions. Written

questionnaires, however, allow the respondent to reconsider initial responses. This second-guessing compromises the integrity of candid responses.

9. Group discussions using the personal interview method are also possible. The interviewer can usually control which person (or persons) answers the questions.
10. Personal interviews allow the respondent to become better oriented to the survey topic. This can increase the quantity and quality of information obtained from the respondent.
11. An experienced interviewer can strategically place sensitive questions directed toward the respondent. The interviewer can change the subject if necessary or explain the survey problem further if the respondent is avoiding a question.
12. The personal interview method dedicates more of the respondent's time to answering questions, since the interviewer is responsible for recording the responses.
13. An interviewer can adapt the terms used to suit the ability or educational level of the respondent. This helps to avoid misinterpretations or misleading questions.

2.5.3 Disadvantages of Personal Interviews

Consider the following disadvantages of the personal interview method [6]:

1. Transportation costs and the time required to locate respondents may make the personal interview method economically infeasible.
2. The human element may distort the returns. If any interviewer has certain biases, this person may unconsciously ask the questions in a way that confirms personal views.
3. An improperly trained and supervised interviewer may record incomplete and inaccurate data.
4. The organization required for selecting, training, and supervising the interviewer is more complex than that required for surveys conducted by other methods.
5. The costs associated with each personal interview are higher for face-to-face data collection than with telephone or mail surveys. This may, however, not be true if the location of the interview is not a problem for the parties involved. Additionally, if the population studied is large, the costs of obtaining a representative sample by

telephone or mail will probably equal or exceed the cost associated with the personal interview method, since personal follow-up will be necessary.

6. Personal interviews usually require more time than telephone interviews and, therefore, do not yield results as quickly.

2.6 Preparation of Questionnaires

Selecting opinion questions suitable for a given survey is as complex as obtaining representative samples of user populations, and may influence the results more than a poor sample. Developing unbiased questions requires experimenting with different types of question formats and phrasings.

The type of questions used in a survey depends on the following factors [6]:

1. Quantity of background research needed to identify the nature and scope of the survey topic.
2. Extent to which the survey topic is in a crystallized (or vague) state.
3. Potential of population studied to formulate opinions and biases concerning this subject matter.

4. Complexity of the survey topic and qualifications associated with it.

2.6.1 Questionnaire Format

Each survey addresses the above factors and the limitations of different forms of questions in the development and design of a suitable questionnaire. One question type focuses on the magnitude of the opinion held by the respondent regarding a particular subject (e.g., excellent, good, average, fair, and poor categories). A second type deals with a format that allows the person surveyed to respond freely and at length about the topic. The third question type restricts the respondent to choosing between two or more alternatives [2].

2.6.2 Open-End Questions

The preliminary phases of a survey require using questions which give the respondent more flexibility in answering them. Effective planning and development of questions entail a thorough understanding of specific aspects of the survey topic that have made a significant impression on the respondent. Additionally, the specific terms used by the population studied are an important consideration.

Large studies using open-end questions yield such a variety of responses that the task of classification and analysis is both time-consuming and difficult to handle statistically. Small surveys or pilot studies more effectively use the open-end questions. However, the development of improved techniques for analyzing narrative materials systematically will increase the objective handling of data regardless of survey size.

CHAPTER III

RESEARCH METHOD

3.1 Introduction

This chapter describes the procedures followed in obtaining information regarding the documentation associated with two current microcomputer word processing programs. Four university students provided background information regarding the personal characteristics (attributes) associated with a given user population. Specifically, these students evaluated the word processing documentation under close supervision. The research method (i.e., experiment) incorporated the basic information gathering techniques presented in the literature review (Chapter II); however, most of the research experiment evolved in response to specific problems encountered by each student.

3.2 Description of Research Experiment

The research experiment used to meet the objectives (stated in Chapter I) incorporated several survey experiments currently in practice or under investigation by large microcomputer software vendors. This experiment recognized the advantages connected with conducting personal and group

interviews for obtaining specific user population information. The use of a questionnaire also assisted the information gathering process. The procedures that appear in Sections 3.2.1 through 3.2.9 are the principal steps of the research experiment.

3.2.1 Selection of Software/Hardware

The selection of two word processing packages (i.e., Applewriter II and Screenwriter II) provided greater flexibility in scheduling the student users' documentation review sessions. Their popular appeal and similarity were bases for selecting these particular packages for use with the Apple II+ microcomputer.

3.2.2 Selection of Research Location

Several factors made the Productivity Evaluation Center (Department of Industrial Engineering and Operations Research, Virginia Polytechnic Institute and State University) a suitable location for conducting the documentation research:

1. The center restricts access to its microcomputer hardware, software, and resource materials. A posted hardware/software operating schedule reduces the waiting time for students using the center.

2. Sufficient illumination and climate control create comfortable environment for users.
3. Adequate supervision by authorized university representatives (e.g., professors, graduate students, and secretaries) reduces noise and confusion, and ensures a professional work setting.
4. A Graduate Research Assistant is available to assist users in operating the hardware and software, in addition to locating specific reference information from user guides and periodicals.
5. Each microcomputer workstation has enough table space for software manuals and notebooks.
6. A centrally located conference table provides ample seating for small user meetings and group discussions.

3.2.3 Identification of User Population

Four Industrial Engineering students (enrolled in an independent study course) volunteered to use the word processing software and review the associated documentation. These students had basically similar educational backgrounds and capabilities, based on preliminary information obtained from their professor. Each user's course grade was dependent on the degree of cooperation exhibited by the student

during the conducting of the research. This helped to maintain the integrity of the user's participation in the research.

3.2.4 Orientation of Users

An initial orientation meeting with the volunteer users included an open discussion of the following topics:

1. Nature and Scope of Research. The first meeting with the sample users focused on the definite need to improve the documentation associated with this software based on the results of the preliminary literature review. A statement of the research objectives provided the users with the motivation for soliciting their assistance.
2. Apple II+ Operating Procedures. An open-ended questionnaire asked about the users' operating experience with this microcomputer. Three of the users (i.e., Users B, C, and D) indicated an understanding of microcomputer fundamentals, while admitting only limited exposure to the Apple II+. User A had a general knowledge of microcomputers and demonstrated proficiency with this specific brand of personal computer.

3. Applewriter II and Screenwriter II Software Packages. Each user received information regarding the reasons for selecting these packages. Only User A was familiar with their general contents. The location of these software packages (and other helpful reference documents) was highlighted.
4. Preliminary Questionnaire. This questionnaire (Figure 6) requested the following personal information from each user:
 - i. Microcomputer experience and proficiency
 - ii. Formal microcomputer classroom instruction
 - iii. Engineering coursework completed

Each user received a questionnaire to be completed and returned during the second orientation meeting (discussed in Section 3.2.5). Additionally, the preliminary questionnaire contained a brief description of the types of potential errors and deficiencies which each user could encounter during the documentation review.

3.2.5 Final Orientation Meeting

The questionnaire also asked each user to indicate how much time was available in the user's class schedule for

reviewing the documentation. After developing an operating schedule (Figure 7) for using the Apple II+, each user was provided with the following review guidelines:

1. Obtain the designated software package and proceed to the Apple II+ workstation. Based on a random selection, Users A and B would use Screenwriter II, and Users C and D would use Applewriter II.
2. Organize the documentation manual and support materials for easy referencing during the review session.
3. Record the exact START and STOP times associated with each session. A 12-hour maximum was the total amount of time allowed to review the documentation. Each user agreed that this maximum amount of time closely approximated the time available in a real-world business setting for learning to operate a word processing program. Additionally, a three-hour maximum was set for each review session to reduce the effects of fatigue and increase the operating efficiency of the user.
4. Record the date on which the review session occurred. The users were asked to complete the review sessions within a two-week period. This

requirement would enable the results of each user's documentation review to be compiled within a reasonable length of time. It would also approximate the scheduling constraints and budget limitations set by real-world software vendors.

5. Enter any comments or observations regarding the documentation in a logbook format (i.e., cite specific pages of the documentation manual and record remarks immediately following them).
6. Use flowcharts, if desired, to illustrate the organization of the manual's contents or the direction (i.e., forward or backward) the user must follow to access the information needed to operate the program.
7. Demonstrate any acquired proficiency in using the software by generating a printout of some or all of the recorded logbook information.
8. Consult the Graduate Research Assistant regarding Apple II+ operating procedures. Each user should formulate questions related to the documentation to obtain supplemental instructions as needed.
9. Minimize interruptions during the review session (e.g., coffee breaks and casual conversations) to maintain the integrity of the session's allotted time.

10. State all documentation shortcomings, personal attitudes and preferences, and operating experiences in a candid and specific manner.
11. Recommend modifications, additions or deletions for improving the documentation.
12. Submit this recorded information to me as prescribed during the final orientation meeting.

3.2.6 Periodic Supervision of Users

During the two-week documentation review period, the review sessions were periodically supervised to assess initial user responses to documentation irregularities and shortcomings. Each user had an opportunity to raise questions regarding review and recording procedures. Careful observation of the operating habits of each user provided a better understanding of what motivated specific responses to the documentation.

3.2.7 Personal Interviews

Personal interviews with each user prior to the actual review session supplemented the responses generated by the preliminary questionnaire. These interviews provided additional information regarding difficulties experienced by each user in using the documentation, and each user's

personal limitations in operating the Apple II+. Section 3.3.2 discusses these difficulties based on personal interview information.

3.2.8 Review of User Information

The users' notes indicated individual attributes and documentation preferences. This review consisted of a line-by-line reading of all available information to note specific items that were significant to the characterization of the user population and the improvement of documentation material, redundant items of information, and lengthy passages in the logbook entries that could later be organized into concise, manageable topics (introduced in Section 3.2.9).

3.2.9 Organization of User Information

The information was reviewed and compiled by grouping specific types of information obtained from the users. For example, computer-oriented education listed by Users A, B, C, and D was grouped into manageable units of information. This task was facilitated by instructing the users to organize their comments in a manner compatible with the experiment. This organization provided a simplified framework for referencing specific items of information rapidly. The

simplified units of information also provided a basic foundation for developing information summaries.

3.3 Demonstration of Method

The research experiment required Users A, B, C, and D to fully document their documentation review efforts. Each user followed a prescribed set of recording instructions. During the periodic supervision and personal interviews, each user received assistance in developing a recording format compatible with these instructions. Appendices A, B, C, and D (outlined below) contain the information submitted during the final orientation meeting and at the completion of the documentation review period:

1. Appendix A: Profiles of Users A, B, C, and D that contain brief descriptions and composite lists of computer-oriented classroom instruction.
2. Appendix B: Personal data sheets and resumes for Users B, C, and D that provide supplemental information regarding supporting coursework and microcomputer experience.
3. Appendix C: Software review logbooks submitted by Users A, B, C, and D that consist primarily of any observed documentation weaknesses, inconsistencies, or errors.

4. Appendix D: Recommendations proposed by Users A, B, C, and D that include personal documentation preferences and requirements.

(Note: The appendices contain unrevised questionnaire and logbook information. However, minor typing changes have been made to improve the neatness and general appearance of the logbook entries.)

CHAPTER IV
EXPERIMENTAL RESULTS AND PRELIMINARY FINDINGS

4.1 Introduction

The research experiment evolved from a comprehensive documentation review to a structured data collection process. A principal objective of the experiment was to provide a framework for testing the acceptability of two common word processing manuals and the overall user/documentation relationship. The discrepancy between actual and expected research gains justifies improving data collection techniques and recommending specific procedural guidelines for future documentation reviews (as discussed in Chapter V).

This chapter presents the major results of the research experiment and preliminary findings supported by the collected data. Evaluating these results and findings requires identifying the final steps of the original research method (summarized in Section 1.6). These steps (deleted from the experiment due to time constraints) appear below because they help to explain the significance of collected data:

1. Classification of user attributes to establish a basic framework for developing a user population profile.

2. Categorization of documentation topics.
3. Cross-matching of user attributes with documentation topics to determine suitable relationships.
4. Reorganization of lengthy documentation text into independent modules of instructional material.
5. Development and design of an organizational framework that accommodates each instruction module without compromising its independence.
6. Incorporation of boilerplate material into the above framework to soften the transition from one module to another.

4.2 Evaluation of Results and Preliminary Findings

This section evaluates the feasibility of each deleted step based on the quantity and quality of collected data. It also discusses the organization of user data using matrix summaries developed at the conclusion of the data collection process. (Note: These summaries appear in Appendix E.)

The collected data consists mainly of information related to individual user attributes and documentation preferences. As a result, the evaluation concentrates on how well the data helps to increase the understanding of user needs. The data supports certain preliminary findings that provided a basis for the recommended procedural guidelines (Section 5.3).

4.2.1 Organization of User Information

Matrix summaries simplify referencing of basic user information and documentation preferences. These summaries indicate attributes and preferences exhibited by one or more of the sampled users. Common attributes or preferences (i.e., those exhibited by each user) are readily identifiable in each matrix summary.

The quantity and quality of collected data dictate the use of matrix summaries to reveal trends in user responses to the Preliminary Questionnaire. These trends promote a better understanding of the user population's capabilities and limitations. Additionally, the trends provide a basis for distinguishing one user population from another.

Each summary represents a starting point for isolating attributes or preferences shared by different user populations. These shared attributes or preferences reduce the degree to which documentation must be personalized. This reduction favors a least-cost method of personalizing documentation while maintaining certain boilerplate material.

4.2.2 Classification of User Attributes

A detailed classification of user attributes would require a much larger sample population than that used during the course of this research. Information contained in the

summaries, however, highlights the basic areas which impact the documentation process. The summaries also establish a basic classification framework. Supplemental data collection can further define the individual items listed in each matrix summary. Such data increases the accuracy and precision of the documentation that evolves in response to specific user attributes or preferences.

4.2.3 Categorization of Documentation Topics

Logbook entries recorded by Users A, B, C, and D do not categorize documentation topics beyond the level of basic format, content, and terms used. User responses reflect limited knowledge of alternatives. These responses further reveal that each user was preoccupied with only the general operating instructions. The users recommended only simple changes to the documentation (e.g, changing the sequence of major sections). Only User D's modified flowchart introduced a more sophisticated documentation framework that addresses specific function capabilities and commands.

4.2.4 Cross-matching of Collected Data

Matching user attributes or preferences and documentation topics entails the development of correlations based on the collected data. Ideally, additional data collected from

a larger sample population would ensure greater statistical validity regarding the determination of suitable relationships between the user population and documentation provided. The matrix summaries generated by this research do, however, reveal some basic correlations (e.g., above-average technical aptitude and advanced program applications).

4.2.5 Reorganization of Documentation

Users A, B, C, and D indicated a preference for independent documentation modules. Such modules would allow the reorganization of lengthy documentation into chunks of instructional material. Specifically, each user expressed dissatisfaction with having to read lengthy text passages in order to search for key commands.

4.2.6 Development and Design of Documentation Framework

User responses regarding the development and design of a framework for organizing modules are not specific. Flowcharts submitted by Users B and D suggest a basic restructuring of the manual's contents and an emphasis on more manageable chunks of information. These flowcharts do not, however, recommend specific contents or sequencing of the modules.

4.2.7 Incorporation of Boilerplate Material

The incorporation of boilerplate material into a user manual recognizes the significance of certain standardized text regardless of the user population. User logbook entries did not consider, for example, the advantages of this material in softening the transition from one module to another. User responses did, however, support the use of specific items of information (e.g., a table of contents for each section).

4.3 Data Collection Problems and Findings

The following problems and findings support the personal recommendations for improving documentation review procedures. These problems and findings illustrate the significance of developing comprehensive data collection methods. The sections below also contain some suggested research improvements justified by user responses.

4.3.1 Sample Population

Users A, B, C, and D represented a relatively small sample of the Industrial Engineering student population. They did reflect similar educational backgrounds, however, based on limited user profile data. Each user regulated the time expended and logbook quantity/quality under the strict

guidance of their course instructor who outlined course requirements.

4.3.2 Collected Data

The data collected suggests the types of information that are necessary to personalize the documentation of microcomputer software. For example, user responses indicated a need for more user-interactive tutorials. However, four data points represent a relatively small sample size, and the statistical validity of the data collected is obviously very low.

4.3.3 Open-End Questions

These questions did not provide sufficient information to assist each user in responding effectively to the Preliminary Questionnaire. A more structured questionnaire format (e.g., multiple-choice questions) probably should have been used. This would limit user responses to precise documentation topics and would, therefore, simplify data compilation. (Note: The research experiment did not use a structured format to avoid the possibility of coaching the users; however, a structured format would eliminate the difficult process of reviewing lengthy logbook entries.)

4.3.4 Control Group

The lack of a control group of users reduced the significance of conclusions regarding learning time. Each user's above-average technical aptitude may have decreased the user's performance level in using the word processing documentation, due to simple errors in judgment (e.g., attaching dual meanings to simple terms based on prior software experience). A control group of nontechnical users might have experienced less difficulty since they would have simply followed each instruction as directed by the manual.

4.3.5 Time Constraints

The 12-hour time constraint imposed on each user for reviewing the documentation turned out to be totally impractical. Each user indicated (i.e., during personal interviews) that recording information in the logbook was very time-consuming and disruptive to the learning process. At least 25-30 hours should have been allotted to allow the organization and recording of detailed responses. This increase in the maximum allowable time would also provide flexibility for each user to learn basic hardware operations.

4.3.6 Performance Rating System

User logbook entries were unsuitable for establishing a performance rating system. A rating system would compare the users' progress in effectively using the documentation. This system would also provide an estimate of the time required by future users exhibiting similar attributes.

The entries do reveal that prior computer coursework did not eliminate the need to follow the manual's instructions on a line-by-line basis. Minor variance between Users A, B, C, and D did occur, however, regarding the types of problems encountered. Specifically, Users A and B recorded less difficulty in locating vital pieces of information than Users C and D.

4.3.7 Quick-reference Modules

Users A, B, C, and D indicated a preference for quick-reference documentation modules. User logbook entries and responses made during personal interviews reflect this preference. Each user, however, failed to identify the principal components or the precise ordering of the desired modules within the documentation framework. As a result, user recommendations support the use of instructional summaries less sophisticated than a well-defined documentation module. Development and distribution of a basic module

would have allowed each user to more systematically complete each component.

4.3.8 Personal Interviews

Personal interviews indicated considerable interest in gaining hands-on experience with basic program applications. Specifically, each user recommended a set of basic instructions (e.g., for simple letter generation). These basic instructions would not include all available options provided by the program for a given application. Users A, B, C, and D recognized the advantages of basic instructions in generating simple output for use as a learning base.

4.3.9 Function/Command Sequence Modules

Users B, C, and D favored the development of function/command sequence modules. These modules would provide each user with only the commands needed to perform a specific function. The result of a given command would appear immediately adjacent to the command. This alignment of commands and results would reinforce correct command inputs and indicate errors in judgment for each step of a given function.

4.3.10 Documentation Prototype

The documentation manuals selected are not suitable for incorporating major modifications recommended by each user. Minor modifications (e.g., basic command summaries) are possible; however, many text deletions are required to reduce the documentation to a step-by-step instructional format. A documentation prototype (i.e., simplified instructional framework) would allow greater flexibility regarding additions, deletions, and modifications of the basic information provided. The user manual would evolve during a series of overlapping modifications in a building-block fashion. Each user would also test each modification immediately after incorporation into the basic framework.

4.3.11 User Interaction

During the experiment, the users interacted very little during the documentation review process. Each user was requested not to communicate with the other users in order to maintain the integrity of individual responses. A closely monitored coordination of user activities, however, may have increased the operating efficiency of the users without ignoring the capabilities and limitations of each. This arrangement would also recognize the team approach to problem solving commonly used in real-world settings.

4.4 Procedural Guidelines for Documentation

As stated in Section 4.1, the research experiment evolved from a comprehensive documentation review to a structured data collection process. This discrepancy between actual and expected research gains justifies the documentation review procedures proposed in Section 5.3. These proposed procedures do not represent a comprehensive list of documentation guidelines; however, they provide a starting point to those responsible for documenting a given software program. Moreover, treating these procedures as a comprehensive list would misrepresent the actual results obtained by the experiment.

CHAPTER V
ANALYSIS AND CONCLUSIONS

5.1 Introduction

This chapter provides a detailed analysis of the research experiment and conclusions related to the documentation's effectiveness. Additionally, it proposes procedural guidelines designed to improve the experiment's data collection techniques and help future documentation writers more accurately gauge user capabilities and limitations. The detailed analysis, conclusions, and proposed procedural guidelines are the result of the following:

1. Personal evaluation of experimental procedures using simple logic.
2. Critique of experimental design based on new reference standards (i.e., physical layout and instructional components).
3. Comparison of personal evaluation and new literature critique.
4. Discussion of costs and problems associated with full-scale experimentation.

(Note: The reader is reminded that "new" literature refers to documentation reference standards available today in the microcomputer software marketplace. These standards were not in use at the time this research was initiated.)

5.2 Personal Evaluation of Experiment

The research experiment was designed to test the acceptability of two common word processing manuals. This experiment, however, evolved from a comprehensive documentation review to a structured data collection process. The discrepancy between actual and expected research gains is the basis for modifying the experimental procedures.

Several problems occurred during the research experiment. The users experienced certain difficulties in operating the Applewriter II and Screenwriter II software and recording their observations. These problems and difficulties appear in the following sections.

5.2.1 Research Design Problems

The research experiment was originally designed to provide a comprehensive set of user attributes and documentation preferences, based on a review of the user manuals associated with the Applewriter II and Screenwriter II software. Each orientation meeting included a discussion of

the required recording procedures. During the two-week documentation review period, however, the following problems occurred:

1. Preliminary learning-time estimates (i.e., 12-hour maximum) did not allow for items such as the time needed to record responses.
2. The quantity of information obtained from each user was considerably less than expected. User A contributed much additional information during personal interviews; however, only a small amount of recorded material was submitted. The original plan was to compare the progress of Users B, C, and D against that achieved by User A. A motivation for this exercise was to determine if a strong correlation existed between demonstrated microcomputer proficiency and learning time.
3. Recorded information from each user varied in quality. User B made detailed logbook entries for each page of the documentation, and illustrated difficult concepts using flowcharts. User C generated a printout of the recorded information using the Applewriter II program. User D's responses were more closely in line with the requested format. These different recording formats

decreased the ability to quickly reference similar information from different users.

4. The more experienced microcomputer users (i.e., Users A and B) repeatedly made assumptions regarding Screenwriter II's basic instructions. Additionally, each was guilty of experimenting with the program based on the design capabilities of other software. Each of these users compounded simple documentation weaknesses by selecting commands on a trial-and-error basis.
5. Users A and B (reviewing Screenwriter II) indicated dissatisfaction and frustration regarding the large amount of introductory material provided in the documentation. During the periodic supervision of these users, they repeatedly attempted to randomly select various passages in the documentation in learning the program. Each user wasted valuable time and was forced to reread earlier items of information.
6. Users C and D (reviewing Applewriter II) provided only a minimal amount of additional information during personal interviews. Each of these users was cooperative, but was unable to discuss documentation weaknesses beyond the level of basic program applications.

7. Due to the limited progress realized by each user, the review of user preferences and requirements addressed only those initial sections of each program's manual. Similarly, the scope of the research experiment was restructured to include only information gathering, review, and organization procedures.

5.2.2 User Difficulties

Each user experienced difficulties in reviewing the word processing software and recording individual observations. These difficulties accounted for some of the inconsistencies in the logbook entries. Each of the difficulties listed below decreased the users' overall effectiveness:

1. The documentation repeatedly failed to reinforce correct inputs by illustrating in the text how the screen display should appear. Each user admitted that considerable time was wasted in continuing to read and follow instructions which later revealed an earlier incorrect command entry.
2. Users B and C complained about their inability to clearly distinguish between a documentation weakness and a simple misinterpretation of the instructions.

3. The more experienced users (i.e., Users A and B) attributed dual meanings to many of the terms used by the Screenwriter II program. The users agreed that their prior software operating experience caused them to second-guess even simple terms.
4. Each user found the process of recording exact error types (introduced by the Preliminary Questionnaire) extremely difficult and time-consuming. Despite continued assistance, the users often could not readily identify these errors in the text.
5. Each user was unable to assess quickly and accurately what information should be entered into the logbook.

5.3 Recommended Procedural Guidelines

Chapter IV highlighted the major experimental results and introduced some potential improvements to the research method. Similarly, Section 5.2 indicated specific problems related to the experimental design. The most significant conclusion regarding the research experiment is the need for more user data. Specifically, the user information compiled in the matrix summaries does not address detailed documentation topics. The findings from this research suggest that

the following procedural guidelines for improving the data collection process are needed:

1. Identify the documentation manual to be reviewed.
2. Categorize the software program's specifications (e.g., functions, commands, and potential applications).
3. Organize the manual's instructions into more manageable chunks of information.
4. Develop one independent instruction module for each chunk.
5. Sequence the modules according to the manual's basic outline to provide a framework for incorporating recommended modifications.
6. Design a documentation prototype that includes the instruction modules and essential boilerplate material (e.g., a general table of contents, glossary, and index).
7. Target specific audiences (i.e., user populations) for the documentation.
8. Isolate one population for the documentation review process.
9. Select qualified representatives of the user population (e.g., practicing accountants, attorneys, engineers, and physicians) to act as paid consultants during the review process.

10. Design and distribute to each user consultant a detailed preliminary questionnaire containing structured user response categories (e.g., work experience, educational background, general documentation preferences, and desired program applications).
11. Use the following steps to develop a user population profile based on attributes shared by the user consultants:
 - i. Develop a comprehensive list of user attributes for each user response category on the preliminary questionnaire.
 - ii. Isolate and compile the attributes shared by all user consultants.
 - iii. Classify groups of users within the population (e.g., a project team member, project manager, executive, and independent user).
 - iv. Identify sub-classes of users (i.e., beginner, intermediate, or advanced).
12. Select a suitable real-world setting (e.g., a corporate training department or small office) in which to conduct the documentation review process.

13. Schedule a group meeting with the user consultants and perform the following steps:
 - i. Introduce the software and its associated documentation prototype.
 - ii. Highlight the program's potential applications.
 - iii. Discuss the basic contents of popular manuals (e.g., general and specific tables of contents, independent instruction modules, a glossary of key terms, and a detailed index).
 - iv. Describe the types of errors commonly found in current documentation.
 - v. Illustrate a simple module framework and its key components (e.g., sequenced commands, screen display illustrations, precautions, and warnings).
 - vi. Present a standardized format for recording logbook information and encourage detailed responses to documentation deficiencies.
 - vii. Introduce the microcomputer hardware selected for the review process and discuss the basic operating procedures.
 - viii. Recommend cooperation between the user consultants to promote the exchange of basic

- operating information and encourage brainstorming of new ideas.
- ix. Coordinate the documentation review schedules of the user consultants to reduce conflicts regarding use of the hardware or software.
 - x. Establish time constraints that allow for items (e.g., coffee breaks, interruptions, and recording user responses) not directly related to operating the software.
-
- 14. Conduct personal interviews with each user consultant to monitor individual progress, clarify recommended changes, and maintain prescribed recording procedures.
 - 15. Schedule group interviews to gain a consensus of opinions regarding operating difficulties or recommended modifications.
 - 16. Contact the documentation's technical writers to address problems commonly encountered during the review process and troubleshoot minor documentation weaknesses.
 - 17. Establish a testing program to gauge user competence and evaluate recommended modifications.
- (Note: Software suppliers today commonly use "alpha" and "beta" testing procedures for

developing effective documentation. Basically, alpha users test a prototype version of a given manual to identify problem areas. New sample, or beta, users are introduced to further test modified versions of the manual [26].)

18. Distribute a final questionnaire to each user consultant requesting final suggestions and recommendations.

The procedural guidelines listed above introduce some potential improvements to the research experiment's data collection process. These improvements require, however, additional research to assess their significance in writing effective documentation. User responses obtained from a larger sample base (i.e., via mailed questionnaires) may, for example, increase significantly the statistical validity of recommended modifications.

5.4 Critique of Experimental Design

During the course of this documentation research, the microcomputer software marketplace has dictated fundamental changes in the overall quality of user manuals. New software literature has introduced two documentation reference standards commonly used to evaluate the effectiveness of these manuals. The two reference standards (i.e.,

physical layout and instructional components) include numerous criteria for assessing the merits of current documentation.

Data collected from the research experiment indicates a basic deficiency in its design. Specifically, the experiment did not include detailed guidelines regarding aspects of the physical layout. User logbook entries do reveal, however, specific comments concerning the limitations of the tutorials provided in the manuals reviewed.

Each section below describes some of the more visible criteria influencing the documentation review process. These criteria provide a basis for critiquing the research experiment's design. The criteria presented also demonstrate the expanding role of the software supplier in producing user friendly manuals.

5.4.1 Loose-leaf, Hardbound Manual

A loose-leaf, hardbound manual stays open by itself. It also provides users with greater flexibility in flipping back and forth between sections and increases the lifespan of the pages. Quality manuals include the following [25]:

1. Function key template to visually reinforce vital keys.

2. Quick-reference booklet for rapid accessing of major commands.
3. Tutorials to test user understanding of subject matter.
4. Easel for supporting instructional material and providing more available workspace.
5. Supplementary disks (i.e., containing the tutorial) compatible with text to increase overall operating efficiency of program.

The criteria above are examples of the manual's physical layout (defined in Section 1.2). The following sections address the instructional components associated with these examples.

5.4.2 Function Key Template

A color-coded template placed on the keypad clearly identifies the major commands performed by each key. This template guides users when pressing each key individually or with the shift, alternate, or control keys. One prescribed color directs users to press a function key and shift key. A different prescribed color directs users, for example, to press a function key and the alternate key. The manual's instructions follow this color-coded system to assist users in moving between the manual and the keypad while learning each command.

The research experiment did not examine the benefits of using a function key template. Each user indicated that the transition from manual to keypad (i.e., without template) was extremely difficult. The keypad simply did not reinforce the required function commands, and the users were unable to recommend an organizational structure to remedy this situation.

5.4.3 Quick-reference Booklet

This booklet provides users with brief summaries of major function commands and required keystrokes. It also outlines the instruction categories contained in the manual's text. A quick-reference booklet is a useful tool for users because it:

1. Introduces the basic categories of the documentation.
2. Highlights the program's basic applications.
3. Indicates the predetermined module sequence found in the manual.
4. Demonstrates the program's overall level of sophistication.
5. Provides a basis for evaluating the quantity and quality of documentation provided by the manual.

During the course of the research experiment, the users could not effectively incorporate into the review process the quick-reference booklet included. Each user experienced considerable difficulty learning basic commands contained in the booklet or the abbreviated command sequence. The users also could not reference the manual for a detailed explanation of the booklet's required keystrokes, because the former was equally confusing.

5.4.4 Supplementary Disk-based Tutorial

An effective tutorial is an extra learning aid that gauges user understanding of the manual. The text instructions associated with the on-screen tutorial prepare users to perform major functions. Each page of the manual's tutorial section contains an accurate illustration of what appears on the screen. This allows users to quickly compare the screen and text illustration to determine if a discrepancy exists. Color-coded drawings of the keys required by each command further reinforce correct user responses.

The users involved in the experiment indicated the need for a comprehensive tutorial requiring more user participation. The experiment did not yield detailed recommendations for modifying the tutorial provided. It highlighted, however, the users' interest in documentation that challenges

their competence. Specifically, the users requested realistic exercises that provide a foundation for the program's practical applications.

5.4.5 Page Layout

As defined in Section 1.2, a manual's physical layout may consist of several components. One critical component of the physical layout is the typed page. Page design (or layout) is important to the overall impression the manual makes upon the user. A single page of documentation cluttered with finely printed information, for example, does not provide ready access to vital pieces of information. Larger type sizes, boldface titles, and contrasting print colors reduce visual fatigue. Generally, better use of "white" space (i.e., unprinted page area) assists users in more readily locating specific items. Consider the following page design criteria [26]:

1. Page size
2. Paper weight
3. Type specifications (i.e., type faces and sizes)
4. Print characteristics (i.e., number and contrast of colors)

Results of the documentation review process reflect the users' dissatisfaction with the congestion of instructional items. However, the users did not recommend the use of white space by limiting, for example, the number of instructional lines per section. The experimental design did not include provisions for training the users to identify page design deficiencies.

5.5 Personal Evaluation vs. New Literature Critique

This section summarizes the advantages and disadvantages associated with the two different evaluations of the experimental design. The comparison of the personal evaluation versus the new literature critique is a fundamental issue regarding documentation in general. Documentation writers must continue to balance personal writing techniques and widely accepted documentation standards.

The research experiment tested the acceptability of two common word processing manuals. It demonstrated the importance of incorporating user comments into the documentation review process. The personal evaluation of the experimental design recommended procedural guidelines which include improved data collection techniques. One principal disadvantage of an evaluation based on simple logic is its failure to statistically support each recommended guideline. The

novelty of the experimental design may, however, partially justify this problem.

The new literature critique highlighted the need for a more structured set of criteria for reviewing documentation. Specifically, two reference standards provided a basis for recognizing the limitations of the experiment's review process. These standards are the result of software suppliers listening to market feedback regarding their user manuals. This feedback ensures, therefore, at least some statistical validity. One principal disadvantage associated with this critique, however, is its failure to differentiate between a program's potential users.

A major contention of this research is the need for more personalized documentation. Documentation based on specific reference standards and writing techniques is important; however, the role of the documentation writer must include a complete understanding of the program's audience. The following sections briefly describe the primary advantages related to the two formats for evaluating the documentation review process.

(Note: Software suppliers today are developing programs and documentation that recognize differences in user needs and capabilities. Multimate International, for example, markets two different versions of its word processing

program, Just Write and Multi Mate Executive. The former package offers a lower-end program with fewer features and a simplified user manual. The latter package provides more sophisticated users with advanced word processing applications and additional documentation [26].)

5.5.1 Advantages of Personal Evaluation

The personal evaluation of the experiment focused on direct user input for developing effective user manuals. This evaluation recognized the value of effective questionnaires in characterizing a given user population. Additionally, this format for evaluating the experimental design does the following:

1. Emphasizes the critical role of users in reviewing documentation.
2. Recognizes the need for a larger sample group to improve the experiment's statistical significance.
3. Promotes greater cooperation between users and the documentation writer.
4. Recommends real-world settings to review documentation.
5. Introduces the concept of a documentation prototype for incorporating users' recommended modifications.

6. Develops the use of a simple module framework to assist users in completing instructional summaries.
7. Establishes a basis for classifying and categorizing user attributes and documentation topics, respectively.
8. Reduces the dependence upon structured documentation standards for reviewing user manuals.
9. Examines the use of non-technical writers in reviewing and modifying user manuals.

5.5.2 Advantages of New Literature Critique

The two documentation reference standards commonly used (i.e., physical layout and instructional components) serve as checklists for writing comprehensive user manuals. These standards are primarily the result of user reactions to popular programs. Specifically, information gathered by customer service representatives provided a basis for generating practical documentation criteria. Some software suppliers have incorporated user comments and suggestions into earlier versions of their manuals. As a result, current versions demonstrate a more user-interactive format.

The development of the two reference standards does the following:

1. Provides structured criteria for users in assessing the manual's overall readability and the effectiveness of its instructions.
2. Categorizes major documentation topics required by most user manuals.
3. Establishes a basic linearity for the documentation modules needed.
4. Highlights the use of the standards in developing a documentation prototype.
5. Indicates documentation criteria of common interest to users based on feedback regarding manuals in the marketplace.
6. Promotes a checklist approach for users to follow in preparing detailed recommendations.
7. Reinforces the role of a technical writer in researching the average reader's attention span and level of comprehension.
8. Recommends a more structured format for user log-book entries.
9. Increases the significance of matrix summaries by identifying specific criteria requested by users.
10. Establishes an intricate network of decision points (i.e., alternative criteria) for users to follow in determining individual preferences.

11. Provides greater statistical validity for modifications made to the documentation reviewed, based on the frequency of user requests for given criteria.

5.6 Full-scale Experimentation

Based on the results of this research, full-scale experimentation can contribute significantly to a better understanding of the documentation process. There are certain costs and problems, however, associated with this suggested research. Software suppliers must decide if long-term user satisfaction (based on effective documentation) justifies short-term capital expenditures for experimental research.

The following sections discuss full-scale experimentation costs and problems based primarily on the research experiment's findings. Additionally, user feedback mechanisms that aided the development of the new reference standards support this discussion.

5.6.1 Experimentation Problems

The research experiment encountered several critical problems related primarily to its design. Section 5.2 recommends procedural guidelines to specifically improve data collection techniques and the overall experimental design.

These guidelines do not, however, address the following real-world limitations imposed on full-scale experimentation:

1. Selection of a statistically valid user sample group involves extensive market research.
2. Qualified user consultants may not have the available time required to conduct a comprehensive documentation review.
3. A real-world setting for the experiment may interfere dramatically with normal office production.
4. The documentation prototype or preliminary version of a manual requires user consultants to demonstrate operating proficiency despite missing instructions.
5. Time constraints may dictate assigning particular manual sections to each user consultant, thereby eliminating a start-to-finish appraisal of the documentation.
6. Collecting spontaneous user reactions to the documentation requires time-consuming techniques for compiling large quantities of data.
7. A more precise and structured format for recording user responses simplifies the generation of matrix summaries; however, it may encourage user

consultants to discard information valuable to the software supplier.

8. User consultants may not have the capacity to discuss at length the documentation's nonverbal messages (e.g., quantity of white space used, number of lines per major topic, and overall page layout).
9. Data obtained from several potential user populations may provide a basis for developing a user classification system; however, this system requires marketing diverse software packages to these populations.
10. User feedback mechanisms (e.g., customer service calls) require training company representatives to effectively record user comments.
11. Incorporating and testing recommended modifications may entail reiterations of the drafting, review, and testing cycle.

5.6.2 Experimentation Costs

The principal costs associated with full-scale experimentation reflect the need for paid user consultants and effective methods of managing large quantities of user data. A software supplier must organize a documentation project

team consisting of contract writers or in-house writers who are responsible for developing specific documentation modules. Additionally, a software supplier will realize costs related to incorporating and testing recommended modifications to an initial prototype or preliminary version.

Full-scale experimentation requires more elaborate systems for managing user data. Microcomputer disk storage of a user classification system may provide documentation writers with a concise and accurate profile of each user population. Similarly, basic module frameworks stored on a disk can accommodate recommended changes more readily. The costs associated with compiling and analyzing data collected from user logbooks represent a significant initial investment. A software supplier can use, however, a well-designed user classification system for developing future manuals.

As stated earlier, the introduction of new reference standards has improved software suppliers' understanding of specific documentation topics. Full-scale experimentation can incorporate these standards into a comprehensive data collection process. The additional costs of this experimentation are the result of the following:

1. Additional project team members
2. Increased manhour estimates
3. Paid user consultants (i.e., non-technical writers)

4. Preliminary questionnaire design
5. Relocation of experimentation setting (e.g., real-world office)
6. Disk storage of user classification system
7. Disk storage of basic module frameworks
8. Incorporation of recommended modifications
9. User testing of changes
10. Final questionnaire design

5.7 Conclusion

The research experiment's results and findings identify the focal points of the documentation review process. Specifically, they establish the significance of isolating a particular user population and reviewing documentation with the assistance of sample users from this population. This cooperative effort promotes the development of user manuals that are logical to a given user population.

Each result or finding represents a decision point for the documentation writer. Minor procedural variations are possible without disrupting the overall data collection process. This flexibility allows the documentation writer to increase the data sources in response to program complexity and user limitations.

As stated earlier, a microcomputer software revolution has taken place during the two-year research period. New documentation reference standards are now available to assist documentation writers in developing quality user manuals. These reference standards are the direct result of software suppliers responding to user feedback.

Section 5.6 recommended full-scale experimentation to gain a better understanding of the documentation process. Despite costs and problems associated with this recommended future research, full-scale experimentation can contribute significantly to the quantity of available user data. Ideally, a large group of sample users would review a documentation prototype and later versions that have incorporated recommended changes.

Future research may, for example, investigate the role of university students in testing user manuals. Students enrolled in a given course may have the opportunity to complete course requirements on a self-paced basis using only a specific software program and its associated documentation. Comparative studies of self-paced students and students learning under the guidance of instructors may reveal some inherent limitations or hidden advantages of user manuals in the learning process.

The university setting may act as a proving ground for software and documentation development using professors as paid consultants. Professors in a given specialty may provide software suppliers with specific information critical to the development of feasible program applications. Specifically, professors may contribute significantly to the writing of quality user manuals by recognizing first-hand the capabilities and limitations of their students.

Finally, the recommended procedural guidelines (Section 5.3) may stimulate future research regarding the storage and retrieval of large quantities of user profile data. The rapid transfer of this information between a software supplier's documentation writers may make the independent modules developed by each more compatible. This coordinated effort in developing instruction modules may decrease significantly the need for generating several drafts in developing an effective user manual.

BIBLIOGRAPHY

Books

1. Alwin, Duane F. (edited by). (Sage Contemporary Social Science - Issues 46) Survey Design and Analysis - Current Issues Beverly Hills: Sage Publications, 1978.
2. Belson, William A. The Design and Understanding of Survey Questions. Hants, England: Gower Publishing Co., Ltd., 1981.
3. Hunkins, Francis P. Questioning Strategies and Techniques. Boston: Allyn and Bacon, Inc., 1972.
4. Kish, Leslie. Survey Sampling. New York: John Wiley & Sons, Inc., 1965.
5. Mullins, Carolyn J. The Complete Manuscript Preparation Style Guide. Prentice Hall, 1982.
6. Parten, Ph.D., Mildred. Surveys, Polls, and Samples: Practical Procedures. New York: Cooper Square Publishers, Inc., 1966.
7. Warwick, Ph.D., Donald P., and Charles A. Lininger, Ph.D. The Sample Survey: Theory and Practice. New York: McGraw-Hill Book Company, 1975.
8. Weisberg, Herbert F., and Bruce D. Bowen. An Introduction to Survey Research and Data Analysis. San Francisco: W. H. Freeman and Company, 1977.

Articles

9. Collopy, David. "Software Documentation: Reading a Package by Its Cover." Personal Computing, February 1983.
10. Horn, Robert D. "Information Mapping." Datamation, January 1975.
11. Houghton-Alico, Doann. "Creative Thinking Aids Documentation Focus." Software News, March 1984.
12. Houghton-Alico, Doann. "Developing a Documentation Methodology." Software News, April 1984.

13. Mace, Scott. "SIGDOC Members Work to Improve Documentation." In Focus, August 30, 1982.
14. Margolis, Neal. "Do Bend, Fold or Mutilate - Your Manual." Infoworld, Volume 5, Number 15.
15. Margolis, Neal. "What You Can Do When All Else Fails - Call for Help." Infoworld, Volume 5, Number 16.
16. Milton, John. "Documentation." Infoworld, August 30, 1982.
17. Mullins, Carolyn J. "Do you hear what I say? Do you read what I write." Cause/Effect, September 1983.
18. Rosenberg, Scott. "Documentation Comes of Age - New Approaches to Making Manuals User - Friendly." Popular Computing, March 1984.
19. Shea, Tom. "Technical Writers Talk About Trials and Tribulations." In Focus, August 30, 1982.
20. Snyders, Jan. "User Manuals that Make Sense." Computer Decisions.
21. Swaine, Michael. "Documentation of Tomorrow: 'Malleable Manuals'." In Focus, August 30, 1982.
22. Swaine, Michael. "Reviewing Documentation." In Focus, August 30, 1982.
23. Tucker, Sheila. "Why Computer Manuals Drive Everyone Stark Raving Mad!" PC Magazine, August 1983.
24. Weiss, Edmond H. "Usability: Toward a Science of User Documentation." Computer World, January 10, 1983, Vol. XVII, #2.
25. Zilbergeld, Bernie. "Word Perfect 4.0." Infoworld, February 25, 1985, Volume 7, Issue 8.

OTHER

26. Mullins, Carolyn J. Class Notes. English 3720, Technical Writing, Virginia Polytechnic Institute and State University, January, 1984.

APPENDIX A

PROFILE

User A

I have a B.S. degree in Industrial Engineering
(General Option)

Relevant classes include:

EF 1020 - Fortran
IEOR 3460 - Fortran
CHEM 4150 - Electronics for Scientists (8080A Assembler)
CHEM 4151 - "
CHEM 3010 - Microcomputer/Minicomputer Interfacing
IEOR 4150 - Simulation
IEOR 4980 - Microcomputer Applications
*IEOR 5160 - Simulation Applications

Other Qualifications:

- 2 years (approx.) experience w/ microcomputers including:

Programming
Maintenance of software and hardware
Instruction of Independent Study Students
Instruction of Seminars

- Familiar with TRS-80 (Radio Shack) Models I, III,
& II, Apple II+, and IBM PC (minimally)

*not yet completed

PROFILE

User B

Major: IEOR

QCA: Major 3.61

Overall 3.27

Computer Experience:

Besides the 2 computer courses required by all IEOR's (Fortran Intro and Comp Methods), I have taken 3 quarters of independent study on Radio Shack Micro-Computers working on a user-oriented self documented Queueing Program. I am also currently taking Model Simulations (another required IEOR course), self study in a computer language (Cobol), and another quarter of independent study.

PROFILE

User C

Computer Experience:

Introduced to FORTRAN in my Engineering Fundamentals course, then used FORTRAN throughout the IEOR Computational Methods course still using the cards rather than the CMS terminals. I am presently enrolled in IEOR Simulation course in which I will use the CMS terminal, and FORTRAN to solve technical problems. Also, I am presently working under Dr. Agee for an Independent Study in which I have worked on the Radio Shack TRS 80 Microcomputer using the BASIC language.

PROFILE

User D

Computer classes:

- Simulation
- Independent Study - Mini/Micro (TRS 80)
- Fortran IV
- Computational Methods

APPENDIX B



PERSONAL DATA SHEET

Line # _____
 Employer _____

Name: First Middle Initial Last User B		Social Security Number:	Check One: U.S. Citizen <input checked="" type="checkbox"/> Perm. Resident <input type="checkbox"/> Student Visa <input type="checkbox"/>
Present Address (Street, City, State, Zip):		Permanent Address (Street, City, State, Zip):	
Present Phone:		Permanent Phone:	

Type of Work Desired: - Technical sales with an interest in management - Manufacturers Field Representative	Location Restrictions (State "None" or "Preferred"): Date Available for Employment:
---	--

COLLEGE INFORMATION: Name and Location of College(s) Attended	Date (mo/yr)		Degree Expected Or Earned	Completion Date	Field of Study	G.C.A.		Grade Basis
	From	To				(1) Major	(2) Overall	
Virginia Polytechnic Institute and State University Blacksburg, Va.			B.S.		I.E.O.R.	(1) ----- (2) -----		A = 4
						(1) ----- (2) -----		A =
						(1) ----- (2) -----		A =

Honors, Activities, Interests, Organizations and Offices Held, Hobbies:
HONORS:

ACTIVITIES/INTERESTS:

HOBBIES: Music * Golf * Dungeons & Dragons * Painting * Camping

EMPLOYMENT EXPERIENCE (Include Permanent, Cooperative, Intern, Part Time, Summer, Volunteer, Military Service):				
Name and Location of Employer(s)	Job Title and Responsibilities	Hours Per Week	Dates (mo/yr)	
			From	To
	CO-OP STUDENT 1st & 2nd Qtr: Prepared test Plans for a new environmental connector test stand 3rd Qtr: Research on Electrochemical machining (ECM)	40 40		
	Acting in "The Owl and the Pussycat"	20		
	Waiter	20		
	Ass't. Manager: In Charge of Sales and Bookkeeping	50		
	Ass't.: Cleaned the kennel, issued medications, helped in general	30		

Job Related Information, Skills, Chief Accomplishments, Strengths, Interests
COMMUNICATION: Tutored Math and Chemistry in High School and College * Acting since the age of 6 * Debate Team (high school) *

SALES AND SUPERVISION: Manager for sales
COMPUTERS: 5 Qtrs. of Independent Study on Radio Shack, Apple, and IBM microcomputers * Strong programming skills * Senior Design Project on the Implementation and Application of Microcomputers for Small Businesses

University Placement Services

Name: User B

Courses Supporting Your Employment Interests

COURSE NAME	Cr.	Gr.	COURSE NAME	Cr.	Gr.
<u>MAJOR</u>			<u>SUPPORTING</u>		
Independent Study (2 Qtrs.) (microcomputer applications)	3	A,A	Public Speaking	4	A
Computation Methods in I.E.	4	A-	Organizational Communication	3	A
Industrial Economy	3	A	Human Communication (P/F)	3	P
Human Performance in I.E.	4	A-	Advanced Interpersonal Comm.	3	*
Industrial Cost Control	4	A-	Personal Selling	3	*
Manufacturing Processes	3	A-	Sales Management	3	*
Probabilistic Operations	3	A	Introduction to Computer Engineering	3	*
Research Models			Advanced Computational Techniques	3	*
Methods Engineering	4	A-	Personnel Administration	3	A
Engineering Economy	3	B	Language & Logic	3	A-
Deterministic Operations	3	A	Persuasion	3	*
Research Models			Theatre Practicum (4 Qtrs.)	1	A,A,P
Introduction to I.E.O.R.	3	B	A		
Statistical Quality Control	4	*	Public Communication	3	*
Industrial Systems Engineering	3	*	Principles of Economics (2 Qtrs.)	3	A,A-
Industrial Systems Design	4	*	American Government	3	A
Production Planning & Control	3	*	Industrial Automation	3	*
Plant Design & Layout	4	*			
Theory of Organization	3	*	<u>SUPPORTING MAJOR</u>		
Independent Study (3 Qtrs.)	3	*	Statistics for Engineering (3)	3	A-,A.
(microcomputer applications)			A		
			Electrical Theory	5	A-
			Statics of Rigid Bodies	3	A-
			Dynamics of Rigid Bodies	3	B+
			Mechanics of Deformable Solids	5	A-
			Engineering Thermodynamics	5	B

*Will be completed this year

Q.C.A. by Year: 1st 3.14 ; 2nd 3.04 ; 3rd 3.49 ; 4th _____

COMMENTS and/or ADDITIONAL INFORMATION:

User C

Present Address

Permanent Address

EDUCATION B.S. Industrial Engineering and Operations Research,
Virginia Tech, Blacksburg, VA.
Major G.P.A. 2.8/4.0. Overall G.P.A 2.64/4.0.

Computer languages: FORTRAN, BASIC. Also experienced on
TRS-80 microcomputer.

Courses include:

Engineering Economy
Computational Methods
Methods Engineering
Human Performance in
Industrial Systems
Probabilistic Models
Systems Analysis
through Simulation
Industrial Systems
Engineering

Manufacturing Processes
Math Methods
Deterministic Models
Production Planning and
Inventory Control
Industrial Cost Control
Plant Design and Layout
Statistical Quality
Control
Theory of Organization

EXPERIENCE Salesclerk

Engineering Intern

- performed research and made recommendations for capital equipment cost justification project
- developed and revised process specifications
- compiled inventory list of equipment for future data base applications.

Cashier

ACTIVITIES Assistant Social Chairman-

President-

Club.

OTHER DATA Date of Birth: . Single. Willing to relocate.

REFERENCES AVAILABLE ON REQUEST

User D

Present Address:

Permanent Address:

CAREER
OBJECTIVE:

Desire position in Industrial Sales with special interest in computer hardware/software field.

EMPLOYMENT:

Intern,

- Responsible for reviewing and recommending mini/micro-computer software packages for company use.
- Extensive analysis of IBM, Apple, and Radio Shack hardware/software systems.

Pre-professional Intern,

- Designed, implemented, and supervised computerized inventory system for high technology detailed parts.
- Experience with application of software packages utilizing IBM CICS.

Engineering Assistant,

- Responsible for materials procurement and inventory.
- Entailed supervising, accounting and scheduling duties.

Engineering Assistant,

- Duties included the completion of Department of Defense contract for review and revision of naval technical manuals.
- Experience with BASIC and FORTRAN computer languages.

EDUCATION:

B.S., Industrial Engineering and Operations Research,
Virginia Polytechnic Institute and State
University, Blacksburg, Virginia.
Q.C.A.: 2.9/4.0 (major) 2.4/4.0 (overall)

Earned 80% of college expenses through summer and part-time employment.

HONORS AND
ASSOCIATIONS:

Representative,
Chairman,
Vice-President,
Outstanding Service Awards,

APPENDIX C

SOFTWARE REVIEW LOGBOOK

User A

Thursday, February 9, 1984

Flowchart (page iii) makes little or no sense right now. I'm assuming it will all be clear later.

When you type C for Catalog, page 6, gives you list of files on (Drive 1) disk, with corresponding #. Manual tells you to type TUTORIAL <CR>, but typing the number next to TUTORIAL (4) works, too.

Doesn't tell you how to get out of insert mode on same page it tells you how to get in (on TUTORIAL).

<CTRL C> Change

<CTRL D> Delete

<i> Insert

<s> Saves to output named at beginning of file

<CTRL i> Back to editor

Additional Commands, page 9. Should be included in tutorial file. N - (under Additional Commands) - what happens if you haven't saved your file and want to? [else type N?]

Copy Text Block, page 11. Manual doesn't tell you whether to be in insert or edit. No problem if you're doing exactly what the manual says.

Can you put several embedded commands sequentially?
(What about Cr's?) Doesn't say. Trying it now. [YES]

Can't get out of file without saving and leaving RUNOFF
- maybe forgot how.

Macros, page 36. I don't understand how to start
the MACRO option. Still in Screenwriter Initial Menu -
by following instructions, I type <SHIFT> <%> and instead
of a Command Line or anything useful, I get ?REENTER.

If I go into Editor, I get the "What character do
you wish to replace?" prompt (page 37), but when I enter
<CTRL> <0> the computer requests "Input rep. chars. j
CR to end". I press CR and get nothing instead of a request
to "input character string".

Program doesn't work like manual says it does.

I cannot figure out Macros at all, maybe it only works
with the SHIFT KEY modification - it would be nice to have.

Figured it out, but don't want to save it. It requires
saving on Master Disk (or Library Disk - which the manual
talks about in Chapter 5, currently in Chapter 4). Have
to get into Editor to use Macro.

There are several things in Chapter 5 that look
interesting, but not for right now - they'll be used later
when I start writing my paper.

Finished through Chapter 6 today (1:50 - 3:40 p.m.)

Friday, February 10, 1984

Decided to try making a library diskette.

Question from detailed instructions for FIX SECTOR COUNT (or screen instructions). What is a VTOC?

Problem in that you can't run FIX SECTOR COUNT after initializing Drive 2 until you switch control back to Drive 1 and then can't do a 1-normal text disk (fixing sector counts) because it tries to do this to Drive 1 and the disk is write-protected.

Have to put library disk in Drive 1 and then run FIX SECTOR COUNT. If this is in the manual, it isn't obvious.

Finished this - now typing a MACRO (again). Have saved MACRO - can't get rid of it. I think I figured it out - instructions reasonably clear, but ought to take you through step-by-step!!

Monday, February 13, 1984

A rather confusing string of instructions meant to be followed step-by-step rather than remembered, on pages 54-55 concerning the saving of large files. I hope it is put somewhere later in the manual also (like after we've read the manual and are ready to begin typing in something the size of a dissertation.)

Spent about 1 hour, 40 minutes in here and only read through Chapter 6. Slow going, but no problems. Finish up later.

Last 20 minutes - no problems, but getting bored just reading.

Friday, February 17, 1984

Page 63 - I've forgotten what ND is. Going looking for it.

ND turns date printing off (big deal) - page 29.

UL sets an underline character (page 64). The manual has not told us this before and by the end of page 64 still hasn't. Had to go to the reference card to figure this out.

I'm about ready to learn how to delete files.

Never got that far.

SOFTWARE REVIEW LOGBOOK

User B

February 6, 1984

<u>Page</u>	<u>Comments</u>
Cover Page	What is an "Integer Basic Machine"
i	<u>Foreward</u> - Understood
ii	Understood
iii	Editor flowchart confusing - the difference between "choose output file name" and "choose input file name" is not clear. I'll assume it will be made clear, however, on page 5.
iv	Understood
1	Understood
2	Understood
3	Chapter 1 - The Editor
4	Does this mean we need one right now? I assumed we did, and made one up. (Took about 5 minutes) I followed the instructions by hitting 'l' and Return (which I'll abbreviate as R from now on), but I didn't get the results I was supposed to. So, I tried again. I tried it again, but I still didn't get the correct results. It keeps giving me the menu back I don't know what to do!! I tried taking the disk out and starting all over, but I still can't get out of the menu screen. I think there might be something wrong with this disk, so I'm trying to find the original one.

Page

Comments

The original disk won't be available until tomorrow, so I guess I'll wait until then.

February 7, 1984

4

I have the original disk and manual. I will be using the disk, but I'll still be using the copy of Screenwriter (in the blue book) so I can write some notes directly in the book.

I repeated the instructions on pages 1-4, and this time I got the correct prompt. So, here I go again.

5

I didn't see the note about going to Drive 2 until after I made the command. I'll try again.

6

Had to go back once to page 5 to understand exactly what was going on.

7

They now send me to the documentation within the "Tutorial" Program. (Screen)

When they put you in Insert Mode, they don't really say how to continue reading the instructions. I had to assume that you type <CTRL P>.

They don't explain how to move the cursor more than one space at a time.

Just found out that by pressing the <REPT> key, I can move down fast.

I still can't figure out how to move the cursor back (or up) fast.

Note: This whole bit about using Control <CTRL> and <A> simultaneously is a pain. Awkward.

I missed some instructions because they were off the screen - when I advanced to the next page, I lost what I was

Page

Comments

doing.

By the way, they keep talking about the "SHIFT KEY" modification - this should have been referred to in the beginning of the written manual.

Again, it's a pain using the <ESC> key, but the shift modification probably would have helped.

Again, problem with cut-off instructions.

Have no idea what <SHIFT PCTRL D> means.

In change mode, the use of the space is very confusing. It doesn't work for spaces.

They should explain what happens when you get rid of <CR>, (i.e., ignores the command to go to next page).

7

The summary in Chapter 1 should have been reviewed before going to Tutorial and referred to in case of confusion while running Tutorial.

8

Understood

9

Understood

10

They talk of slots, drive types, drive #'s without explaining what they are.

In going back to review "Output File" and (another file). Why would you want to save it on "Another File"?

11

Realized I was out of Tutorial when I shouldn't have been. They should have said "Stay in Tutorial".

I didn't know where I was supposed to be, so I saved my file.

Finally figured it out:

1. They failed to explain where you are to be doing this.

Page

Comments

2. If you are in input or change mode, you must use <CTRL G>, not just 'G'.
- 12 I still am confused on what command lever is.
- 13 Again, confused on command mode. Nothing was found when I tried to find "Tutorial".
- Stopped - Ready for Chapter 2.

February 13, 1984 - Chapter 2 - Runoff

- 15 Understood
- 16 Understood
- 17 They never explain what the "Text File" is.
- 18 When I selected 6, they didn't tell me that I had to press R again to reload the machine - they also didn't tell me that it would be "Output File"?
- What they have done is assume that you know what to do at steps they have committed, like telling the computer 'Yes' its OK to create the new file 'sample'. While this will become second hand soon, I think it's a little too soon to assume the user knows what to do.
- One thing that is wrong is that they take you through a series of things to do, but they don't explain why you are doing it, which can make it difficult to understand and to remember.
- 19 See Manual
- 20 Understood
- 21 See Manual.

<u>Page</u>	<u>Comments</u>
22	See Manual.
23	When I went to test print, nothing happened. Now I'm going to go back and see what went wrong. I tried to go back to Main Menu, but it won't accept the 'Q' command. I couldn't figure out what was wrong - I think my file is empty. I went back to the beginning of Chapter 2 and ran through the process. This time it worked.
24	Understood
<u>Chapter 3</u>	
25	Understood
26	Understood
27	Understood
28-31	Understood
<u>Chapter 4</u>	
33	Understood
34	Understood
35	Would have been nice to know that if you get stuck in "The Twilight Zone", you can enter 'Run' R to get initial menu back. They didn't let you know what to expect. What is an Initialization Parameter?
36	When I pressed 'Y', it didn't accept it. To customize the disk, you must take the write-protective tab off - they

Page

Comments

37 don't tell you this.
I got stuck because I have no idea where I am or how to get out of where I am.
Skip Macros for now.

February 14, 1984

Chapter 5 - Introduction to the Creative Writer

43 Understood
44 Understood
45 What is 'Fix Sector Count'? They don't explain it very well. Why would you want it?
How do you do that? I had to go back.
What they should have is a review that goes through step-by-step how to use the system. A practical example from start to finish - perhaps a flowchart diagram.
46 What does 'BRUN etc' do? Why don't they tell me what is going on?
47 Understood
48 Understood

Chapter 6 - Advanced Editing

49 Understood
50 I took about an hour and went back to practice everything.

<u>Page</u>	<u>Comments</u>
<u>February 15, 1984</u>	
51	Understood
52	What is "Spontaneous" - Bad explanation. When the tell you the command <SHIFT CTRL N> they don't explain where you are supposed to use it. (Command, insert, change, etc.) Same with other ones. (<CTRL N> <SHIFT CTRL C>)
53	Bad explanation of the 'Z' command. What is Packing? When they talk about 'Fix Sector Count', they talk about 'Do Sectors' - but they never explain what it is. I guess they assume you know.
54	Understood
55	Understood
56	Understood
57	Understood
58	Understood
59	What the 'Get Buffer'?
60	The Help function is <u>GREAT!!</u> Why didn't they introduce it at the beginning???
61	Understood

Chapter 7 - Letters and Mailing Labels

63	Understood
64	Understood
65	Understood

Page
February 16, 1984

Comments

- 66 What is the 'EOR' marker? Oh, I see.
Why don't they be consistent?

Are we supposed to get to this sample?
If so, why don't they tell me?
- 67 Understood
- 68 Understood
- 69 Understood
- 70 I think they should have spent a little
more time on this. For instance, they
could have given an example of how to
insert material in the body of the letter.
70. I skipped over the 'Compare' Section
because I really wasn't interested.
It would have taken too much time.
- 71 Skip.
- 72 Understood

Chapter 8 - Reports and Manuscripts

- 73 Understood
- 74 Understood
- 75 Understood
- 76 Not a very good explanation on why we
would want to use it.
- 77 What are they talking about??

I don't understand why we would need
to use this.
- 78 Skimmed.
- 79 Understood
- 80 Understood
- 81 Understood

<u>Page</u>	<u>Comments</u>
<u>Chapter 9 - Formatting</u>	
83	I plan on going through this chapter rather fast since I am running short on time and the material is pretty user-specific. This is nice since they are giving an overview of the material and how to use it.
84	Understood
85	Understood
86	Understood
87	Understood
88	Understood
89	Understood
90	Understood

Chapter 10 - Editing Basic Programs

Again, for this chapter to have any use, I would need to create a basic program and then manipulate it. This would take alot of time that I don't have, so I will just skim the chapter.

Chapter 11 - Editor Commands

97	Understood
98	I think this section would have been nice after Chapter 6. The rest of this chapter and the rest of the text is simply a review. Very helpful, expecially if I plan to use this later.

SOFTWARE REVIEW LOGBOOK

User C

February 7, 1984 (1:00 p.m. - 3:00 p.m.)

My first attempt at Applewriter II.

Page 4 explains how to select menu options, but does not give instructions to do so in the documentation or the program. One of the options was to hit "Return" so I did.

Page 5 refers to additional functions menu for a list of editor options. Since I'd not typed anything to be edited these commands had little meaning. So far the screen had been on with the cursor flashing for fifteen minutes and nothing had been done. I was still waiting for instructions to enter something but was forced to continue reading.

The next section talked about the data line on the top of the screen and how to remove it. It was my first opportunity to try something. I followed the instructions but was not successful after several attempts. At this point the instructions are slightly ambiguous because it does not instruct you to hit "return" after "CTRL Q".

Finally on page 6 capitalization was discussed. So far I'd only seen capital letters. This section had no application for this system. Much later, by chance, I learned how to get both upper and lower case.

By this time I'd thought I was on the wrong track

or wasn't following the documentation correctly. I was suspicious that something was missing.

Page 7, reading on, I learned how to move the cursor. Again, neither the program or documentation instructed me to do anything, but for my own satisfaction I would experiment with each command. For example, when I first tried to move the cursor, nothing happened because (I found out later) there was no text for it to move along. This is the point at which the documentation was breaking down considerably.

At wits end, I entered some more text to edit. I typed a line of letters using the "repeat (rept)" key. To my dismay the cursor locked into the bottom r.h. corner. I skimmed the pages looking for a command to release the cursor. On page 12 I learned about the "help" command. No response from the computer at all. Finally, I typed "ESC ESC ?" for help and returned to the original menu. Tried the load option but it didn't make much sense at the time because I had nothing to load.

In attempt to find a program to load, I skimmed through to page 77. Tried copying a command to load a file but nothing happened. I felt I had skipped over too much.

February 8, 1984 (10:00 a.m.)

Decided to enter some of my own text to practice on, but wanted capital letters. Flipped to page 15 on "capitalizing using the shift key" (found in table of

contents). Reference was made to the "shift key modification". I didn't know what it was but I assumed it hadn't been made. I looked again in the table of contents for a heading for "shift key modification" but did not find one. Went back to the beginning of the chapter because I remembered something about the shift key mentioned. Still no luck. Since I was near page 5 anyway, I tried removing the data line. In this section they suggest that the data line may be annoying; therefore, I decided it was useless and was successful in removing it on the third try. I read some more on "capitalizing using the shift key". The cursor changed as expected but lower case was never displayed on the screen.

Flipped to page 21 to clear the screen and was successful.

February 10, 1984 (10:00 a.m - 11:00 a.m.)

Up to this point there was no direction to the documentation except page numbers so I kept reading. I tried to remove data line again but this time was unsuccessful. Tried adding and deleting words but the results were inconsistent. Referred to the "help" screen on delete/recover text. Then tried saving/loading segments. It was beginning to look hopeful.

NOTE: The caution note could be misinterpreted. From previous experience I know the command "new" will erase the memory. Some could follow the directions and press

"new" just before loading and erase their entire input. It means to clear the memory before creating the file.

February 10, 1984 (1:15 p.m. - 3:15 p.m.)

I had saved some files on the main disk just for practice. The section on deleting files was not a major heading in the table of contents, but I did find it thumbing through the manual (page 29 under "CTRL-O DOS Commands).

Page 25 Reference to another section of the manual. This practice is very inconvenient. Read through page 34.

February 15, 1984 (10:00 a.m. - 11:00 a.m.)

Page 33 described the underlining procedure. Underlining only shows up on the printed output; therefore, I needed to experiment with the printer which has not been mentioned yet. I turned to page 68 (print procedures). I encountered two problems:

1. This section referred to format procedures which I'd not seen yet.
2. The instructions did not explain exactly what keys to press. I didn't have problems with this but many beginners will.
3. It printed out in lower case while only upper case showed on the screen.

February 15, 1984 (3:30 p.m. - 5:30 p.m.) New Video Display

By this time, I was typing in alot of text and editing it using commands I had learned. The most frequently used key is the left arrow to erase mistakes, but the left arrow also saves what the cursor erases. Nowhere does it explain how to erase characters permanently.

Finally the lower case command works and prints out correctly. This could be due to the new screen, because the same commands were used.

I tried to save some programs but I kept getting "DOS: i/o Error". Referred to operator's manual for initializing instructions. Somehow I erased the main disk, following instructions on page 30.

February 16, 1984 (3:00 p.m. - 6:00 p.m.)

Practice inputing and printing out text. Once the paper got messed up in the printer and there was no emergency note in the printing instructions. The first half of my file was printed; therefore, I only needed to print the second half. Backtracked to page 24 on loading segments.

Page 47 defines data line which should have come in the beginning because I tried to delete it and it proved to be very useful.

I skimmed the remainder of the manual and found the same format as the rest of the book. There were no step by step instructions. It never told you to do anything.

The manual forced you to read it in it's entirety and then strike out on your own. This type of manual leads to boredom from only reading and not experimenting, and to frustration from incomplete or ambiguous instructions.

SOFTWARE REVIEW LOGBOOK

User D

Wednesday, February 8, 1984 (12:00 p.m. - 2:00 p.m.)

Familiarize myself with Applewriter system in general. Start going through documentation page-by-page.

Documentation covered: Pages 1-15

NOTE: Much of this time was spent preparing examples and demonstrating the software to myself as I went through it. I felt I needed to do this since the Applewriter software allows for little user interaction.

Thursday, February 9, 1984 (4:00 p.m. - 6:00 p.m.)

I originally was scheduled for the 2:00-4:00 slot but we were unable to find the documentation for Applewriter (we did have the software disk). Continued going through documentation.

Documentation covered: Pages 15-30

NOTE: Much of this time was spent referring back to prior sections for reference.

Examples: capitalization, CTRL and ESC keys.

Saturday, February 11, 1985 (2:00 p.m. - 4:00 p.m.)

Documentation covered: Pages 30-60

NOTE: Spent the majority of the time simply reading the documentation. It was extremely dry and I experimented with printing out paragraphs, saving files, loading file, and using special Applewriter functions.

Wednesday, February 15, 1984 (12:00 p.m. - 2:00 p.m.)

Since I hadn't worked with the software since Saturday, I often had to review material and use the documentation as a reference.

Documentation covered: Pages 61-100

By now I was skimming the material instead of reading it in depth. This is a particularly large downfall of Applewriter. The user feels that he is wasting his time with meaningless jargon and special techniques when it is evident that the user can't even type a simple paper with the correct punctuation and output format.

Thursday, February 16, 1984 (4:00 p.m. - 6:00 p.m.)

NOTE: I was scheduled for the 2:00-4:00 slot, but was unable to come at that time.

Documentation covered: Pages 101-finish

Was not experimenting with various output. Tying to get paragraphs typed out correctly. Changing the output settings to see their effect, etc.

Monday, February 20, 1985 (6:00 p.m. - 8:00 p.m.)

Typed out the comments on page-by-page analysis of the software. Note section titled "Comments on Typed Output".

PAGE-BY-PAGE ANALYSIS

User D

Below is an analysis of the first 40 pages of the Applewriter documentation which consists of the majority of the basic Applewriter functions.

Entered here are only the problem areas encountered in the documentation. Especially well-written material is also noted.

Page 1-3

No real problem. On page 3 you are instructed by computer to enter "Do you have a lowercase display (Y/N)?" Does this mean it is available on the terminal? Do I want it on the word processing output?

Page 4

After reading the function of the CTRL it was clear that I was not using the menu command correctly. For example, you must press CTRL and l in order to load a file to your disk.

Page 6

Under the "capitalizing using the ESC key" it refers to "first boot apple writer." What does this mean? Also, what if you entered "No" to capital/lowercase question

at the beginning of the program? Will these functions still work?

Page 7

Does "but not beyond" really mean "but not prior to"?
Very unclear.

Page 10

Correcting typing errors. Documentation is good as a reference. An example would be helpful.

Page 12

"Help" function requires master diskette. It would be helpful to outline at the beginning what other functions require use of the master diskette.

Page 13

The command (p) do help80 and do help40 are impractical. How can these possibly be remembered unless used constantly?

Page 15

What is a shift-key modification? Following procedure is useless unless you know the answer to the above question.

Page 16-19

Adding files to a disk. Good summary but an example would be helpful. The "delimiters" and "(s)aving a segment of text" sections are poorly written. It is difficult to get an idea of what they do without better documentation or a thorough example.

Page 18

(S)ave: = Command. Why do this? Why not just put (s)ave and old name? Examples would clarify the sometimes subtle differences that abound in the different types of options.

Page 21

"Loading files into memory" section is confusing. Does the (n)ew command erase all old memory? What does "at this position of the cursor" refer to? Very vague.

Page 23-24

Readable. Need examples. "(L)oad: # copying text in memory" is hard to understand. This may be a desirable option on the PC but exactly when is it used?

Page 26

At last, a quick summary of the preceding information. A summary would be helpful at the end of each section in

remembering, constrasting, and referring back to the previous material.

Page 27-30

"Sections 1-6" are readable but again an example is needed for each of the six cases to strengthen skills and ensure appropriate execution of commands.

Page 30

"Nine tab stops already set at every eighth position" and "default positions" are unclear phrases. What is being referred to?

Page 31-32

What exactly is a tab setting? Is it the same as on a typewriter? Why then would you want to save and retrieve tab settings? Why display a return?

Page 33

"Underlining" with this software package is so confusing that you have to refer to the documentation for reference. Why are these procedures so difficult?

Page 34-39

Again, although there are no faults in the documentation

you have no examples to show you exactly what will happen when executing the commands. In addition, you are not able to practice on any documentation without returning to the insert mode and working on text that you have to develop.

Page 40

Summary of (f)ind commands is good as a reference. Again, it would be helpful if every summary section included an example in lieu of a verbal definition.

Page 41

The "special delimiters" section is vague in that they detail what the delimiters are capable of but don't go into enough depth in instructing when, where, why, and how they should be used.

COMMENTS ON TYPED OUTPUT

User D

There are a few things about the output that I think are interesting:

- All the letters are in lowercases. Why? I don't know. Every other session with the Applewriter provided output which was entirely in uppercases. I examined the type settings in the software program to be sure they were set correctly. According to the documentation I executed the capitalization functions correctly. This leads me to believe that there is a big problem with the documentation (poor learning guide) or a mechanical problem with the hardware. Note that I have included an example of typed output from a learning session on February 16, in which all of the output was uppercase (pagey). I tried using the documentation as a reference to correct the problem. It was no help.
- Page #'s and page length are not what was desired. I attempted to start a new page with page # at the top of a new sheet of paper. I followed the documentation exactly. The result? It is evident that I didn't get what I wanted.

In short, I have spent 12 hours on the Applewriter

reading and reviewing the documentation. I was unable to produce an output that was desired. Either I didn't read the documentation well enough or it does a poor job of explaining certain features.

Basically, the documentation is good only as a reference - as a teaching guide it leaves a lot to be desired. I have prepared a flowchart and some comments on what type of documentation I would like to see. Read on.

Note: Although I read the remainder of the software documentation, I made no specific comments regarding its quality after those comments made above up to page 41. The reason I chose not to detail any problem areas was that I was becoming increasingly repetitive. In essence, the same problems occur through the documentation from beginning to end. They are as follows:

- a. Use of examples. Examples are seldom used and when they are it is still vague as to the function of that particular software section.
- b. Layout. The documentation is often difficult to follow and seems to lack a logical flow (see comments with flowchart). A user searching for a specific section or key command is often frustrated by the "jargon" associated with the Applewriter.
- c. Interaction. A good software package should keep the users attention throughout the learning process.

With Applewriter the documentation often reads like an encyclopedia and has no interactive sections which would help in learning the Applewriter functions.

- d. Time. A minimum amount of time should be spent in learning a software package so that productive work can begin. With Applewriter, the documentation is time consuming to digest and often requires the user to refer back to previous sections for specific commands as key words.
- e. Troubleshooting. What if you have a problem with the Applewriter commands? What will happen to your current program? Applewriter seldom advises the user on a sequence of commands to follow, their use and their outcome. In short, the user is left to guess certain instances.

APPENDIX D

RECOMMENDATIONS

User A

The manual is set up so that in order to use all commands effectively, you must go through the entire program in one sitting and have a perfect memory.

Given the choice, I would like to see several things that are not easy to locate placed at the beginning of the manual (or at least at the front of each section):

1. A summary of the commands used in this order:

Editor commands

Filing commands

Text commands

Print (runoff) commands

2. A somewhat more forced tutorial. The tutorial included in this package is cute, but doesn't demonstrate many of the commands. (I spent a great deal of time simply typing in text).
3. The examples they go through are more interesting to someone interested in spending the rest of their lives doing word processing. A rearrangement of the examples (or at least a summary of the available examples and where they start) would be more useful.
4. A unified list of the contents should be at the

beginning. As is, the look is divided into sections and each section has a table of contents which, therefore, can't be found easily.

5. Every example (or at least the first several) should tell you what every command does. I found myself several times stopping at a command, flipping backwards to review, returning to the example, only to be flipping again for an explanation of the next line.

I found only one near-fatal problem with the manual, the example for using the macros. No where in that section is there a hint to tell you to return to the editor before trying to save a macro. It took disgust with the program and a good bit of luck to figure that one out.

RECOMMENDATIONS

User B

As you will see by reading my notes and following through the manual, most of my objections were directed at poor writing, misleading directions and ambiguous statements. I think most of these problems could be solved by letting an English Major who knows the computer well write the document, not the other way around.

There were, however, some structural problems with the documentation. Most of those were caused by not reviewing enough, especially at the beginning there is a point I should make here. Since I was a little pushed for time, I might have gone through some of the chapters faster with not much review and practice on my part, which of course is essential in learning anything that is somewhat complex. However, there are still a few recommendations I could make.

Figure 1 shows the process that I took in going through this manual. If you compare it to the sequence of chapters in the manual, you will see that they are identical. There are a few alterations I would make (see Figure 2).

First of all, I would have had an overview of the entire system, how edit and runoff interact, and what might be a typical sequence of events in an office situation. This initial review wouldn't have to go into details, but just enough information to give the user a feel for the

FLOWCHART OF SCREENWRITER II SOFTWARE

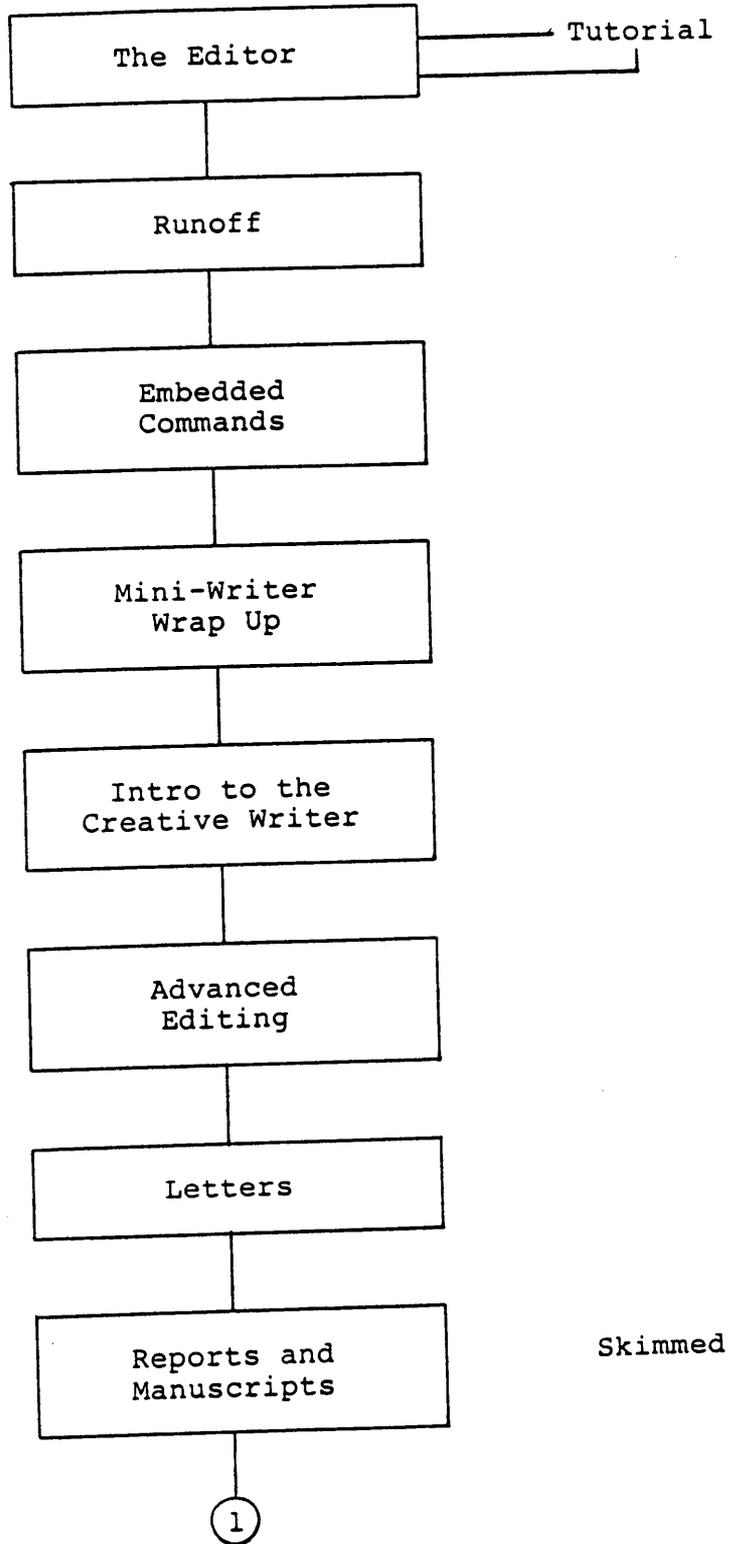
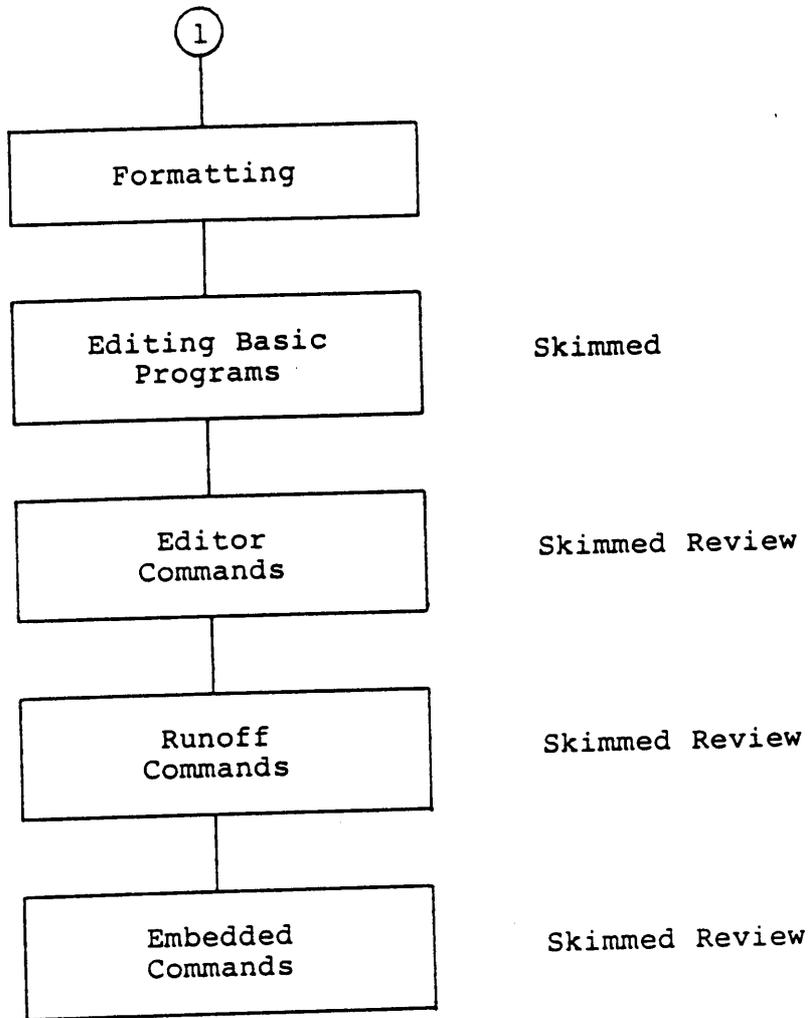


Figure 1

(Continued Next Page)

FLOWCHART OF SCREENWRITER II SOFTWARE
Page Two



When I say "skimmed", it means I spent very little or no time actually practicing the information on the computer.

overall system and how it works. This was a major frustration throughout the first half of the manual.

I also would have referred the reader to chapters 11, 12, and 13 very early since these chapters are basic review sections and would have been handy to flip to throughout the manual.

I would have made the letters and reports and manuscripts chapters optional (so the reader can learn them when they needed them).

RECOMMENDATIONS

User C

1. I would like to see step by step instructions. For example, in the beginning of the manual it should instruct the user to type in a given page of text and as the manual describes different functions have the user perform them.
2. The format functions come late in the manual and these functions are the basis for word processing. I think they should be seen earlier in the program.
3. It would have been helpful to have instructions on how the cursor worked. In other words, the manual should have explained that all capital letters would appear on the screen but lower case would be printed unless a special key was used. After a long period of trial and error and a lot of frustration, I realized how the upper/lower case was obtained.
4. Often the documentation would describe a function of the package, and ten pages later give the information which would enable you to use the function. For example, the manual describes the underlining function on page 33 and the print commands are not discussed until page 68.

5. I think the manual should mention common user errors as they might occur.
6. Aspects of the computer, such as the data line, should be explained earlier in the documentation. The first time it mentions the data line, you are told how to delete it if it distracts you. Much later in the manual the need for the data line is realized.
7. Again, I must emphasize the step by step instructions for the inexperienced user. When the manual describes a function, it should very explicitly tell you which keys to punch and then tell you what will be seen on the screen.
8. My overall impression of the documentation was unfavorable. If I were studying the package for immediate use (like a secretary who needs to get letters out by the next day) I would not have spent more than a few hours and then given up or called the company who made the package. As it turned out, I forced myself to continue reading until I could do at least something on the word processor. Even then I did not absorb everything and I do not feel proficient at word processing.
9. I never found a command which told how to stop the printer in progress. I tried turning it off

but it would continue where it left off when turned back on. This is a crucial command since you frequently realize that the margins were set incorrectly or the heading was forgotten as soon as you see the first printed page. When I was confronted with this problem, I referred to the section on "Loading Segments". I followed the instructions carefully, but it did not work.

RECOMMENDATIONS

User D

Additional Comments - Revising the Applewriter Software

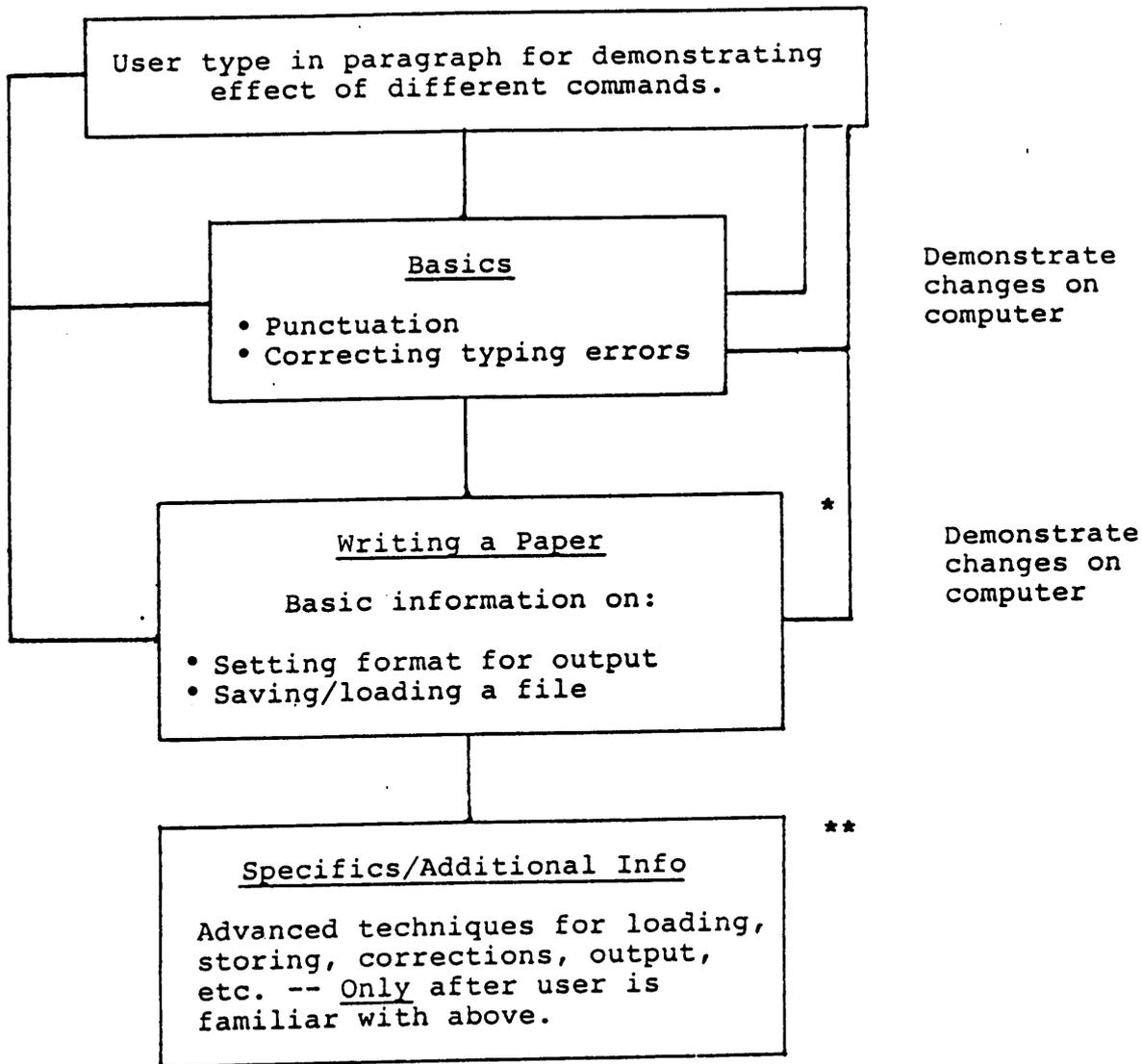
The Applewriter could stand a lot of improvement in the documentation area. Good documentation should lead the user through the basics of the material with a minimum of tedious memorizing and a minimum of time. As it stands, the Applewriter documentation also does not cover basic concepts well, requires alot of review and reference to former material and takes alot of time. I have spent 12 hours on the software and do not feel that I have a strong grasp of the basics. I could type a paper but it would take awhile for me to become completely familiar with Applewriter's functions and overcome any difficulties.

What, then, should good software do? A better plan for software documentation, in my opinion, would be one that is completely user interactive. For example, at the beginning I would have the user type in a paragraph or long phrase. Then, as the user goes through the material I would have him execute the commands that he is reading about in the documentation on portions of the paragraph on the screen. I would detail step-by-step what changes should be occuring visually on the screen or when the material is typed out. Thus, the user's attention would be kept but more importantly the user's knowledge of the material would be strengthened. I found as I went through

the documentation that I was reading about things that were very unclear since I had no example - either through computer interactive example or a book example.

In short, I would have a documentation that was user interactive, included step-by-step processes, noted what each command would do, and provided you with the basics for performing a given task. Then, as you became more comfortable you could refer to the documentation for better techniques or special functions.

REVISED FLOWCHART OF APPLEWRITER II SOFTWARE



* After learning the basics I feel the user should be able to perform certain functions -- type and get output for a letter, save it on a disk, and retrieve it.

** I would introduce more complex commands only after above is complete. With current Applewriter, however, you are told about all of the storing, loading, correction, and special techniques initially. It is too much to consume and the basics get lost.

Figure 3

APPENDIX E

USER/COMPUTER COURSE SUMMARY				
USER COURSE	A	B	C	D
Advanced Computational Techniques		X		
COBOL (Independent Study)		X		
Computational Methods in Industrial Engineering		X	X	X
Electronics for Scientists (8080A Assembler)	X			
FORTRAN (Engineering Fundamentals)	X	X	X	X
Introduction to Computer Engineering	X			
Mini/Microcomputer Applications (Independent Study)	X	X	X	X
Mini/Microcomputer Interfacing	X			
Simulation	X	X	X	X
Simulation Applications	X		X	

USER / ENGINEERING SUPPORT COURSE SUMMARY				
USER COURSE	A	B	C	D
Deterministic Operations Research Models	X	X	X	
Dynamics of Rigid Bodies		X		
Electrical Theory		X		
Engineering Economy	X	X	X	X
Engineering Thermodynamics		X	X	X
Human Performance in Industrial Systems		X	X	
Introduction to Industrial Engineering and Operations Research	X	X	X	X
Industrial Cost Control		X	X	
Industrial Economy		X		X
Industrial Systems Design		X		
Industrial Systems Engineering	X	X	X	
Manufacturing Processes	X	X	X	X

USER / ENGINEERING SUPPORT COURSE SUMMARY				
USER / COURSE	A	B	C	D
Mechanics of Deformable Solids		X	X	X
Methods Engineering		X	X	X
Plant Design and Layout	X	X	X	
Probabilistic Operations Research Models	X	X	X	
Production Planning and Inventory Control	X	X	X	X
Statistical Quality Control		X	X	
Statistics for Engineering	X	X	X	X
Statistics of Rigid Bodies		X		
Theory of Organization		X	X	

USER / DOCUMENTATION REQUIREMENTS SUMMARY				
USER REQUIREMENT	A	B	C	D
Command Summary (for each section)	X	X	X	X
Structured Tutorial	X		X	X
Summary of Examples (including text locations and contents)	X			X
General Table of Contents	X	X	X	
Specific Table of Contents (at beginning of each section)	X	X		
Detailed Examples of Commands	X			X
Macros (with "saving" features)	X			
Program Overview		X	X	X
Step-by-step Instructions (i.e., "cookbook style)	X		X	X
Function/Command Sequence Modules		X	X	X
Common Error Messages (including precautions)		X	X	X

USER / DOCUMENTATION REQUIREMENTS SUMMARY				
USER REQUIREMENT	A	B	C	D
Screen Display Illustrations	X	X	X	X
Basic Instructions (e.g., for simple letter generation)	X	X	X	X

USER / ADVANCED APPLICATIONS REQUIREMENTS SUMMARY				
USER REQUIREMENT	A	B	C	D
Special Letters, Reports, and Manuscripts Information (in optional chapters)		X		X
24-hour Toll-free Vendor Assistance			X	
Command Variables (e.g., inputting format changes during print function)			X	X
Step-by-step Office Automation Instructions	X	X	X	X
"Quick-reference" Modules (for key commands only)	X	X	X	X

**The two page vita has been
removed from the scanned
document. Page 1 of 2**

**The two page vita has been
removed from the scanned
document. Page 2 of 2**