DOCUMENT RESUME

ED 272 189 IR 012 315

AUTHOR Gray, Peter J.
TITLE Microcomputers: Software Evaluation. Evaluation Guides. Guide Number 17.
INSTITUTION Northwest Regional Educational Lab., Portland, Oreg.
SPONS AGENCY National Inst. of Education (ED), Washington, DC.
PUB DATE [84]
CONTRACT 400-80-0105
NOTE 26p.; A product of the Research on Evaluation Program.
PUB TYPE Guides - Non-Classroom Use (055) -- Tests/Evaluation Instruments (160)

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS *Computer Literacy; *Computer Software; *Evaluation Criteria; *Evaluation Methods; *Microcomputers; Needs Assessment; Services; Specifications; Task Analysis; Technological Advancement
IDENTIFIERS *Software Evaluation

ABSTRACT

This guide discusses three critical steps in selecting microcomputer software and hardware: setting the context, software evaluation, and managing microcomputer use. Specific topics addressed include: (1) conducting an informal task analysis to determine how the potential user's time is spent; (2) identifying tasks amenable to computerization and matching them with existing programs; (3) hardware options; (4) screening of software; (5) sources for software evaluations; (6) criteria for software evaluation, e.g., ease-of-use (set-up, installation, initial learning, and use after initial learning), program performance (speed, error handling, and versatility), and support (printed documentation, on-screen documentation, and supplier/manufacturer); (7) software selection; and (8) services, training, and facilities necessary to manage computer use. Appendices include a software evaluation summary form and a software evaluation worksheet, which provide suggestions for rating software on the specific components of the major evaluation criteria specified in the guide. Three references are listed. (JB)

EVALUATION GUIDES

Guide Number 17

MICROCOMPUTERS: SOFTWARE EVALUATION

Peter J. Gray

Three critical steps in selecting microcomputer software and hardware are discussed, including:

- Setting the Context
- Software Evaluation
- Managi. g Microcomputer Use
- Software Evaluation Summary Form
- Software Evaluation Worksheet

Research on Evaluation Program Nick L. Smith, Director

Northwest Regional Educational Laboratory
300 S.W. Sixth Avenue • Portland, Oregon 97204 • Telephone (503) 248-6800

Other guides in this series focus on the potential benefits of using microcomputers in evaluation. For example, the features of word processing programs are described that can faciliate the development of proposals and of evaluation project reports. Statistical programs are shown to bring much of the power of main frame computer programs to individual researchers at a fraction of the cost, and in a very convenient way. Implicit in those guides are criteria for evaluating specific types of software since their desired features were described.

However, it is all too easy to get overwhelmed by these seemingly irresistable features. In order to avoid being overwhelmed, selection should be based on a clear understanding of the potential benefits and problems of microcomputer use. Therefore, in this guide three topics are discussed. One is the context of computer use. The second is the evaluation and selection of appropriate software and hardware. And, the third is the management of microcomputer resources.

The work from which this guide is adapted, A Guide to the Evaluation of Microcomputer-Based Administrative Software, is being jointly developed by the Research on Evaluation Program and the Computer Technology Program at the Northwest Regional Educational Laboratory.[1] Since the Guide is still in development, any suggestions for its improvement as it appears in this guide would be greatly appreciated.

SETTING THE CONTEXT

There is a lot of pressure on all of us as individuals and as part of organizations to join the microcomputer revolution. Advertisements loudly proclaim software and hardware capabilities and colleagues brag about their latest purchase and what it can do. However, you may not feel comfortable with nor especially attracted to microcomputers. The fact is that many people are simply not interested in microcomputers because they do not see how they could benefit from using them. Others shy away from microcomputers because of the apparent complexity of integrating them into their everyday lives. Some people have a fear of being replaced by a computer. This may in part be stimulated by a few of the microcomputer advertising campaigns.

[1] Thanks to Phil Griswold, Don Holznagel, Rex Hagans, Bob Rath, and Mark Shermis for their comments and suggestions regarding earlier drafts of this guide.

Such "people" concerns are important when we consider purchasing a microcomputer. They are also important when microcomputers are about to be introduced into an organization. The basis cf many of these concerns is the fact that microcomputers represent a new and rapidly evolving technology that is at times bewildering. Seeing how micrccomputers can help accomplish daily tasks is an important ingredient in being able to make a reasoned decision about joining the microcomputer revolution. Therefore, before you consider evaluating specific software programs, you should think about why and how you could use a microcomputer in your professional life.

Software

A question often asked by people just learning about microcomputers is, "What can a computer do?" This question is best answered by looking at what software can do, since it is the software programs that direct the operation of the microcomputer. That is why many of the microcomputer guides in this series focus on software that can make a computer do things that are useful in evaluation research.

A good way to understand the potential benefits of using microcomputers is to become familiar with their capabilities in relation to how your time is typically spent. Therefore, you might start your "evaluation" of microcomputer use with an informal task analysis. It is important to be fairly specific in this regard. While it is tempting to stop at such big categories as manage people, make decisions, or plan and coordinate evaluation research projects, computers are not yet sufficiently intelligent to give you <u>direct</u> assistance in accomplishing these larger tasks. Rather, they are tools that can help with such tasks as: maintain personnel files so that you can manage people more efficiently; store, manipulate and display data to help you make decisions; and provide current and projected budgets to help you plan and coordinate projects.

Identifying such tasks can be done as part of a simple task analysis procedure whereby you sort the things you do into several categories. You may be satisfied, for example, with the way your manual filing system is set up to keep track of study participants. Therefore, this task is not a likely candidate for computerization at present. But you may be dissatisfied with the budgeting and accounting system. However, if this system is tied into a main-frame computer system, it might not make sense to develop a parellel system on a microcomputer.

Another example is a situation in which a system is needed that can help develop written reports and oral presentations, as well as do the statistical analyses which form the basis for these reports. If this need is not currently being met very well, and it is something that you or others have shown some enthusiam for in terms of using microcomputers, then it should be considered as a likely candidate for "microcomputerization".

The tasks amenable to computerization may cluster around particular roles such as office/clerical (repeated typing of proposal drafts), statistician (flexible and varied interactive manipulation of data), consultant (development of transparencies and handouts summarizing plans or results), research specialist (instrument design and data collection), project director (budget tracking and projections and ongoing reporting). Involvement in the task analysis process of the people who occupy these rules will help them become clearer about both the alternatives to computer use and the alternative ways computers can be used. It will also become a mechanism for providing information about microcomputers and will open lines of communication about them so that people's anxieties and concerns can be aired.

As a result of the task analysis, you may have tasks sorted into three groups: (1) tasks that you are satisfied with and for which there is no immediate need for change; (2) tasks that are a concern, but for which it is not currently appropriate to use microcomputers; (3) tasks that are both in need of change and which would potentially benefit from microcomputerization.

The next step toward improving your understanding of the microcomputer's potential usefulness is to identify the types of programs that can help improve efficiency and effectiveness regarding those tasks identified as high priority through the process just described. By using a simple matrix like the one shown in Figure 1 you can begin to match tasks to be improved (along the left hand side) with types of software (along the top).

You may want to refer to the other guides in this series for background information on specific types of software. Other resources include Talley (1983), Spuck and Atkinson (1983), and Huntington (1983). In addition, such magazines as BYTE, Popular Computing, and Personal Computing frequently have feature articles which explain the characteristics and uses of different types of programs (e.g., Special Report on Information Management in Popular Computing, June 1984).

Figure 1
Tasks/Software Matrix

	Word Processing	Calc/ Statistics	Data Base Management	Communications & Networking	Graphics
Proposal & Report Writing					
Oral Presentations					
File Maintenance					
Statistical Analysis					
Budget & Accounting					

Hardware

Microcomputers vary in their speed of processing, internal and external storage, printing and display capabilities, and other features. However, the importance of these features depends on the software program chosen, that is, whether a particular program requires, or can even take advantage of, a specific feature.

Figure 2 is a list of hardware specifications. Included are the range of specifications regarding the basic microcomputer, such as random access memory (RAM), operating system(s), and the availability of co-processors. There are also various types of peripheral devices. Input devices include keyboard, mouse, touch screen, numeric key pad, and touch pad. Output devices include the monitor (monochrome, color) and printer (dot matrix, letter quality, ink jet) and their ability to handle graphics. Auxiliary storage covers flexible disk, mini-disk, and hard disk drives with varying capacities. Finally, communication capabilities include telecommunication via modem and local area networks.

While it is exciting to think about the hardware with all of its bells and whistles, in fact, it is more appropriate to consider software capabilities first, and then buy the hardware which allows you to get the most from the software you select. Of course, it is possible that because of prior purchases, or organizational policies, a particular brand of hardware is required. If this is the case, your choice of software will be somewhat limited, but there are usually many competing titles

Figure 2
Hardware Specifications

Microcomputer (brand, model): ______________________________

Random Access Memory (RAM):

____ 64k ____ 128kb ____ 256kb ____ 512kb

Auxiliary Storage:

____ 3-1/4" ____ 5" ____ 8" diskettes

____ 10M ____ 10M hard disk

Operating Systems(s):

____ Apple DOS ____ PRO-DOS ____ CP/M (e.g., 86)

____ MS/PC-DOS ____ TRS-DOS ____ UCSD p-System

Other Hardware:

____ Numeric Data Processor (e.g., Intel 8087)

____ Printer buffer (spooler) ____ Voice synthesizer

____ Other (specify): ______________________________

Input

____ Standard keyboard (QWERTY) ____ Cursor keys

____ Special function keys ____ Numberic keypad

____ Mouse ____ Touch screen ____ Touch pad

____ Graphics tablet/light pen ____ Mark sense reader

Output:

Monitor: ____ RGB ____ Color (composite) ____ Monochrome

____ B&W ____ Liquid crystal

Printer: ____ Dot matrix ____ Ink jet ____ Character

____ Graphics printer/plotter

Carriage: ____ 8-1/2" ____ 11" ____ 14"

Feed: ____ Friction ____ Pin ____ Single sheet

Communication:

Modem: ____ 300 baud ____ 1200 baud ____ Other

____ Acoustic coupling ____ Direct connect

Networking (system): ______________________________

within a given machine group and software type. In addition, there are still decisions to be made about specific system components such as printers and auxiliary storage devices, as well as about the capabilities of the basic system itself in regard to the amount of memory, the addition of a co-processor, and so on. These decisions will depend on the tasks to be performed and the software selected.

When the specification of hardware requirements is completed for a number of programs as part of the evaluation process described next, it can give you a picture of the total system needed.

Summary

The software and hardware issues just discussed form the basis for software evaluation since they describe the context in which the software is to function, and the part that it is to play in regard to the major tasks of evaluation. By considering them before beginning to evaluate software, it is possible to decribe any constraints that should be kept in mind. For example, software may need to run on a particular machine or operating system. Another constraint may be that program files such as those produced by word processing programs will need to be compatible across an organization. In this case, either one program must be found which meets everyone's needs, or the files from different programs must be interchangeable. Understanding the context of microcomputer use lays the foundation for software evaluation.

SOFTWARE EVALUATION

There are three steps that lead to the selection of particular software. The first step is to initially screen the various types of software, and the second step is to conduct systematic evaluations of them. The third step is to summarize the evaluation results for making selections.

Screening of Software

The initial screening of software may be constrained by prior decisions. For example a particular hardware brand and/or operating system may have been established as a standard. Therefore, only compatible software need be considered. Or only one set of tasks, such as those related to word processing, might be designated as appropriate for immediate implementation.

Therefore, only programs with word processing capabilities need be considered. At the top of the Software Evaluation Summary form in Appendix A, there is a set of blanks that can be used to describe the requirements of software relative to such constraints.

There are two ways to proceed from here. One is to use the Software Evaluation Worksheet in Appendix B. This worksheet reiterates the discussion of evaluation criteria presented in the main part of this guide. It also provides suggestions for rating software on the specific components of the major evaluation criteria (e.g., program ease of use: set-up/installation, initial learning).

The second way to proceed is to read the following discussion and then use the Evaluation Summary Form in Appendix A. The Summary Form should also be used with the Worksheet as a means of concisely presenting the evaluation results.

Program Description

The first step in describing a program is to indicate the name of the program and the version (e.g., dBASE III). Include the name of the manufacturer and/or supplier (not the specific store or vendor from whom you bought the package).

You should also give a brief description of the program in terms of its major function, such as word processing or data base management. If it is an integrated or multi-function program, include all of the major features (e.g., spreadsheet, graphics, telecommunications).

It is especially important to describe the requirements needed to run the program. These requirements include the following:

1. any hardware brand specific constraints,

2. the operating system of the version you evaluated, (e.g., PC/MS-DOS 2.10),

3. the amount of random access memory required (e.g., 512Kb),

4. any special languages necessary (e.g., integer BASIC),

5. the number of capacity of disk drives (e.g., hard disk of 10 Mb or greater, or two double-sided, double density 5-1/4" flexible disk drives with 360 Kb capacity),

6. an' special peripheral devices (e.g., card readers, co printer/plotter, high resolution color monitor).

7. any co-processors that are necessary or recommended to enhance performance (e.g., Intel 8087)

It is also valuable to know whether the program can be examined using a demonstration disk or for a specific trial period. Finally, the price structure of the program for single, multiple and backup copies, as well as new versions or u 'ates, should be indicated.

Once this basic information is ascertained, the following criteria should be considered. The results should be summarized on the back of the form in Appendix A.

Evaluating

There are many sources that can be used to identify likely candidates. For example, the PC Buyers Guide, "the independent guide to products and services for the IBM personal computer," lists a myriad of programs under various categories. Computer magazines such as INFOWORLD, Interface Age, T.H.E. Journal, Personal Computing, and Popular Computing have reviews which compare and contrast a dozen or more similar programs (e.g., word processing programs). There are also electronic compilations of software titles, such as that provided by SOFSEARCH (San Antonio, Texas, 512-340-8735). This data bank has over 19,000 software products listed, with 90,0000 operating variations. These can be searched electronically, much like the ERIC library. Last but not least, friends and colleagues who use microcomputers are generally more than happy to nominate software for your consideration, too.

Once a pool of programs has been identified that appears to fit within the constraints of your context, arrangements can be made for their review. One way to use the Summary Form is to summarize reviews conducted by others. There are many sources of reviews, particularly of business packages, such as word processing, data base management, spread sheet, graphics, and so on. They often use the same criteria as those on the Form. One could search for reviews of the programs identified through the screening process, and then simply consolidate them using the evaluation worksheet. It is also possible to interview current users of particular packages to get their ratings. In this case the worksheet could serve as an interview guide. The advantage of this approach is that one would quickly find which packages are used and appreciated by those people who are likely to be future sources of support. Additionally, using this approach would give some recognition to those who have already gained expertise.

Of course, it is possible to sit down with a number of packages and do the kind of evaluation suggested by the evaluation form. But the range and number of programs would have to be severely limited to make this feasible. That is one of the reasons for doing the task analysis suggested earlier. It is not reasonable for an individual or small group of people to think about computerizing many tasks, using many different types of software within a short period of time. Onlv the software related to those tasks that are of highest priority should be candidates for evaluation.

A good way to conduct an evaluation in a large organization is to have people with a strong desire to computerize similar tasks using particular types of software form groups to evaluate software. Each group's results could then be compiled and shared. This would not only accomplish the task of evaluating specific software packages, but it would also develop a pool of local experts with knowledge about a variety of programs. These people could then become leaders in educating others.

The reviewers should be representative of a variety of constituencies throughout an organization. The reviewers should be those who will eventually use a package (e.g., office and clerical workers, warehouse workers, administrators). Teams should consist of novice computer users, experienced users and technical people, since they will have different insights into a program's quality.

The evaluation form can be used to rate individual programs and to record notes and comments regarding three evaluation criteria. The major evaluation criteria are program ease of use, program performance, and program support.[2] The criteria and elements are applied in very similar ways to most types of software. The only elements that change dramatically from one type to another are ease of use after initial learning and versatility. These elements refer to the features of a program (e.g., word processing, data base management, statistical analysis). A brief description of each of the criteria and their related elements follows.

[2] These criteria and their elements are based in part on the Software Digest Ratings Methodology, Software Digest Inc., One Wy newood Road, Wynnewood, PA 19096.

Ease of Use

Ease of use includes three elements: (1) the ease with which you can set up the program on your computer, called installing a program; (2) the ease in learning the program; and (3) ease of use once you know how. Each of these aspects is described below.

Set-up/installation. Programs differ in the number and types of tasks required before the program can be used. Programs also differ in just how difficult it is to carry out these tasks. This can be measured, for example, in terms of the time it takes to complete installation. In rating a software package for ease of use regarding set-up/installation, you should check to see whether the following tasks are required and how easy it is to complete those that are required: copy master disk, select monitor type, add operating system, set printer parameters, create data disk, indicate number of disk drives.

Initial learning. Factors that influence initial learning include the presence or absence of a computer-based tutorial of program features, sample-runs and examples of actual data with written descriptions, levels of help available on screen, and inclusion of an initial learning section in the written documentation (in contrast to technical sections which fully describe all program features). As a general rule of thumb, a program is "easy" to learn if you can get an overall sense of its complexity within two hours.

Use after initial learning. Evaluating use after initial learning should involve testing the program in terms of the typical task(s) you anticipate doing. For example, you might create a standard form letter to be edited for a particular occasion, merged with a limited number of names, and printed, along with mailing labels. The point is to go beyond the tutorial and/or examples provided with the program. The more realistic the test is in terms of scope, time, users, and so on, the better it will indicate the ease of use after initial learning.

Program Performance

Evaluating program performance entails a more detailed look at program capabilities.

Speed. There can be a remarkable difference in how quickly programs perform tasks. In most situations it will not be possible or desirable to do highly sophisticated bench-mark testing of a program's speed of execution. Instead, the time it takes to complete each part of the trial tasks associated with initial learning and/or use after initial learning should be measured (a stopwatch should give results that are accurate enough). This might involve finding a record in a data base management system, merging an address and a form letter with a word processing system, or recalculating a budget projection using a spreadsheet program.

Error handling. The way a program reacts to input is the area of greatest potential frustration in using microcomputers. One aspect of error handling concerns how a program responds to commands entered by a user. In the worst case, the program will "crash" or "freeze," that is, refuse to respond further. Current data are often lost in such a situation. In some cases more or less intelligible error messages will be given by the program. These range from a beep, to operating system messages, to helpful advice. Error handling also includes procedures for getting back to the main menu or other starting point when a menu item or a command are erroneously selected.

Another set of error handling procedures relates to the software and operating system. Can the program close a file if the user leaves the program or one of its parts without saving current data? Error handling also refers to the program's ability to deal with situations in which: the file disk is full, the disk directory is full, or the system memory (RAM) is full. The following situations can also cause problems: when the disk drive door is open, when the system is directed to write to a write-protected disk, or when it is directed to print with the printer disconnected or turned off.

Versatility. A program is versatile to the extent that it has a variety of basic and advanced features. These features fall under several general categories. One category is data management, including data entry and editing. Another is data manipulation or the transformation of data. A third is data output to such devices as monitors, printers, storage units, and communication equipment.

The manifestation of features varies considerably from program type to program type. For example, word processing programs typically have many editing features not found in other types of programs. Spreadsheet and statistics programs typically have many data manipulation features not appropriate for word processing programs.

Too many features can burden a program. It is important to evaluate programs in terms of their versatility regarding those features you need most. Some common advanced features to look for include hard disk compatibility, the ability to link commands together into macro-commands and to create standard files (ASCII), mouse compatible, and the ability to support many different printers.

Support

Three elements of support should be evaluated. Two are related to the documentation provided with the program, that is, printed documentation and on-screen documentation. The other kind of support is the technical advice and training provided by the supplier and/or manufacturer and others.

Printed documentation. The manuals that come with many programs have been criticized for being unreadable and unintelligible. In some cases, they are the only information that is available. Therefore, the printed documentation should be carefully evaluated to determine its quality. Some valuable printed documentation features include:

table of contents,
index,
summary of commands,
summary of error messages,
description of how software is organized,
basic information about each program feature,
detailed information about specific features and when to use them,
written examples of how to use features,
clear illustrations of screens,
examples of results of using features,
technical information on the source code or algorithm used in the program,
information on how to modify the program to fit individual needs if appropriate.

The evaluation of printed documentation should include an indication of the types of information offered and the organization of the documentation. An assessment of the writing style should also be made because, regardless of the presence or absence of a particular type of information, the language used and clarity of writing have an important impact on the extent to which the documentation is helpful. That is, the writing style should be evaluated on how clear, concise, and jargon-free it is.

On-screen documentation. In addition to written information, many programs include documentation that is an integral part of the software itself. It includes various kinds of messages and directions, the format of the screen menus and other workscreens, as well as specific help features. Help features may include directions on using the software or definitions of terms. They may be organized into several levels with the intent of providing different degrees of assistance to novice, intermediate, and advanced users. These levels are associated with decreasing detail in the help messages appearing on the screen until at the advanced level help is only available if called for. The evaluation of on-screen documentation should start with the title page of the software as it appears on the screen.

Other support. With complex programs, sometimes more information is needed than is included in printed and on-screen documentation. Problems may occur that are not covered in the documentation. Particular uses may be desired for which there are no examples. Or, a person may simply need to talk to another human being to understand how to use the program. This is where support by the supplier and/or manufacturer, by independent trainers, by users groups, and by publications comes in. This support takes many forms: actual installation of a program, a hot line for questions, short-term training, a contract for extended support, a quick reference guide, or a detailed description of program functions and use. It is helpful for others interested in a program to know about any special support that helped you learn a program. Or, you might note on the evaluation form any lack of support that hindered your learning a program.

As you review a program, you should summarize your evaluation by rating it relative to the criteria on the back of the Evaluation Summary Form in Appendix A. On the Front of the form you should summarize your recommendations and describe yourself so that others will have an idea of your experience.

Selection.

Selection may well come down to the program(s) with the best ratings on those features of greatest importance as opposed to those with the best overall ratings. This notion of the highest ratings on the most important features is worth considering. Sometimes pricing, especially in regard to multiple copies, is the deciding factor among programs of generally equal ratings. In other cases it may be that speed, error handling, and versatility (i.e., program performance) is more important than either ease of use or support. Therefore, lower ratings in these areas would not disqualify a program if it was a strong performer.

Any combination of features is possible. Selection should be based, therefore, on a consideration of the combination of features most desired for the types of tasks to be performed using the program.

If the screening process left open the possibility of selecting programs that would run on different operating systems and/or machines, this might be a good time to narrow down those restrictions. By identifying all of the strong programs that run on a particular system, some guidance can be given to hardware selection. For example, if the most desired specialized programs (e.g., statistical analysis, project management) run on one machine, other programs, such as word processing, that are not necessarily the strongest in their group, might be selected so that only one type of machine or operating system has to be purchased.

The checklist in Figure 2 can be used to summarize hardware related requirements including internal memory (RAM), auxiliary storage (e.g., flexible disk and hard disk drives), hardware (e.g., numeric key pads, monitors, printers, modems) and other software or languages.

A picture of the standard system can be formed by consolidating the hardware requirements relevant to the leading software programs. Of course, specialized sub-systems can be put together to perform specific tasks such as extensive statistical analysis. In this case, for example, the standard system might be augmented by additional internal memory, a mathematic co-processor, and extensive external storage capacity, plus an automatic mark sense reader for entering data directly into the program from standardized test answer sheets.

MANAGING MICROCOMPUTER USE

Planning for microcomputer use involves more than just selecting hardware and software. For example, thought must be given to the provision of service, maintenance, and support of hardware and software, since special technical expertise is needed in each area. Support includes both initial and ongoing training for people in regard to the hardware and software selected.

Another related issue concerns the provision of facilities for using microcomputers. Major facilities issues include:

- the physical space allocated for microcomputers;
- the electrical requirements (i.e., the power to run one or more systems with central processing units and peripherals, precautions against power surges and power losses);

- furniture needed to make the use of machines convenient and comfortable;

- noise reduction in areas where many keyboards or printers are being used.

Security versus availability is also a concern. Microcomputers are vulnerable to abuse, vandalism, and theft, since they are essentially stand-alone units. However, overly tight security, such as locking them in windowless rooms, may discourage their use. The security of information stored on a microcomputer (e.g., flexible diskettes or hard disk) is also important. Access to personnel records, student records, and financial records must also be protected.

Security, facilities, services and support are important considerations whether you are planning to buy one computer or a hundred. Some thought about these early in your planning for microcomputer use will help you to avoid problems later.

REFERENCES

Huntington, F. (1983, Fall & Winter). The microcomputer in the administrative office. AEDS Journal, 17(1&2), 91-97.

Spuck, D. W. & Atkinson, G. (1983, Fall & Winter). Administrative uses of microcomputers. AEDS Journal, 17(1&2), 83-90.

Tally, S. (1983, Fall & Winter). Selection and acquisition of administrative microcomputer software. AEDS Journal, 17(1&2), 69-82.

APPENDIX A

Software Evaluation Summary Form

PROGRAM DESCRIPTION

Product Name: Version:

Manufacturer and/or Supplier:
(address and telephone)

Major Function(s):

Requirements

Operating System/Machine: Memory:

Languages: Disk Drives:

Peripherals: Co-processors:

Examination Options

Demonstration Disk: Trial Period:

Cost

Single Copy: Multiple Copies:

Backup Copies: New Versions:

RECOMMENDATION

Would you recommend this program to other evaluators?

_____ yes, for the following purposes ____________________________________
____

_____ not sure (why?) ____________________________________

_____ no (why?): ____________________________ ____________
____

EVALUATOR

Please rate yourself on the following items:	novice				expert
1. level of expertise with this program	1	2	3	4	5
2. general level of expertise with microcomputer software	1	2	3	4	5

Name:________________________________ Position:________________

Address:____

________________________________ Phone:________________

EVALUATION
Please rate the program on each of the following criteria. Also give reasons
for your ratings and/or other key information about the program.

Criteria	Rating (? = don't know)
Program Ease of Use	
Set-up/Installation Comments:	? poor fair good excellent
Initial Learning Comments:	? poor fair good excellent
Use after Initial Learning Comments:	? poor fair good excellent
Program Performance	
Speed Comments:	? poor fair good excellent
Error Handling Comments:	? poor fair good excellent
Versatility Comments:	? poor fair good excellent
Support	
Printed Documentation Comments:	? poor fair good excellent
On-screen Documentation Comments:	? poor fair good excellent
Cther Support Comments:	? poor fair gooc excellent

APPENDIX B

Software Evaluation Worksheet

This worksheet reiterates the discussion of evaluation criteria presented in this guide. It also provides suggestions for rating software on the specific components of the major evaluation criteria (e.g., program ease of use: set-up/installation, initial learning).

The worksheet can be used to guide the detailed evaluation of a package. The results can then be summarized on the back of the Evaluation Summary Form in Appendix A.

Program Ease of Use

Program ease of use includes three elements. One is the ease with which you can set up the program on your computer. This is also called installing a program. Program ease of use also covers how easy the initial learning period is. Ease of use after initial learning is the third part.

Set-up/Installation.

Programs differ in the tasks they require before the program can be used. The number and type of tasks required is one indication of a program's ease of use. Programs also differ in just how difficult it is to carry out these tasks. This can be measured, for example, in terms of the time it takes to complete installation. Below, indicate the tasks required (Y) or not required (N), and the time it takes for installation Then rate overall ease of set-up/installation and note special positive or negative characteristics on the evaluation worksheet.

Tasks (required = Y; not required = N)

_____ copy master disk

_____ add operating system

_____ create data disk

_____ other (specify)____

_____ select monitor type

_____ set printer parameters

_____ indicate number of disk drives

Time (minutes):____

Initial learning.

Factors that influence initial learning include the presence or absence of a computer-based tutorial of program features, sample-runs and examples of actual data with written descriptions, levels of help available on screen, and inclusion of a helpful initial learning section in the written documentation (in contrast to technical sections which fully describe all program features). In addition, program with menus are often easier to learn that those controlled by key commands (e.g., having to press the control and W keys together to delete a word). Of course, if key commands are used they should be consistent throughout the program.

Within a two-hour maximum time period, it should be possible to get a sense of a program's complexity and the ease with which it is initially learned.

Below, indicate if various aids to learning are present (Y) or not present (N), and the time it takes for initial learning (with a maximum of 2 hours). Then rate the overall ease of initial learning and note special positive or negative characteristics on the evaluation summary form.

Aids to Initial Learning (present = Y; not present = N)

_____ tutorial program

_____ sample run/example

_____ initial learning section in written documentation

_____ help screens

_____ levels of help

_____ menu driven

_____ other (specify) ______________________________ ____________________

Time (up to a two-hour maximum) ____

Use after initial learning.

Evaluating use after initial learning should involve testing programs in terms of the typical task(s) you anticipate doing with a particular type of software. To provide a consistent test, it is necessary to develop a standard task. For example, you might create a standard form letter to be edited for a particular occasion, merged with a limited number of names, and printed, along with mailing labels.

The point is to go beyond the tutorial and/or examples provided with the program. The more realistic the test is in terms of scope, time, users, and so on the better it will indicate the ease of use after initial learning. Of course, any test will fall short of total realism, but if used consistently, it can indicate in a relative way this aspect of program ease of use.

A set of typical task features can be generated from the basic software features described in the other guides in this series. (Advanced program features will be used to assess the program performance element called versatility). A checklist like the one below could be created to indicate if the features are present (Y) or not present (N). A rating scale like the one on the evaluation worksheet could be associated with each feature. The time it takes to do the task should figure into the rating. The overall ease of use after initial learning should be rated and any special positive or negative characteristics noted on the evaluation summary form.

Basic Word Processing Features (present = Y; not present = N)

Editing Control	(Y/N)	Rating			
cursor control	( )	poor	fair	good	excellent
text control	( )	poor	fair	good	excellent
page control	( )	poor	fair	good	excellent
inserting	( )	poor	fair	good	excellent
deleting	( )	poor	fair	good	excellent
search and replace	( )	poor	fair	good	excellent

Comments on time to complete task(s) (minutes):
(A checklist of features should be developed for each software type.)

Program Performance

Evaluating program performance entails a more detailed look at program capabilities.

Speed.

There can be a remarkable difference in how quickly programs perform tasks. In most situations it will not be possible or desirable to do highly sophisticated bench-mark testing of a program's speed of execution. Instead, the time it takes to complete each part of the trial task should be measured (a stopwatch should give results that are accurate enough). By adding a column for recording the speed of execution to the checklist of program features, it can become the means of rating this element of program performance. Summarize the results regarding speed by noting special positive or negative characteristics and give an overall rating on the evaluation summary form.

Error handling.

The way a program reacts to input is the area of greatest potential frustration in using microcomputers. One aspect of error handling concerns how a program responds to commands entered by a user. In the worst case the program will "crash" or "freeze", that is, refuse to respond further. Current data are often lost in such a situation. In some cases more or less intelligable messages will be given by the program regarding the error. These a range from a beep, to operating system messages, to helpful advice. Error handling also includes procedures for getting back to the main menu or other starting point when a menu item or a command are erroneously selected.

In addition, error handling refers to such conditions as when a data disk is full, when the disk drive door is open, trying to write to a write-protected disk and so on.

Below, indicate the reaction of the program to the various error situations, crash (c), unintelligable message (um), or appropriate message (am). It is assumed that error messages from the operating system level or lower (e.g., interpreter or compiler) will be generally unintelligable and not very helpful to most users. Then rate the overall error handling ability of the program and note any special positive or negative characteristics on the evaluation summary form.

<u>User Input Errors</u>	<u>Reaction</u> (c, um, am)
Wrong input (e.g., character rather than number)	________
Wrong command or menu choice	________
Leaving system without saving data	________

<u>System Conditions</u>	<u>Reaction</u> (c, um, am)
Disk full	________
Disk directory full	________
System memory full	________
Disk drive door open	________
Reading or writing to non-existent drive	________
Writing to write-protected disk	________
Printer disconnected	________
Printer turned off	________

Versatility.

Having features beyond basic functions is what makes a program versatile. Of course, these features vary considerably from program type to program type. However, there are some common advanced features such as hard disk compatability, the ability to link commands together into macro-commands, and so on. Using a checklist like the one below and one based on the advanced program specific features described in the other guides, indicate if the feature is present (Y) or not present (N). Then give the program an overall versatility rating and note special positive or negative characteristics on the evaluation summary form.

<u>Advanced Features</u> (present, Y, not present N)

General Features

_____ hard disk compatible
_____ macro commands
_____ creates standard files (ASCII)
_____ mouse compatible
_____ 80-column wide + text
_____ 5 or more printers supported

Program Specific Features

Support

There are three elements of support that should be evaluated. Two are related to the documentation provided with the program, that is, printed documentation and on-screen documentation. The other kind of support is the technical advise and training provided by the supplier and/or manufacturer.

Printed documentation.

The manuals that come with many programs have been criticized for being less than helpful. In some cases they are the only information that is available. Therefore, the printed documentation should be carefully evaluated to determine its quality.

The evaluation of printed documentation includes an indication of the types of information offered and the organization of the documentation. An assessment of the writing style should also be made because regardless of the presence or absence of a particular type of documentation the language used and clarity of writing are important ingredients in making written documentation helpful.

Below, indicate whether various documentation features are present (Y) or not present (N). Describe the writing style. And then give an overall rating for the printed documentation and note any special positive or negative features on the evaluation worksheet.

Printed Documentation Features (present = Y, not present = N)

_____ table of contents
_____ index
_____ summary of commands
_____ description of how software is organized
_____ basic information about each program feature
_____ detailed information about specific features and when to use them
_____ written examples of how to use features
_____ clear illustrations of screen
_____ examples of results of using features
_____ technical information on the source code or algorithm used in the program
_____ information on how to modify the program to fit individual needs

Writing Style	Rating			
Clear (understandable)	poor	fair	good	excellent
Concise (not verbose)	poor	fair	good	excellent
Jargon free (uses plain English)	Poor	fair	good	excellent

On-screen documentation.

This documentation is an integral part of the software itself. It includes various kinds of messages and directions, the format of the screen menus and other workscreens, as well as specific help features.

Starting with the title page of the software as it appears on the screen, use a checklist like the one below to rate the information presented in terms of clarify, conciseness, and freedom from jargon. Also indicate if help on specific features is provided and if it is available in more than one level of detail. Then give an overall rating of the on-screen documentation and note any special positive or negative characteristics on the evaluation worksheet.

<u>On-Screen Documentation</u>	Clear	Concise	Jargon Free
Messages	p f g e*	p f g e	p f g e
Directions	p f g e	p f g e	p f g e
Menus	p f g e	p f g e	p f g e
Work Screens	p f g e	p f g e	p f g e

*poor (p), fair (f), good (g), excellent (e).

Help features: _____Y (present) _____N (not present)
Levels of Help: _____1, _____2, _____3, _____ more

<u>Supplier/manufacturer and other support.</u>

It is sometimes the case with complex programs and novice users that more information is needed besides that included in printed and on-screen documentation. Problems may occur that are not covered in the documentation. Uses may be desired for which there are no examples. Or, a person may simply need to talk to another human being to understand how to use the program. This is where supplier and/or manufacturer support comes in. This support can take several forms, actual installation of a program, a hot line for questions, short-term training, or a contract for extended support.

Below indicated if these forms of support are available (Y) or not available (N), who would supply them, supplier, manufacturer, others, (e.g., users groups, independent trainers or consultants, publications), and their cost. Then give an overall rating for supplier/manufacturer support and note any special positive or negative characteristics on the evaluation worksheet.

<u>Support Checklist</u>

Form of Support	(Y/N)	(who supplied support)	(cost)
Installation	_____	_______________	_____
Problem Hotline	_____	_______________	_____
Short-Term Training	_____	_______________	_____
Extended Service Contract	_____	_______________	_____
Quick Reference Guide	_____	_______________	_____
Detailed Description of Program Functions and Use	_____	_______________	_____

RECENT GUIDEBOOKS IN THIS SERIES

No.	Title
1	Microcomputers and Evaluation
2	Cost-Outcome Analysis: Measuring Costs
3	Microcomputers: Word Processing
4	Cost-Outcome Analysis: Measuring Outcomes
5	Microcomputers: Statistical Analysis Software
6	Investigative Journalism Techniques
7	Microcomputers: Data Base Management Software
8	Committee Hearings: Their Use in Evaluation
9	Microcomputers: Spreadsheet Software
10	Methods of Product Evaluation
11	Microcomputers: Instrument Generation Software
12	Evaluation Design: The Evaluator and the Architect
13	Microcomputers: Communication Software
14	Evaluation Design: Selecting Methods
15	Service Delivery Assessment Techniques

Printed by the Northwest Regional Educational Laboratory, a private nonprofit corporation. The work upon which this publication is based was performed pursuant to Contract No. 400-80-0105 of the National Institute of Education. It does not, however, necessarily reflect the views of that agency.

Research on Evaluation Program
Northwest Regional Educational Laboratory
300 S. W. Sixth Avenue
Portland, OR 97204