

**Software Development
for Computer Applications
in Agriculture and Forestry**

October 6-7, 1982

Atlanta, Georgia

Proceedings of
SOUTHERN RESEARCH DEVELOPMENT COMMITTEE ON
COMPUTER SOFTWARE DEVELOPMENT

Southern Association of Experiment Station Directors
Southern Research Development Committee

on
Computer Software Development

Prepared by the Southern Rural Development Center

Atlanta, Georgia
October 6-7, 1982

TABLE OF CONTENTS

	Page
FOREWORD	ii
I. OVERVIEW--Dr. R. Rodney Foil	1
II. STANDARDS	
A. Dr. Robert S. Sowell	3
B. Dr. Donald C. Huffman	11
III. PEER REVIEW	
A. Dr. J. W. Mishoe	13
B. Dr. Terry C. Keisling	23
C. Dr. Wallace E. Killcreas	29
D. Dr. Rob Martin	33
IV. SOFTWARE INVENTORY AND CLEARINGHOUSE	
A. Dr. Glenn O. Ware	37
B. Dr. Gary Wells	39
V. WHERE DO WE GO FROM HERE? -- Dr. R. Rodney Foil .	41
VI. APPENDIX	
A. Agenda	47
B. List of Participants	49

FOREWORD

This proceedings provides a summary of the third meeting of the Southern Research Development Committee on Computer Software Development. Sixteen persons attended the meeting October 6-7, 1982, at the Capitol Airport Inn in Atlanta, Georgia. The first meeting of the group was in April 1982 in Memphis, and it was followed by a second meeting in July at the Capitol Airport Inn in Atlanta.

This committee has been asked by the Southern Association of Experiment Station Directors to consider the subject of computer software development in agriculture and forestry and to recommend a course of action for regional cooperation. Although most state Extension Services and Experiment Stations are involved with computer applications, very little organized effort exists on a regional basis and no agreement has been reached concerning standards, needs and definition of roles. As on-farm use of computers increases and this technology becomes available for widespread use, more regional coordination may be necessary in software development. One of the goals of this committee is to determine what portion of each state's efforts in this area can best be done on a regional basis and then to pull together a consensus of opinion concerning what kind of cooperative activity is necessary.

OVERVIEW

R. Rodney Foil
Advisor
Southern Research Development Committee
on Computer Software Development

As part of our continuing effort to determine a regional course of action in software development, we have arranged a meeting in February to bring together extension and experiment station directors. This meeting also may include representatives from resident instruction. We have scheduled the meeting for February 9-11 in Atlanta immediately following the meeting of the Southern Association of Agricultural Scientists.

One of the purposes in arranging this meeting is to share information on the current status of software development in agriculture and forestry throughout the Southern region. We also want to estimate the future needs for computer applications in the region and to establish a working relationship between research, extension and resident instruction in the area of computer software development. Bob Kramer of the Kellogg Foundation has agreed to co-sponsor this meeting and has suggested we consider a hands-on demonstration of hardware and software.

There are two things I would like us to consider in preparation for this February meeting:

1. Each state will have to prepare a 15- to 20-minute presentation concerning the status of software development in that state. We need to develop an outline of topics that should be included in these reports, e.g., organization, distribution, hardware, consulting-private sales, review and publications.
2. How will the hands-on demonstration be conducted? Will we bring in computers from one of the land-grant universities or use equipment from vendors in Atlanta? Perhaps those states that have the most advanced programs in software development can take more time for their presentations and demonstrate programs they have developed.

A preliminary outline of the program includes the following:

Wednesday, February 9

noon - 5 p.m. Reports from States

Thursday, February 10

10:00 - 3 p.m. Future Needs--prepared statements from such persons as Bob Kramer of the Kellogg Foundation, the North Central Computing Institute, the USDA

3 p.m. - 5 p.m. Research and Extension Directors have separate meetings

Friday, February 11

8:00 - noon Report from separate groups followed by cross-function speaker, possibly a land-grant vice president to lead discussion

STANDARDS

Robert S. Sowell
North Carolina State University

Frequently during the last two meetings of this committee we have mentioned the "lack of standardization" and the "need for standardization" relative to microcomputers in agriculture. These statements have also been made in many discussions among my colleagues in the North Carolina Agricultural Research Service and the North Carolina Agricultural Extension Service. Unfortunately, due either to a lack of time or knowledge on the subject, these discussions generally have ended with these comments. Our first agenda item for this meeting is to deal with this matter of standardization in more detail and to begin to get into the specifics of what we perceive to be the standards which are most urgently needed in the development of software by researchers in the Southern region.

What do we want to standardize? To begin to give us some structure for discussion, I will categorize the potential areas of standardization as follows:

1. Hardware
2. Operating Systems
3. Programming Language
4. Program Style
5. Documentation

Let us look briefly at each of these.

HARDWARE

We could consider adopting as our standard one "brand" of hardware and decide that we will be in the business of developing software for that

brand and only that brand and that farmers, ranchers, homemakers and other clients that we serve would be expected to buy that brand of hardware if they want to use our programs. Or at least they would be responsible for making the conversions necessary to get the software which we develop to run on their hardware. Perhaps we want to limit software development to two computers. In Successful Farming's Farm Computer News survey (reported in December 1981), 75.7 percent of the subscribers who reported that they owned a microcomputer indicated that it was either an Apple (40.2 percent) or a Radio Shack (35.5 percent). Maybe we could add two or three more brands and take care of 90 percent or so in the foreseeable future. According to Agricultural Computing, Ben Rosen, a noted industry specialist and writer for the Rosen Electronic Letter, indicated that survival in the microcomputer market will be difficult and predicted that by 1985 the top contenders, in order, would be IBM, Apple, Tandy, NEC and Atari.

I am not recommending that we adopt such a standard, but if we did the problems that we will discuss later with respect to operating systems and software would be easier to deal with.

We may, however, want to seriously talk about standards with respect to the minimum complement of hardware for perspective users of software developed by agricultural researchers in the Southern region. Such minimum requirements might be in terms of cassettes vs. disk; minimum RAM, minimum external storage, and printer (should one be required, characteristics, etc.). Again referring to the Farm Computer News survey, 86 percent of computer owners had a disk drive and 81.9 percent had a printer.

OPERATING SYSTEMS

Currently within the Southern region, microcomputer programs are being developed and distributed under at least four different operating systems: Model II TRSDOS, Model III TRSDOS, APPLE DOS, and CP/M. It has been suggested that we should adopt a standard and that the standard should be CP/M. CP/M is available on many microcomputers; however, it is not available on some. Also, it is an additional cost item for many computers.

I would be much more supportive of establishing CP/M as a standard Operating System if the situation with respect to the 16-bit microcomputers was clearer. Will CP/M for the IBM PC be compatible with CP/M which may be available for the TRS-80 Model 16 and other microcomputers using the 16-bit MC68000 processor? Rather than adopt a single standard operating system, I suggest that we should make software available under a few of the most commonly used operating systems. I understand that Oklahoma State University has translated many of its programs (originally developed for the Radio Shack computers) to run on the Apple II⁺ (Farm Computer News, July 1982). At North Carolina State University, most of our programs run under TRSDOS, Models II and III, and under CP/M. Obviously, additional programmer resources are required to make these translations from one operating system to the other. However, if at the beginning of program development it is decided that the program will eventually be run under more than one operating system, efforts can be made to minimize use of language features that are unique to only one operating system. We have found this to present few problems with TRSDOS and CP/M. After programming (in Interpreter BASIC) for one operating system, the program is transferred to the other via the RS-232 and necessary changes are made in the program so that it will run under the other operating system.

PROGRAMMING LANGUAGE

I believe that most of the software currently being distributed by land grant colleges and universities in the Southern region is written in interpreter BASIC. Some are in FORTRAN, some in CBASIC. There probably are some written in PASCAL and other languages, but I am not aware of these. If we choose to adopt a standard operating system, e.g., CP/M (and I believe that this is the only feasible one at this time), then we should probably adopt one or two languages as standards. I suggest that we adopt interpreter BASIC as one and either CBASIC or FORTRAN as a second. Interpreter BASIC is quite adequate for many of the programs that we will distribute, e.g., enterprise budgets, financial analysis, etc. The user generally waits very little for the computer when running these types of programs. By staying with interpreter BASIC for these types of programs, the user can more easily make changes to tailor to a specific situation. There will be many programs (e.g., least cost rations, crop-mix selections, etc.) which utilize optimization routines and records programs requiring file access time. These should be programmed in a compiler language to minimize run time and user waiting time.

If we do not choose a standard operating system, there may be some advantages in choosing FORTRAN over CBASIC as the compiler language. This is particularly true if Radio Shack TRSDOS is to be one of the operating systems supported. Their Compiler Basic is quite different from microsoft CBASIC. I don't believe there is quite as much difference between the versions of FORTRAN available under the various operating systems.

PROGRAM TECHNIQUES

In Dr. Steve Welch's presentation to this group in July, he devoted a large portion of his time to "quality" software. He identified four components

of quality software :

1. Significance of the subject material
2. Sound underlying theory
3. Software engineering
4. Good documentation

I believe that it is more appropriate to discuss the first two of these in the next session on peer review , and I will address the fourth in the last part of my comments. Software engineering has to do with the "coding and techniques that go into production of the program" (Welch). Dr. Welch suggested that "one ought to make them (the programs) modular and one ought to follow certain approaches and do certain things to contribute to the understandability of the program or the ease of writing or ease of transferring."

We have identified a few things that have led to an improvement of the usability of software on computers in our county extension offices and have adopted the following goals for software developed for our extension computers. Many of these will apply to software distributed to farmers.

1. Make them user friendly
 - a. Give clear instructions on the screen
 - b. Give the user an easy way to correct errors as they occur
 - c. Use default data where possible
 - d. Question suspicious input
 - e. Allow the user to change selected inputs after the program has been run so that he (she) can run again without going back through the entire program
2. Have consistency among programs
3. Have optional softcopy vs. hardcopy output

The decision that we must make is how restrictive do we want to be in setting standards with respect to coding and techniques.

DOCUMENTATION

At our last meeting Dr. Wallace Killcreas stated that "documentation is absolutely essential for all software." I agree and there are as many approaches to documentation as there are programming styles. Any one of several would probably be adequate for our purposes; but if we adopt standards for programs and have a regional distribution policy, then we should adopt standards for the documentation. As Dr. Killcreas suggested, the documentation should address the needs of the user audience; it should be a technical bulletin describing what the program does, its limitations, hardware requirements, etc.; and it should provide the information to maintain the program.

There are two types of documentation and each type should address each of the above audiences. Internal documentation is that written in the computer code. Internal documentation intended for the user is printed on the screen. It tells the user what the program is doing as it progresses and provides points of clarification and help when the user is confused. Internal documentation addressed to the programmers and others who will modify and maintain the program is in the form of remarks in the code itself.

External documentation is a hard copy document which should accompany the software but which may also be distributed independent of the software. It should contain general information (why the program was written/what it will do), technical information of the type described above, and a complete example (input data, copy of the interactive session, and result) with explanations.

A great deal of emphasis has been placed on the need for external

documentation. We as a committee need to deal with standards, specifically content and format, for external documentation. However, we should not emphasize external documentation to the extent that it implies that internal documentation is not as important. I tend to believe that internal documentation is at least as important for two reasons:

1. External documentation may become separated from the program
2. Users tend not to read the external documentation.

Therefore, I believe we should place equal emphasis on internal documentation standards. Internal documentation could include much of the same information as contained in the external documentation, including examples of use of the program. We developed a program this summer (a program to store scout reports for the apple integrated pest management program) which included a complete sample run, including movement of cursor to various cells in a table depicted on the screen, and a tutorial session so that the user could both observe and practice before inputting real data to the program. These were options made available to the user. Detailed internal documentation of this nature should always be optional since it can be quite time consuming.

DOCUMENTATION STANDARDS

Donald C. Huffman
Louisiana State University

I would like to begin by saying that we are talking about personal applications of computers and about standards -- but standards for what and standards for whom? For what purpose are we trying to do this?

In terms of hardware, operating systems, and program language, each state or agency has different goals and objectives depending on how they are using microcomputers in information dissemination or educational programs through the extension service. What needs to be standardized within these systems is critical, but this standardization does not necessarily have to cross state lines. Alabama may want to coordinate very closely what they are doing in terms of software language and hardware because they are depending on North Carolina for a lot of basic programs. They are tied closely together. Other than this type of standardization between states, what we are really talking about is documentation. Part of this documentation is internal but a lot of it is external. Documentation provides a level of credibility for a piece of software. The greater the degree of documentation, the greater the potential of credibility. This creates a system that permits us to operate and to let the user beware. If the software is not documented, how good is it? It may be excellent but no one knows it. If it is poorly documented, how good is it? The degree of documentation that goes along with that program shifts the liability to the program provider.

He is liable for the claims he makes in his documentation. In view of this, the standards should apply to the documentation, not the software. What needs to be in the documentation will determine the credibility of the program.

In terms of documentation, I would like to make specific suggestions. First of all, the documentation should include specification of any mathematical computations embodied in the program. Secondly, the documentation should contain a referencing of the authority basis for computational processes and theoretical models encompassed in the program. Who is the authority? In addition, an identification of deviation for normal or standard computational procedures may be necessary because of the particular hardware system being used. The user should be aware that a deviation has occurred. Another item that the documentation should contain is the identification of source and location of user-defined parameters and data sets embodied in the program. For example, if we have a program that is designed to tell us how many acres we need to water and when we need to irrigate, that program may have been designed for Texas but is not workable in Louisiana. The mechanics of the program may be excellent, but what is the evaporation rate? If that is a parameter that is embodied in the program, then we may be able to take the program and use it very well in Northeast Louisiana. But first we have to change that coefficient, and we need to know where it is and what the source of it is. Two final suggestions include a listing of general and/or specific limitations of the software package, and an author or contact agent that may be appropriate for software adaptation or modification.

PEER REVIEW OF COMPUTER APPLICATIONS: AN INTRODUCTION

J. W. Mishoe
University of Florida

I have asked a number of people if we need a review process of computer programs developed by scientists in the university environment. The reaction I have received has been widely varied ranging from "no review is needed" to "yes, a review process is essential." The concept of what needs to be reviewed is also widely varied. In general, one's response to this question depends greatly upon one's perspective of who uses these programs and what type of information is contained in the program. And, of course, these same areas in turn affect the type of review needed. In essence, as a community of agricultural scientists, we do not have a common perspective on what constitutes an agriculturally related computer application and how these applications would and could be used.

The first question that must be resolved is, "Do we need to implement a peer review process for computer applications?"

To answer that question in the affirmative, one only needs to examine the benefits of the review process of scientific publications. First of all, the common benefit of a peer review is the resultant improvement of clarity and organization of the manuscript. Technical errors and omissions can usually be detected readily by a reviewer thus allowing correction before publication. Another equally important reason for the review process is to protect the reader or user from technically poor or bad information. Rarely do we find scientists trying to publish technically unsound information; however, the review process provides the opportunity for knowledgeable persons to examine

the material before it is published and perhaps used by people less able to evaluate the validity of the method and results.

A third reason we currently have a review process is to assure university administrators that the work is at least consistent with accepted scientific methodology thus allowing them to more easily recognize and reward prospective scientists. Review of computer applications by knowledgeable peers will reassure administrators that computer programming researchers maintain professionally acceptable output. This reason for review is somewhat secondary to supplying high quality applications to clientele; however, it is necessary that scientists be rewarded for production of products that are in fact needed and that advance the distribution of agricultural technology. Rewards consistent with conventional evaluations for scientific merit are necessary to assure continued development of new and improved applications.

These reasons for review should apply to all types of computer applications because they are worthwhile guidelines that have few negative aspects. The difficulties arise when one begins to apply the more detailed review guidelines of scientific publications. For example, when reviewing a report, one examines the hypotheses, review of literature, scientific methodology, use of state-of-the-art information, validity of conclusions, new information contained in the report, etc. The reason is that computer application consists of several components including technical documentation, computer code, user's manual, and other material such as testing or operational results and special hardware or software "option." Also, the types of applications vary tremendously in the scientific content of the program. For example, a computer program that is to be used by county agents to sort mailing lists would contain no new information and have little scientific merit. It is, however, an important tool that should

be made available to improve overall delivery of information by the county agent. On the other hand, some applications may represent the availability of a not-before-available technology to the user. In this group, there exists a number of simulation or analytical models that represent various production or biological systems. Examples of these include irrigation scheduling programs keyed to yield loss rather than current soil moisture status, farm level management simulation models, and pest management decision models. Often these types of tools are only usable by the highly trained individual unless implemented in an appropriate computerized delivery framework.

In our current information delivery system for agricultural technology, we have basically depended upon the review process to focus at the research level and deal much less at the extension level. The development of computer programs often requires the subject matter expert or experts to become more involved at the delivery level. It is not as easy to share the burden of putting the information in a form that is directly usable by producers. The problem associated with this issue is the acceptance by peers that are not involved in this type of research of the need to combine the traditional extension role. I must point out that the severity of this issue varies tremendously by institution, department, and individual.

Because computer applications do vary from those that are relatively simple in nature to those that require an interdisciplinary group to develop, the ability to review and the type of review needed does vary. The review process, therefore, must be flexible to allow for a review most appropriate for the particular application.

To address these issues, members of the peer review committee will present their comments related to three general topics. These are as follows:

1. Which components of the computer application should be reviewed?
2. What types of reviewers should review each application?
3. What criteria should the reviewer use to evaluate the computer application?

After this we will open the discussion to the group. I expect each author will overlap somewhat in the material he presents. If differences in approaches arise, it will provide a framework for the group to begin to resolve those issues that currently exist.

IS A PEER REVIEW NEEDED ?

- Improve clarity and determine errors
- Screen for unsound technical material
- Provide framework to reward developers

CONVENTIONAL SCIENTIFIC REVIEW

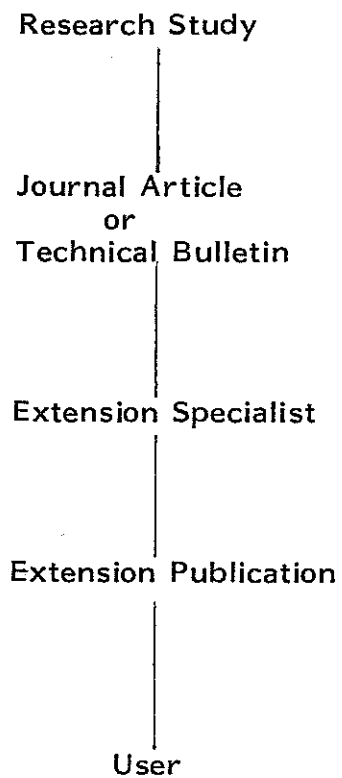
- Hypothesis
- Review of Literature
- Scientific Methodology
- State of the Art
- Validity of Conclusions
- New Information

COMPUTER APPLICATION REVIEW

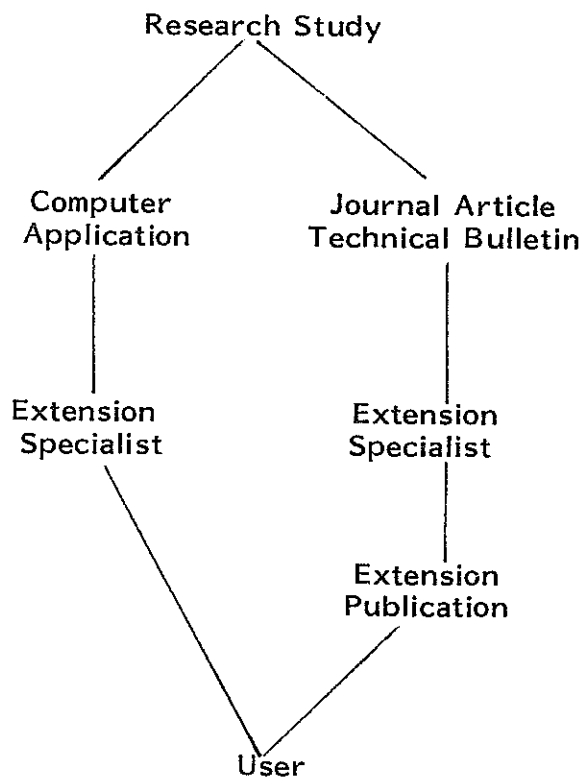
- User's Manual
- Technical Documentation
(subject to conventional
review if appropriate)
- Computer Code
- Program Interface
- Validity of Program to
Technical methodology

INFORMATION DELIVERY

CONVENTIONAL



COMPUTERIZED



WHO SHOULD REVIEW COMPUTER SOFTWARE FOR AGRICULTURE?

Terry C. Keisling
University of Arkansas

This subject can be controversial, but it need not be. The purpose of a review is to solicit opinions on the appropriateness of the subject. We should have adequate resources to select people who can give us these opinions in the Agricultural Experiment Stations and Extension Services in the South. Another reason to review is that the review process acts as an aid in maintaining a high standard of quality in programs. Some quality points of concern would be originality, appropriateness of methods, accuracy, validity, organization, and clarity; and in computer programs we would be interested in adherence to information and engineering principles.

Taking a note from the North Central Computing Institute, (NCCI), we would also be interested in the insurance of adequate information about implementation of the subject as a program. I think we will find that this is different from other categories. To insure implementation, specific details about exact software-hardware environments of operational software can be extremely important in specifying what programs are run on what equipment.

What content are we after for reviewing? Are we interested in reviewing the technical basis or background of a computer program? If we are, then we need technical people to review it. How about the structure of delivery or the

Terry C. Keisling, Extension Agronomist - Soybeans, and Associate Professor of Agronomy, University of Arkansas. Presented at the October 7, 1982, meeting of the Southern Research Development Committee on Computer Applications.

ease of maintenance of a program? This should be high on our list. How about the audience? I feel that if we are talking about reviewing computer code, the audience for which the code is developed must review it.

How about the program type? Just to list a few of the different programs, there are the following:

- . Simulation programs, which are usually lengthy and involved and require much computer memory and fast processing
- . Simple accounting to complex accounting programs
- . Data analysis -- for example, statistical analysis
- . Simple information retrieval -- and this can be simple or very complex
- . Data conversion, which is usually straightforward but could include programs that would require tables and make pie charts or graphs
- . Some sort of data summaries
- . Tutorial programs
- . Combinations -- and we could go on and on if we started looking at combinations. As a matter of fact, the number of types of programs we could have would then be seven factorial.

How about the program applicability area? Do we review for local, state, regional, national, or international? We might need some definition here.

Local program -- developed for a single user in one location.

A state program would be that type that is applicable only in one state. For example, Mississippi State has developed a nice program on estate planning. However, the state laws make the program applicable only in Mississippi.

I think we should realize that some programs fit into the above category. The user should be warned that a program is only applicable within a state. Problems of this nature are especially important for consideration by this committee. By establishing some procedures applicable to the region, we should be able to have better ongoing internal programs as well as be able to interface with other states in a more efficient manner.

Regional -- We are in the regional business here and we want general programs that are applicable across states within (in our case) the Southern region.

National -- international -- These might be out of the scope of this committee, but we should think a little about it.

What should the reviewers review? Should they review the factual documentation? If they do, we are talking about at least two scientists; and in my opinion, they should be from states different from the state where the software originates for regional applications. Structure of delivery and maintenance? If that is what we are reviewing, then I feel that we need at least one qualified person (from a state other than the state where the software originates) to review that particular process.

Reviewing code? If we review code, that in itself is a tremendous problem. After delving into the subject and studying it for quite some time, I have come to the conclusion that the North Central Computing Institute was wise not to review the actual code. We need to have dedicated people spend quite a bit of time doing nothing but looking at computer code. One small change in the code can cause tremendous changes, and I feel that the code should be verified by the person who is putting the code together.

Who should the other state use as their personnel to review the program? I would think it would either be a scientific or Extension type person who is well versed in the subject matter area. They need to also make sure that they select a person who is somewhat well versed in computers and computer type delivery. Someone who is familiar with algorithms and this type thing. I personally think we have people in most states qualified to look at the structure of delivery and maintenance.

As far as getting audience review, when we are looking at output, I feel that this could be handled by the state that originates the program. The selection of particular personnel will depend upon the program type that we are talking about. For example, the people who need to review a simulation type program usually are not at all familiar with the techniques involved in information retrieval. The technical bases behind information retrieval having to do with file structures and access times and techniques really tend not to lend themselves to simulation type programs in which we take a few basic variables and attempt to grow a soybean crop and this type thing. I think it is fairly obvious that tutorial programs should be reviewed by the teaching community or the community that hopes to use them for educational purposes, which in our case could be Extension.

Other program types which would originate in private industry leave somewhat of a question mark that I have not touched on in any of my discussion so far. But, essentially, let me summarize by giving an example.

When we are programming a computer and we say multiply X by Y which is a straightforward example and it returns the answer, do we worry that the computer multiplies X by Y? Do we go in and check the machine language instructions, of which there may be 45 or 50, to accomplish the simple

multiplication or do we accept the fact that it multiplied X times Y and we got the answer? In other words, we have a situation here where we are willing to accept that I put X and Y in with a star command or an asterisk between them, and I got back the product of X and Y. So I put in a product. A black box process goes on. I know not where; I do not care. I get out an answer. I assume that the people who put the program together did it in a technical enough fashion that the internal operation is correct, and I can be interested less in all the processes that are going on in the simple multiplication of X times Y.

In our case here, we are talking about producing programs which people will put into a computer. They will put some things in up front, and they will get an answer. What we are interested in is maintaining through the review process a quality of program which is accurate and valid and which will give us an answer without our being concerned about what went on from the beginning to the end. I feel that we have the resources of people in the Agricultural Experiment Stations, Extension Services, related federal government agencies and computer science departments of various schools to help us to obtain this objective.

WHAT SHOULD BE REVIEWED BY PEER GROUPS
IN THE SOUTHERN REGION?

Wallace E. Killcreas
Mississippi State University

I will move from my perception of the most general definition of what should be peer reviewed to the most specific.

Most General Definition of What Should Be Peer Reviewed

Agriculture computer software and supporting documentation intended for wide distribution in the Southern Region.

All material should be thoroughly reviewed locally prior to submission for peer review. I recommend 3-5 local reviewers followed by 1-2 regional reviewers.

Why computing software only? Agencies already in place, such as SAAS (Southern Association of Agricultural Scientists), can perform necessary publication functions for non-computing papers.

Why limit the review to software intended to be widely distributed? Local applications, as well as some specialized applications such as citrus or catfish, should be limited to states in which they are germane. This does not preclude birds-of-a-feather (Florida, Texas, California for citrus and Mississippi, Alabama, Arkansas, Louisiana for catfish) getting together in joint efforts.

Computer programs submitted for review should be complete and tested. Sufficient operating instructions and data should be provided so that the reviewer can review the program as part of the review procedure. Since programs are the main product, their review is essential to the review process.

Intermediate Definition of What Should Be Peer Reviewed

Agricultural computer software and supporting documentation to be distributed (directly or indirectly) to individual producers in the Southern region.

This definition would not include software developed to expand computational techniques (mathematical algorithms, specialized sorting or storage techniques, etc.). Such software could conceivably be published in computing publications or agricultural disciplines such as SAAS.

Most Specific Definition of What Should Be Subjected to Peer Review

Agricultural microcomputer software and supporting documentation intended for direct distribution to individual producers.

This definition requires slightly more heroic assumptions, but not necessarily illogical ones.

The real reason that we are here today looking into peer review of computer software is the rapid development of microcomputer hardware. There is a tremendous demand for microcomputer software by individual agricultural producers. Most individual producers cannot afford to buy minicomputers or large-scale computers.

Developers of large-scale and minicomputer software can generally publish through conventional channels due to the relationship of their work to the more traditional agricultural areas of interest.

General Comments

1. Everyone submitting information for review should be willing to perform reviews themselves.
2. We need to immediately specify equipment and supporting computer language configurations at each university involved in the review process.
3. Time should be set aside during our meetings for committee interaction.

PEER REVIEW CRITERIA

Rob Martin
Auburn University

When we met in Memphis, I strongly supported peer review and I still do today. I believe there are two purposes for peer review. One is to receive credit for the work, and the other is to avoid duplication. There is no reason for me to do something that has already been done on a separate computer if I can pick it up from another source. Peer review goes right along with an inventory; both help in avoiding duplication.

If we examine the documentation outline of the North Central Computer Institute, we find that they look first at the problem. They make a statement about that problem, and then they look at the solutions and subject matter vocabulary. They also look at solutions in computer terms to facilitate implementation on an arbitrary computer system. In addition to the problems and two sets of solutions, they provide information that will be useful to the program users. NCCI provides validity tests and information as well as testing aids. They tell how to use the test in order to see whether or not the program is suitable to particular needs, and then they provide lots of references and knowledge.

In terms of peer review, we have the conventional route used for journal articles and then the computerized route. In the conventional route, general review criteria used for journal articles do not insure that procedures and results can be strictly reproduced without input from the authors. If I read a journal article, I believe all these things were done that were stated in the procedures. I do not have enough information to

check and see whether or not that was the case. Should we do this with software documentation or should we go further?

One other aspect of peer review concerns the credibility of the developer. We need an inventory of available software. If we go beyond the land-grant system or the government agencies that are like the land-grant system, the credibility of the developer becomes very important. In this generalized area, if a graduate student puts together a good journal article that stands on its own, it does not matter who he is or whether or not he ever got through the program. What is credibility here? I wonder about that. When we get to the tree market area, I know we need credibility of the developer.

I have put together a peer review outline for our consideration. The first part should be a good statement on the area being covered and its importance. In terms of the importance of the program area, we should look at the general needs of the program as well as the specific needs. Another item to include in the outline would be the potential benefits that the software might bring about. I think the documentation should describe this. Is the program for a farm firm, a commodity group, communities or institutions? There may be some software developed for microcomputers that is useful in an institution at the university, maybe in an administrative budgeting sense. I do not know that we could leave that out. As we state a problem area, let us put the objective in relation to the problem area. Then let us look at the objective in terms of clarity of purpose and whether or not we've really got an extension role or research role. This needs to be said out front so we can evaluate whether or not the program does its job.

In this documentation outline, most of you naturally want to look at the methodology from a subject matter standpoint. We also have to look at it from a computer software standpoint. This is where the inventory ties in. We can look at this as a review of other work, identifying the capabilities and limitations present in existing software and then identifying the improvements that might be made to this particular software.

Next in the documentation is a discussion about implementation. In implementation I would expect to see some demonstration in case application(s). Then I would expect to see a report on field testing. This report would explain how well the program implements or can be implemented. Next would be a description of the program, and we would have written documentation with a concern for quality and clarity. Then we have internal versus external. I would call the display documentation external. Here we need quality, clarity and amount. Do you have enough display to carry you through the program -- menus and that type of thing?

The last criteria that I would suggest would be quality, and I would measure quality in terms of meeting objectives and limitations. Are there objectives that have not been met and should not have been part of the objectives in the first place? I would also measure quality in terms of ease of implementation. I want to look at menus and samples of problem data. We should have some sample problem data to go in this implementation. I also want to look at operational efficiency. I do not have to have a code to determine that, but if I get a program in which I make an error, do I get to recover it or am I blown out of the program and have to start over again? I hate these programs where I

have to go back and go through grounds that I have already been over. I like to have plenty of escape and recovery. Range of application is another way to measure quality. In other words, I think we should have a lot of use and we should use a lot of user-assigned coefficients. In that way, the range is much larger. An example might be a simple budget that contains in the fixed part of the machinery coefficient an item called property tax on farm machinery. Alabama does not have that; I expect Mississippi does, and I know Georgia does. So you really ought to be able to assign that percentage rate as a user if the program is going to cross state lines. I believe we are talking about a regional effort in which programs would cross state lines. Another measure of quality would be the use of computer capability. Are we fully utilizing computer capability in the micro senses? A final measure of quality would be the range of hardware applicability.

I do not have suggestions for a format that could be followed for an editorial reviewer sheet, but I think many of the items I have mentioned could be rated using the numbers one through five. One is excellent, five is poor, and three is average.

SHARING COMPUTER SOFTWARE RESOURCES

Glenn O. Ware
University of Georgia

Many of our state universities have begun to provide a response to the computer software needs of the agricultural community. Although a variety of computer software programs has been developed by most states, only a limited number of universities have established organizational units to develop, maintain, support, and distribute computer software. As other states begin to develop similar capabilities, they may wish to benefit from the established units and avoid the high costs of computer software development through resource sharing.

Resource sharing offers the most logical and natural means for continuing and improving the delivery of computer software services to the agricultural community. Although very little organized effort exists between the states in the development of computer software, working relationships between various institutions are beginning to evolve.

For the purpose of opening the discussion in this session, I would like you to consider a computer software information system because it provides one basic means for resource sharing. The concept of a computer software information system for agriculture will evoke markedly different responses from the various members of this research committee. To be a little more specific, I will define the system to be a regional or national inventory of computer software developed by cooperating institutions. The system could be as sophisticated as an interactive, on-line retrieval system providing information on available programs and how to use them.

It could also be defined simply as a published catalog listing an inventory of available programs. Updates to the catalog could be provided on a monthly or quarterly basis. The information system could document the following major components of computer software development:

- 1) Problem Definition
- 2) Method of Solution
- 3) Program Design and Development
- 4) Program Verification
- 5) System Implementation Instructions
- 6) User Instructions
- 7) Sample Input/Output
- 8) Machine Operational Requirements
- 9) Other Appropriate Information

One goal of this committee could be to design the framework for implementing a computer software information system for agriculture. However, it is important for this committee to explore and evaluate various alternative approaches in which computer software resources can be shared on a regional basis.

SOFTWARE INVENTORY SUGGESTIONS

Gary J. Wells
Clemson University

- I. Explicit consideration to tie program inventory to CRIS.
- II. Develop a system that utilizes currently existing discipline classification systems (e.g., CRIS or those used by professional societies).
 - A. Make use of optical scan equipment to reduce entry problems of the agency administering the clearinghouse.
 1. Same scan form would be used to supply and request software information.
 2. The optical scan form would include
 - a. area of primary concentration
 - b. area of secondary concentration
 - c. topic area within primary concentration
 - d. topic area within secondary concentration
 - e. space for a one-paragraph description of software if the scan form is being used to supply software information to the clearinghouse
 - f. name and address of supplier or requestor of software information
 - g. space for a list of keywords
 - h. general software characteristics (e.g., language, K requirements, machine compatible, etc.)
 - i. date (if software is being supplied).
 3. Users guides would be provided to subscribing institutions.
 - B. Provide to requestors a list of programs and their authors based on a keyword and area of concentration search.
 - C. Require occasional updating of files based on entry date.

- D. System run on fee basis with nonsupplying requestors paying a premium.
- E. Suppliers of software listings would be expected to recover costs if software is requested. A small profit would not be discouraged. (Profit may encourage listings.)
- F. Clearinghouse would not directly provide software.
- G. Cooperating institutions may consider requiring all publicly used software developed within their institution to be supplied (i.e., the minimum requirement for receiving credit for work done).

WHERE DO WE GO FROM HERE?

R. Rodney Foil
Mississippi Agricultural and Forestry Experiment Station

Let me recap where I think we are and what commitments we have made. As a committee we have agreed to the following: (1) to provide our states individually and collectively with suggested procedures for selecting, inventorying, maintaining and distributing software; (2) to provide a mechanism for review and publication of software that will be acceptable as evidence of professional and technical accomplishments, utility and need; and (3) to design and establish a regional inventory and clearinghouse function.

It seems to me that the order in which these three things are listed would be the order in which we should proceed. The regional clearinghouse activity presupposes a peer review of publications. Once we have the peer review and publication process, then we know what we are going to inventory. I see these three initiatives as the ones that have been more or less accepted or defined for future involvement of this committee.

In terms of timetables, we must remain flexible until after the February meeting of the directors and administrators. I think we should have a first draft of the committee recommendations in these three areas for their consideration at that meeting February 9-11. We have put the mechanism in place for doing that. It will be a tremendous task writing back and forth and will possibly even involve a subcommittee meeting to capture the substance of what should be presented. We all have our own pictures of what is needed, and it is going to take some reflection and quiet contemplation before we put together what is really needed. We are close to being able to meet the first of these suggested guidelines and procedures.

Looking to the future, I suspect that following the February meeting we will be closer to a consensus among all of the land-grant functions as to what kind of regional organization is needed. I expect there will be something different that emerges from this particular research development committee. From the very beginning, we have been a unique committee in regard to representation and disciplinary outlook as well as functional responsibilities. We probably will have a regional, broad-based organization that will involve teaching, research and extension. Hopefully there will be specific functional kinds of committees that will be needed in the future, e.g., a research committee, an extension committee. But as I see it now, I would point towards this committee submitting more or less a final report and going out of existence about April. This would be consistent with a one-year assignment by the Regional Experiment Station Directors Association resulting in recommendations for a replacement kind of mechanism based on our deliberation. I would not pre-suppose what kind of replacement mechanism will be needed because that will depend on what the individual states decide to do.

Obviously, if this committee or a separate committee is the editorial board and/or review committee and serves the function of putting a stamp of approval on software, then we will need a different kind of title and have a method of appointment and continuity. There have been similar kinds of activities within the Southern region. One example is the soil testing group. It is a joint research and extension group which meets and more or less adopts standard ways to test soil to see whether or not it needs more nitrogen. This group has been in existence for 30 years. They are a quiet group. Many people do not know they exist but they have served a regional need. I see something parallel to that growing out of this activity. I also see the application of computer

technology through research and extension functions moving into being part of the normal course of doing business just like publishing or making speeches. It is a tool that we use in getting our work done. We will move from being a computer science group to a situation in which a computer science group exists within each of our other regional committees. We have a good deal of work to do to meet those three major challenges!

VI. APPENDIX

AGENDA

October 6, Wednesday

- 1:00 p.m. Introductions
 --Brief discussion of each state's activities
 and computer applications in research
- 2:00 p.m. Standards
 Discussion leader: Robert Sowell
- 3:15 p.m. Break
- 3:30 p.m. Peer Review
 Discussion leader: Wayne Mishoe
 Each state's position on professional credit
- 5:00 p.m. Adjourn

October 7, Thursday

- 8:00 a.m. Software Inventory and Clearing house
 Discussion leader: Glenn Ware
- 10:00 a.m. Break
- 10:15 a.m. Where Do We Go From Here?
 Discussion leader: Director Foil
- 12:00 noon Adjourn

PARTICIPANTS

October 6-7, 1982

Rodney Foil, Director
Mississippi Agricultural and
Forestry Experiment Station
Drawer ES
Mississippi State, MS 39762

Donald C. Huffman
Department of Agricultural Economics
Louisiana State University
Agricultural Administration Bldg.
Baton Rouge, LA 70803

Terry C. Keisling
University of Arkansas
P. O. Box 391
Little Rock, AR 72203

Wallace E. Killcreas
Mississippi State University
Box 5187
Mississippi State, MS 39762

William W. Linder
Southern Rural Development Center
Box 5406
Mississippi State, MS 39762

Rob Martin
Department of Agricultural Economics
Auburn University
Auburn University, AL 35830

Bryan Maw
Coastal Plains Experiment Station
Box 748
Tifton, GA 31793

Jim McGrann
Department of Agricultural Economics
Texas A&M University
College Station, TX 77843

John A. McGuire
Research Data Analysis
Auburn University
Auburn University, AL 35843

H. DelVar Peterson
USDA, Agricultural Research Service
P. O. Box EC
College Station, TX 77840

Bill Sanders
University of Tennessee
P. O. Box 1071
Knoxville, TN 37901-1071

Robert S. Sowell
Biological and Agricultural Eng. Dept.
North Carolina State University
P. O. Box 5906
Raleigh, NC 27650

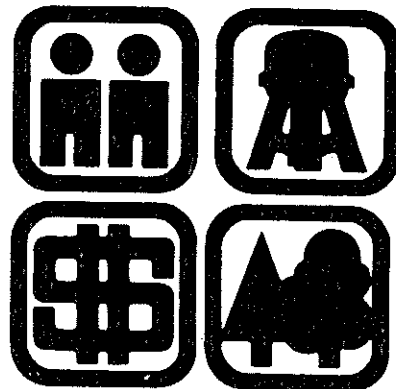
Alfred D. Sullivan
Mississippi State University
P. O. Drawer FR
Mississippi State, MS 39762

Tom Tice
Department of Agricultural Economics
Oklahoma State University
AGH 408
Stillwater, OK 74078

Glenn O. Ware
School of Forest Resources
University of Georgia
Athens, GA 30602

Gary Wells
Department of Agricultural Economics
and Rural Sociology
Clemson University
Clemson, SC 29631

The SRDC is one of four regional rural development centers in the nation. It coordinates cooperation between the Research (Experiment Station) and Extension (Cooperative Extension Service) staffs at land-grant institutions in the South to provide technical consultation, research, training, and evaluation services for rural development. This publication is one of several published by the Center on various needs, program thrusts, and research efforts in rural development. For more information about SRDC activities and publications, write to the Director.



**Southern Rural Development Center
Box 5406
Mississippi State, MS 39762**

The Southern Rural Development Center is an Equal Opportunity Organization providing research, educational information, and other services only to individuals and institutions that function without regard to race, color, sex or national origin. SRDC is an Equal Opportunity Employer.