

# GUIDELINES FOR TECHNICAL REVIEWS OF SOFTWARE PRODUCTS

MASTER

---

---

## Hanford Engineering Development Laboratory

HANFORD ENGINEERING DEVELOPMENT LABORATORY  
Operated by Westinghouse Hanford Company  
P.O. Box 1970 Richland, WA 99352  
A Subsidiary of Westinghouse Electric Corporation  
Prepared for the U.S. Department of Energy  
under Contract No. DE-AC06-76FF02170

## DISCLAIMER

**This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.**

## **DISCLAIMER**

**Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.**

## NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the U.S. Department of Energy (DOE), nor any of its employees, nor any of its contractors, subcontractors or their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for any third party's use or the results of such use of any information, apparatus, product or process disclosed in this report, or represents that its use by such third party would not infringe privately owned rights.

Printed in the United States of America  
Available from  
National Technical Information Service  
U.S. Department of Commerce  
5285 Port Royal Road  
Springfield, VA 22161

NTIS price codes

Printed copy: A 03

Microfiche copy: A01



# GUIDELINES FOR TECHNICAL REVIEWS OF SOFTWARE PRODUCTS

DISCLAIMER

This book was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

HEDL-TC--2132

DE82 013411

---

## Hanford Engineering Development Laboratory

N.P. Wilburn  
March 1982

**HANFORD ENGINEERING DEVELOPMENT LABORATORY**  
Operated by Westinghouse Hanford Company  
P.O. Box 1970 Richland, WA 99352  
A Subsidiary of Westinghouse Electric Corporation  
Prepared for the U.S. Department of Energy  
under Contract No. DE-AC06-76FF02170

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

248

**THIS PAGE  
WAS INTENTIONALLY  
LEFT BLANK**

GUIDLINES FOR TECHNICAL REVIEWS  
OF SOFTWARE PRODUCTS

N. P. Wilburn

ABSTRACT

*A guideline is given for technical review of products developed during a software life cycle. Purposes and benefits of reviews are given. Varieties of reviews, when they should take place, roles of the reviewers and products of the review are described.*

**THIS PAGE  
WAS INTENTIONALLY  
LEFT BLANK**



CONTENTS

	<u>Page</u>
Abstract	iii
Figures	vii
1.0 INTRODUCTION	1
2.0 PURPOSES OF A REVIEW	2
2.1 STAFF IMPROVEMENT	2
2.2 TEAM BUILDING	2
2.3 OTHER STAFF BENEFITS	3
2.4 PROGRESS MEASUREMENT	3
2.5 SOFTWARE QUALITY IMPROVEMENT	3
2.5.1 Error Removal	4
2.5.2 Other Improvements	5
2.6 ADDITIONAL BENEFITS	6
3.0 GUIDELINE DESCRIPTION	7
3.1 GUIDELINE APPLICABILITY	7
3.2 VARIETIES OF REVIEWS	7
3.2.1 Walkthroughs	9
3.2.2 Big Room Reviews	9
3.2.3 Inspections	10
3.2.4 Speed Review	10
3.2.5 Round Robin Reviews	11
3.2.6 Peer Rating	11
3.3 WHEN REVIEWS SHOULD TAKE PLACE	12
3.4 REVIEW TEAM	14
3.4.1 General Principles of Selection	14
3.4.2 Representation	15
3.4.3 Review Team Roles	15
3.4.4 Functional Representation	18

CONTENTS (Cont'd)

	<u>Page</u>
3.5 CONDUCTING THE REVIEW	18
3.5.1 Basic Principles	18
3.5.2 Preparation	20
3.5.3 Review Materials	21
3.5.4 Mechanics of the Review	21
3.5.5 Successful Reviews	25
3.6 PRODUCTS OF THE REVIEW	25
3.6.1 General	25
3.6.2 Summary Report	26
3.6.3 Issue List	26
3.6.4 Related Issues Report	28
3.6.5 Decisions/Action Taken in the Review	28
3.7 POSSIBLE PROBLEM AREAS	29
3.8 MANAGEMENT RESPONSIBILITIES	30
3.9 MANDATORY ITEMS	31
4.0 BIBLIOGRAPHY	32

FIGURES

<u>Figure</u>		<u>Page</u>
1	Software Life Cycle and Development Products	8
2	Technical Review Summary Report	27

GUIDELINE FOR TECHNICAL REVIEWS OF SOFTWARE PRODUCTS1.0 INTRODUCTION

The purposes of this guideline\* are to describe the technical review process for products developed during a software life cycle, to identify those elements which are of key importance and to provide a procedural framework for carrying out the technical review.

The identification of defects in a product is an accepted, expected part of the software development cycle. No product is perfect; no product is expected to be. The purposes of a technical review are to improve a product (which can be any one of a number of elements created during the software development process) by identifying issues (defects, errors, omissions, ambiguities, inconsistencies, or incorrect assumptions), and to identify these issues as early as possible in the development process. Technical review of a product by a well-prepared group of the producer's peers is one of the most effective methods identified to date for finding defects and for assuring adherence to the product's requirements and established guidelines.

---

\*This guideline is primarily based on "Ethno-Technical Review Handbook," by Daniel P. Freedman and Gerald M. Weinberg and the book "Structured Walk-throughs," by Edward Yourdon, but includes information obtained from many other software engineering references which are listed in the Bibliography. It is suggested that participants in the review process read both of the above books for a more comprehensive explanation of the ideas and techniques which are presented here.

## 2.0 PURPOSES FOR A REVIEW

In addition to the stated purposes for the review, there are a number of additional benefits which result directly from the process.

### 2.1 STAFF IMPROVEMENT

A major improvement which almost always occurs is in the overall staff performance. Since all technical members of a project team from the most senior to the most junior should have their products reviewed, the benefit is obtained throughout the staff. Participants in the software development process collectively improve their techniques and styles as they share concepts at the review. Not only do the junior people learn techniques from the senior people, but the senior people often get ideas and advice from the junior people. Thus, people who regularly participate in technical reviews will have many more and diverse opportunities to learn from other people than does an isolated software developer. It has been shown that a programmer who regularly participates in formal and/or informal reviews of his own work and of the work of others gains experience at the rate of perhaps three times as great as a programmer working in solitude.

### 2.2 TEAM BUILDING

The review process also promotes team building. It becomes one of the first steps toward establishing a good development team, by substituting for an environment where programmers work alone throughout their career, a programming team environment in which each individual feels free to discuss and critique everyone else's program. Implicit in the concept of a team is the notion of working closely together, reading each other's work, sharing responsibilities, learning each other's idiosyncracies both on a technical and personal level, and accepting altogether as a group shared responsibility for the product where each member can expect similar awards if the project is a success and similar penalties if the project fails. If the attitude of a team can be instilled, the effect is usually one of synergism -- that is, five people working together on a team may produce twice as much as they would individually.

### 2.3 OTHER STAFF BENEFITS

In addition to the staff training and team building benefits of the review process, there are others that have been identified. It has been shown that the review process tends to increase the staff productivity in software development, and that the morale and general spirit of the staff increases significantly as a result of the review process. Also, better professional attitude, reduced turnover and more accurate estimating and scheduling are seen.

### 2.4 PROGRESS MEASUREMENT

Without reviews, there is no reliable method for measuring the progress of a product. The review serves as a formal commitment by technically competent and unbiased people that a piece of work is indeed complete, correct and dependable. It is the "rite of passage" of a product in the software development environment -- the review process marks the change from one status in the software life cycle to a more advanced one. With reviews required, there is no way the project can slip unnoticed from, for instance, the design stage to the coding stage, as was often the case in the past. Each module will be incorporated into the system based on a review and each document will enter the same way. Nothing will become part of the overall system until it has passed a review.

### 2.5 SOFTWARE QUALITY IMPROVEMENT

It has been stated that the review process is one of the most effective ways known to improve the quality of a piece of software. This comes about for a number of reasons, but one of the principal reasons is simply that the expectation of a review process tends to improve the quality of a software product (design, specification, or test data). Clearly, one does not want to risk embarrassment among one's peers and will thus do a far superior job than would be done in isolation.

In addition to this, there are other reasons for the improvement and the quality of the software.

### 2.5.1 Error Removal

Walkthroughs or reviews have been found to be highly successful in producing reliable, bug-free programs, reducing the number of errors in a production program by as much as a factor of 10. It may well be true that if the claim of error free software is valid, it is due more to peer code review than any other element of the structured programming or software engineering methodology. Thus, one of the major purposes of a walkthrough, or review, is not the major purpose is to find errors or bugs in the product.

Why does the review process work for error detection as opposed to a software developer working in isolation? The author of any product, especially the author of a computer program, has a number of mental blocks which prevent him from seeing errors in his product as quickly as a group of peers. In addition, he may not be motivated to invent test cases that will demonstrate that his product is wrong. Instead he is motivated to demonstrate that his product works. If, for example, a programmer forgot to check for exceptional conditions in his code then he'll probably forget to invent a test case which will cause the exceptional condition to occur. Whereas, the psychological orientation of a walkthrough is highly conducive to error detection, the reviewers have little emotional stake in the product under their review so they are less inclined to cognitive dissonance (which is the failure to notice what they don't want to notice). In addition, simply the process of the programmer explaining the operation of his program or the producer describing his product under review causes him to discover flaws that had previously eluded him.

A number of benefits result from finding errors in software products during the review. When bugs are found in the design or specifications/requirements walkthroughs, some recent studies have indicated that it is roughly ten times less expensive to fix errors detected at this phase of the project than it is to fix them after the code has been written. It follows then, that finding errors by an inspection/review process and reworking them earlier reduces the overall rework time and thus increases the productivity



over the whole process as well as reducing maintenance substantially. During a code review process, a whole module or a substantial part of the code is considered at one time and several bugs or errors will probably be identified. This results in a definite cost improvement when the turnaround time for a typical batch operation is considered. Often a programmer will have to wait several hours for his program to compile or for a test run to be complete (sometimes even overnight). When the process is done this way, and a single error at a time is found, the time to complete removal of the known errors can be substantial as compared to the review process.

### 2.5.2 Other Improvements

Several other improvements in quality also occur as a direct result of the review process. For instance, it has been demonstrated that typical programs which have been thoroughly reviewed during development and for which maintenance changes are also reviewed, show a five-to-one reduction in maintenance cost. Reviews and walkthroughs can help spot gross inefficiencies in the design or implementation of the program as well as design strategy or coding techniques that would seriously detract from the maintainability of the program. In general, group reviewing increases the product reliability, maintainability, adherence to standards, and adherence to schedule, while encouraging the social interaction, training and morale improvement described above.

Once reviews are established and producers understand that their work inevitably will be reviewed, they tend to bring it out earlier for public scrutiny on a more informal basis. As a result, the errors and deficiencies in design are identified even earlier and the product can therefore be improved at a lesser cost.

Finally, it should be noted that very few one-person projects remain with one person forever. Sooner or later the single producer moves on to bigger and better things and his software system must be turned over to someone else. (At this time it probably would be good to again have a total system

review regardless of the system history. A fundamental objective of the review process is to broaden the knowledge of the products and thus change programs from private masterpieces to a corporate asset and, hence, minimize the chance of throwing away a piece of software if someone leaves the project.

## 2.6 ADDITIONAL BENEFITS

Some additional qualitative benefits identified in the Bibliography and attributable to use of the review process are:

- More reliable estimating and scheduling,
- Increased user satisfaction,
- Decreased debugging time,
- Increased program readability.

### 3.0 GUIDELINE DESCRIPTION

This guideline provides a brief, high-level description of the technical review process. It outlines the responsibility of the participants and the tasks which must be performed, but deliberately avoids detailed discussions on exactly how each task should be performed or what review process should be used. That type of detail varies with circumstances, and is, therefore, left to the judgment of the participants.

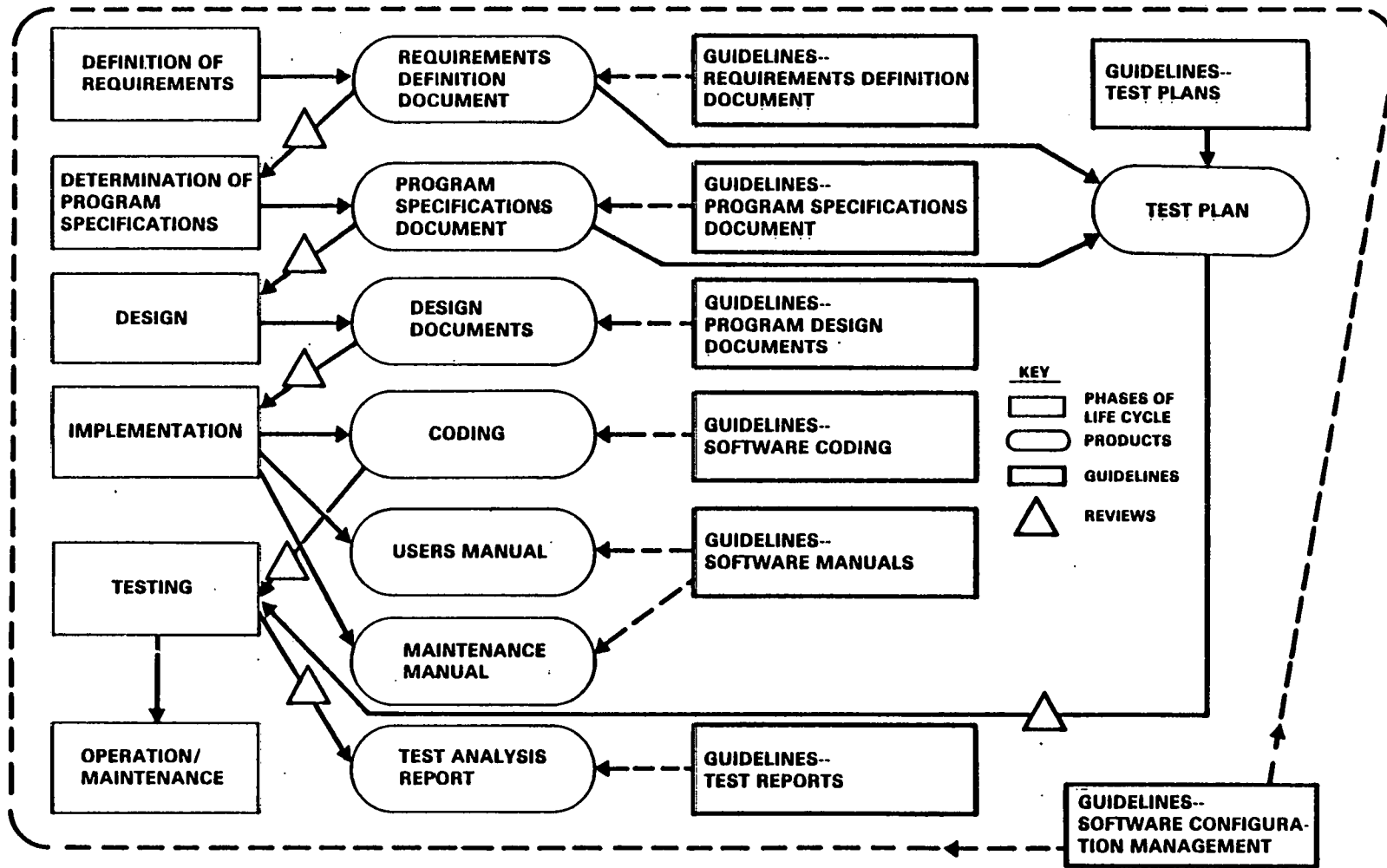
### 3.1 GUIDELINE APPLICABILITY

Technical reviews are applicable to both the end products in each phase of the software life cycle as shown in Figure 1, but are certainly also applicable to any subproducts within a given phase. As illustrated in Figure 1, each product is judged during a review in two ways: 1) it must be consistent with the preceding documents (products) in the chain; 2) it must also satisfy the appropriate standard or guideline. (Until the standards/guidelines shown in Figure 1 are developed, the technical review process should rely on the judgment and experience of the development staff.)

Considerations of practicality indicate that formal technical reviews should not be used for small changes to the computer code, or design, or specifications. New codes, or substantial changes to an existing product, do require a formal process, but small changes need only an informal review by one or more other persons. This guideline will not attempt to define the boundary.

### 3.2 VARIETIES OF REVIEWS

The review process itself can take place in a number of ways. There are walkthroughs, inspections, big room reviews, speed reviews, round robin reviews, and peer ratings. These differ in the roles of the various participants, their responsibilities, the output produced, and input required. Each of these types of reviews can be formal or informal. The basic difference between a formal and informal review is that a management summary



HEDL 8203-076

FIGURE 1. Software Life Cycle and Development Products.

(Section 3.6.2) should be issued in the former case. Which kind of review should be used for which product is at the producer's discretion. However, it seems from references given in the Bibliography that the "Walkthrough" has the most general applicability.

### 3.2.1 Walkthroughs

A walkthrough is a generic name given to a series of reviews of software products each having a different objective and each occurring at different times in the software development cycle. It can be described as simply a peer group review of any product generated during the software life cycle where people at roughly the same level in the organization meet to review and discuss the product. It is predominately a tool for analyzing the functional design or coding of a developing system or associated documentation. It permits the critical review of these elements against the system specifications or the intended function by members of the project team. The primary purpose of the walkthrough is to detect errors in the logic of the program design or code or the documentation and to eliminate the errors before they are formally established.

There are a number of variations of walkthroughs all of which are characterized by the producer being present and guiding the progression of the review. In its usual form, the walkthrough is a stepwise simulation through the designed software procedure, for example, as when walking through a code line-by-line with the simulation of inputs or a section-by-section discussion of a document.

### 3.2.2 Big Room Reviews

The big room review is often used and can be exemplified by a design review. All of the design documents are posted on the walls of the room in some meaningful sequence. Participants are "locked" into the room to wander about studying the system in any manner they choose. Participants freely discuss the issues with one another with no attempt to have everyone in on the

same discussion. The circulation of the individuals probably should be combined with some explicit assignment to the participants, such as identifying the product's features or errors, by use of some form of checklist.

Another variant of the big room technique is to make a simultaneous review of several alternate designs or specifications or other product. As in an art show, the reviewers circulate freely, examining the alternatives in any order or manner that they desire. At another follow-on session it probably would be beneficial to have a walkthrough of the same materials presented in the big room review. At that later session, the participants would be thoroughly familiar with the material and additional problems or errors or issues can be identified and recorded in the manner described in Section 3.2.1.

### 3.2.3 Inspections

An inspection is a method of rapidly evaluating the material under review by confining attention to a few selected aspects, one at a time. Inspections are a commonly used form of peer review. The key feature of an inspection is that it is driven by the use of checklists to facilitate error detection. An inspection should, therefore, be short because they are often quite intensive. Each participant should be informed of the primary area of concern -- standard adherence, modifiability, portability, coding errors of certain types and so forth. When possible, the participants should be supplied with a checklist of the items for which to be on the lookout. Before the inspection starts, each participant should at least be familiar with the overall structure of the product so that no time is wasted.

### 3.2.4 Speed Review

In a speed review, the work under surveillance is divided into equal parts in some fashion, after which each individual in the review team spends a short time -- say three minutes -- studying each part and making notes of issues identified. At the end of this interval, the work units are rotationally passed around the room with sufficient intervals being scheduled to give the

desired level of redundancy. For instance, suppose it is desired to find out if a particular module is easily understandable for maintenance purposes. The module is divided into pieces and speed reviewing takes place on each piece. In this manner, the first impressions of a half a dozen people can be rapidly collected.

### 3.2.5 Round Robin Reviews

A round robin review is a peer review where each participant is given an equal and similar sized share of the product being reviewed to study, to present, and to lead in the evaluation. Round robin reviews emphasize a cycling through by the various participants, with each person taking an equal and similar share of the entire task. Round robin reviews are especially useful in the situations where the participants are at the same level of knowledge; a level which may not be too high. For the most part, round robin reviews are an excellent way of raising the knowledge level of a relatively inexperienced group, especially one that lacks even one strong person who can teach the others.

Though it is practiced in different installations under a variety of names, the most common round robin review is similar to the child's game of musical chairs. The missing chair is the leader's chair which each person takes a turn occupying. Another method of division of the effort is based on the idea of feature chiefs. This is similar to the above, but the division is made on the basis of the function, rather than form. The various features in a specification are allocated among the participants, each of whom must demonstrate how the work at hand contributes to this feature or leaves that feature unaffected.

### 3.2.6 Peer Rating

Peer rating is a technique of evaluating anonymous programs in terms of their overall quality, maintainability, extensibility, usability and clarity. The purpose of the technique is to provide a programmer with some degree of



evaluation. Each participant is asked to select two of his own programs to be reviewed. One should be representative of what he considers to be his finest work, the other should be a program the programmer considers to be rather poorer in quality. Once the programs have been collected, they are randomly distributed to the participants. The evaluation form is then provided which asks the reviewer to answer, on a scale of one to five (one meaning definitely yes, five meaning definitely no), such questions as:

- Was the program easy to understand?
- Was the high level design visible and reasonable?
- Was the low level design visible and reasonable?
- Would it be easy for you to modify this program?
- Would you be proud to have written this program?

The results of the review are then collected and an overall average across the participants reviewing each piece is computed. From this, the producer of the item can obtain a good idea of the evaluation by his peers of the specific product quality.

### 3.3 WHEN REVIEWS SHOULD TAKE PLACE

The resulting product from each phase of the software life cycle (shown in Figure 1) should correspond to a key goal and must be reviewed for its consistency with previous documents and products prepared during the course of the software development life cycle. The product also must satisfy the requirements established by the appropriate guideline or standard. For example, a software "Design Document" must provide for all of the functionally mandated requirements established in the "Requirements Document", as well as the functions established in the "Functional Specifications Document" for the code. It also must meet the guidelines established for design documentation.

In addition, since reviews are needed to establish quality for all products, there should always be something to review, whether formally or

informally. Whether it is standards, procedures, specifications, design, coding with listing, documentation, test plans, test results: all should be reviewed.

The review should be scheduled late enough so that the various products are sufficiently well developed and well documented so some sense can be made of them. Yet reviews should be scheduled early enough so that the producer of the product does not invest too much of his ego or time in the product, and that there will be some bugs and improvements to find in the product.

As a basis for estimating how much time and effort should be spent in the review process, it has been noted that approximately 2 to 10% of the total labor allocation for the software lifecycle should go into the formal technical reviews. If there is doubt, one should be conservative and select the 10% figure. Whatever figure is selected, it must have full management support.

In addition to the formal reviews, a number of informal reviews also can, and should, take place during the process of the software life cycle. For instance, a code walkthrough can be held in any of the following states: before the code is key punched or entered on a terminal; after it is entered on a terminal, but before it is compiled; after the first compilation; after the first clean compilation; after the test case has been executed successfully; or after the programmer thinks all test cases have been executed successfully. At other times, where testing identifies an excessively error-prone code, it may be quite economical and saving of schedule to select the most error-prone module and inspect it by means of a walkthrough before continuing the testing process.

Reviews can also be done in other parts of the life cycle, such as in the maintenance environment. If the product is large (i.e., a program containing many modules or thousands of lines of code) such a review is likely to have all of the benefits that were previously described, especially if the original product was developed in a nonstructured fashion several years ago. What appears to a maintenance programmer to be an innocent change to a "rat's nest"

system may have subtle ramifications in other parts of the system. Reviews will have the advantage of minimizing the nasty surprises that make maintenance programming such a pain. Also, experience has shown that modifying an existing program is a very error-prone process in terms of errors per statement written, more so than when writing a new program. Hence, all program modifications should be subjected to a review and a thorough testing process.

Another time where reviews are helpful is during a training phase for use of a system. A number of systems which have looked good on paper have failed because the people involved were not adequately trained in their use. This is less likely to happen when training materials are explicitly reviewed as part of the system development process.

### 3.4 REVIEW TEAM

#### 3.4.1 General Principles of Selection

The following general principles may be stated for selection of participants in the review process:

- Select reviewers to assure that the material on review is adequately covered.
- Nobody should be in the review who lacks the qualifications to contribute to the review.
- Avoid using technical reviewers who themselves are "above" review.
- A very effective approach is to select a team of reviewers for each project at its inception. This type of review team brings in management continuity to the project as they move from the specifications through the design reviews to code and testing reviews. They can then develop the effectiveness of the team, yet retain some of the broad views and objectivity of outsiders.

- If a participant feels that he will be unable to devote the time to review the products, it is better that he not participate rather than perform a superficial review.
- The minimum review group should have at least four present and these should not all be good buddies.

#### 3.4.2 Representation

Participants in a formal review are often selected from the following types of groups of people:

- Project manager
- Project technical leader
- Other project team members -- analysts, designers, programmers
- Client, or user
- Outside reviewers -- quality assurance personnel, or experienced people from other projects
- Others -- configuration management representatives, maintenance representative.

#### 3.4.3 Review Team Roles

The review team should include at least the following four roles: a review leader, a recorder, the producer and one or more additional reviewers. Generally, the size of the team should not exceed seven. In addition, across the four roles, a number of different functions could be represented, such as a user representative, or a standards bearer. The responsibilities for each of the four roles are described below as well as the responsibilities of the functions representative.

#### 3.4.3.1 Review Leader

The review leader is usually selected by the project manager. The review leader must not be the producer for reasons of objectivity. The primary responsibility of the review leader is to obtain a good review; one which produces an accurate assessment of the product, or to report the reasons why the good review was not obtained. Other responsibilities of a review leader include:

- Selection of review team members, including the recorder;
- Format and content of the review (see Section 3.5 below);
- Discipline during the review -- keeping the meeting on track, restricting discussion to finding problems rather than fixing them; separating technical issues from management and political issues, limiting discussions of style, verifying participation of the participants;
- Logistics -- distributing the documents such that the participants have at least two working days to prepare for the review; setting up the meeting place and time, notifying the participants, and providing the supplies (easel, pens, tape).
- Serve as a moderator during course of the review.

#### 3.4.3.2 Review Recorder

The principal responsibility of the recorder is to maintain an accurate record of the issues raised. The list should be public, preferably on an easel. The recorder is also responsible for the distribution of the issues list and the summary report described in Section 3.6 below. The recorder is also responsible for helping the review leader control the meeting by keeping the discussion focused on the identification of issues. The recorder can

short-circuit long-winded participants by asking for a short statement of the issue; likewise, the recorder can terminate the design discussions by asking that the issue be identified. He should record all errors identified, discrepancies, exposures, and inconsistencies that are uncovered during the walk-through review so that nothing that will be needed later is forgotten.

#### 3.4.3.3 Reviewers

Generally, all members on the review team, including the leader and the recorder, will also function as reviewers. Reviewers have several responsibilities:

- Advance preparation is absolutely essential. Adequate time must be budgeted into project schedules by management and the reviewer must use that time to give the product a thorough examination;
- Self-discipline -- keep the meeting on-track; keep the remarks concise; stick to finding problems; avoid solutions; stick to technical issues, avoid political issues, review the product, not the producer; bring out the positive as well as the negative; arrive at the review on time.

#### 3.4.3.4 Producer

The producer has the responsibility at the beginning of the meeting to present an overview of the product, placing it in its appropriate context. However, the producer does not run the meeting. The producer primarily attends in order to get a first-hand understanding of the issues and to provide information initially and when asked. The producer is responsible for the resolution of items on the technical review issues list (see Section 3.6 below).

#### 3.4.4 Functional Representation

In addition to the four roles that have been identified above, a number of functions should be represented. There should be a reviewer who can be called the maintenance oracle. This function is an obvious one, to review the product from the viewpoint of future maintenance. Another important function is that of a standards bearer. As the name implies, the person who plays this role has a primary concern with the adherence to standards -- programming standards, design standards and any other kind of standards or guidelines that have been defined in the software engineering guideline sequence. There should also be included a user representative, i.e., someone who can ensure that the product is meeting the customer's needs. The user should be expected to play a major role in the specification review and possibly the design review, and he definitely could play a constructive role in test reviews.

### 3.5 CONDUCTING THE REVIEW

#### 3.5.1 Basic Principles

A number of basic principles can be enumerated for conducting a technical review.

- The review involves the time of four to seven highly paid people. Their time should not be wasted; indeed the review should be run as if every minute was precious.
- Everyone in the review must agree to follow the same set of rules and procedures.
- Participants must agree and understand that it is the product being reviewed -- not the producer.
- Experience has shown that reviews are most successful when the individual with responsibility for directing the review is knowledgeable about the process and its intended results.



- The responsibility for the quality of the written review should be shared among all participants.
- A technical review should have active and open participation of all participants under the direction of an experienced group leader.
- The review process itself is to raise issues, not resolve them. Thus, the producer should limit his response to any of the review team comments during the review process to asking for clarification of a reviewer's comment. Once the issues list is generated, a group of interested individuals can then form a working group to address the specific problems.
- Most reviews are not attended by management (see Section 3.8). The presence of management tends to inhibit participants since they feel they are being personally evaluated. This is contrary to the intent of the review, which is studying the product itself. The presence of management also interferes with a frank open exchange of views between peers. However, there is a situation where the manager may feel strongly that his presence in the review is required. For instance, he might attend if he has considerably more technical knowledge and experience than the producer or any of the other participants. A solution for this dilemma is to let the manager attend a second review of the product. In other words, let the producer and participants have a review by themselves first, in which they will find many of the "stupid" errors that they wouldn't want the manager to see. Once they are satisfied with the product, then the manager can conduct a more formal review.
- Attitude is most important among participants. Those who cannot be cooperative and tactful and nondefensive while striving for technical excellence should be excluded from the reviews. It should be noted here that an excellent phrase that has stood the tests of hundreds of emotionally charged reviews is, "I don't understand -----".

This should be uttered in a sincere tone of voice. This phrase begins with the humble and quite likely true assumption that the difficulty rests in the reviewer's own head rather than in the product. Also, the statement "watch your language" should be adhered to. Be humble and avoid, if possible, critical, negative phrasing.

- At least one positive and one negative statement should be made by each participant at the outset.
- Discussions of stylistic issues should be limited except where dictated by guidelines (questions of style can be handled on an individual basis outside of the review itself).
- Remember that education of junior members is a goal of the review process.

### 3.5.2 Preparation

The first and foremost requirement of the review process is the advanced preparation by everyone involved. At least 80% of all review failures can be traced back to lack of adequate preparation. Therefore, the review leaders should be constantly monitoring to ascertain how well advance preparation was carried out. Participants should expect to spend approximately one hour or more reviewing the documentation for a review that will last approximately thirty to sixty minutes. A review is meant to be a short highly technical meeting. It is a waste of everybody's time if people in the review are learning the material for the first time during the actual review process. (Note the exceptions above in Section 3.2 which are specifically pointed toward the learning process.) Therefore, the material must be distributed ahead of the review a reasonable length of time but not so far ahead so as to be lost or forgotten. This allows the participants to spend the requisite amount of time reviewing the materials to become better prepared for the review. Finally, in the course of the preparation for the review, the

location and time of the meeting must be established, the participants identified and the particular agenda planned.

### 3.5.3 Review Materials

The materials for the review process should include the items identified in Figure 1 (the appropriate standards and the product from the particular phase, such as requirements or design specifications or program listing), the summary report form, backup documentation, any checklist for the review of the particular phase, and any comments from past reviews. In addition to these items required for all reviews, some other items are optional. For instance, a flip chart which can be displayed in full view of the participants can prove to be very useful. Notes are written on the charts by the recorder which capture the essence of each issue. With flip charts, the filled page may be ripped off and attached to a wall with masking tape. Some organizations get the same effect, more or less, by using view-foils with an overhead projector. One advantage of this method is that the view-foils may be copied on ordinary copiers, something that cannot be done with flip charts. However, the view foils have the disadvantage that it is not possible to post previous sheets for public scrutiny which is possible with the flip charts.

Another possibility is the use of a large screen video projection system tied to a time-shared text editor. The recorder then sits at a terminal in the review room and types the issues directly into the text processor. Some advantages are that the typing is faster and clearer than writing and everyone can see exactly what has been written on the large screen. Changes can be made instantaneously and easily because of the text processing capability. Then later, the review report can be constructed more easily.

### 3.5.4 Mechanics of the Review

A suggested agenda for the review meeting should be as follows: first, the producer should give a brief tutorial overview. Next, the reviewers should comment on the completeness, accuracy and general quality of the

product; after all the comments have been made, the recording secretary circulates the issues list among the attendees (if a flip chart or other common media is not used) so that they can make sure that all their comments have been noted; finally, the review team makes sure that all disputed points are resolved. The results of all actions taken by the producers should be communicated to the reviewers. This may or may not require another walkthrough.

Some specifics about the mechanics of the review follow:

- 1) A review will typically include three to five people and last a pre-specified amount of time (usually one or two hours at the maximum). The two hour limit is really built into human beings. Adults generally can handle no more than two hours in a review. This is a simple psychological fact about human beings in our culture, not an arbitrary limit. In fact, even the two hours should be relieved by a ten minute break in the middle so that people can get their second wind.

If it appears that a review threatens to go beyond the capacity of the review team, its organization may be broken down into parts by one of two means: either by form or by function. Breaking a review down by function is easiest to understand. If the product is too big to review at one sitting the product is simply decomposed into pieces that can be reviewed. Breaking down by form is not so obvious. In essence, different points of view on the same material are sought. For instance:

- Conformity to standards can be reviewed separately from conformity to the specifications.
- Coding and other documentation can sometimes be reviewed separately.
- Efficiency can be reviewed separately from other specifications.
- User interfaces can be reviewed separately.
- Maintainability can be reviewed separately from conformity to functional specifications.

- Operating convenience may be a separable review item.
- 2) To avoid wasting time, each participant should give to the producer and the recorder a list of errors that require no explanation, for example syntax errors in the program listing or spelling errors in the specification. Each reviewer then presents his comments about the product, making sure that his comments are directed at the product rather than the producer.
  - 3) Review leaders often have difficulty enforcing time limits on speechifying. The difficulty is in interrupting somebody who is all wrapped up in a presentation but sometimes it has to be done to keep the review flowing. A good technique for timid leaders (or even aggressive ones) is to use a small alarm clock to set time limits on each speech.
  - 4) It is sometimes useful to adopt the concept of the devil's advocate. One person is assigned the task when the review materials are passed out of making the strongest case against the product. The devil's advocate doesn't have to be fair, compassionate or nice in any way, but it is understood that it is the devil speaking and not the person, so that no stigma is attached to having an argument produced that didn't hold water. The rest of the review group must convince themselves that on each negative point raised by the advocate "is this a real issue or not?" Most often the tactic is found in design reviews or in code reviews where major design decisions have been left to the programmer producing the code.
  - 5) "Bebugging" is a way of testing the reviewers as they are testing the product. The usual bebugging procedure is for the producing team to leave some number of known problems in the product submitted for review. Then a list is made of these known problems and it is sealed in an envelope which is placed on the table in plain sight as the review begins. When the review is finished the leader opens the envelope and compares the producer's list with the issues list generated by the review. If the reviewers have caught all the issues, there is some evidence the review

was thorough. If some issues went undetected or were dismissed as not serious, then the quality of the review may be questioned.

- 6) It is beneficial that each participant have at least one positive and one negative remark. This ritual may seem silly but actually it serves a number of useful functions. By insisting that each participant make a negative comment the leader ritualizes criticism. Once the ritual nature of criticism has been established no person needs to feel that making negative comments is impolite.
- 7) If someone is being cut off or shut off by others, the leader must take explicit steps. One way is to ask that the meeting take one point from each person in rotation. If one person has more points to raise than others, the excess points can be brought up after everyone else is finished. The leader may also ask that each person in rotation give their most important point and then each successive rotation will take the next most important point, so that if there are too many points to consider, at least the most important ones will be identified and included in the issues list.
- 8) The leader must have complete control of the review. The only way the group can avoid risk of turning the review into a noisy brawl is to agree ahead of time that they will respect the leader's role and they will stop arguing if he asks them to do so.
- 9) During the review the list of problems and questions is recorded. This issues list (described in Section 3.6.3) is distributed to all participants and is used by the producer as the basis for subsequent changes to the product.
- 10) Finally, when the review has finished, the leader should ask the group for their recommendations concerning the product that has been reviewed.

### 3.5.5 Successful Reviews

The success of the technical review process is dependent on these key elements:

- Strong Management Support of the review process.
- The reviewers must arrive well prepared. Most of the effort in the review process is expended prior to the review itself, as the reviewer studies the material to be reviewed, questions its assumptions, considers its ramifications, and looks for defects and ambiguities.
- The review must be limited to identification of issues. The purpose of a technical review is to find problems, not to fix them. A review is a problem-finding session, not a design or coding session. The exercise of creativity is the privilege and responsibility of the Producer, not the review team.
- The identified issues must be documented (see Section 3.6) for later follow-up. A list doesn't forget.

## 3.6 PRODUCTS OF THE REVIEW

### 3.6.1 General

All products of a formal review should be filed. Most organizations believe that the detailed comments from the review plus a copy of the management summary and the review issues list should be filed along with the official copy of the product or document itself. Also, the leader has the responsibility of delivering a copy of the summary report and issues list to each of the participants in the review.



### 3.6.2 Summary Report

The summary report is a report to management describing the results of a formal review, the problems revealed, proposed solution and any upper management assistance required. The summary report must identify three items: what was reviewed, who did the reviewing and what was their conclusion. The summary report is prepared by the recorder at the end of the meeting using the report form shown in Figure 2. The signatures of the reviewers indicating agreement with the report signify that they share with the producer the responsibility for the quality of the product. If the review team cannot agree on an acceptance or rejection of the product, the review leader will document the reasons for disagreement in the summary report and will request a decision from the appropriate manager. Copies of the summary report are distributed by the recorder to all participants of the review, the producer or producers, the project manager and appropriate unit subsection managers. The project manager will add a copy to the project file as evidence of the review and its result.

### 3.6.3 Issues List

The issues list is primarily the report to the producers that tells the producers why the work was not fully accepted by the review committee as is, hopefully in sufficient detail to allow the producers to remedy the situation. It is simply a list of all the technical issues raised during the review. An issue is any indication (controversial or not) that the document, product or code will not do the job as it is supposed to. Typical issues are defects, errors, omissions, ambiguities, inconsistencies and incorrect assumptions. These issues were raised and thought important by a "jury of the producer's peers" not by someone whose opinion they don't value. The recorder is responsible for recording the issues, identifying who raised the issues, having the issues list typed, and having it distributed to the review team and the producers within one working day of the review. The distributed list should agree closely with the list prepared during the review. Thus it is important that the original list be understandable.

DATE: \_\_\_\_\_ START TIME: \_\_\_\_\_ END TIME: \_\_\_\_\_

MATERIAL REVIEWED: \_\_\_\_\_

PRODUCED BY: \_\_\_\_\_

BRIEF DESCRIPTION: \_\_\_\_\_

PARTICIPANTS:

NAME:

SIGNATURE:

LEADER: \_\_\_\_\_

5. \_\_\_\_\_

6. \_\_\_\_\_

7. \_\_\_\_\_

APPRAISAL OF MATERIAL REVIEWED:

ACCEPTED (no further review)

NOT ACCEPTED (new review required)

\_\_\_\_\_ AS IS

\_\_\_\_\_ MAJOR REVISIONS

\_\_\_\_\_ WITH MINOR REVISIONS

\_\_\_\_\_ REBUILD

\_\_\_\_\_ REVIEW NOT COMPLETED

\_\_\_\_\_ NO DECISION

REASON: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

SUPPLEMENTARY MATERIALS PRODUCED:

DESCRIPTION AND/OR IDENTIFICATION

\_\_\_\_\_ ISSUES LIST

\_\_\_\_\_

\_\_\_\_\_ RELATED ISSUES LIST

\_\_\_\_\_

\_\_\_\_\_ OTHER

\_\_\_\_\_

\_\_\_\_\_

FIGURE 2. Technical Review Summary Report.

The issues list is not distributed to management or the project manager unless a copy is specifically requested. Note that an issues list is generated whether the product is accepted or not by the review team. If the product is accepted, the producer is responsible for following up on the issue list on his own. If the product is not accepted, it will be re-reviewed later, preferably by the same team and the old issues list must be supplied to the reviewers as input to the second review. The job of the review team ends with the identification and documentation of the issues. Resolution of issues is the responsibility of the producer. If the product fails because an issue wasn't properly resolved, the responsibility lies with the producer. If the product fails because the problem wasn't properly identified, the responsibility rests with the review team.

#### 3.6.4 Related Issues Report

This is a list of issues that arise during the review that are either nontechnical or not directly related to the product being reviewed. This list is prepared by the recorder during the review and distributed by the recorder (in typed form within one working day) to the review team, to the producers and the project manager. The project manager then is responsible for resolving these related issues or for communicating the issue to someone who can. If it isn't communicated to the latter in some standard official form they may not even recognize it. Therefore, if the related issue is not going to pass directly into a wastebasket it must be given some official status.

#### 3.6.5 Decisions/Action Taken in the Review

In the "Appraisal of the Material Reviewed," (see Figure 2) the following definitions are assumed.

A revision is considered minor if the review committee feels confident that it can be made correctly without the necessity to reconvene to review it. If, however, they are any changes that the committee feels should be reviewed after completion, the decision must be a major revision and a new review must be scheduled. When the committee says rebuild they are saying in their most conservative technical opinion it would cost more time and money to try and fix the product than it would be to build it over.

In the final analysis, we have to assumed the producer probably knows more about the product than the participants, so he is the one who will have to decide ultimately which suggestions can be easily accommodated and which are impractical. Thus, the producer should seriously and objectively consider each suggestion, each criticism and each complaint and accept those suggestions which seem appropriate, compromise where it seems reasonable and reject those suggestions which turn out to be unreasonable. If he strongly disagrees with the suggestions, he has to be able to come up with a rational argument that he can present in the next walkthrough.

### 3.7 POSSIBLE PROBLEM AREAS

Major problem areas which can occur in the review process and have been identified in the references listed in the bibliography, are the following:

- The participants goof-off.
- The participants become involved in long arguments about minor -- if not trivial -- points. Arguments about the style of the products are especially common.
- The participants become involved in personality clashes which obscure the real subject of the walkthrough.
- Some of the participants and the producer are unwilling to cooperate.
- Inexperience in giving or taking criticism.
- Unenthusiastic reviewers who do not try hard enough to find errors or defects in the product.
- Reviewing too much at one time.
- Fear that walkthroughs will be used to judge performance.

- Reviews becoming a management tool for evaluating programmers rather than problems. This will certainly defeat the purpose of the review which is error detection and reliability assurance.
- The inclusion in the product of bad fixes. The human tendency is to consider the fix or a correction to the problem to be error-free itself. Unfortunately, this is all too frequently untrue and, in the cases of fixes to errors found by inspections and reviews, the inspection process clearly has an additional obligation called follow-up to try and minimize any bad fixes.

To alleviate some of these problems and because the participants will not be expert reviewers right from the onset, a good strategy is to initially review something for which the outcome is not critical. For instance, one can start by reviewing an existing production program or a piece of a production program or its associated documentation.

### 3.8 MANAGEMENT RESPONSIBILITIES

Management (defined here as those persons having responsibility for the project) has several responsibilities with respect to technical reviews:

- Budget adequate time and manpower for both reviews and the preparation for them. Preparation will usually require as much time as the review itself.
- Understand the difference between a good review and a good product and judge the review team on the quality of the review rather than on the quality of the product submitted for review.
- Consider the results of the technical review very seriously. Ignore the judgment of a review team only in the rarest of circumstances; the long term benefit of the process is too valuable to be undermined.

- Be aware of the political and psychological effects of attending technical reviews.
- Provide logistic support - adequate conference space in particular.

### 3.9 MANDATORY ITEMS

A good deal of the methodology for technical reviews can be changed or adapted to fit the product or the particular circumstances and experimentation is encouraged. It is equally important, however, that certain guidelines be followed in all cases. The following should be considered mandatory:

- All technical products will undergo technical review.
- Technical reviews will be used to identify issues, not resolve them.
- Reviewers will prepare thoroughly in advance of the review. (See exceptions Section 3.2).
- All technical reviews will employ the review leader and a recorder except informal reviews.
- A technical review summary report and a review issues report will be prepared and distributed (except for informal reviews).

4.0 BIBLIOGRAPHY

1. Richard W. Adrion, Martha A. Branstad, and John C. Cherniavsky, "Validation Verification and Testing of Computer Software," NBS Special Publication 500-75, February 1981.
2. V. R. Basili, "Structured Walkthroughs," Tutorial on Structured Programming: Integrated Practices, Victor R. Basili and F. Terry Baker, editors, IEEE Computer Society, Los Alamitos, CA, 1980, p. 213.
3. Tom DeMarco, Concise Notes on Software Engineering, Yourdon Press, New York, NY, 1979.
4. Daniel P. Freedman and Gerald M. Weinberg, Ethnotech Review Handbook, Ethnotech, Inc., 1977, 1979.
5. Peter Freeman, "Toward Improved Review of Software Designs," Tutorial on Software Design Techniques, 3rd Edition, Peter Freeman and Anthony I. Wasserman, editors, IEEE Computer Society, Los Alamos, CA, 1980, p. 434.
6. Robert L. Glass, Software Reliability Guide Book, Prentice-Hall, Inc. Inglewood Cliffs, NJ, 1979.
7. Randy L. Merilatt, Mark K. Smith, and Leonard L. Tripp, Computer Software Validation and Verification: A General Guideline, Boeing Computer Services Company, BCS-40342, June 1981.
8. Glenford J. Myers, The Art of Software Testing, The Wiley-Interscience Publication, John Wiley and Sons, New York, NY, 1979.
9. Ben Shneiderman, Software Psychology: Human Factors in Computer and Information Systems, Winthrop Publishers, Inc., Cambridge, Mass., 1980.
10. Edward Yourdon and Tom DeMarco, The Software Engineering Lectures, (2nd edition), Yourdon, Inc., New York, NY, October 1979.
11. Edward Yourdon, Structured Walkthroughs, 2nd edition, Yourdon Inc., New York, NY, 1978.
12. Edward Yourdon, Managing the Structured Techniques, Yourdon Press, New York, NY, 1979.
13. Edward Yourdon, Structured Systems Development, Yourdon, Inc., New York, NY, 1978.

DISTRIBUTIONDOE/RL-NPQAO

J. Maclaren, Mgr (50) W/B-110

HEDL (85)

PS Abel	W/E-8	DE Lessor	W/A-4
HJ Anderson	W/C-39	DE Lucas	W/C-75
GR Armstrong	W/C-83	GW Main	W/C-114
JM Atwood	W/C-45	FM Mann	W/A-4
CL Bennett	W/D-27	RE Masterson	W/C-81
RA Bennett	W/C-43	BJ Miller	W/C-58
SI Bennion	W/B-109	LD Muhlestein	W/221T
FH Berting	W/C-24	DH Nguyen	W/C-81
RJ Bliss	W/B-65	LG Niccoli	W/C-83
GR Bloom	W/221T	LD O'Dell	W/C-81
ET Boulette	W/C-75	RP Omberg	W/E-8
PB Bourne	W/B-118	A. Padilla	W/C-81
WL Bunch	W/B-47	RD Peak	W/C-83
WF Carlson	W/E-19	RE Peterson	W/C-80
RF Carney	W/C-58	MG Piepho	W/C-83
WL Chen	W/C-83	CA Rogers	W/C-75
RW Colley	W/C-82	AR Shade	W/A-44
GE Culley	W/E-4	RE Schenter	W/A-4
JW Daughtry	W/B-45	FA Schmittroth	W/A-4
JS Davis	W/A-44	Se Seeman	W/C-82
JT Denovan	W/C-75	WF Sheely	W/A-61
DS Dutt	W/E-11	DE Simpson	W/B-63
SR Fields	W/E-9	DE Smith	W/C-81
JA Figg	W/A-4	JC Sonnichsen	W/C-32
GL Fox	W/C-81	DE Stepnewski	W/C-3
GR Franz	W/C-27	JN Strobe	W/C-35
RL Fritz	W/C-89	CD Swanson	W/C-79
RD Gaines	W/A-44	JC Van Keuren	W/C-75
JW Hagan	W/E-18	AE Waltar	W/C-79
TG Halverson	W/C-3	AB Webb	W/A-44
FJ Heard	W/C-75	NP Wilburn (2)	W/C-79
NC Hoitnik	W/C-24	GA Worth	W/E-34
JR Holmes	W/C-58	RM Wu	W/E-8
FE Holt	W/C-37	SC Yung	W/C-81
JA Hubbard	W/C-24	WH Yunker	W/C-53
EN Hurd (2)	W/B-110	Central Files (5)	W/C-110
RL Jackson	W/E-7	Publ Services (2)	W/C-115
EC Koschik	W/B-110	Doc Management	W/C-125
CN Krohn	W/C-29		
NR Laughlin	W/C-24		