Management Services: A Magazine of Planning, Systems, and Controls

Volume 5 | Number 5 Article 3

9-1968

Pitfalls in Planning an EDP Installation

Michael R. Moore

The well planned EDP installation has its roots in a condition and management attitude that must precede any planning for the installation itself. It is only when the system's objectives are clearly spelled out that a good system can be developed —

PITFALLS IN PLANNING AN EDP INSTALLATION

by Michael R. Moore
Arthur Young & Company

THIS IS A story about pitfalls in the path to a successful EDP system—and how to avoid them. Many managements today are puzzled and frustrated by the lengthy, seemingly endless, turmoil involved in converting basic information systems to EDP. It is the purpose of this brief article to identify some of the more common problems associated with the conversion to EDP systems and to suggest how they can be minimized by a planned, controlled, systematic approach to the design and installation process.

Success in EDP systems design and installation begins long before an analyst first lifts his flow-charting template to sketch out the initial rough approach. The roots of success or failure are to be found in the management philosophy and organizational framework within which the EDP system is to provide service.

Early management involvement

From the very outset the systems analyst must be assured of management's commitment to the scope of the system which he is responsible for designing. If management has carefully specified its information requirements, the systems analyst will have a large head start toward a successful job.

Such tangible evidence of management's commitment will enable the analyst to develop a well considered set of system specifications and reasonable estimates of the time and cost required to meet these specifications. Further, management itself will appreciate, to a much greater extent, the complexity of the work involved in design-

ing and installing a system which will successfully meet the specifications it has established.

It is difficult to find much sympathy for the unhappy management which doesn't get seriously interested in defining its objectives until the EDP project has almost reached operational status—only then to discover that the system isn't designed to produce the information that management really needs or wants. The practical lesson here is that the user must remove himself from the conceptual level of definition and get into the specific details at as early a point as possible in the systems definition and design effort. If he does not, the implementation cycle will become a merry-go-round for everyone concerned.

Third-generation computer equipment is a far cry from punched-card tabulating machines. This equipment has truly enormous processing capability, and its most effective use requires the design of systems which integrate or pool the information needs of several users whenever feasible. This need to "blend" the requirements of several different users increases the importance of top management's involvement in and commitment to the project.

Ideally, an applications advisory group should be established which includes representatives of all the potential users of the data processing system. The first task of this group should be the development of an inventory of proposed projects to (1) evaluate feasibility, (2) assign priorities, and (3) develop plans of action and cost estimates for implementing approved applications.

One of the practical advantages of this approach is that it should substantially decrease the frequency of crisis-type requests which otherwise could be expected from any one (and typically all) of the users. If all requests are identified in the projects inventory and priorities are established by top management, the data processing organization will have a reasonably effective defense against the overly aggressive manager who pushes for specialized applications, on a "top priority" basis, which serve only his limited needs.

One of the greatest roadblocks to the effective integration of information systems has historically been the mistake of placing the EDP function and its control at too low a level within the overall organization structure. Recently, however, there has been a trend toward the consolidation of responsibility for information systems at a relatively high level in the corporate hierarchy. Ultimately, it seems clear, the information systems function will be raised to the status of a major corporate activity reporting to a top-level executive (e.g., an administrative vice president) and operating as a service to all line organizations. The advantages of this type of organization are obvious:

Specific responsibility for the development of integrated applications is assigned at a high level in the company.

Authority is granted to cross departmental lines as necessary for analytical and other purposes.

The possibility of the system's becoming dominated by any one major user is minimized.

Consistent documentation and programing standards can be established and maintained on a centralized basis.

Opportunities for consolidating files and applications are substantially increased.

The essential steps involved in planning, designing, and installing an EDP-based management information system are not much different from those required in any major systems design effort. Briefly, these steps include the following:

1. Define objectives in terms of specific, measurable results or outputs. It is admittedly difficult to be specific but critical to do so—especially about the following:
   a. Report formats, frequency, timing, planned uses, etc.
   b. Cost objectives for system development and for continuous operation
   c. Benefits to be realized—in dollars, if feasible; otherwise, the improved timing, specific new or summarized information to be received, etc.

2. Establish responsibility and authority for managing the job through to completion. A committee can be expected to operate as an organized diffusion of responsibility, although it should be able to fulfill a useful advisory role. One individual must be designated to take responsibility for getting the job done.

3. Develop a specific plan of action with interim milestone events. Systematically identify the input forms, processing routines, and output reports which need to be defined and estimate the time requirements in terms of specific skills—accounting systems analyst, machine systems analyst, programer, etc.—and staff up for the job.

4. Develop a timetable for accomplishment by adding the estimated time requirements for all the scheduled tasks and spreading these time requirements across the available manpower. The final schedule should show the relative timing and interdependence of events, and the total time commitment should be expressed in a manpower and dollar budget.

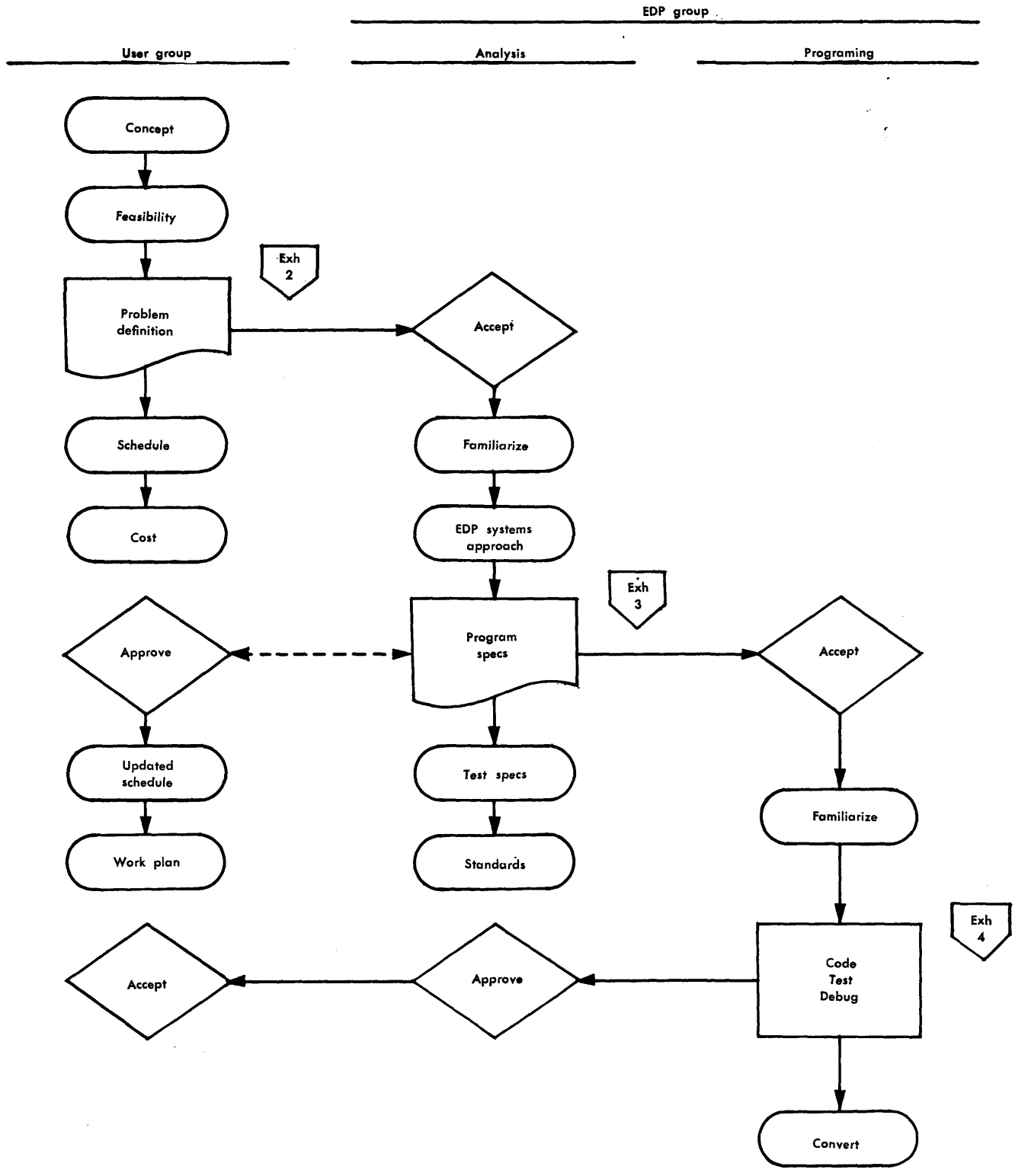
5. Summarize the plan into a formal document and establish a progress reporting and review procedure. Progress should be measured by:
   a. Actual vs. estimated com-

MICHAEL R. MOORE, CPA, is a partner in the Santa Ana, California, office of Arthur Young & Co. Mr. Moore is a member of the American Institute of CPAs, the American Accounting Association, the California Society of CPAs, and the Systems and Procedures Association. He has written a number of articles on electronic data processing for technical publications. Mr. Moore received his B.A. degree in 1956 from St. Ambrose College in Davenport, Iowa.

EXHIBIT I

EDP SYSTEM DESIGN AND INSTALLATION

(SIMPLIFIED FLOW)



pletion date of each interim milestone event

b. Actual vs. estimated time and cost expenditure in accomplishing each milestone event.

The development of a list of tasks in the form of a critical path network can be of great value in planning and scheduling the steps necessary to complete the job. The primary benefit of this approach lies in the detailed planning necessary to construct a meaningful network. The analysis required to develop such a network will instill a large and helpful measure of realism into the project and establish most of the necessary controls over the job schedule.

Management of the project is much simpler if it entails measurement of progress against a well formulated set of plans. Effective management is much more difficult when the job is planned "as you go along."

The major pitfall to avoid in the use of the network approach is the preparation of overly detailed networks. For most EDP installations it is far more desirable to diagram only major milestone events, then supplement each milestone with summary narratives or Gantt charts.

Exhibit 1 on page 27 shows the major steps and the logical flow of procedures in the design and implementation of an EDP system. The flow of procedures emphasizes the necessity for formal communications and continuous user involvement in three highlighted areas on the chart:

1. Application definition or problem statement
2. Program specification
3. Detailed programing and conversion cycle.

Application definition

In a small data processing installation, or for small applications, it would be appropriate to combine the application definition and program specification statements into a single effort-and-output document. In all installations, these formal definitions should be the joint product of the user and EDP groups.

The primary responsibility of the user is to make a comprehensive inventory of his information requirements and express these in a formal application definition. This document should reflect the transition in the user's thinking from a general concept of his information requirements to the detailed expression of these requirements in EDP terms. In making this transition the user will, in most cases, need to be guided by the EDP systems analyst.

Exhibit 2 on page 29 suggests the organization and content of a formal application definition statement. In theory, any EDP process can be performed by manual means. The user should think through the basic processing steps and present an abstract of them to the EDP organization. The application definition statement is intended to accomplish this communication.

The primary emphasis of this statement is the definition, in detail, of the input and output requirements of the system. The processing steps should be stated in summary form but should be relatively specific as to (1) control and balancing procedure, (2) main-line processing steps, and (3) error detection and correction procedures.

Definition of the application at the level described in Exhibit 2 requires that the user have a reasonable understanding of data processing terminology and techniques. The method of achieving the suggested standards of the application will differ from company to company. In some few instances, the user will have the necessary background to complete the application definition himself. More commonly, the application definition will be a joint product of the user and the systems analyst. In such cases, the systems analyst assumes primary responsibility for review and approval of the program specification.

It must be clearly understood, however, that the user cannot delegate his responsibility wholly to the EDP services group and expect that this approach will satisfy his needs. Rather, it is up to the user to recognize and pursue his responsibility to define his own needs as best he can.

The application definition also serves as a means of communication with top management, emphasizing the management uses of the recommended reports and the cost/benefits relationships of the recommended approach.

Programing and conversion

The second point of emphasis on the flow chart of procedures (Exhibit 1) is the preparation of a formal program specification statement. One of the principal objectives of this specification is to restate the problem statement prepared by the user. This restatement by the EDP organization, in terms understandable to the user, will go a long way toward assuring that a clear, correct communication of the system requirement has in fact been accomplished. Exhibit 3 on page 30 shows a suggested organization and content of the program specification.

The organization of this specification is very similar to the application definition. However, the content of the program specification is intended primarily for the programer and therefore emphasizes the specific input and output formats, data field sizes, processing steps, edit routines, and controls.

Approval of the program specification by the user organization should go a long way toward assuring that the system which is ultimately developed will satisfy the user's needs without further substantial changes.

One of the most significant elements of programing expense is the cost of making changes in programs after the specification has been "frozen." The formal program specification will minimize such eleventh-hour changes and will

serve as the basis for developing a specific work plan, time estimates, and programing standards for the project.

Exhibit 4 on page 31 shows the basic sequence of events in putting together the programs required to get the system on the air. Its principal points of emphasis are:

1. The importance of adequately designed test cases in which the user should participate to assure that the resulting program will do the job
2. The cyclical nature of desk checking and machine testing of written programs to identify and correct deficiencies in the written program
3. The requirement for some form of comprehensive parallel run or pilot test and formal acceptance by the user of the completed program.

The most common pitfalls

The basic steps in the planned, controlled, systematic approach to EDP systems design have been outlined in Exhibits 1 through 4. Successful implementation will require attention to, and avoidance of, the common pitfalls discussed below.

Communication

The single most significant pitfall in the design and installation of EDP systems is the problem of communication. A user presumably knows what he wants, but he typically cannot communicate with the machines and has great difficulty in explaining his need to the man who can make the machines talk. The EDP analyst-programer presumably can make the machine sing if he wants to, but he typically has a problem scoring the music because he can't figure out the user's composition.

The planned, controlled, systematic approach to communication via the application definition (the user's problem) and the program specification (the programer's approach) is an absolutely necessary price to pay for successful installation.

APPLICATION DEFINITION

(Outline)

TITLE PAGE
Title, date, author.

TABLE OF CONTENTS
Sections, exhibits, appendices, page references.

ABSTRACT OF APPLICATION
Introduction, description of major objectives, scope, relationships to other systems, major users and uses of the system output, special problems, summary of costs and benefits.

OUTPUT REQUIREMENTS
Definition of final content of all reports, draft version of formats, description of use of each report, number of copies and distribution of reports.

INPUT REQUIREMENTS
Complete definition of all input data to be captured, source of data, field sizes, master files and transaction files, constants, draft version of formats.

PROCESSING SUMMARY
Definitions of input editing criteria, major calculations and logical operations to be performed, history files to be maintained, special conditions, formulas, exceptions, restrictions and limitations, overall flow diagram.

CONTROL AND BALANCING
Batch balancing and other controls over input preparation, data conversion and capture, programed edits, run-to-run controls, checking procedures on output.

ERROR CORRECTION
Procedures and criteria for error detection, display, suspense account treatment, correction and re-entry.

SCHEDULE
Timing requirements for input, processing, output.

GLOSSARY
Definition of any special user-oriented terminology.

EXHIBIT 2

The steps described in the exhibits accompanying this article represent a tested approach to the solution of the communications problem. A few additional points worth mentioning are as follows:

1. It is important to have some type of orientation meeting in which the user organization presents its business application requirements to the data processing people and the analyst-programers present basic EDP concepts and approaches to the user organization. Both groups should recognize the usefulness of the jargon (or shorthand) that each of them uses in its own line of work. Emphasis in the orientation meeting should be on trying to express concepts in *the other man's language.*
2. The systems analyst function must be user-oriented. For most organizations today, this means the separate maintenance of a systems function in the user organization, outside the so-called data processing family. In the long run, it seems clear that the functions should be consolidated in the higher-level MIS (management information systems) organizations, which are not yet as common as they should be. A frequent problem in the implementation of EDP systems is the shortage of people who understand the user's needs and can bridge the communications gap.
3. There should be frequent (e.g., weekly) meetings to measure progress against some reasonable schedule. These meetings should not be Hollywood-type production affairs. They can be informal. They should emphasize problems.

The previous discussion has emphasized the absolute necessity of

PROGRAM SPECIFICATION

(Outline)

TITLE PAGE
Title, date, author.

TABLE OF CONTENTS
Sections, exhibits, appendices, page references.

ABSTRACT OF PROGRAM
Restatement of application definition in user-oriented EDP terms, description of major objectives, scope, relationship to other systems, special problems.

PROCESSING METHOD
Flow chart of EDP systems approach showing equipment used and major processing flows, summary of input balancing and editing, major calculations and logical operations, history files, special conditions, exceptions, limitations, formulas, constants, sort keys, tables, checkpoint and restart procedures.

INPUT FORMATS
Specific record layouts for card or tape input, master file and transaction records, tables.

OUTPUT FORMATS
Specific print formats, card and tape output, record layouts, console messages.

CONTROL AND BALANCING
Batch balancing, programed edit checks and edit tables, run-to-run controls, output checking procedures.

ERROR CORRECTION ROUTINES
Procedures for display of error conditions, program halt conditions, suspense account maintenance and reports, management flags, correction and re-entry procedures.

SCHEDULES
Due-in, due-out schedules, run-timing estimates, EAM and keypunch estimates.

PROGRAM WORK PLAN
Estimate of manpower requirements, staffing, elapsed time, timing of systems test and conversion.

FILE CONVERSION
Estimate of data history file conversion requirements, special program requirements.

GLOSSARY
Definition of any special EDP-oriented terminology.

EXHIBIT 3

management understanding and support for a successful EDP installation. Without such support, it is only a matter of time before the ship bubbles down beneath the waves.

Management support does not always come easily. One very common lament about the new EDP system is that "We have lost the flexibility that we had with the old card system." There is an education job to perform with this type of manager. For the most part, that old "flexibility" was actually the source of most of the problems and limitations of the old system. The systems analyst has a real job to do in selling management on the concept of *controlled flexibilty* which is feasible in third-generation systems design.

A very common pitfall in the implementation of EDP systems is the attempt to skimp on the quantity and quality of people assigned to getting new systems on the air. A realistic, adequate allowance must be made for system analysis, design, programing, and documentation staff and for adequate clerical support for these people. It is dangerous to expect one person to do all the detail work and still expect him to maintain the necessary overview of the basic purpose and objectives of the system being developed.

Specialists essential

Recognition also must be given to the contribution which specialists can make in the programing area. Although it is tempting to think about training a bookkeeper or a clerk to be a programer, it should be recognized that the larger third-generation systems require a skill and knowledge of equipment which are independent of the application to be programed. Application knowledge may be useful, but it doesn't necessarily qualify one to cope with the programing of a sophisticated EDP system. One of the most significant buried costs of EDP installations lies in the underutilization of equipment capability because of poor systems design and an inadequately trained technical staff.

Progress monitoring

The importance of planning and progress monitoring has been clearly established. Emphasis should be on the plan. A well conceived plan suggests that a manager has really thought the job through and has established most of the control that he will need to perform the job effectively.

Control should de-emphasize formal reporting of progress but should require that progress be covered frequently and informally, always in comparison with the plan that was so carefully developed.

Testing of programs

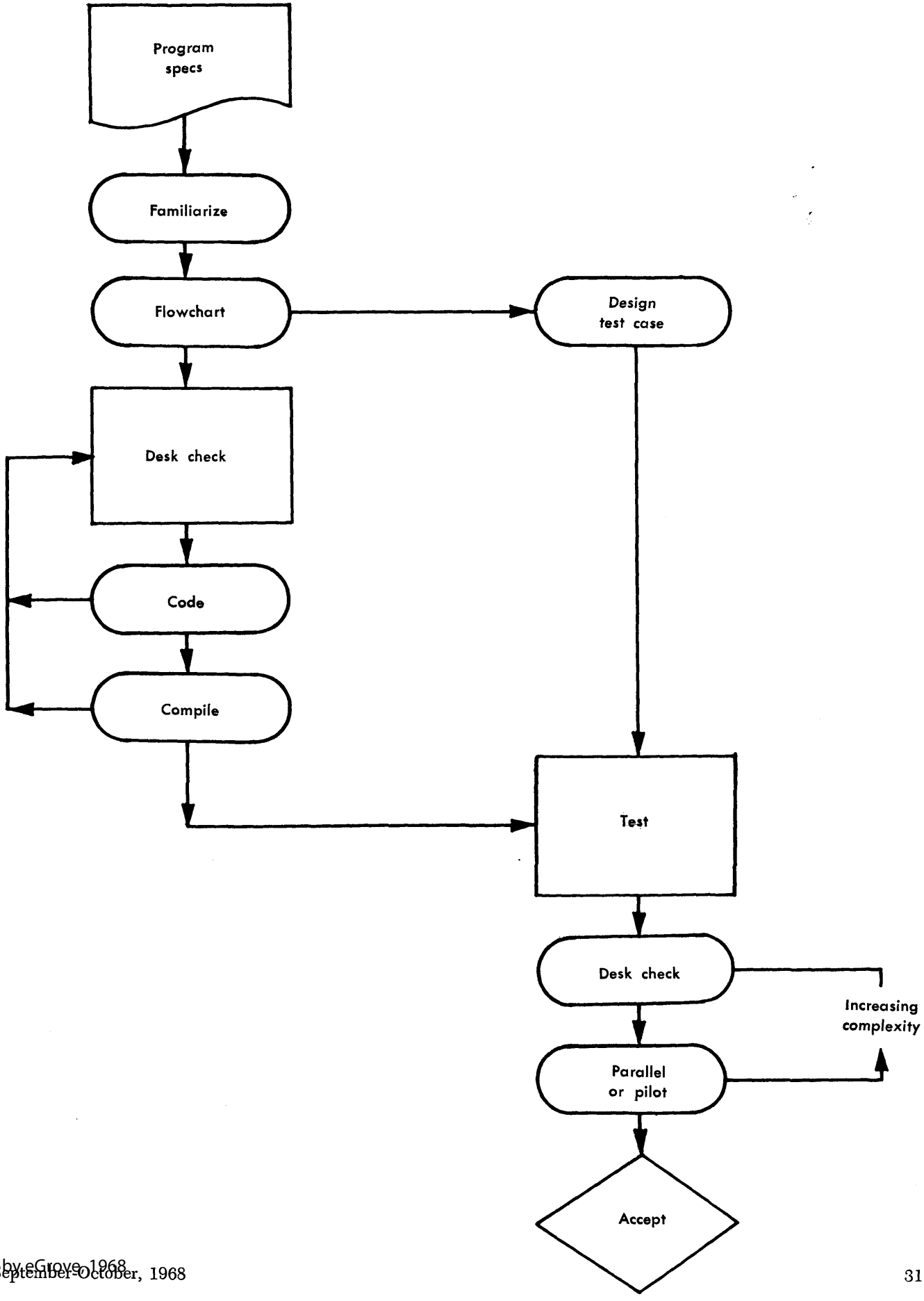
The user must involve himself as much as possible in the actual testing of the EDP system. Too often the user treats this as an area of responsibility exclusive to the EDP organization. In many ways, however, it is the user who is in the best position to interpret the adequacy and the results of tests. He should plan to participate heavily in the design of the test data.

Documentation

As a practical matter the only way to achieve good documentation is to do it as you go along. There are two very significant advantages to this approach:

EXHIBIT 4

FLOWCHART OF PROGRAMING FUNCTIONS



1. The communication is improved because documents such as the application definition and the program specification are prepared in a comprehensive, formal manner.

2. These same formal documents which were used primarily to communicate now become the basic data for preparation of formal documentation and manuals—both for the user organization and for the programing and operations groups.

Conversion of files

It is extremely important to achieve an early definition of the history files and master files which will be needed for the new system. The processes of file cleanup and reorganization to suit the new application can be very lengthy and difficult to control. In some cases, the file conversion job is so large that the special programs which must be written to accomplish conversion represent nearly as large a programing task as the effort to write the continuing main-line processes of the new system.

The conversion problem exists in every EDP installation. Sometimes it is minor; more often it is huge. Treat it like any other application problem:

1. Define it.
2. Plan and schedule it.
2. Control it.

Every manager considering a "crash" program should be given the opportunity to walk through the ruins of one of the many data processing installations which have never recovered from the implementation of their basic systems on a crash basis.

Crash programs

The very words "crash programs" strike terror in the hearts of system analysts everywhere.

Prepare for trouble if you are going to have a crash program. Recognize that the output in many of the programs will be of the "quick and dirty" variety, with the emphasis on "dirty."

All kinds of justifications have been offered for crash programs, but most of them are founded in sand. Crash programs are usually encouraged by a management which simply does not understand the complexity of the job which it is asking to be done. Too often, the basic problem stems from a desire to use a new piece of equipment before the organization is really ready for it.

No simple answer

There is no happy solution for the analyst who is faced with this kind of problem. And the only sound advice that can be offered to management is "Don't do it." The sad effects of the crash program are lasting; they are not one-time events. Every manager considering a crash program should be given the opportunity to walk through the ruins of one of the many data processing installations which have never recovered from the implementation of their basic systems on a crash basis.

Conclusion

Some people appear to have been able to get through the design and installation of an EDP system by flying it blind. It is not clear whether these people are geniuses or are somehow protected by Divine guidance. They call to mind some of the famous chases in the old slapstick comedies, where the hero, in an uncontrolled car, is careening down the street between and around streetcars, buses, trucks, etc. At the moment of crisis, he closes his eyes and charges blindly forward—and in the next scene the two trucks which have been converging upon him have crashed head on and our hero is in the midst of another calamity.

Most of us, however, do not live in the world of the Keystone Cops. We live in the real world, where a successful EDP installation is simply not achieved by haphazard methods.

EDP systems design and installation requires a planned, controlled, systematic approach. That's the only way to fly. The dollars involved are too many and the systems involved too vital to the basic information and control needs of the business to do it any other way.