

THE MANAGEMENT OF END USER COMPUTING

John F. Rockart

Lauren S. Flannery

September 1981

CISR No. 76
Sloan WP No. 1259-81

© J. F. Rockart and L. S. Flannery 1981

This paper was presented at, and was published in the Proceedings of, International Conference on Information Systems, December, 1981, Cambridge, Mass.

Center for Information Systems Research

Sloan School of Management

Massachusetts Institute of Technology

The Management of End User Computing

John F. Rockart
Massachusetts Institute of Technology

Lauren S. Flannery
Massachusetts Institute of Technology

Seven years ago, Dick Nolan and Chuck Gibson provided the information systems (IS) community with a stage theory for IS management (Gibson & Nolan, 1974). Despite an initially small research base (and some concomitant academic jeers), the model rang true to practitioners. Since 1974, it has served as a useful tool for many IS managers. They have used it as a means to describe the current status of computer use in their organizations and as a basis for future information systems planning.

The model postulates four stages of IS development. In stage 1, the initiation stage, a few basic well-structured applications, typically accounting, are successfully automated. This success leads to a proliferation of application development giving birth to stage 2, the contagion stage. Unfortunately, while widespread penetration of the new technology is achieved during this stage, it is most usually "too much, too fast." Inexperienced programmers having too much to do produce poorly documented, inept, and expensive systems, often much behind schedule. To ensure adequate management of information systems, stage 3, the control stage, begins. Controls are put in, standards developed, and the frantic pace of new application development is slowed to a walk. Finally, now being able to computerize in a managed, controlled manner, the company enters stage 4, the integration stage. New development once again begins and the company moves ahead with its use of computers into advanced applications.

Recently, Nolan has added two additional stages to the model. He has been appropriately challenged as to the usefulness of this extension. It has been pointed out that his original stage theory is, in effect, a particularization of the general phenomena of the "learning curve" (Keen, 1981; McFarlan). Adding credence to this view, Zisman has described a four-stage learning curve phenomenon he sees in the field of office automation (Zisman, 1979). Keen has suggested concurrent learning curves for several new technologies such as data resource management, integrated communications, personal computing using microcomputers, and teleconferencing (Keen, 1981). The purpose of this paper is to present the results of some ongoing research which strongly suggests that we have yet another managerially significant multi-stage "learning curve" taking place today in the information systems field. This one is in the area which can be termed "end user computing."

It is becoming increasingly evident to us that the Gibson-Nolan stage theory, originally developed for application to the management of traditional data processing (the "COBOL shop"), is equally applicable to the management of end user computing (the "APL, RAMIS, EXPRESS, BASIC, etc., shop"). Most companies today are in the first, or "initiation," stage with regard to end user computing. They are just beginning to offer a set of "tools" such as APL to their end users. The percentage of use of the computing facility by end users in those stage 1 companies, in terms of MIPS, is somewhere between zero and ten per-

cent. In general, the users are a highly capable group of expert staff people who know precisely what they want to do. They have mastered, on their own initiative and often with little assistance, the necessary languages and techniques. But there are relatively few of them. End user computing has yet to "take off" in these companies.

There is today, however, a very clear set of companies who have moved into the second, or "contagion," stage. These are the companies which have been offering extensive time sharing and other facilities to end users for the past five to ten years. In these companies, the use of MIPS by the end user community now ranges from twenty to forty percent of the entire computing facility. What is more, the growth of computer use by end users in these companies is clearly accelerating. Bob Benjamin of Xerox, a company in this stage, estimates that end users will be utilizing 75% of the computing facility in his company by the end of this decade (Benjamin, 1981).

The evidence we have gathered in our field study of end user computing suggests that Benjamin is at least roughly correct in his estimate for 1990. It also suggests, however, that unless significant attention is given to managing this area that we are highly likely to repeat the most adverse effects of the initiation-contagion-control-maturity cycle of traditional data processing. In fact, the cycle itself probably is inescapable. Yet, unmanaged, it has some significant negative consequences.

Perhaps the worst effect, from an IS management viewpoint, of an unmanaged learning curve, is the debilitating--and often devastating--transition to the third stage. In too many companies the COBOL shop went through a stultifying crunch of human and financial resources as it entered the "control" phase. Observing things to be "out of control" top managers of these

companies demanded immediate pruning of many projects and a closely cost-justified approach to those remaining. In many cases, much worthwhile work was terminated. Some companies fell far behind in their use of the computer as Draconian control measures were put into place. From the evidence we see, this "dark age" could again be imposed--this time on end users, unless adequate management attention is given to the process of end user computing now. There is an opportunity today, with most companies in stages one and two, to foresee the issues which might invoke a counter-productive stage three in end user computing and to develop managerial processes and techniques to deal with these issues.

Given this opportunity, we began a research project on the management of end user computing in early 1980. The research involves field interviews in seven companies in which significant end user computing is being done. In our view, all but one of these companies are in stage two. One of the company studies (at APEX, name disguised) has been completed. Its findings are reported here. They are of interest since the issues and analysis noted appear to generalize well to the initial findings from our other six sites.

COMPANY BACKGROUND

APEX is a multi-billion dollar company that develops, manufactures, and markets a wide range of information processing products. APEX has followed an information systems strategy of almost complete hardware centralization and systems development decentralization. As Figure 1 illustrates, the majority of APEX hardware is located in "DP Operations" under the General Services division. Systems development support personnel, however, are distributed throughout the organization, broken into timesharing support and commercial (data processing) support

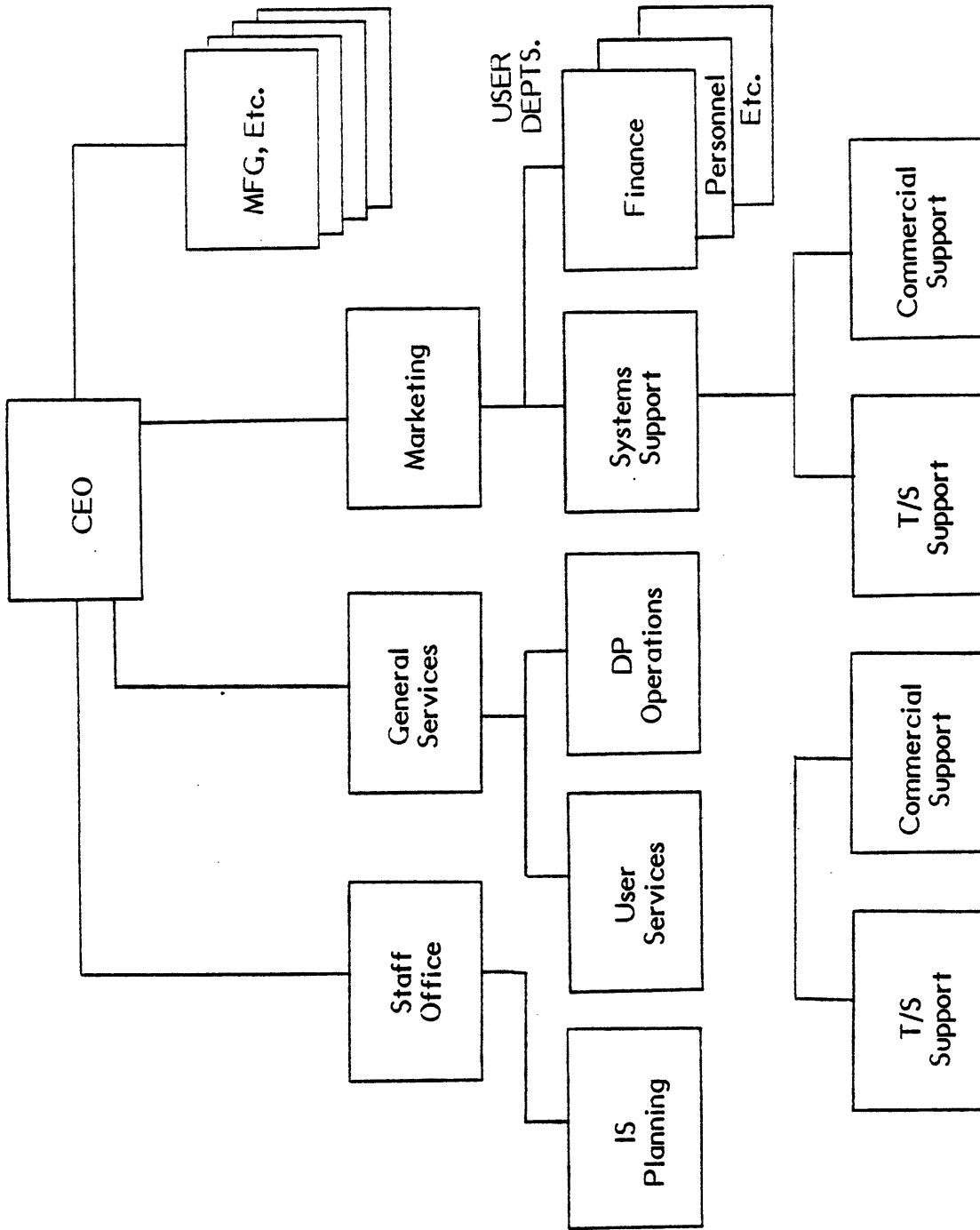


Figure 1. APEX--Organization Structure (Abridged)

groups. Not all divisions have timesharing support.

Over the past five years, the total data processing budget at APEX has increased by only 10% per year. During the same period, however, the usage of the internal timesharing system for end user computing (which at APEX principally involves APL) has increased by almost 25% per year. From 0% of the operational budget in the early 1970's, end user computing has grown to the point where it now accounts for 25% of each operational data processing dollar.

Management at APEX encouraged our study for three main reasons. These were (1) the need to understand the factors underlying the surprisingly high rate of growth of, and therefore the company's increasing (and perceived as somewhat out of control) financial commitment to, end use computing; (2) the belief that significant risks were involved in having end users develop some systems; and (3) the desire to understand how end user computing should be managed.

Under existing policies, the growth of end user computing had been relatively uncontrolled. It had been assumed that the budgetary constraint imposed on each user would be a sufficient deterrent to unjustified use of the resource. Internal timesharing had grown, however, in a very short number of years, to over 3,000 users spending well into eight digits annually. Budgets were being exceeded throughout the company, and the rate of growth actually appeared to be increasing.

Several major questions were raised concerning the potential risks involved in end user computing. The key questions were: did a segment of the timesharing applications actually represent systems critical to the ongoing performance of the business? If so, were they adequately documented and maintained? How adequate were data security and data integrity?

Finally, IS management wanted to understand better how to manage the end user programming area. Specifically, the manager of Timesharing User Services was concerned about how to structure a support organization that would effectively serve the end users and anticipate their product and service requirements.

STUDY METHODOLOGY

This initial study was aimed at gaining a qualitative understanding of timesharing use and at developing a set of hypotheses which could be evaluated in later quantitative research. The approach taken was one of indepth interviews with a relatively small cross-section of approximately thirty end users. The interviews, which were confidential, began with an open-ended discussion of each participant's computing activities. The approach has aimed at facilitating the surfacing of the key issues with regard to end user computing as perceived by the users themselves. To ensure that data were collected on each of a set of previously hypothesized issues, however, the interview was then directed to a structured discussion of each of the issues not raised by the interviewee. We sought in particular to understand four things. These were the reasons underlying the rapid growth of timesharing, who the end users were and whether they could be usefully categorized, the types of systems that were being built, and, most important, the range of issues involved in the effective management of end user computing.

STUDY RESULTS

Our results in each of these four areas are as follows.

The Reasons For Timesharing Growth

Five factors emerged from the study as the principal reasons for the impressive growth

The Application Profile

The sample of users interviewed utilized the timesharing resource for a wide range of applications, from traditional batch systems to interactive, "what-if" models supporting managerial decision making. We identified six fairly broad categories of applications. They are:

1. APL Prototypes. Among individuals who develop the systems, "APL prototyping" was a common theme of many applications. In theory, prototypes are developed quickly in APL in order to test a concept and/or to establish its benefits to the user. Many of these systems are true "decision support systems." The initial prototype is revised as early usage occurs. The APL advantages of quick development and ease of change are especially important for these applications.

In theory, where these applications prove to be of use, it is assumed that they will be converted to COBOL to be run in the formal systems environment, where it will run more cost effectively. In practice, this often does not occur. First, once the users have paid to have an application developed in APL, they are often reluctant to pay to have it reprogrammed in COBOL. They are willing to forego any future cost savings in favor of using what they have without a further investment. Second, some users insist on keeping this type of application on the timesharing system because they hate to "lose control" of the ability to run it on their own schedules and to make changes when and as desired. As one professional support staffer noted, "The word 'prototype' has become a buzzword which is often used to mask a quick and dirty system that ends up lasting years."

2. Operational Batch Systems. These systems, performing functions important to the ongoing effectiveness of the organization, were not at all uncommon in the end user environment. One user, who provides timesharing support for his functional area, observed, "We have one clerk who does nothing but run standard inventory-control APL batch jobs." The manager of another functional support group said "Most of my applications would be better done in COBOL--but management wanted to get them running quickly, and this was the only way." Even the professional timesharing support staffs develop some applications that are essentially batch systems better suited for the COBOL environment. In the words of one project leader, "I have certain criteria I use in determining whether an application should be developed on the timesharing systems. If the application doesn't meet those criteria, I try to dissuade the user on the basis of cost and try to convince him to go to the COBOL people. However, if someone insists on APL, and many do...I'll implement just about anything."
3. General-Purpose Models and Statistical Routines. The third category of use is that of models and statistical routines, such as the available SPSS package. In general, these models and routines consist of purchased packages or systems developed by the local functional support groups or professional staffs. Their use by unsophisticated end users requires ongoing education and assistance.
4. Report Reformatters. Several of the users described applications that essentially amounted to the

rate in end user computing. Some of these reasons are well known. First was the fast development time afforded by APL--the ability to produce results quickly in response to current needs of management. There is little doubt today that, where data are available and appropriate, end user languages can be used, systems can be developed more quickly, and, in many cases, more economically.

Tied in with fast development time is the ability to easily prototype an application. Several interviewees considered this extremely important because a prototype provides an opportunity to establish the rough benefits of a project before the decision to proceed with full blown implementation is made.

The third important factor influencing timesharing growth at APEX was the ease of access by end users to the systems serving them and the quick execution of jobs desired at the time desired. APEX provides access to both the data and programs available in its internal timesharing system through an international communications network. This is a significant drawing card for geographically dispersed users. Although many of the applications we looked at were not actually "interactive," on-demand, fast execution was perceived as important because it allowed the user to request output directly through his terminal and to bypass the processing queue in the batch environment.

The non-responsiveness of the traditional IS community was viewed, without exception, as a fourth major reason, and quite possibly the principal reason, for using the timesharing system. The frustratingly long lead times for COBOL support were contrasted with what was generally perceived as a "cooperative and helpful" timesharing support group. Further, many users felt that the COBOL programmers they had worked with did not understand their problems and could not respond to them if they

did. As one APL programmer explained, "The ability to develop systems quickly and 'on the spot' enables us to deliver exactly what the user wants. He'd be facing at least an eight month lead time in the traditional systems group before development even began--and then the design and programming might take years. His needs would probably have changed by the time the application was up and running."

The fifth, and perhaps the most interesting, reason for the growth in timesharing is the expandable supply of programmers within the user organization. Although the professional timesharing support staff, like the traditional DP shop, is restricted by a tight lid on head count, users in the timesharing environment are free to hire programming help from outside the company or to train employees within their departments to develop and maintain applications. On the organization chart, these people are "financial analysts" or "market researchers." In effect, many are full-time programmers. They can be switched to this function from other duties as needed.

The End User Profile

The image of an end user community as a set of individuals who are both developers and users of their own computer programs did not prove to be the case at APEX. Instead, the majority of the timesharing users we interviewed merely access systems developed by any one of several small groups of designer/programmers. Most "end users" with whom we talked had neither the inclination, nor the knowledge, to use APL to develop systems on their own. The market for timesharing services can thus be segmented into four distinct groups of users (McLean, 1979). These are:

1. Unsophisticated end users, who seek development and maintenance support from one of the user support groups (either through for-

mal or informal channels) and who do no actual programming themselves. They often interact with the system in a highly structured and limited "menu-driven" environment. This segment of end users is by far the largest and our current multi-site data suggest that it represents an estimated 70% to 80% of the users.

2. Sophisticated end users, who utilize the timesharing resource directly for their personal information needs and who develop their own applications. The use of the applications they develop is likely to proliferate through informal channels and they may end up providing limited support to other end users. They do not, however, consider this an important aspect of their job. About 10% of the user population appears to fall into this category.
3. Local functional support group personnel, consisting of those individuals, who, by virtue of their APL prowess, have become informal centers of design and programming expertise within their functional areas. These pockets of APL support have spontaneously grown up to meet the demand created by the unsophisticated end users in segment (1). An example of this kind of ad hoc support group, functioning as a surrogate for more traditional programmer/analysts, is a group of four individuals within the Corporate Planning Department at APEX which develops APL-based interactive models for use by financial analysts throughout the company. The manager of the group explained the trend away from actual end user programming as follows: "When APL first was available, people did

their own programming, but as usage matured, small pockets of programmers formed in each functional organization, and end users gradually ceased to write their own code, turning instead to these groups. Many of the groups, in order to legitimize and strengthen their positions in the organization, now actively solicit new business."

In spite of the large percentage of their time that these individuals spend coding (several of them estimated over 80%), they do not view themselves as programmers or anything akin to data processing professionals. Accordingly, most feel that providing documentation is not part of their job. This is a highly interesting segment of the end user population. Representing approximately 5-10% of the users, these ubiquitous local functional support group personnel present some interesting managerial challenges, as we note later.

4. Professional support staffs, located within Systems Support for each division (only the marketing division is shown in Figure 1), whose chartered responsibility is to develop and maintain timesharing applications for end users. Also included in this segment is the Timesharing Support staff, located centrally in the General Services Division, which does engage, although selectively, in development and/or maintenance activities. These professionals account for less than 5% of the user population.

Predictably, the users in each of these four segments had a somewhat different perspective of end user computing. Each raised some unique issues and had different value judgments on those issues commonly perceived by all groups.

reformatting of reports originally produced by the formal IS systems. The reformatting of available data is necessary since different managers have different information requirements. However, the number of systems developed to do so at APEX is an anomaly springing from the company's heavy reliance on APL, the lack of a good end user oriented database system, and the concomitant lack of an easy-to-use extraction and report generating language.

5. Inquiry and Status Reporting. Inquiry applications are characterized by a large specialized database which is accessed through a menu of subroutines heavily oriented to the provision of canned reports. In general, the inquiry systems we looked at had a large number of geographically dispersed users, requiring a network to permit wide-ranging geographic access.
6. Text-Oriented Applications. There were a number of applications that, as opposed to the above essentially numerically-oriented categories, were oriented toward textual transfer and processing. One of these was the electronic mail package available on the timesharing system. Other systems were aimed at text editing and processing.

The Key Managerial Issues At APEX

In addition to providing a skeletal classification framework for timesharing users and applications at APEX, the study also uncovered a number of characteristics of end user computing at APEX which can be described as significant, company-specific issues needing attention by those desiring

to more efficiently manage end user computing. Since many of these findings also tend to generalize to other sites, we note them here. We list seven major findings and the key managerial issues each raises. They are:

1. Many applications vital to operational decisions have been programmed by end users. In general, these applications are poorly documented and supported. Documentation and coding standards have been formulated at APEX for APL programs by the Systems Methodologies group in the Corporate IS Department. However, according to several managers of timesharing support groups, only systems that are perceived as "critical to the business" are subject to these corporate standards. Since the judgment of whether or not an application is "critical" rests with the user, the use of the standard is rarely invoked.

An example of this situation is a Plan/Allocation model, developed and maintained, but not documented, by an individual who provides information support services to marketing and sales vice presidents. The company's budget and performance plan is fed into the model, which produces targets for each region. Regional analysts write their own programs in APL to access the data. The regional data are then further broken down to the branch level, at which point they are fed into operational systems which provide the basis for salesmen bonus reporting. Should any part of that APL processing fail, salesmen would not have the targets which are the basis for their reward structure. The system, however, is not adequately documented.

The developer of another apparently critical, yet undocumented, system that produces short range forecasts used by multi-

ple areas of the company states the situation succinctly. "If this system didn't run, it would have a tremendous impact. I don't think short range forecasts can be developed by hand anymore."

Even an application which has been developed by a professional support group according to corporate coding and documenting standards is not assured of being maintained to the same standards. If it is turned over to the end user, and it must be released if the user insists, he is under no obligation to keep the documentation up-to-date--and typically does not. The clear issue for management here is to develop a procedure to ensure that "critical" applications are developed with adequate standards and controls.

2. Informal support groups, who spend all their time performing systems development and programming functions, but who do not perceive themselves as IS professionals, pose several managerial problems. As noted earlier, these informal groups are small pockets of users within each department who are in reality a local, home-grown, replication of the professional user-support function in the timesharing organization. In general, these groups started with a single financial analyst (market researcher, etc.) within a functional department who learned to program and achieved excellent results with an end user language. His expertise was quickly recognized and colleagues then asked him to build a system for them. To do so, he added one or more additional people to help him. The group was born. In many cases, it has expanded significantly, as additional demand justified additional personnel. Many of the individuals perceive they have considerable power--and they have been well-

rewarded for their highly useful expertise and the quick results they obtain. Some of the groups we saw have expanded to the point where they are even marketing their services within their departments in order to maintain and increase the demand for their services. Yet these groups do not consider themselves bound by any of the normal constraints on computer professionals. Documentation of systems is limited. Controls in systems are often not explicitly considered. Justification processes, the development of educational materials, and other procedures which increase the time from project initiation to completion (and thus decrease their effectiveness in users' eyes) are given short shrift.

Significant issues for IS management with regard to these "amateur" programming groups include the definition of the documentation, the standards, the educational materials, and the controls necessary for particular classes of applications. Equally important is the definition of the appropriate methods for the day-to-day management of this new type of computer semi-professional. Perhaps most important is the development of a career planning process for these people. While admitting to not being computer professionals, they are also, in many cases, not keeping up with the new knowledge in their former functional fields. Many feel that they are becoming "citizens without a state." Some, while enjoying what they are doing at present, are quite concerned about their future careers.

3. Users are willing to pay significant hardware running cost premiums to get systems up and running quickly under their control. The users we interviewed were relatively insensitive to the cost of hardware

cycles. They clearly perceived that the cost to run systems programmed in end user languages on the timesharing network was significantly higher than the cost of running a corresponding system developed for the batch environment. However, they perceived the value of bringing systems up in this environment to be very high. The ability to get systems up quickly, the facility to make changes as they become necessary without going into the "waiting line" caused by the COBOL shop's backlog, the ability to eliminate much non-productive, time-consuming interaction between their organizations and the systems support staff, and the capacity to easily control the allocation of development resources to particular projects -- all of these factors are considered by users when deciding in what manner to develop a particular system. In short, users perceive the "net-cost" (value minus cost) of developing and running systems. The equation is increasingly swinging away from the COBOL shop as better end user languages are developed, as they are better understood, as increasing support is given to end users, and as the cost of hardware cycles declines.

The managerial issues here include the ability to estimate and keep up with demand, the definition of guidelines to assist line managers to more accurately assess the real "net-cost" of developing systems for each environment, and the ability to assist end users to adequately plan and budget in the end user sphere.

4. Locating data, and coordinating its collection, are problems to both new and experienced users. Since a number of the timesharing appli-

cations we looked at were utilizing at least some data from existing batch systems, the data interface with the formal IS world is an important issue. The responsibility for engineering a smooth interface at APEX today rests with the user, although the timesharing support staff in each function, if asked, will help a user figure out "where the data is and how to get at it." When asked how a new timesharing user, unfamiliar with the IS organization or the location and contents of data files would go about finding data for his application, a member of the support staff answered, "It would be pretty tough. Hopefully, there would be someone in his area who knew enough to point him in the direction of one of the formal or informal support groups--which would be the best place to start. But a significant number of end users don't know about us. In addition, many reinput their data since it is difficult to get at some files. I believe there is an incredible amount of data duplication here--with a lot of costs and error-potential associated with it."

Locating data and coordinating its collection are not problems confronting only new users, however. As a member of the timesharing support staff lamented, "I have to collect data from several different locations. The data turns out to be of differing currency and consistency. This is a real problem."

The implications of all these data problems for IS management are obvious. Perhaps clearest of all the implications of this study is the need to support users in their desire to know what data is available, to provide simple mechanisms to allow users to obtain the desired data, and to provide means to aid users to understand the attributes concerning the quality of the data

they receive (source, time, or origination, etc.).

5. Little attention is paid to transferability of applications, resulting in considerable duplication of development efforts. With the exception of some of the products developed by the timesharing support staff, the applications we looked at were not designed with an eye toward sharing the code with other users. Nor was there any significant managerial effort to encourage sharing. The applications, once developed, were only occasionally made available to other users through informal channels or through the APL User's Library with minimal support. It was widely felt that much redundant system development was taking place.

There are many significant reasons not to share code (including the costs of maintenance of shared code, coordination costs, etc.). Yet the managerial implication here is that each organization must determine for itself the degree of code-sharing desired—and the best ways to assist users to avoid the cost of reproducing simple, ubiquitous programs and subroutines. (This is similar to the "common system" question in the COBOL shop. There is no one right answer. It is organizationally dependent.)

6. There is a mismatch between the types of applications being programmed and the end user languages available. End user support at APEX revolves almost entirely around APL. However, as noted earlier, there is a wide variety of types of applications being developed by and for users. Some of these applications are well served by APL. For others, however, APL is very suboptimal although it can be used.

The implications of using APL for high volume often-run typical batch reporting systems are obvious. The execution time penalty resulting from this interpretative language, together with the difficulty of documenting APL, makes it less than an ideal choice for these systems.

Others of the application classes noted above would be better developed and used through non-procedural, end user oriented, data-management systems such as RAMIS, NOMAD, FOCUS, and EXPRESS. Simple report generators and inquiry systems also seem to have their place for some of the applications.

Despite this need for a richer portfolio of end user tools, it will be difficult to institute new tools for a variety of reasons. At APEX, the user community is extremely adamant in its use of APL for nearly all classes of programs. This is not surprising since user education is slanted almost exclusively toward APL. Further, support personnel (both professional and informal) are well versed in the finer points of APL and only superficially familiar with alternate approaches. The process of weaning all concerned away from APL to the most appropriate end user tool for each application type will be long and difficult at APEX.

The key implications for IS management here start with the evident need to develop an understanding of the types of applications (both current and future) applicable to end uses and to provide the current range of types of tools to meet these diverse application types. In addition, the training of support personnel who can assist users with alternate products is necessary. The development of education programs to aid users in both selecting and using the correct tool from the end user armament is highly desirable. Ultimately, the provision to the users of an all-encompassing, unified end user language providing single-point access to a complete range

of end user tools would appear highly desirable. (But, first, this language must be developed by computer or software vendors!)

7. The current approach to the management of end user computing is unclear to both end users and IS management. At APEX, end user computing has grown like TOPSY. Much attention has been given to effective and efficient management of the COBOL shop. End users, however, are served under an unclear management structure by a variety of groups with unclear charters. Fundamental issues with regard to the previous six issues are just being investigated by IS management. In fact, because users are able to do what they want with the tools they know, there is not much overt discontent. But there is a distinct feeling among all end users interviewed, and among the professional IS staff, that "things can be done a lot better" and that "it is time to develop a well-thought-out approach to the management of the end user side of the house."

SUMMARY — THREE KEY TASKS

It appears to us, as a result of the work at APEX, and similar findings at our other six sites, that there is a need for the development of exactly such a well-defined managerial approach to end user computing. The companies we have seen are experiencing an "explosion" in the demand for end user computing. Budgets in many are being significantly overspent. In some, line managers are asking searching questions as to ways to control, or at least to ensure the most efficient use of, this new resource. We believe that answers must be forthcoming or Nolan's "third stage" will hit with more impact than is necessary.

In our view, policies and procedures must be developed today in each company for the management of end user computing. There are three distinct and separable sets of tasks that must be accomplished. Taken together, these three will define a management structure for end user computing. They are (1) the development of a set of managerial policies for the end user computing "market," (2) the determination of the set of "support" processes needed by end users, and (3) the determination of the appropriate "controls" to be placed on end users. While we do not yet feel that we can provide answers in each of these areas, the issues in each are reasonably clear.

Developing a set of policies. Clearly, the first step in developing a set of managerial policies with regard to end user computing is the understanding of the "marketplace." IBM's BSP and other similar processes have allowed us to map the basic "paperwork-processing" (COBOL shop) applications so that they are well understood today. No such understanding exists as to the type of computer based work being performed, and its probable future extent, in "end user land." This type of understanding (developed through careful analysis) must be done to further the development of policies concerning the types of tools which should be offered, the support mechanisms which are needed, and the methods to be used to control the expenditures being made by end users. User management must be strongly involved in the development of all these policies.

Determining the necessary user-support procedures. This is perhaps the most obvious need. Procedures must be developed to provide appropriate data administration services, formal and informal education, and adequate system selection, design, and programming support. Moreover, a determination must be made as to where, how, and by whom these services are rendered. Finally, care must be taken

to adequately manage the careers of each different set of support personnel.

Determining the appropriate controls over end user computing. This appears to be a neglected, yet vital, area which needs managerial attention. Controls must be developed to ensure that "critical" applications are well documented and contain the necessary audit and control procedures. Justification procedures for large end user systems must be developed and implemented (or at least made available for use, or non-use, by end user management). One can think of several other controls, such as a mandating of the use of standards modules (e.g., for present value calculations) by end users, which would improve the efficiency of the use of the end user resource. While it is perhaps not yet appropriate to mandate such efficiency-inducing measures today, it may be appropriate "tomorrow" as end user claims on the computing resources grow even greater.

Attention to all three of these areas should result in a "managed" not simply an "exploding" use of computer resources by end users. If this attention is not provided, a more difficult than necessary stage three

on the learning curve and its negative repercussions appears inevitable. What is more, users will be less than adequately served in all stages.

REFERENCES

- Benjamin, R. Xerox Corp., Speech at Endicott House, Dedham, Massachusetts, CISR seminar on February 26, 1981.
- Gibson, C.F. and Nolan, R.L. "Managing the Four Stages of EDP Growth," Harvard Business Review, January - February, 1974, pp. 76-85.
- Keen, P.G.W. "Telecommunications and Business Policy: the Coming Impact of Communications on Management," unpublished working paper, May 1981.
- McFarlan, F.W. Personal Communication.
- McLean, E.R. "End Users as Application Developers," Proceedings of the Guide/Share Applications Development Symposium, October, 1979, p. 50.
- Nolan, R.L. "Managing the Crises in Data Processing," Harvard Business Review, March-April, 1979, pp. 115-126.
- Zisman, M.D. "Office Automation: Revolution or Evolution?" Sloan Management Review, Massachusetts Institute of Technology, Volume 19, Number 3, Spring 1979, pages 1-16.