Association for Information Systems
AIS Electronic Library (AISeL)

ICIS 1984 Proceedings

International Conference on Information Systems (ICIS)

1984

Organizational Designs for Software Maintenance

E. Burton Swanson
University of California, Los Angeles

Follow this and additional works at: http://aisel.aisnet.org/icis1984

Recommended Citation

Swanson, E. Burton, "Organizational Designs for Software Maintenance" (1984). *ICIS 1984 Proceedings*. 9.
http://aisel.aisnet.org/icis1984/9

This material is brought to you by the International Conference on Information Systems (ICIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ICIS 1984 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Organizational Designs for Software Maintenance

E. Burton Swanson
Graduate School of Management
University of California, Los Angeles

ABSTRACT

Organization design theory is applied to issues in information systems (IS) management of application software maintenance. Following the suggestion of Galbraith (1973, 1977), it is argued that inceased uncertainty in the maintenance task requires organizational adaptation to reduce the IS need for information processing in support of this task, or, alternatively, to increase the capacity for the same. Sources of uncertainty in the maintenance task are identified, and a number of related propositions are developed and enumerated. A corresponding set of design alternatives for maintenance management is also presented and given theoretical interpretation.

Introduction

The maintenance of computer application software is a subject which has received increasing attention. An early article suggested that maintenance was similar to an "iceberg," with low visibility but high impact potential for the success or failure of the information systems organization (Canning, 1972). Survey research has since assessed the magnitude and dimensions of the maintenance problem (Lientz, *et al*, 1978; Lientz and Swanson, 1980). Solution-oriented books now follow (e.g., Martin and McClure, 1983).

Much research remains to be accomplished. A particular need is for theory which unites the fragmented findings and anecdotal wisdom. The present paper aspires to this end.

Organization theory is applied here to the maintenance task of the information systems organization. Specifically, a theory of organization design proposed by Galbraith (1973, 1977) is applied to an analysis of the maintenance task, and to the identification of design alternatives for organizing the maintenance function responsible for this task. Galbraith's theory argues broadly that increased uncertainty associated with any task requires an information processing solution in the supporting organization design. Thus, our application of this theory begins with the indentification of sources of uncertainty in the maintenance task, and follows with the indentification of alternative information processing solutions. In the course of this application, the literature on software maintenance is examined. The result, it is hoped, is a fresh view of application software maintenance and its management.

The Maintenance Task

Software maintenance consists of three forms of work activity termed corrective, adaptive, and perfective. (Swanson, 1976). In corrective maintenance, failure in software processing, performance, and implementation are addressed. In adaptive maintenance, the software is improved, eliminating processing inefficiencies, enhancing performance, or increasing maintainability itself.

The effective and efficient management of software maintenance requires that information and decisions be organized and deployed in support of all three forms of maintenance work. The task uncertainty associated with this work may be argued to be the crucial determinant of

Acknowledgement: Partial support for this research was provided by the Information Systems Program, Office of Naval Research, under contract N00014-75-C-0266, project number NR049-345. Helpful comments on an earlier version of the paper were made by Cynthia M. Beath and three anonymous reviewers. A number of the ideas presented were developed in a course taught at the IBM European Systems Research Institute, La Hulpe, Belgium.

the information and decision support requirement (Galbraith, 1977). From this "contingency perspective," uncertainty should differ across IS organizations and, accordingly, require varying approaches to IS organization design.

The amount of uncertainty associated with the maintenance task may be understood to rest upon three factors: task variability (over time), task diversity (the range of service provided, and the associated division of labor), and task difficulty (in meeting performance objectives) (Galbraith, 1977). In the following paragraphs, these three factors are examined in detail as applied to the maintenance task.

The basis of the maintenance task is understood to be the installed application system portfolio. This portfolio consists of all operational application systems for which the IS organization has ongoing maintenance and development responsibility.

In Table 1, the elements of uncertainty in the maintenance task are summarized. Each will be discussed in turn. Propositions relating to task variability, diversity, and difficulty are developed and enumerated.

Table 1

Sources of Uncertainty in the Maintenance Task

- —Portfolio size (1-3)
- —Portfolio reliability (4)
- —Portfolio integration (5)
- —Portfolio architecture (6)
- —Portfolio age (7, 8)
- —Portfolio language (9, 10)
- —Hardware and system software base (11, 12)
- —Use of software development tools (13, 14)
- —Level of innovation (15)
- —User population (16, 17)
- —Skills of the maintenance staff (18)
- —System experience of the maintenance staff (19)
- —Documentation of the systems portfolio (20, 21)

Note: numbers refer to relevant propositions presented in the text.

Portfolio size is obviously the major contributor to the magnitude of the maintenance task. Three propositions related to task uncertainty are suggested. (1) The larger the portfolio, measured in terms of number of systems, the greater the diversity of the maintenance task. This follows from the range of applications represented within the portfolio itself. (2) Further, the larger the systems in the portfolio, measured in terms of amount of source code, the greater the task difficulty (Lientz and Swanson, 1980; Gremillion, forthcoming). This is a result of the complexity which is characteristic of larger systems. And (3) the greater the variance in system size within the portfolio, the greater the task diversity. In this case, it is recognized that systems of different size characteristics pose different maintenance problems.

It is also well understood that (4) the less reliable the software in the portfolio, the greater the maintenance task variability and difficulty. Lack of reliability may be attributed to software faults (i.e. bugs), and these lead to unpredictable outcomes, and require maintenance of the corrective type.

Level of integration of the system portfolio should also affect the maintenance task. Specifically, (5) the greater the level of integration of the portfolio, the greater the task difficulty. This follows from the necessity to maintain consistency of assumptions across systems.

Whatever the level of integration of the portfolio, however, consistency of architectural design will also be an influential factor. The use of alternative approaches to modularization and structured program design should be particularly important, for example (Vessey and Weber, 1983). Thus, (6) the less the commonality of design within the portfolio, the greater the task of diversity.

Portfolio age is also an important consideration (Lientz and Swanson, 1981; Guimaraes, 1983). Two additional propositions follow. (7) The older the systems in the portfolio, the greater the task difficulty. (8) The greater the variance in age within the portfolio, the greater the task diversity. In general, the effects of age upon the maintenance task are indirect, rather than direct. Thus, for example, variance in age within the portfolio tends to be associated with lack of design commonality, and thereby greater task diversity. For this reason, the propositions relating to age may be useful primarily for their diagnostic, as opposed to their explanatory utility.

The language of the portfolio source code also affects maintainability (9). The more aged the languages of the portfolio, assessed in terms of language generations, the greater the task difficulty. On the other hand, transition to more current languages often entails another consideration. Namely, (10) the greater the number of languages employed within the portfolio, the greater the task diversity.

A similar argument follows for the portfolio's system software and hardware base. (11) The more aged the system software and hardware base, assessed in terms of equipment generations, the greater the task difficulty. By the same token, transition to current generations involves the problem that (12) the greater the number of systems in the base, the greater the task diversity.

The employment of software tools in the development of the portfolio is also a factor of significance. In principle, (13) the greater the use of appropriate development tools, the less the maintenance task difficulty. However, (14) the greater the variety of development tools applied, the greater the task diversity. Again, a balance must thus be sought between two basic considerations.

A closely related consideration is the level of innovation in the portfolio. In general, (15) the greater the level of innovation within the portfolio, the greater the maintenance task difficulty and variability. This is inherent to the nature of the innovation process. In the case of application system software, innovations in both software development technology and functionality are of relevance in this regard.

The composition of the user population served is of particular importance in understanding the maintenance task. In general, (16) the larger the user population served, the more diverse the maintenance task. And (17) the greater the turnover in the user population served, the more variable the maintenance task. These consequences follow naturally from the usual variance in user need within a user population, on the one hand, and those changes in priorities characteristic of normal organizational turnover, on the other.

Summarizing to this point in the discussion, elements of maintenance task uncertainty have been identified as associated with the application system portfolio, the technological base, and the user population served. These elements are the important determinants of the amount of information required in task performance. However, task uncertainty is understood to be the difference between this requisite information and that which is already possessed by the organization. (Galbraith, 1973). Thus, other elements of task uncertainty are to be indentified by considering the knowledge base of the organization.

The maintenance staff itself is an important source of task knowledge, and task uncertainty. Fundamentally, (18) the lesser the skill level of the maintenance staff, the greater the task difficulty. On the other hand, skills which are inapplicable to the systems maintained provide no advantage.

Experience by the maintenance staff with the systems being maintained is particularly important. Thus, (19) the less the systems experience of the maintenance staff, the greater the task difficulty. This occurs because systems experience is the primary basis for retention and extension of knowledge about the application system portfolio. Evidence in support of this proposition is reported in Lientz and Swanson (1980).

Knowledge about this application system portfolio may also be shared by means of documentation methods. Therefore, (20) the more complete, consistent, and current the documentation, with respect to the software being maintained, the less the task difficulty. On the other hand, (21) the more demanding the documentation process, the greater the difficulty of the maintenance task. One difficulty must thus be borne in amelioration of the other.

As summarized in Table 1, thirteen sources have been related by means of twenty-one propositions to maintenance task variability, diversity, and difficulty, and thereby to overall maintenance task uncertainty. For present purposes, the suggestion is that these propositions be admitted as working hypotheses by means of which maintenance task uncertainty may be indirectly assessed (e.g. through measures of portfolio size and age, user population size, etc.). This uncertainty presents an information processing requirement to the IS organization. The necessary response to this requirement takes the form of a choice among organizational design alternatives. In the section to follow, the theory associated with these alternatives is sketched.

Basic Organizational Design Alternatives

A summary of Galbraith's basic organization design strategies is presented in Table 2. The general thesis is that an organization composes its particular strategy from the alternative components listed, according to the level of task uncertainty it faces. A brief discussion of the basic component alternatives follows. For a complete treatment, the reader is referred to Galbraith (1977).

The organization's hierarchy of authority provides the fundamental means of resolving organizational uncertainties. Because the hierarchy is suject to overload, even where substantial discretion is exercises at lower levels, rules and procedures are further adopted, providing policy guidance for decision makers. Repetitive decisions may thus be moved to lower levels of the organization. Planning and goal setting are a third design alternative, and are used for the same general purpose. Their particular virtue is the efficient use of the information processing capacity of the hierarchy, through a focus of the substance of the communication.

Table 2

Galbraith's Basic Organizational Design Strategies

—Hierarchy of authority
—Rules and procedures
—Planning and goal setting
—Narrowing span of control
—Environmental management
—Creation of slack resources
—Creation of self-contained tasks
—Investment in management information system
—Creation of lateral relations

Note: adapted from Galbraith (1977).

Where the hierarchy is subject to increasing overload, the span of control of the manager may also be reduced. This should be recognized as essentially equivalent to an increase in the size of the hierarchy itself.

In response to yet more task uncertainty, the organization must adopt other alternatives for reducing its need for information processing, or for increasing its capacity for the same. Environmental management provides one basic alternative for reducing uncertainties about external events. Cooperative schemes including contracting, coopting, and coalescing are included here among others in this category of activity (see Galbraith, 1977).

The creation of slack resources may also be used to reduced the need for information processing. This is achieved through a relaxation of constraints upon performance, e.g., through the extention of project due dates, or the easing of quality controls. Such relaxation of constraints makes problematic situations less frequent, and thereby reduces the information load which must be borne by the hierarchy.

Self-contained tasks constitute another design alternative for reducing the amount of information which must be processed. Here, the idea it to shift the organization from a functional division by labor type to one in which each unit has all the resources needed to perform its task. Organization according to product or service category typically results.

In contrast to the above strategies, investment in management information systems may be undertaken to increase the organization's capacity to process information. Such investment enables managers to replan more frequently than would otherwise be possible, and to apply greater computational sophistication to the production of the plan as well.

Lastly, lateral relations may be employed to enhance the information processing capacity of the organization. In this case, the establishment of lateral communication channels provides means for uncertainty resolution across organizational units, as an alternative to upward hierarchical referral. Among the various lateral relations alternatives are: direct lateral contact, the liaison role, the task force, the team, the integrating role, the managerial linking role, and the matrix organization (Galbraith, 1977).

Importantly, the basic design mechanisms sketched above are posited by Galbraith to be exhaustive of the alternatives open to the organization. A strong prediction follows: Faced with an increase in task uncertainty, the organization necessarily adopts one or more of the alternatives. The default alternative is the creation of organizational slack through a reduction in required level of performance. The rational alternative is the least-cost one, among those which suffice to meet the information processing requirement. Relative costs are understood to vary according to circumstances, and thus the strategies actually chosen may reasonably vary across organizations with comparable levels of task uncertainty.

Organizational Design for Maintenance

The problem of organizational design for maintenance arises with the natural growth of the application system portfolio. The installed systems which constitute this portfolio shape the nature and magnitude of the maintenance task, in addition to providing the point of departure for further new system development. The growth of the workforce to maintain and develop the portfolio motivated increasing structural differentiation of the organization, and a corresponding elaboration of the management hierarchy as well. This hierarchy is the ultimate recourse for resolution of problems associated with task uncertainty as described earlier.

Faced with various levels of task uncertainty, the IS management hierarchy may be expected to adopt design strategies across the full range of alternatives described in Table 2. In this section, the literature of software maintenance is examined, in terms of these strategic alternatives. (For a guide to the software maintenance literature, see Swanson *et al.* 1984.) Motivating this examination is the conjecture that our common knowledge about management of maintenance, that which has been obtained both from research and from practice, had indentifiable foundations in the organization theory sketched above.

Basic alternatives for organizing the maintenance hierarchy of authority have been described by Markus (1984):

(i) the inclusion of the maintenance unit within the computer operations function; (ii) the inclusion of the maintenance unit within the software development function; and (iii) the distribution of maintenance across projects, in a project form or organization. Various hybrids of these basic alternatives are often adopted in practice.

The adoption of maintenance rules and procedures is frequently recommended by practitioners concerned with the managment of maintenance (see e.g., Perry, 1981). Consistent with such advice, widespread use of procedures for the recording and documention of user change requests, operational trouble reports, and changes to the programs themselves have been confirmed by Lientz and Swanson (1980). Such basic recording and documentation procedures serve to order the execution of the maintenance task, facilitating coordination and minimizing the need for management referral.

Additional examples of procedures include the acceptance test procedures by which software is formally transferred from development to operation and maintenance, and the establishment of life cycle audits which are employed to provide continuing, periodic "cradle to grave" reassessment of all application systems in the portfolio. Evidence of the usefulness of formal, periodic audits in managing the software maintenance burden has been reported by Lientz and Swanson (1980).

The adoption of criteria for maintenance management, such as those proposed by Arnold *et al.*, (undated), constitutes an excellent example of planning and goal setting in support of maintenance. As discussed above, this form of activity is designed to focus management communication about the maintenance task, and hence to reduce the information load borne by the hierarchy.

No evidence on the adjustment of the span of control in maintenance management was uncovered in our survey of the literature. However, in one case currently under study in which the maintenance unit was recently subdivided, thus halving the span of control, advantages for managerial information processing were well understood.

Strategic alternatives for reducing the organizational need for information processing include environmental management, creation of slack resources, and creation of self-contained tasks. Applications of these concepts in a maintenance context are described next.

The use of scheduled maintenance, the batching and implementation of software changes on a regular, periodic basis has been recommended by Lindhorst (1973) and others (including McNeil, 1979, who advocated the concept of a "system release discipline"). This technique provides a basic means of environmental management. The important effect is the reduction of maintenance task variability. The demand for maintenance, which is characteristically incremental, fragmented, and unpredictable, is channeled by means of a preexisting decision structure into a pattern around which maintenance work may be organized with increased certainty.

Various organizational means for encouraging "user involvement" may also be seen as fundamentally directed toward environmental management. An example is the user steering committee (Nolan, 1982). Established by a number of IS organizations to assist in setting work priorities, it also provides a basic means of user cooperation in that senior user representatives are presented with the full set of work demands placed upon IS management and asked to share in the responsibility for allocation of scarce IS resources. The desired long-term effect of such cooperation is a reduction in maintenance task diversity and variability through the development of a common view.

Perhaps the most extreme form of environmental management is the purposeful adjustment of strategy to create a new domain of organizational activity and exchange relationships. One example of such "environmental maneuvering" in maintenance is the proposal for user-developed (and, by implication, user-maintained) systems (McLean, 1979). Such a strategy, appropriately termed "off-loading," reduces IS task uncertainty through all three of its components. (Whether it reduced task uncertainty for the organization as a whole is another matter altogether!)

The creation of slack organizational resources provides another means of reducing the need for information processing in maintenance management. The avoidance or delay of scheduled commitments through the creation or the "applications backlog" constitutes one example, and characteristically represents the default alternative. As long as user requests for current system improvements are consigned to the backlog, their associated task uncertainty need not be faced. This is sometimes enabled by classifying requested improvements as enhancements, requiring justification comparable to that for new system development, rather than as maintenance to which the IS organization is already obligated. Alternatively, a priority is assigned which ensures that the request continues to fall to the "bottom of the stack." A form of environmental management is thus also achieved by this alternative. Slack resources may also be created by arranging for contract maintenance personnel to be provided to meet those exigencies of demand naturally associated with overall task veriability.

The creation of self-contained tasks also has application in the management of maintenance. Where there is a substantial application portfolio, but relatively low task uncertainty, economies of specialization and scale will tend to favor a functional division of labor, e.g., the separate organization of systems analysis and programming. However, were uncertainty is high for the reasons indicated above, the motivation exists to create self-contained tasks which are resource self-sufficient.

A basic management alternative in this regard is overall organization by area of application, with systems analysis and programming integrated within each area unit. This organizational form achieves a reduction in information processing need through the effective partitioning of overall task diversity. However, responsibility for maintenance of current systems is combined with new system development within each area, offering further opportunity for creation of self-contained work units.

A second alternative, which may be combined with the first, is the integration of maintenance and enhancement work on current systems, again organized within area of application. In some organizations enhancements are treated as "new development," subject to a division of labor other than that applied to routine maintenance work. Integration of maintenance and enhancements thus provides efficiency gains through elimination of the communication required to support hierarchical decision making around this distinction.

The integration of maintenance and enhancements takes an inclusive view of the nature of maintenance work. Where there is substantial task magnitude as well, and reasonable independence of current and future systems, the unitary maintenance organization constitutes another self-contained task alternative. (Lientz and Swanson, 1980). In this case, all maintenance work on a set of current systems is undertaken by one organization with our new system development responsibilities. The result is a reduction in information processing requirements through a corresponding reduction of task diversity and variability.

One final illustrative self-contained task alternative deserves mention—the concept of the information center (Hammond, 1982; McCartney, 1983). A principal motive in establishing such a center is the provision of a special-purpose facility for responding to ad hoc user requests for reports which draw from the organizational database. An intended benefit is the relief of this burden from the shoulders of those already committed to other maintenance and new system development work. This is easily understood as another means of partitioning task diversity, and thereby reducing the need for information processing for this type of service.

Strategic alternatives for increasing the IS organization's capacity to process information include investment in management information systems and the creation of lateral relations as described above. Illustrative alternatives in a maintenance context are described next.

High task uncertainty in maintenance may first be absorbed by means of a computer-based management information system designed specifically to support the maintenance management function. The need to monitor the maintenance effort has been emphasized by Lientz and Swanson (1980). In this context, the growth of the maintenance portfolio and the task diversity thereby generated, makes a maintenance database a particularly worthwhile effort.

The development of maintenance metrics is also an important means of increasing information processing capacity in maintenance management. Such metrics are fundamental to the development of the maintenance database (Swanson, 1976). Illustrative is the "Index of Difficulty" described by Berns (1984) for use by the maintainability analysis tool (MAT).

Use of liaison roles in the creation of lateral relations may also be exploited in organizational design. As one example, in situations where maintenance of existing systems is organized separately from new system development, the maintenance escort provides a basic design alternative (Lientz and Swanson, 1980). The maintenance escort may be based in either the maintenance or new system development unit. In both cases, the individual serves in the new system unit during original development, and then accompanies the system upon its transfer to the maintenance unit. Where the individual is based in the new system unit, he or she returns to that unit after the system is fully absorbed by the maintenance organization. This concept has been borrowed from development and manufacturing engineering, where it is well established in practice. The basic virtue of the escort role is the establishment of a lateral channel of communication between new system development and maintenance, and the provision of a vehicle for knowledge transfer through rotating personnel assigments. The effect is to moderate maintenance task difficulty.

The quick-response team, formed to perform fire-fighting maintenance of the corrective type (Embry and Kennan, 1983), may profit first of all from the virtues of a self-contained task, as described above. Because of its limited responsibilities, it can specialize its skills and procedures accordingly. A reduction of maintenance task diversity is achieved. However, the creation of lateral relations may also be involved. This may occur, for example, where personnel from the data processing center are included on the team in order to facilitate the incorporation of quick fixes without recourse to use of the management hierarchy.

Table 3

Organizational Design Alternatives in Maintenance Management

- —Maintenance hierarchy of authority
- —Basic recording and documentation procedures
- —Acceptance test procedure
- —Lifecycle audit procedure
- —Establishment of criteria for maintenance management
- —Scheduled maintenance
- —User steering committee
- —Off-loading of maintenance
- —Applications backlog
- —Contracted maintenance
- —Organization by area of application
- —Integration of enhancements with routine maintenance
- —Unitary maintenance organization
- —Information center
- —Maintenance management information system
- —Establishment of maintenance metrics
- —Maintenance escort
- —Quick-response team
- —Quality assurance group

A final illustration of a lateral relations design is the quality assurance group, a team of individuals often drawn from maintenance programming, data processing, and user units for the purpose of controlling software quality (Schwartz, 1982). Here, the sharing of quality information by members of the team adds to the information processing capacity of the IS organization as a whole through the provision of this special-purpose communication vehicle.

This completes our brief examination of organizational design alternatives in software maintenance. A total of 19 alternatives has been identified, as summarized in Table 3. While the list is not exhaustive, it should suffice for present purposes. Before concluding, it should be mentioned that other factors of importance, specifically, choices of processes for integrating individuals in the organization (viz. personnel selection and training, and reward systems) are also of significance in the present context, as indicated especially by the research of Couger and Coulter (1983). Here we have focused our attention primarily on what Galbraith terms "organizing modes." This limit to our examination must be acknowledged.

In our conclusion, we draw a number of implications for information systems research from the above discussion.

Conclusion

As documented above, recent years have seen the emergence and adoption of a variety of organizational designs for software maintenance management. In the present context, this penomenon may be understood to be a necessary organizational response to the task uncertainties associated with the continuing growth and gradual recognition of the "maintenance iceberg." As applied here, the Galbraith theory provides the basis for this understanding. Further, it predicts that as maintenance portfolios continue to accumulate, task uncertainties will motivate further development and refinement of organizational design alternatives across the range of types identified above. Future research should monitor and assist this process, in support of practicing information systems managers.

Two complementary avenues of empirical research are proposed. The first is directed toward the examination of comparative maintenance environments, organizational design choices and associated consequences, by means of multiple field studies conducted in a variety of host organizations. These field studies should gather common quantitative data across organizations, but should also seek to identify and document those qualitative aspects of the organization which are necessary to the understanding of each unique case. In this way, a collection of cases might be assembled to provide the foundation for a well-grounded and developed contingency theory of software maintenance management.

The second avenue of research would build from that of the first by studying, over time, the process of change in maintenance task environments, organizational design choices, and associated consequences for a few selected organizational cases. This work would seek to understand the dynamics rather than the statics of maintenance management. Of particular interest in this regard should be the study of organizational design experiments conducted for learning purposes. Thus, this work should also contribute to a well-grounded and useful contingency theory.

In summary of this paper, a review of the software maintenance literature provides informal, but supportive evidence for the proposition that growth of an application system portfolio is characteristically accompanied by increased uncertainties in the software maintenance task, and that the information systems organization responds accordingly by adapting its design so as to reduce its need for information processing in maintenance tasks, or, alternatively, to increase its capacity for the same. That is, the Galbraith theory of organization design finds ready application to the management of software maintenance.

A variety of sources of uncertainty in the maintenance task have been identified, as have a number of organizational design alternatives in maintenance management. In the latter case, each alternative is seen to be interpretable within the Galbraith theory. Further, the theory itself is fully illustrated by the alternatives identified.

Most importantly, parsimony of theory has been achieved. It has been previously recognized that evolving maintenance environments pose complex problems and alternatives for management of the maintenance task. But these problems and alternatives were more a tangle of apparent relationships, than a coherent whole. This tangle may now be absorbed and understood. Further, new relationships may seek their place within the established structure.

Finally, the virtues of drawing upon existing organization theory are illustrated in the present case. Management of the information systems organization is seen to be not wholly unlike the management of organizations of other types. Hard-won theory extracted from the study of the latter should therefore not be ignored in seeking answers to questions which arise relative to the subject of our principal concern.

REFERENCES

Arnold, R.S., Parker, D.A., and Yeh, R.T. "The Dimensions of Healthy Maintenance," unpublished manuscript, undated.

Berns, G.M. "Assessing Software Maintainability," *Communications of the ACM,* Volume 27, Number 1, January 1984, pp. 14-23.

Canning, R.G. (ed.), "That Maintenance Icebergy," *EDP Analyzer,* Volume 10 1972.

Cougar, J.D. and Colter, M.A. *Motivation of the Maintenance Programmer,* Cysys, Inc., Colorado Springs, Colorado, 1983.

Embry, J.D. and Keenan, J. "Organizational Approaches Used to Improve the Quality of a Complex Software Product," Proceedings, Software Maintenance Workshop, Naval Postgraduate School, Monterey, California, December 6-8, 1983, pp. 87-89.

Galbraith, J. *Designing Complex Organizatins,* Addison-Wesley, Reading, Massachusetts, 1973.

Galbraith, J. *Organization Design,* Addison-Wesley, Reading, Massachusetts, 1977.

Gremillion, L. L. "Determinants of Program Repair Maintenance Requirements," *Communications of the ACM,* forthcoming.

Guimaraes, T. "Managing Application Program Maintenance Expenditures," *Communications of the ACM,* Volume 26, Number 10, October 1983, pp. 739-746.

Hammond, L. W. "Management Considerations for an Information Center," *IBM Systems Journal,* Volume 21, Number 2, 1982, pp. 131-161.

Lientz, B. P. and Swanson, E. B. *Software Maintenance Management,* Addison-Wesley, Reading, Massachusetts, 1980.

Lientz, B. P. and Swanson, E. B. "Problems in Application Software Maintenance," *Communications of the ACM,* Volume 24, Number 11, November 1981, pp. 763-769.

Lientz, B. P., Swanson, E. B. and Tompkins, G. E. "Characteristics of Application Software Maintenance," *Communications of the ACM,* Volume 21, Number 6, June 1978, pp. 466-471.

Lindhorst, W. M. "Scheduled Maintenance of Applications Software," *Datamation,* Volume 19, May 1973, pp. 87-89.

Markus, M. L., *Systems in Organizations: Bugs and Features,* Pitman: Boston, Massachusetts, 1984.

Martin, J. and McClure, C. *Software Maintenance,* Prentice-Hall, 1983.

McCartney, L. "The New Information Centers," *Datamation,* Volume 29, Number 7, 1983, pp. 30-46.

McLean, E. R. "End Users as Application Developers," *MIS Quarterly,* Volume 3, Number 4, 1979, pp. 37-46.

NcNeil, D. H. "Adopting a System Release Discipline," *Datamation,* Volume 25, 1979, pp. 110-117.

Nolan, R. L. *Managing the Data Resouce Function*, West Publishing, St. Paul, Minnesota, 1982.

Perry, W. *Managing Systems Maintenance*, Q.E.D. Information Sciences, Wellesley, Massachusetts, 1981.

Schwartz, B. "Eight Myths about Software Maintenance," *Datamation*, Volume 28, August 1982, pp. 125-128.

Swanson, E. B. "The Dimensions of Maintenance," Proceedings of the Second International Conference on Software Engineering, San Francisco, October 13-15, 1976, pp. 492-497.

Swanson, E. B., Wu, C., and Gosnell, T. "Software Maintenance References," unpublished manuscript, March, 1984.

Vessey, I. and Weber, R. "Some Factors Affecting Program Repair Maintenance," *Communications of the ACM*, Volume 26, Number 2, February 1983, pp. 128-134.