Applications: Operations and Management
Howard L. Morgan* Editor

Problems in Application Software Maintenance

Bennet P. Lientz and E. Burton Swanson
UCLA

The problems of application software maintenance in 487 data processing organizations were surveyed. Factor analysis resulted in the identification of six problem factors: user knowledge, programmer effectiveness, product quality, programmer time availability, machine requirements, and system reliability. User knowledge accounted for about 60 percent of the common problem variance, providing new evidence of the importance of the user relationship for system success or failure. Problems of programmer effectiveness and product quality were greater for older and larger systems and where more effort was spent in corrective maintenance. Larger scale data processing environments were significantly associated with greater problems of programmer effectiveness, but with no other problem factor. Product quality was seen as a lesser problem when certain productivity techniques were used in development.

Key Words and Phrases: application software maintenance, maintenance problem factors

CR Categories: 3.50, 4.6

Introduction

What are the principal problems in application software maintenance and how do these problems vary according to the application system maintained and the data processing environment in which maintenance takes place? This paper reports research results which provide some answers to these questions.

The results are based on a sample survey of 2000 managers of data processing organizations. Each manager was randomly selected from the membership of the Data Processing Management Association (DPMA) and mailed a questionnaire on application software maintenance. A total of 487 responses were received, a return rate of 24 percent, considered good given the demanding nature of the questionnaire.

The overall purpose of the survey was to explore five sets of issues associated with maintenance: (*i*) conceptual issues (the meaning of maintenance), (*ii*) scale-of-effort issues (resource level and allocation), (*iii*) organizational issues (task definition and assignment), (*iv*) productivity aid issues (impact on maintenance), and (*v*) problem area issues (relative importance in maintenance). These issues were developed from a review of the literature on maintenance [1].

Basic descriptive results of the survey are reported in [2]. Among the more important findings: (*i*) departments tend to spend about half of their applications staff time on maintenance; (*ii*) over 40 percent of the effort in supporting an operational application system is typically spent on providing user enhancements and extensions; (*iii*) the average application system is between three and four years old, consists of about 55 programs and 23,000 source statements, and is growing at a rate in excess of 10 percent per year; (*iv*) about half a man-year of effort is allocated annually to maintenance of the average system. (See also [3], [7].)

The present paper concentrates on issues associated with problems of maintenance, as assessed by data processing managers in responding to one item of the questionnaire. Analysis of the responses considered in conjunction with responses to other questionnaire items provides insight into both the factors underlying the maintenance problems and the variables that serve as determinants. These insights provide us with some essential building blocks for the development of a theory of software maintenance.

The Problem Items

The questionnaire sought the data processing manager's judgment of the problems in maintaining a selected application system. A list of 26 potential problems was provided and we requested that each be evaluated on a 1 to 5 point scale ranging from "no problem at all" to "major problem." The list of potential problems was identical to that used in a preliminary survey [4], with the exception of the last three items which were added on the basis of comments received in this first attempt.

A summary of the descriptive results is presented in Table I. The computation of means and variances is based upon the assumption of interval item scales. (For a summary of the same results, based upon ordinal item

* Former editor of Applications: Operations and Management, of which Alan Merten is the current editor.

This work was partially supported by the Information Systems Program, Office of Naval Research under Contract N00014-75-C-0266, Project Number NR 049-345.

Author's Present Address: Bennet P. Lientz and E. Burton Swanson, Graduate School of Management, University of California, Los Angeles, CA 90024

Table I. Distributions of Maintenance Problem Items.

Item	Label	Mean	Standard dev	Cases
1	Maintenance personnel turnover	2.2332	1.3356	446
2	Documentation quality	3.0000	1.3103	446
3	System hardware and software changes	2.0404	1.1739	446
4	Demand for enhancements and extensions	3.2018	1.1745	446
5	Skills of maintenance programmers	2.0807	1.1564	446
6	Quality of original programming	2.5897	1.3256	446
7	Number of maintenance programmers available	2.5762	1.3348	446
8	Competing demands for programmer time	3.0336	1.3395	446
9	Lack of user interest	1.8677	1.2171	446
10	System run failures	1.8677	0.9706	446
11	Lack of user understanding	2.6076	1.2670	446
12	Program storage requirements	1.9776	1.2158	446
13	Program processing time requirements	2.5538	1.2562	446
14	Maintenance programmer motivation	1.9170	1.1076	446
15	Forecasting maintenance prog. requirements	2.4552	1.2239	446
16	Maintenance programming productivity	2.0359	1.0866	446
17	System hardware and software reliability	1.8094	1.0438	446
18	Data integrity	1.9036	1.0840	446
19	Unrealistic user expectations	2.5516	1.2616	446
20	Adherence to programming standards	2.1143	1.0905	446
21	Management support	1.8453	1.1141	446
22	Adequacy of system design specs	2.4233	1.2606	446
23	Budgetary pressures	1.9798	1.2075	446
24	Meeting scheduled commitments	2.6861	1.2435	446
25	Inadequate user training	2.7623	1.2387	446
26	Turnover in user organization	2.3610	1.2351	446

scaling, see [2].) Problem items are listed in the order they appeared on the questionnaire. User demands for enhancements and extensions emerged as the leading problem, a result which validates the findings of the preliminary study [1].

The Problem Factors

A factor analysis of the responses to the 26 problem items was performed in order to explore the underlying dimensionality and to facilitate further analysis. The principal factor with iteration option was employed, with varimax rotation [5]. A classical factor analysis based on inferred factors was performed. The method of rotation was the commonly used one.

The factor analysis produced six factors that accounted for about half the total variance in the 26 problem items. These factors are summarized in Table II. The labels attached to the factors are the result of an interpretation of the factor loadings (not shown to conserve space).

On the whole, the results of the factor analysis were striking. The dimensions emerged with unexpected clarity and the dominance of the oft-cited user problem is remarkable. It is not simply that user problems are common to all; it is that user problems account for the major variance in the problems common to all. Particularly noteworthy is that problem item 4, user demand for enhancements and extensions, while it is the major problem item among all those cited, is *not* a significant component in the user knowledge factor which accounts for 59.5 percent of the common variance. The major contributors to the latter are lack of user understanding and inadequate user training.

For purposes of subsequent analysis, the six problem factors were formalized as indices, computed on the basis of problem items with factor coefficients of absolute value 0.200 or greater. Normalized values of the problem item scores were used in the indices [5]. A summary of the six factor indices and their problem item components is presented in Table III.

In interpreting these results and those that follow, the reader should keep in mind that the problems identified

Table II. Summary of Maintenance Problem Factors.

Factor	Label	Eigenvalue	Percent of Var	Cum Percent
1	User knowledge	7.25414	59.4	59.5
2	Programmer effectiveness	1.45230	11.9	71.4
3	Product quality	1.16047	9.5	80.9
4	Programmer time availability	0.97415	8.0	88.9
5	Machine requirements	0.76567	6.3	95.2
6	System reliability	0.58859	4.8	100.0

Table III. Problem Factor Indices and Their Item Components.

Factor Index	Item Component
1. User Knowledge	11. Lack of user understanding (0.363) 25. Inadequate user training (0.237)
2. Programmer Effectiveness	16. Maintenance programming productivity (0.369) 14. Maintenance programming motivation (0.349) 5. Skills of maintenance programmers (0.227)
3. Product Quality	22. Adequacy of system design specs (0.404) 6. Quality of original programming (0.321) 2. Documentation quality (0.272)
4. Programmer Time Availability	8. Competing demands for programmer time (0.785)
5. Machine Requirements	12. Program storage requirements (0.476) 13. Program processing time requirements (0.471)
6. System Reliability	17. System hardware and software reliability (0.440) 18. Data integrity (0.223)

Note: Factor score coefficients shown in parentheses.

here are those expressed by data processing management. User perceptions may or may not accord, a point that we will return to in the Conclusions.

Problem Determinants

What are the determinants of the problems of application software maintenance? The relationships between the problem factors and other maintenance variables were investigated. Among the potential determinants were application system size and age, magnitude and allocation of the maintenance effort, relative development experience of the maintainers of the system, use of productivity techniques in system development, use of a database management system, programming language, use of organizational controls, and the data processing environment. In the sections that follow, the results of the investigation are summarized for each variable.

Application System and Age

Five measures of application system size were obtained in the survey: number of programs, number of source statements, number of files, number of database bytes, and number of predefined user reports. Where parametric analyses were performed, natural logarithm transformations of these measures were judged necessary to meet normality assumption requirements.

Larger systems proved to be significantly associated with greater problems in maintenance. Of 30 first-order Pearson correlation coefficients computed between the six problem factors and the five measures of system size, 26 were positive of which 22 were significant at the $s \leq 0.100$ level and 14 were of magnitude $r \geq 0.100$. Programmer effectiveness demonstrated a notable positive association with all five measures of system size. Product quality was positively associated with four of the five size measures.[1]

The age of the application system, measured in terms of the number of months since the system became operational, was also obtained from the questionnaire. As with system size, first-order Pearson correlation coefficients between the six problem factors and system age were computed.

Older systems tended to be perceived as having greater problems in maintenance. In particular, system age was positively and notably associated with problems of product quality ($r = 0.142$, $s = 0.001$) and programmer effectiveness ($r = 0.128$, $s = 0.003$). The other correlations were not notable, however.

Though system size and age are seen to be strongly associated with the problems of maintenance, this association was shown in subsequent analysis to be explainable in terms of other, intervening variables, viz. magnitude and allocation of the maintenance effort and the relative development experience of maintainers of the system.

Magnitude and Allocation of the Maintenance Effort

The magnitude and allocation of the maintenance effort on the application system described were also included among the data obtained. Two measures of the magnitude of effort were obtained: the total number of individuals assigned (in whole or in part) to maintenance of the system, and the total number of man-hours expended annually. The allocation of the maintenance effort was indicated by a percentage breakdown of annual man-hours according to eight categories: (*i*) emergency program fixes, (*ii*) routine debugging, (*iii*) accom-

[1] Because of the sample size, many correlation coefficients of small magnitude were nonetheless statistically significant. The term "notable" refers here to coefficients of absolute magnitude $r \geq 0.100$ which was found to be an effective cutoff level for the focus of research attention among the many "significant" results.

modation of changes to data inputs and files, (*iv*) accommodation of changes to hardware and system software, (*v*) enhancements for users, (*vi*) improvement of program documentation, (*vii*) recoding for efficiency in computation, and (*viii*) others. The categories chosen were based on the classification system originally proposed by [6]. Within this system, emergency program fixes and routine debugging comprise *corrective maintenance*; accommodations of change represent *adaptive maintenance*; and user enhancements, improved documentation, and recoding for efficiency make up *perfective maintenance*.

As with the measures of application system size, natural logarithm transformations of the two measures of the magnitude of the maintenance effort were judged necessary for parametric analysis purposes.

The problems of maintenance were perceived to be greater, the larger the magnitude of the effort in maintenance. The first-order Pearson correlation coefficients are shown in Table IV. All twelve coefficients are positive, eleven of which are statistically significant at the $s \leq 0.100$ level and eight of which are of notable magnitude $r \geq 0.100$. The correlations between number of maintenance man-hours and the problems of programmer effectiveness ($r = 0.263$) and product quality ($r = 0.240$) are particularly striking.

Problems of maintenance were also perceived to be greater, the more relative time spent in corrective maintenance. Of twelve first-order Pearson correlation coefficients relating the six problem factors to relative time spent in emergency fixes and routine debugging, eleven were positive of which nine were statistically significant

Table IV. Maintenance Problem Factors and Magnitude of Maintenance Effort: First-order Pearson Correlation Coefficients.

	Annual man-hours	Persons assigned
PFactor1 (User knowledge)	0.1158* (451) P = 0.007	0.0965 (461) P = 0.019
PFactor 2 (Programmer effectiveness)	0.2625** (447) P = 0.000	0.2088** (456) P = 0.000
PFactor3 (Product quality)	0.2404** (499) P = 0.000	0.1099* (459) P = 0.009
PFactor (Programmer time availability)	0.1445* (453) P = 0.001	0.0879 (462) P = 0.029
PFactor5 (Machine requirements)	0.0949 (452) P = 0.022	0.0502 (462) P = 0.141
PFactor6 (System reliability)	0.1372* (4452) P = 0.002	0.1186* (461) P = 0.005

Asterisk key: ** indicates $|r| \geq 0.200$; * indicates $0.100 \leq |r| < 0.200$

and five were notable. Relative time in emergency fixes was positively associated with problems of product quality ($r = 0.200$, $s = 0.001$); user knowledge ($r = 0.130$, $s = 0.002$); and programmer effectiveness ($r = 0.117$, $s = 0.005$). Relative time in routine debugging was positively associated with problems of product quality ($r = 0.204$, $s = 0.001$) and programmer effectiveness ($r = 0.132$, $s = 0.002$).

Two other relationships involving the allocation of the maintenance effort with problem factors were of notable significance and magnitude. The problem of machine requirements was positively associated with both recoding for computational efficiency ($r = 0.164$, $s = 0.001$) and accommodating system hardware and software changes ($r = 0.106$, $s = 0.010$). Both these relationships are easily understood.

Interestingly, no notable findings related the percent time spent in providing user enhancements to any of the problems of maintenance, including that of user knowledge.

Relative Development Experience of Maintainers of the System

The questionnaire asked how many of the individuals currently assigned to the maintenance of the application system had previously worked on the development of this same system. The number who had, divided by the total, thus served as a measure of the relative development experience of the maintainers with respect to the system being maintained.

The computation of first-order Pearson correlation coefficients showed relative development experience to be significantly related to perceived problems in maintaining the application system. The most significant relationships indicate greater development experience to be associated with lesser problems with product quality ($r = -0.270$, $s = 0.001$) and lesser problems with programmer effectiveness ($r = -0.171$, $s = 0.001$). Lesser problems with user knowledge and programmer time availability were also indicated for higher levels of relative development experience but correlation coefficients were not of magnitude $r \geq 0.100$. Greater problems with machine requirements were indicated ($r = 0.104$, $s = 0.011$) for which there is no obvious interpretation. No relationship to the problem of system reliability existed.

To summarize thus far, system age and size, magnitude of the maintenance effort, relative allocation of effort to corrective maintenance, and the relative development experience of the maintainers are all shown to be associated with the problems of maintenance. However, these variables proved themselves to be interrelated; thus their impact upon the problems of maintenance must be considered jointly. To examine these effects, a series of multiple regression analyses were performed. Results confirmed that the magnitude of the effort in maintenance, the allocation of this effort to corrective maintenance, and relative development experience are all of importance in explaining the problems of maintenance. In particular, the number of maintenance manhours accounts for a substantial portion of the problem of programmer effectiveness and the relative development experience of the maintainers and the percent time on corrective maintenance (which includes both emergency fixes and routine debugging) are of similar significance in accounting for the problem of product quality.

However, results also indicate that system size and age have little influence upon the problems of maintenance, apart from their established impact upon the magnitude and allocation of the maintenance effort, and the relative development experience of the maintainers. When the latter variables have been entered first into the regression equations, measures of system size have notable influence only upon the problems of machine requirements and system reliablity, which together account for only 11.1 percent of the common problem variance. System age has no notable influence whatsoever.

The network of causal effects that appear to bear directly upon the problems of maintenance is shown in Figure 1.

Use of Productivity Techniques in System Development

The questionnaire also asked which of a variety of tools, methods, and techniques were employed in the development of the application system described. Included in the checklist were decision tables, database dictionary, test data generators, structured programming, automated flowcharting, HIPO (Hierarchy plus Input-Process-Output), structured walk-through, and chief programmer team. Provision for "others" to be indicated was also included.

One-way analyses of variance of the six problem factors according to the use of the productivity tools were performed. The results showed the problem of product quality to vary significantly according to the use of five of the eight tools listed. Specifically, the software product is perceived to be of better quality (in terms of the three components from which the factor is derived: system design specifications, programming, and documentation) where test data generators, structured programming, HIPO, structured walk-through, or the chief programming team have been employed. These results should be heartening to advocates of these techniques. However, it is also noteworthy that the problems of user knowledge and programmer effectiveness which together account for 71.4 percent of the common problem variance, are little affected through the use of these same techniques.

Use of a Database Management System

Whether the application system made use of a database management system was also asked in the questionnaire. Analyses of variance of the six problem factors according to the use of a database management system were thus performed. No significant variances were found and it may be concluded that management's assessment of the problems in maintenance is likely to be the same, on average, for application systems supported by a DBMS as for unsupported application systems.

Programming Language

The variances in the six maintenance problem factors according to the principal language in which the application system was programmed were also examined. Employing one-way analyses of variance, two problem factors were seen to vary significantly according to the programming language used: programmer effectiveness ($s = 0.010$) and system reliability ($s = 0.080$). In the case of programmer effectiveness, problems tend to be slightly greater than average when assembler languages are used and notably greater than average when Fortran or PL/1 is used. When Cobol is used, problems of programmer effectiveness are about average and when RPG is used, notably less than average. It appears at first that the use of certain languages is associated with greater problems in programmer effectiveness.

However, the interpretation of a direct causal relationship between the programming language and the problem of programmer effectiveness proves to be unwarranted. Other analysis suggests that the scale of the data processing environment may largely account for the apparent relationship, since language use varies according to the size of the data processing department which itself is associated with the problems of maintenance (as will be discussed shortly.) Specifically, a two-way analysis of variance of the problem of programmer effectiveness was conducted, controlling for the size of the data processing equipment budget, in addition to the programming language used. Results indicated that the main effects of programming language are not significant at the $s \leq 0.100$ level, when the size of the equipment budget is controlled. Thus, the apparent relationship between programming language and the problem of programmer effectiveness is explained in terms of the data processing environment in which programming takes place. It may be that larger installations are characterized by more complex applications for which certain languages are better suited, and which, at the same time, present greater challenges to effective programming. However, other explanations are also possible.

In the case of system reliability, one-way analysis of variance showed the problem to be greater than average

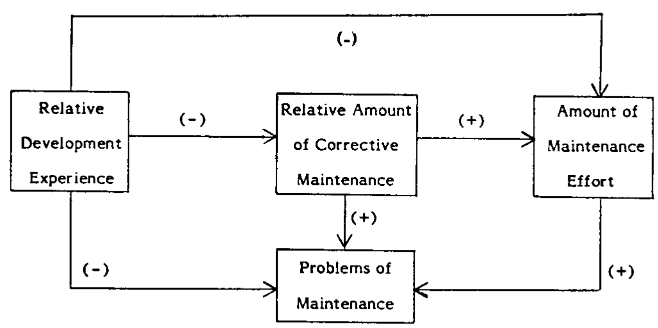

Fig. 1. Some Relationships Involving the Problems of Maintenance.

when Fortran is used, and less than average when RPG is used. For the other languages, the problem is about average. Since other analysis indicates the problem of system reliability does not vary significantly with the scale of the data processing environment, the present apparent relationship may not be explained in the same way as programmer effectiveness. Any interpretation is tenuous. It may be that RPG software is seen as more reliable for certain application programming than Fortran. Alternatively, since Fortran tends to be associated with the older application systems, the reliability problem may simply reflect this age differential.

Use of Organizational Controls

Also included on the questionnaire was a checklist of organizational controls which might be established for the maintenance of the application system. Listed were: (*i*) logging and documentation of user requests, (*ii*) cost justification of user requests, (*iii*) logging and documentation of troubles in operational processing, (*iv*) logging and documentation of changes to programs, (*v*) formal retest procedure for program changes, (*vi*) batching of program changes according to a predetermined schedule, (*vii*) periodic formal audit, (*viii*) equipment cost charge-back system, (*ix*) personnel cost charge-back system. Respondents were asked to indicate which of the above controls were used in the maintenance of the application system described.

One way analyses of variance of the six problem factors according to the use of organizational controls were first performed. A total of 16 relationships were established at significant levels ranging from $s < 0.001$ to $s = 0.070$. Of these, 13 relationships were positive, suggesting at first that the use of organizational controls is associated with *greater* problems in maintenance.

However, further analysis showed these results to be explainable in terms of previously established relationships. When these relationships are taken into consideration, the use of organizational controls proves to be little related to the problems of maintenance. Specifically, multiple regression analysis indicated that when the effects of the magnitude and allocation of the maintenance effort and the relative development experience of the maintainers were first accounted for, few significant associations between the problems of maintenance and the use of organizational controls exist. An exception is the periodic audit which emerges as significantly associated with lesser problems of user knowledge and product quality.

The Data Processing Environment

The first part of the questionnaire sought to establish the data processing environment in which the maintenance of the application system took place. The industry served by the data processing department was identified. The scale of the department was specified, in terms of both personnel and annual equipment budget, and the overall organization and allocation of staff time between maintenance and new system development activities was indicated. The demands of maintenance on the data processing manager's own time was assessed as was the current level of departmental staffing.

The scale of the data processing department as measured by the annual equipment budget proved to be strongly related to the perceived problems in maintaining the application system described. Specifically, analysis of variance showed the problem of programmer effectiveness to be perceived as greater, the larger the scale of the department ($s < 0.001$, linearity also significant at the $s < 0.001$ level). No significant variances in the other five problem factors were found. Thus, the perceived problems of maintenance vary by the size of the organization along the single dimension of programmer effectiveness. Two interpretations are suggested for consideration. The straightforward interpretation is that smaller organizations do have greater programmer effectiveness, possibly because the simplicity of work coordination outweighs the lack of technical specialization. An alternative interpretation is that programmer effectiveness does not vary but awareness of the "problem" is heightened according to the visibility of the data processing budget. Other interpretations may also be possible.

Maintenance problem factors were also correlated to the percent time spent on maintenance in the organization as a whole. As might be expected, problems are seen to be more severe, the more of the organization's time is allocated to maintenance. Four of the six problem factors, accounting together for 88.9 percent of the problem variance, are of notable significance and magnitude: programmer effectiveness ($r = 0.191$, $s = 0.001$); product quality ($r = 0.158$, $s = 0.001$); user knowledge ($r = 0.113$, $s = 0.006$); and programmer time availability ($r = 0.101$, $s = 0.013$). System reliability is also positively related, although the magnitude is not notable. The problem of machine requirements is unrelated.

Finally, analyses of variance of the six problem factors by the perceived level of staffing sufficiency and the demands of maintenance on the manager's own time were performed. It was found that four problem factors varied significantly by perceived level of staffing sufficiency: user knowledge ($s = 0.007$), programmer effectiveness ($s < 0.001$), product quality ($s < 0.001$), and programmer time availability ($s < 0.001$). In each case, linearity was also significant at the $s = 0.003$ level or better and it may be concluded that each problem tends to be perceived as greater, as the staffing level is regarded as less sufficient. These are strong results, though not particularly surprising.

In the case of the demands of maintenance on the manager's own time, four problem factors also varied significantly: user knowledge ($s = 0.005$), programmer effectiveness ($s = 0.055$), product quality ($s < 0.001$), and system reliability ($s = 0.076$). Again, linearity was also significant in each case, here at the $s = 0.016$ or better level and it may be concluded that each problem tends to be perceived as greater, the more of the manager's own time is absorbed by the demands of maintenance. As before, these results are what might be expected.

Conclusions

User knowledge has been seen to account for the majority of the variance in data processing management's assessment of the problems in maintenance. This result provides further evidence of the importance of the relationship between the users and the providers of information systems in the determination of system success or failure.

Lack of user understanding and inadequate user training are the two components of user knowledge and both suggest an estrangement of the users from the systems intended to serve them. It is this aspect of the oft-cited user problem which distinguishes systems with relatively lesser maintenance problems from those with greater maintenance problems.

User demands for enhancements and extensions is seen as the most severe problem item overall. This item is *not* a component of the user knowledge factor, although it may be somewhat related. It does not account for variance in the problems of maintenance. Rather, it is the most common complaint. That it is such is perhaps understandable when it is recalled that user demand is not controllable by data processing management, but has substantial implications for its resource allocation decisions.

The potential determinants investigated proved most strongly related to problems of programmer effectiveness and product quality. When the nature of these determinants is considered, this result is, in part, understandable. The determinants consist primarily of the characteristics of the software, the programming staff, the programming effort, and the programming environment. Characteristics of the user environment were not included. Such characteristics would presumably be more strongly related to the problem of user knowledge.

The direction for future research seems clearly indicated. Attention should be focused on the users of data processing applications. Characteristics of user environments in which relatively successful applications exist should be identified. These characteristics should then be related to their impact upon the maintenance of the application software as performed by the data processing department.

A likely approach would be to study the problems of ongoing systems from the viewpoints of users as opposed to data processing professionals. Problem factors from the users' perspective might be identified and correlated to the corresponding problem factors as seen by data processing management. An interesting question is whether the user knowledge problem factor would be recognized by the users themselves. If not, how would the users see the problem? Further, what characteristics of user environments would explain the variance in their problem perceptions?

By approaching the user problem from both sides, a more enlightened understanding of the problems of application software maintenance should ultimately be expected.

The development of a unifying theory of software maintenance remains a long-term prospect. In the present paper we have taken some initial inductive steps in this direction. Variables of importance have been identified, and certain relationships established. The picture is by no means complete; much remains to be done. However, a number of building blocks for the theory have been provided.

Received 8/79; revised 9/80; accepted 5/81

References

1. Lientz, B.P., and Swanson, E.B. Discovering issues in software maintenance. *Data Management, 16*, 10, (Oct 1978), 15–18.
2. Lientz, B.P., and Swanson, E.B. Software maintenance: A user/management tug-of-war. *Data Management, 17*, 4, (April 1979), 26–30.
3. Lientz, B.P., and Swanson, E.B. Impact of development productivity aids on application system maintenance. Proceedings of a Conference on Application Development Systems, *Data Base, 11*, 3, (Winter–Spring 1980), 114–120.
4. Lientz, B.P., Swanson, E.B., and Tompkins, G.E. Characteristics of application software maintenance. *Comm. ACM, 21*, 6, (June 1978), 466–471.
5. Nie, N.H., Hull, C.H., Jenkins, J.G., Steinbrenner, K., and Bent, D.H. *SPSS: Statistical Package for the Social Sciences.* 2nd Ed., McGraw Hill, New York, 1975.
6. Swanson, E.B. The dimensions of maintenance. *Proc. 2nd Int. Conf. on Software Engineering*. San Francisco, October 13–15, 1976.
7. Lientz, B. P., and Swanson, E. B. *Software Maintenance Management*. Addison-Wesley, Reading, MA, 1980.