Application Software Maintenance Characteristics

Dr J Rajaram, Professor Ph.D ----CSE
E Ravi Assistant Professor M.Tech—CSE
M Arya Bhanu, Assistant Professor
Dr M Bal Raju, Professor
Nagole institute of engineering and technology, Hyderabad.

abstract

aintenance and enhancement of application software consume a major portion of the total life cycle cost of a system. Rough estimates of the total systems and programming resources consumed range as high as 75-80 percent in each category. However, the area has been given little attention in the literature. To analyze the problems in this area a questionnaire was developed and pretested. It was then submitted to 120 organizations. Respondents totaled 69. Responses were analyzed with the SPSS statistical package. The results of the analysis indicate that: (1) maintenance and enhancement do consume much of the total resources of systems and programming groups; (2) maintenance and enhancement tend to be viewed by management as at least somewhat more important than new application software development; (3) in maintenance and enhancement, problems of a management orientation tend to be more significant than those of a technical orientation; and (4) user demands for enhancements and extension constitute the most important management problem area.

Key Words and Phrases: software maintenance, use of productivity aids, management and technical issues CR Categories: 3.50, 4.6

introdutions

maintenance and enhancement of operational application software systems is frequently viewed as a phase of lesser importance than the design and development phases of the system life cycle. Maintenance and enhancement are generally defined as activities which keep systems operational and meet user needs (see, for example, Riggs [20]). A characterization of three types of maintenance activities has been presented by Swanson [22]. Briefly, these activities are: corrective maintenance (performed in response to the assessment of failures); adaptive maintenance (performed in anticipation of change within the data or processing environments); and perfective maintenance (performed to eliminate inefficiencies, enhance performance, or improve maintainability). There have been a number of estimates of the amount of effort that goes into maintenance and enhancement. Riggs [20] cites a range of 40-60 percent of total systems and programming resources. Similar figures have been given in [5, 8, 11, 23]. An estimate as high as 75 percent of resources has been cited in [9] and [19]. A more conservative estimate of 40 percent has been given in [12, 13], and by Boehm [2]. A more recent estimate by Boehm [4] is 70 percent. Some of the specific problems in maintenance and enhancement have been the effect of hardware changes (Boehm [21]) and errors introduced with modifications (Kosy [14]). Studies involving specific software systems include [21] and the excellent analysis of OS/360 by Belady and Lehman [1]. Some interesting ideas on maintenance have been stated by Brooks [6]. Other sources which take a management and implementation point of view include [7, 10, 16, and 17].

Collection

This section summarizes the data collection process as well as the general profile of respondents. The questionnaire appears in [13]. The process of data collection began with the construction of an initial questionnaire and a field test of five organizations. Refinements were made and the form used for the survey finalized. Some 120 organizations were contacted by telephone and asked to participate. Managers of systems and programming departments were identified and requested to complete the questionnaire with staff assistance. Questionnaires were then mailed out to those expressing an interest in participation. Follow-up calls were made if no response was pleted questionnaires returned was 69. This is a substantial percentage considering the length (35 pages) and depth of the questionnaire.

The questionnaire is composed of two parts. Part I deals with the systems and programming department and contains 12 questions in the following areas:

-- industry category

-- annual budget for software and hardware

-- number of personnel in department (systems analysts and programmers as well as aggregate)

-- division of tasks among staff in maintenance and new application work, and in analysis and programming

-- management structure -- current percentage of effort in maintenance

-- relative importance of maintenance compared to development

-- reallocation of effort between maintenance and development, given hypothetical budget increases and decreases

-- evaluation of adequacy of current levels of staffing

Analysis Results

This section is organized into the following categories: profile of respondents, tools and techniques employed, evaluation of maintenance, and interrelation of variables.

Profile of Respondents

Each respondent was asked to indicate the industry segment of their organization. A classification of the responses indicated: manufacturing, 27 (39.1 percent); and nonmanufacturing, 42 (60.9 percent). This distribution corresponds closely to that associated with a recent classification analysis of the organizational distribution of the journal of the Data Processing Management Association (37.7 percent manufacturing, 62.3 percent nonmanufacturing). However, some caution is in order in interpreting our selected sample as representative. Several questions were asked relative to data processing equipment and expenditures. The response on equipment was similar to the division of the market and was IBM (73.9 percent), Burroughs (8.7 percent), Honeywell (5.8 percent), NCR (4.3 percent), Univac (4.3 percent), and others (2.8 percent). The distribution of annual organizational budgets for hardware is given in Table I. It should be noted that these figures reflect total company expenditures, not simply departmental expenditures. Several questions were asked on how development and maintenance effort would be redistributed if the systems and programming staff were increased or reduced by certain percentages. The results are summarized below (Table II) and indicate that most additional resources would go to new development. Also, as expected, most budget reductions would occur in new development. Tools and Techniques Employed Respondents were asked to distribute the percentages of source code lines by language.

Tools and Techniques Employed

Respondents were asked to distribute the percentages of source code lines by language. As expected, the preponderance was in Cobol and Assembler. The distribution was:

Cobol	58.1 percent
Assembler	18.5 percent
RPG	10.2 percent
PL/I	3.1 percent
Fortran	2.6 percent
Algol	1.5 percent
Other	6.0 percent

A somewhat frequently made assertion in the literature is that productivity tools in design and programming are not yet widely employed in practice. This is substantially borne out in the percentages given in Table III. In Table III the most frequently used tool is decision tables (46.4 percent). Other tools in use by at least 30 percent of respondents included test data generators, online programming, and chief programmer teams. It is interesting to note that approximately one quarter of the sample indicated that they use structured programming. Responses other than those in Table III include modular programming, top-down testing, online simulator, copy library, and technical design review. It should be noted that the percentages from Table III reflect operational application systems; for systems currently being developed, the figures might be somewhat higher.

Evaluation of Maintenance

The respondents were asked to contrast the relative importance of maintenance with new system development within their organizations. A response summary appears in Table IV. It indicates that most view maintenance as more important than new development. More strikingly, few view new system development as more important. Respondents were further asked to rank possible problem areas in maintenance. This is summarized in Table V. The table colums are arranged by problem area, statistics, and relative frequency. The statistics are based on the coding: l--not a problem; 2--somewhat minor problem; 3--minor problem; 4--somewhat major problem; 5--major problem. Items marked with an asterisk indicated technical problem areas. The only problem cited by the majority as more than minor is that of user demands for enhancements and extensions. Following this are two technical issues (quality of original program and quality of documentation) and one management issue (competing demands for personnel time). Frequently mentioned problems such as hardware change, turnover of maintenance personnel, and motivation of maintenance personnel showed up surprisingly low (means of 2.14, 2.46, and 2.03, respectively).

It is particularly interesting that maintenance programming productivity is not considered by management to be more than a somewhat minor problem. Given that quality of original programs and quality of documentation rank relatively high as problem areas, it would seem reasonable to expect that an increased investment in quality in the design phase would yield subsequent productivity increases in the maintenance phase. It is not clear whether management recognizes such a potential for productivity increases. In ranking the maintenance productivity problem relatively low, management may simply be saying that the programmers are productive, given what they have to work with. In addition to the 24 areas that are mentioned in the questionnaire, respondents were encouraged to list other problem areas. Areas mentioned included quality of operations personnel, turnover in user organization, high learning curve due to large system, and retaining personnel at implementation time. Maintenance and enhancement Percentage By far more important 33.3 Somewhat more important 21.7 Equal importance 34.8 Somewhat less important 5.8 By far less important 4.3 It is of interest to determine if management issues are more important than technical issues. This would serve as a guide in efforts to improve maintenance procedures and tools. Statistical tests indicate that management problems are more significant.

To carry out the tests, the average problem rating was computed for technical and management areas for each respondent. The Mann Whitney-Wilcoxan and sign tests were selected to test the hypothesis that the distribution of the average response in each category was the same. These tests do not depend on actual scores but relative ratings. For the Mann Whitney-Wilcoxan test the hypothesis was rejected at the c~ = 0.10 level. For the sign test it was rejected at the c~ = 0.01 level. Both results indicated higher values for the management areas. A second hypothesis is that the response to the problem of user demands for enhancements and extensions is significantly larger than the average for all problem areas. The same nonparametric tests were applied, and the hypothesis of the same distribution was rejected at the ct = 0.10 level. This indicates user demands for enhancements and extensions is more of a problem than other areas. As was mentioned in Section 2, some of the questions were followed by questions on the quality of the data on which the answer was based. The results are summarized by average and relative frequency in Table VI. An asterisk indicates technical subjects. A question here is whether there is less data available for management-type questions than for technical-type questions.

The results indicate respondents had firmer data for technical management types of questions. The statistical test was to test that the average responses to the management questions are based on data of a quality average equal to that of responses for technical questions. The nonparametric tests applied were the sign test and the Mann-Whitney Wilcoxan test. Both tests rejected the hypothesis at the a = 0.10 level. Similar tests (at c~ = 0.10 level) indicated that respondents knew more about effort in maintenance and enhancement in general than specific tasks within maintenance and enhancement. Interrelation of Variables The previous subsections of this section were concerned with responses to individual questions. This subsection examines the responses for interrelationships between response items. The analysis indicated that system characteristics.

Conclusions

From the analysis of the survey data several tentative conclusions are suggested. It should be emphasized that these are based on the limited sample. The conclusions are: Maintenance and enhancement do consume much of the total resources of systems and programming groups. -- Maintenance and enhancement tend to be viewed by management as at least somewhat more important than new application software development. -- In maintenance and enhancement, problems of a management orientation tend to be more significant than those of a technical orientation. -- User demands for enhancements and extensions constitute the most important management problem area. In general, more attention should be given to management problems associated with maintenance. In practice, maintenance work should be categorized to permit the gathering of more detailed management information. Project reporting systems should be detailed with respect to the type and tasks of maintenance and enhancement. The handling of user requests for enhancements should be examined to determine means Of better evaluating and satisfying requests. Research into software design and program construction techniques should give fundamental consideration to issues of maintainability. In particular, consideration should be given to designing with future enhancements and extensions in mind. Based upon the results reported here, the authors are currently pursuing a larger survey effort in cooperation with the Data Processing Management Association (DPMA).

References

!. Belady, L.A., and Lehman, M.M. A model of large program development. IBM Syst. J. 3 (1976),

2. Boehm, B.W. The high cost of software. Proc Symp on High Cost of Software, Monterey, Calif., 1973, pp. 27--40.

3. Boehm, B.W., Brown, J.R., and Lipow, M. Quantitative evaluation of software quality. Proc. 2nd Int. Conf. on Software Eng., Oct. 1976, pp. 592-605.

4. Boehm, B.W. Software engineering. IEEE Trans. Comptrs. C-25 (Dec. 1976), 1226-1241.

5. Brantley, C.L., and Osajima, Y.R. Continuing development of centrally developed and maintained software systems. IEEE Comptr. Soc. Proc. 45 (Spring 1975), 285-288.

6. Brooks, F.P. Jr. The Mythical Man-Month. Addison-Wesley, Reading, Mass., 1975.

7. Bucher, D.E.W. Maintenance of the computer sciences teleprocessing system. Proc. 1975 Int. Conf. on Reliable Software, April 1975, pp. 260--266.

8. Ditri, A.E., Shaw, J.C., and Atkins, W. Managing the EDP Function. McGraw-Hill, New York, 1971.

9. Elshoff, J.L. An analysis of some commercial PL/I programs. IEEE Trans. Software Eng. SE-2 (June 1976, 113-120.

10. Gildersleeve, T.R. Data Processing Project Management. Van Nostrand Reinhold, New York, 1974.