THE EFFECT OF SOFTWARE QUALITY ASSURANCE FUNCTION ON SOFTWARE MAINTENANCE COST

Rebecca Chung-Fern Wu *

ABSTRACT

Software maintenance is becoming the most expensive part of the software life cycle. This study attempts to examine the following research questions:

1. Does the SQA function have any impact on software maintenance cost reduction?
2. In which systems development phase does the SQA effort have the greatest impact on the reduction of software maintenance cost?

Software quality assurance function's performing review in an early software development phase is theorized in this paper to be effective for reducing maintenance cost. A field study of business application systems in financial institutions confirmed the importance of SQA function's conducting early reviews, especially in the requirement definition stage.

Key Words: software maintenance, software quality assurance, software engineering, system development review

* Associate Professor Department of Accounting College of Management National Taiwan University Special thanks go to professor E. Burton Swanson at University of California, Los Angeles for the constructive comments.

I. INTRODUCTION

Software maintenance, which refers to all modifications after an application system becomes operational, was estimated to cost about 70 percent of the total software life cycle cost, which includes software development, operation and maintenance costs (Brantley and Osajima, 1975; Zelkowitz, 1978). Lientz, Swanson and Tompkins (1978) found that 48 percent of annual person hours was spent on maintenance activities as opposed to 46 percent on new developments. They also indicated that over 20 percent of the cases in their survey allocated 85 percent of personnel hours to maintenance activities. Software maintenance is becoming the most expensive part of the software life cycle.

The keys to reducing the maintenance effort are these: accurately defining the user's requirements, assembling better systems documentation, using more effective designing methods, making better use of tools and techniques, better project management. However, even if the good software development practices are followed carefully, the analyst must not assume that the necessary quality standards have been met.

Software quality assurance is the review of software products and related documentation for completeness, correctness, reliability, and maintainability. It includes assurance that the system meets specifications and the requirements for intended use and performance.

A review performed by an independent and technically competent software Quality Assurance (SQA) function before a system becomes operational has been proposed to be a powerful technique for building in software maintainability, detecting discrepancies and monitoring corrective actions (Hamilton 1982; Martin and McClure, 1984). Lientz and Swanson (1980), from their survey of 487 data processing (DP) organizations, reported that periodic audits contribute to a reduction in software maintenance. They recommended review during early development

phases in order to identify various problems such as unclear user information requirements, design compromises and poor documentation, which were identified as causes of serious software maintenance problems.

II. THE STUDY

The establishment of a SQA function in the organization is one approach to aid systems development and evaluate information systems effectiveness. However, the effectiveness of SQA has never been empirically tested and little adequate research is directed toward guiding SQA in the system development phases. The purpose of this study is to understand and investigate the impact of SQA in three system development phases: definition, construction, and implementation (Ahituv and Newmann, 1982).

This study attempts to examine the following research questions:

1. Does the SQA function have any impact on software maintenance cost reduction?
2. In which systems development phase does the SQA effort have the greatest impact on the reduction of software maintenance cost?

In order to answer the above questions, a study was conducted of the system maintainers and the SQA staff. This study investigated computerized application systems in financial institutions located in Southern California that have SQA functions conducting the system development review for those application systems before their being turned over to production. The application must have been operational for one year or longer.

Forty-nine questionnaires were mailed. Each manager responsible for the system received the questionnaire. The total number of completed and useful questionnaires was thirty four. The questionnaire provided information relating to the application system and software maintenance. The questionnaire was pre-tested by three managers.

Table 1. The Study Variables

Variable	Operation Definition
Maintenance costs	the size of the maintenance budget for an application system.
SQA in definition phase	the total actual SQA person-hours in the construction phase.
SQA in construction phase	the total actual SQA person-hours in the construction phase.
SQA in implementation phase	the total actual SQA person-hours in the implementation phase.
System age	the total number of months since the application system became operational.
Maintainer's Relative Development Experience	the total number of maintainers who worked previously on the development of an application system
	the total number of maintainers of an application system
System size	the total number of programs currently included in an application system maintained.

SQA review effort on each development phase of a system was gathered from the project time-reporting system, which records all hours worked by the SQA staff allocated by project and by activity within project.

There are studies about the factors that contribute to the software maintenance effort. In general, the software maintenance effort was found to be explained, in part, by system age (Guimaraes 1983, Lientz and Swanson, 1980), maintainer's relative development experience (Lientz and Swanson, 1980), and system size (Vessey and Weber, 1983). Therefore these determinants must be controlled in the study in order to examine the unique contribution of SQA effort. Table 1 presents variables in the study and their operation definitions.

Table 2. Pearson Correlation coefficients among variables

Variables	1	2	3	4	5	6	7
1. System size	1.00						
2. Experience	-.22	1.00					
3. System age	-.19	-.49*	1.00				
4. SQA in Definition phase	-.28	-.08	-.17	1.00			
5. SQA in construction phase	.14	.03	-.16	.03	1.00		
6. SQA in implementation phase	-.31	-.26	-.00	.51*	.19	1.00	
7. Maintenance costs	.60*	.20	.30	-.56*	.06	-.59*	1.00

* significant at .10 level

The data were evaluated first for the four basic assumptions of the multiple regression method, namely, normality, linearity, homoscedasticity, and multicollinearity. The Shapiro-Wilk statistic, W, were tested for normality (Conover, 1971; Shapiro and Wilk, 1965). At a significance level of .01, the assumption of normality was rejected for maintenance costs and system size. To meet the normality assumption, the natural logarithmic transformation for both variable was used throughout. The assumption of normality was satisfied after the transformation. Other assumptions were satisfied by examining the scatter plots.

III. FINDINGS

Table 2 presents the Pearson correlation coefficients among the study variables. System size, SQA in definition phase and in implementation phase are significantly associated with maintenance costs showing correlation coefficient of $.60, -.56, and -.59$.

Table 3. Multiple Regression on Maintenance costs

Variables	b	SE B	Standardized Beta	t	P value
Intercept	-8.71	1.58		5.53	.0001
System age	-0.03	0.17	.30	1.55	.11
Experience	-3.27	1.40	.41	2.34	.030*
System size	-0.76	0.19	.63	3.94	.0009*
Total SQA effort	-0.0009	0.0004	-.34	-2.02	.052*
Adjusted $R^2 = .60$					

The SQA effort in the three system development phases: definition, construction and implementation were aggregated as the total SQA effort. The results of a multiple regression of the software maintenance cost on system age, maintainers' relative development experience, system size and total SQA effort are presented in Table 3. The results showed that the SQA effort has significant impact on the reduction of software maintenance cost.

In order to control the effect of other variable on maintenance cost, the system age, maintainers' relative development experience and system size were entered at Step 1. Table 4 presents the results of a multiple regression of the software maintenance costs on various independent variables.

As shown by the adjusted R^2, the proportion of variance in the software maintenance costs explained by these three controlled independent variables is equal to 56%. SQA in definition phase, construction phase and implementation phase were entered at Step 2. This increased the value of adjusted R^2 to 0.60. The R^2 change of 0.04 is significant at the 0.10 probability level. The R^2 indicated that there was unique maintenance costs variance accounted for by SQA effort, even after the effects of system size, system age, and maintainers' relative development experience had been partialed out. In addition, the study found that SQA in definition phase is the only audit variable that has significant effect on software maintenance cost.

Table 4 Multiple Regression on Maintenance costs

Variable	b	SE B	Std Beta	t	P value
Step 1:					
Intercept	2.03	2.49		.82	.42
System age	1.64	0.59	0.53	2.77	.01
Experience	5.00	1.41	0.62	3.56	.002
System size	0.99	0.18	0.82	5.39	.0001
Adjusted $R^2 = .56$	F value=10.7		P value=.0002		
Step 2:					
Intercept	2.03	2.49		3.55	.003
System age	1.10	0.63	0.32	1.74	.09*
Experience	3.45	1.56	0.43	2.21	.04*
System size	0.75	0.21	0.62	3.55	.003*
Definition audit	-0.0046	0.002	-.37	-2.30	.03*
Construction audit	0.0009	0.003	0.05	0.36	.72
Implementation audit	-0.001	0.001	-.15	-1.062	.30
Adjusted $R^2 = 0.60$	F value=2.30		P value=.01		

The result also indicated that the regression coefficient of definition audit effort is significantly greater than that of construction audit effort ($t = 5.0$),[1] and that of implementation audit effort ($t = 2.25$). In addition, investment of SQA effort in the definition phase yields the greatest reduction in software maintenance cost.

[1] The T test for differences between two reqression coefficients (Pedhazur, 1982)

$$t = \frac{b_1 - b_2}{S_{b1-b2}}, S_{b1-b2} = Var(b_1) + Var(b_2) - 2Cov(b_1, b_2)$$

IV. CONCLUSION AND DISCUSSION

The results of this study clearly showed the importance of SQA effort in the reduction of software maintenance costs. Especially, the SQA effort in the definition phase showed the most significant impact on software maintenance.

The study result justifies for the software quality assurance function. However SQA functions will not be effective unless they are properly managed and applied. Martin and McClure (1983) provides some software review guideline: (1) SQA reviews should not be performed at every life-cycle point but instead selectively performed at those points where quality concerns have proven to be the most serious for an organization, (2) automated tools such as CASE (Computer Aided Software Engineering) tools should be considered to use whenever possible to reduce the staff needed to apply quality measures, (3) follow-up actions to investigate and to correct quality deficiencies uncovered during audits should be required and enforced, (4) there should be a commitment from management recognizing the resources needed to perform reviews, (5) decisions concerning what SQA review will be conducted, and who will participate in the review should be made as early as possible in the software life cycle.

In this empirical study, we found that software quality assurance reviews performed in the development phases reduced software maintenance effort. Additional research is needed to provide evidence for the effect of SQA function on the ultimate information system outcome measure, which is information system success. Also, further research is needed to determine the most useful methods and techniques to perform the SQA function, and the organization of the SQA process. Study for designing methods for measuring software quality and software productivity is another important area for investigation.

REFERENCES

Ahituv, N. and Neumann, S., *Principles of information Systems for Management*, William C. Brown Inc., Dubuque, Iowa, 1982.

Boehm, B.W., Improving Software Productivity, *Computers*, September 1987, pp. 43-57.

Brantley, C.L. and Osajima, Y.R., Continuing Development of Centrally Developed and Maintained Software Systems, *IEEE Computer Society Conference Proceedings*, Spring ,1975.

Case, Albert F., Computer-Aided Software Engineering (CASE): Technology for Improving Software Development Productivity. *Data Base*, Fall 1985, pp. 35-43.

Chrysler, E., Some Basic Determinants of Computer Programming Productivity, *Communications of the ACM*, Volume 21, Number 6, June 1978, pp. 472-483.

Conover, W., *Practical Nonparametric Statistics*, John Wiley & Sons, Inc., New York, 1971.

Edward, C. *Information Systems Maintenance: An Integrated Perspective*, MIS Quarterly, December 1984.

Gilb, Y., *Software Metrics*, Withrop Publishers, Inc., Cambridge, Mass., 1977, pp.60-64.

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

Gustafson G. and Kerr R., Some Practical Experience With a Software Quality Assurance Program, *Communications of the ACM*, January, 1982.

Hamilton, S., EDP Quality Assurance: Selecting Applications for review, *Proceedings of the Third International Conference on Information Systems*, 1982.

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

Lientz, B. P. and Swanson, E. B., and Tompkins, G.E., Characteristics of Application Software Maintenance, *Communications of the ACM*, Volume 21, Number 6, June 1978, pp.466-477.

Martin, J. and McClure C., *Software Maintenance: The Problem and Its Solution*, Prentice-Hall, New Jersey, 1984.

Orlikowski, W. J. Case Tools and the IS Workplace: Some Findings form Empirical Research, *Proceedings of the 1988 ACM SIGCPR Conference on the MIS personnel*, April 1988.

Pedhazur, E., *Multiple Regression in Behavioral Research, Explanation and Perdition*, Holt, Rinehart, and Winston, New York, 1982.

Shapiro, S. and Wilk, M., An Analysis of Variance Test for Normality, *Biometrika*, Vol.52, pp.591-611.

Swanson, E.B., The Dimension of Maintenance, *Proceedings of the Second International Conference on Software Engineering*, Piscataway, New Jersey, 1976, pp.492-497.

Vessey I. Weber R., Some Factors Affecting Program Repair Maintenance: an Empirical Study, *Communications of the ACM*, February 1983, pp.128-134.

Zelkowitze, M. V., Perspectives on Software Engineering, *Computing Survey*, Volume 10, number 2, June 1978.

軟體品保對軟體維修成本影響之實證研究

Rebecca Chung-Fern Wu

吳琮璠

摘　要

就整體軟體生命週期之支出而言，軟體維修成本已逐漸成爲最昂貴、負荷最重的軟體支出。本研究提出假說：在系統開發期間對軟體執行軟體品質保證複核，有助於降低軟體維修成本，特別是在系統功能定義時執行軟體品保複核最見績效。本研究實證34個銀行業應用系統於系統上線前，執行軟體品保複核，利用調查研究方法蒐集資料，並建立多元迴歸模式，控制其他相關變數，以檢驗假說。實證結果確認本研究所提假說。

關鍵詞: 軟體品保複核、軟體維修、軟體工程、系統開發生命週期