A Structural Model of Software Maintainer Effectiveness

Wui-Gee Tan, Taizan Chan, Guy G. Gable
Information System Management Research Centre,
Queensland University of Technology
GPO Box 2434 Brisbane, QLD Australia 4001.
Email: wuigee@iss.nus.edu.sg

Abstract

Through soft modeling of data collected from a survey of 83 Information Systems (IS) organizations in Singapore, this paper develops and tests a Partial Least Square (PLS) model that determines the impact of user knowledge, application quality, and maintenance control procedures on maintainer effectiveness. It was found that application quality has a very strong influence on maintainer effectiveness while neither the use of maintenance controls nor user knowledge has much impact on maintainer effectiveness. These results suggest that IS organisations should accord the same, if not greater, priority to application quality in maintenance as they do in the development phase.

Keywords
Software Maintenance Management, Maintenance Programmer Effectiveness, Predictive Modeling, Partial Least Squares

INTRODUCTION

Software maintenance is a highly resource-intensive and critical Information Systems (IS) activity. On average, 50% of an IS budget is assigned to software maintenance, sometime consuming up to 95% of IS resources (Banker, Datar, and Kemerer, 1991; Swanson and Beath, 1989). While there are several resource inputs to software maintenance, the main and most costly input to the software maintenance function is the effort of the software maintainers.

Existing literature aimed at reducing software maintenance effort can be classified into two streams. The first stream of literature (for example, Banker, Datar, and Zweig, 1989; Banker and Slaughter, 1994) aim at improving maintenance productivity at the individual task level. Focusing on productivity at the individual task level, however, is inadequate as it ignores factors, such as the turnover rate of maintainers within a maintenance function, that may not affect individual tasks or maintainers but could affect the effective functioning of the maintenance function as a whole. Furthermore, maintenance productivity has been found to be only *one* of the problems associated with the management of software maintainers (Swanson and Beath, 1989). Thus, focusing on the individual task or maintainer level can only help in reducing part of the maintenance cost.

The second stream of literature, notably those of Lientz and Swanson (1980) and Swanson and Beath (1989), focus on a broader set of managerial problems associated with software maintenance at the maintenance function level. For example, Lientz and Swanson (1980) conducted a survey targeted at the IS managers and supervisors in 487 US organisations. The survey attempted to identify, among other things, the use of maintenance controls, procedures, methods, and tools to manage software maintenance, and the management problems faced by IS managers (Lientz and Swanson, 1980).

Based on the respondents' rating of twenty-six problem items, Lientz and Swanson (1980) found that, in addition to maintenance productivity, motivation and personnel turnover are other important factors in the overall effectiveness of the maintenance function.[1] In addition to problems associated with software maintainers, other serious problems were users' lack of understanding of applications and implications of their demands on maintenance, and poor application quality, e.g. outdated documentation and poor reliability associated with the applications. Factor analysis of the twenty-six problems reveal that these problems can be classified into six categories: maintainer effectiveness[2], user knowledge, application quality[3], maintainer time availability, machine requirements, and system reliability, with the first three being the most severe. Later studies by Nosek and Palvia (1990), and Swanson and Beath (1989), using a very similar survey instrument, found that despite a decade of advances in software maintenance technology and new insights into software maintenance problems, the nature and ranking of the software maintenance problems remain relatively unchanged.

This stream of research yielded useful insights into managerial and organisational problems faced by IS managers at the maintenance function level. However, it did not attempt to determine if problems in one area (for example user knowledge) might be associated with problems in another area (for example maintainer effectiveness), even though they have been repeatedly found to occur *together*. In particular, despite the importance of maintainers' effort as the main resource input to the software maintenance task, there has been no research aimed at determining if and how the problems of user knowledge and application quality might have affected maintainer effectiveness. Furthermore, while these studies have provided insights into the types of maintenance control procedures, methods, and tools employed in the organizations surveyed, no effort has been directed at determining if the use of these procedures and tools were indeed effective in managing these problem areas. This is perhaps due to the difficulty associated with analysis of the variables involved, as the main constructs of interest are not directly measurable, and there is no guiding theory for the formulation of a model involving these constructs.

By using a survey instrument adapted from Lientz and Swanson (1980), and Partial Least Squares (PLS) to overcome the data analysis problems, this paper reports the results of a study[4] that aim to fill this gap in the literature. More specifically, this paper addresses the following research questions:

1. To what extent does the level of user knowledge of the application systems affect the effectiveness of the maintainers?
2. What is the impact of the quality of the application systems on the effectiveness of the maintainers?
3. Does the use of maintenance controls, procedures, methods, and tools affect the user knowledge, application quality, and ultimately effectiveness of the maintainers?

This paper, in addressing these questions, contributes to the field of software maintenance by providing a conceptual model for relating maintenance management problems that have already been identified to be significant but until now treated as disparate problems. While the instrument may seem to be dated in that the current computing environment appears to be quite different from that two decades ago, it should be pointed out that the

[1] Section 3 describes in full detail the concept of maintainer effectiveness used in this study.

[2] In Lientz and Swanson (1980)'s study, the term "programmer" is used instead of "maintainer". This is because the IS staff in question could be doing both new development and maintenance work. However, as the questions in their survey (as well as ours, to be discussed in Section 2) ask the respondents specifically about the problems associated with software maintenance, we feel that it is more precise to refer to the maintenance roles of the IS staff directly by referring to them as maintainers.

[3] Lientz and Swanson (1980) used the term "product quality" in their work. It is argued here that "Application Quality" as reported herein is a more precise term.

[4] The results reported in this paper is based on part of a study that aimed to identify software maintenance problems in Singapore organisations (Tan, 1999).

mainframe/centralised environments targeted at by the survey instrument are still prevalent.[5] Indeed, as pointed out in Section 2, all the respondent organisations have centralised IS functions using mainframe or mid-rage computers. Hence the instrument used is still relevant.

The rest of the paper is organized as follows. The following section describes the research methodology employed in this study. The third section describes the research model we are testing. The fourth section describes PLS, the methodology used in testing the research model and the analysis results. The fifth section discusses the results. Finally, the last section concludes by discussing the limitations of this study.

RESEARCH METHODOLOGY

This study represents an extension to prior research by Lientz and Swanson (1980), Nosek and Palvia (1990), and Swason and Beath (1989). The survey instrument used is adapted from that originally employed by Lientz and Swanson (1980). Similar to Lientz and Swanson (1980)'s study, the target respondents are maintenance managers.[6] However, unlike the Lientz and Swanson (1980) survey, respondents in the current study were required to describe their application portfolio rather than a single existing major application. It was felt that the data collected for an arbitrarily selected application would not realistically reflect the overall state of software maintenance practice nor the general problems respondents are confronted with.

The survey instrument consists of three parts, all of which are adapted from the Lientz and Swanson (1980) instrument. Part 1 contains general questions on the respondent's organisation. Part 2 includes questions on software maintenance practices (e.g. use of various maintenance controls to ensure that only justifiable changes are made to the code and in a manner that would preserve the quality of the application software). In part three, the twenty-six problem items are presented. In addition, anticipating that these twenty-six problems may not exhaustively account for all concerns, provision is made for respondents to indicate other problems they are experiencing. Similar to Lientz and Swanson (1980)'s questionnaire, the evaluation of these problems is based on a 5-point Likert scale in which 1 represents "no problem" and 5 represents "major problem".

The survey instrument was pre-tested on three known subjects taken from the list of selected target respondents. The final questionnaire was posted to the target respondents accompanied by a cover letter explaining the purpose of the survey and inviting the respondents to avail themselves of the opportunity to benchmark their current software maintenance practices against those of other Singapore organisations. Attempts were made to improve the response rate by sending reminders to the non-respondents, emphasizing the importance of the study and the advantage of obtaining a copy of the results.

Taking advantage of the close links between the Institute of Systems Science (ISS)[7] and the local IT industry, 263 people with key managerial appointments in centralised IS installations in government, local commercial corporations, and Multi-National-Corporations (MNC) were identified from the ISS contact database. A total of 88 completed questionnaires were

[5] See Tan (1999) for a detailed discussion on the continued relevance of the survey instrument.

[6] It is recognised that a criticism associated with Lientz and Swanson (1980) study is that the respondents consist of only managers, who usually did not work on the maintenance tasks directly, and that maintainers should also be surveyed. This paper, however, is concerned with only the identification of relationships among the problems and issues identified by the *previous* studies which consisted of only maintenance problems from the managers' perspectives. The criticism has been addressed in a broader study that also looks at software maintenance problems from the perspective of the maintainers (Tan, 1999).

[7] ISS (established in 1981) is an 18-year old government-funded organisation that has, through its education, consultancy, and research programs, played an important catalytic role in Singapore's computerisation growth. Its contact database is believed to be broadly representative of the Singapore IS community as a whole.

returned giving a response rate of 34%. Eight responses were unusable: these respondents having indicated one of the following circumstances in their organisation: (1) source code maintained by vendor, (2) source code maintained by corporate headquarters, or (3) only application packages used. Including the three pretest cases which also yield useful data, a total of 83 sample points were available for further analysis.

It is noted that 15% of the respondents are senior corporate executives (e.g. CIO), 69% are IS executives and systems managers who have direct control over maintenance, while 15% have analyst titles (e.g. Project Leader). Responses from these three groups for all interval data (e.g. the twenty-six maintenance problems) were compared using the t-test, revealing no significant differences at the .05 level thus suggesting that all respondents had a common level of awareness of maintenance management problems. It is also noted that 24, 25, and 34 responses were received from the public sector, local commercial organisations, and MNCs respectively. Approximately half of the respondents are from the manufacturing and financial industries. The median number of IS employees is 22 and in one organisation an IS staff size of 350 was indicated. Half of the respondents (42) operate mainframe computers while the other half (41) operate midrange computers. A chi-square test for non-response bias by sector, reveals no significant difference at .05 level (C^2 = 14.1, d.f. = 2), suggesting that the sample is representative across the sectors.[8]

THE RESEARCH MODEL

As described in Section 1, this study aims to determine the impact of the problems associated with user knowledge and application quality on maintainer effectiveness, as well as to determine the impact of the use of maintenance control procedures on these problem areas.[9] Thus, the predicted construct of interest is maintainer effectiveness, while user knowledge, application quality, and maintenance control procedures represent the predicted construct. The following sections describe the research constructs, the operationalisation of these constructs, and the hypothesized relationships among these constructs.

Research constructs

The research constructs, namely, maintainer effectiveness, user knowledge, application quality, and maintenance controls are labeled as Maintainer Effectiveness, User Knowledge, Application Quality, and Maintenance Controls respectively.

Maintainer Effectiveness

Maintainer Effectiveness, the predicted construct in the research model, refers to the overall effectiveness of the maintenance function at accomplishing maintenance tasks efficiently. It is a measure of not only the productivity of the maintainers but also other factors such as the skills of the maintainers, the motivation level of the maintainers, and the turnover rate of the maintainers. The skills of the maintainers affect the maintenance function as a whole because the lack of certain skills in certain maintainers would place burden on other maintainers who have those skills (for example, in training the less skilled maintainers). A low motivation level among the maintainers means that even if the maintainers are skillful and potentially productive, they may not be performing at their best (Couger and Colter, 1985), thereby lowering the performance level and the effectiveness of the entire maintenance function.

[8] No further test of non-response bias is possible, as a large proportion of the organisations surveyed was comprised of government and MNCs, neither of which have statistics that are publicly available.

[9] Although "maintenance control procedures" is not one of the maintenance management problems identified from the twenty-six problem items, it is included in this model since the use of control procedures have been known to influence software development activities and it is useful to determine if maintainer effectiveness and the other maintenance problems are similarly affected by the use of maintenance control procedures.

Finally, it is clear that with a high turnover rate, the motivation level of the maintainers will be affected, skilled maintainers who are more productive will be lost, and new maintainers will need to be trained, thereby draining the already limited maintainers' effort to perform maintenance tasks effectively.

Application Quality

Similar to studies by Lientz and Swanson (1980) and Swanson and Beath (1989), the Application Quality construct is conceptualised in terms of application quality from both the maintainers' perspective and the users' perspective. From the maintainers' perspective, application quality includes the quality of the application system components, that is, of programs, data, documentation and specifications, that are used and updated by maintainers in performing their functions. From the users' perspective, application quality reflects the root concerns of the user such as performance of the application software (e.g. application failures and data integrity), validity of design (e.g. adequacy of design specifications), and adaptability of application software to include new functions (Evans and Marciniak, 1987).

User Knowledge

From results of their factorial analysis, Lientz and Swanson (1980) attributed the problems of user knowledge to the adequacy of user training in the use of the application systems, users' expectations of the effort required to fulfill their maintenance requests, and users' understanding of application systems. These in turn affect users' interest of the applications and management support of the maintenance function. Following Lientz and Swanson (1980), the construct User Knowledge is conceptualised as encompassing the users' understanding of and interests in the application systems, and their expectations of and support for the maintenance function.

Maintenance Controls

Maintenance Controls as used in the current study focuses on change management measures, such as configuration/change request procedures and provision of maintenance standards.

Operationalisation of Constructs

The three constructs: Maintainer Effectiveness, User Knowledge, and Application Quality, were first identified as *a priori* constructs in our model through the logical grouping of problems from Lientz and Swanson (1980)'s twenty-six problem items. Operationalisation of these constructs was attempted without resorting to the use of any data reduction tools or statistical techniques. Rather, the problem set was analysed then logically grouped according to the analysis of the concepts underlying the constructs found in the literature, and from prior personal experience of the authors. The fourth construct, Maintenance Controls, is operationalised with indicators drawn from the list of maintenance controls in Part 2 of the survey instrument. Table 1 summarises the operationalisation of the constructs.

Research hypotheses

Instituting appropriate maintenance controls will not only ensure that change requests are properly justified and approved but also that the integrity of the application system components i.e. code, documentation, data and specifications is not lost during the implementation of the desired change. It is hypothesised that these measures will bring about greater confidence among the maintainers of the application systems, improve their productivity, raise their motivation, and reduce their frustration and propensity to leave the company. This leads us to the first hypothesis:

H1: Maintenance Controls directly predicts Maintainer Effectiveness.

Maintenance Problem Items/Maintenance Controls Items

MAINTAINER EFFECTIVENESS			
m1	Turnover of maintenance personnel	m14	Motivation of maintenance programming personnel
m5	Skills of maintenance programming personnel	m16	Maintenance programming productivity
APPLICATION QUALITY			
m2	Quality of application software documentation	m18	Data integrity in application software
m6	Quality of original programming of application	m20	Adherence to programming standards in
m10	Application software run failures	m22	Adequacy of application software design
USER KNOWLEDGE			
m4	User demand for enhancements/extensions to	m21	Management support of application software
m9	Lack of user interest in application software	m25	Inadequate training of user personnel
m11	Lack of user understanding of application s/w	m26	Turnover in user organization
m19	Unrealistic user expectations		
MAINTENANCE CONTROLS			
p2	Changes are cost-justified	p5	Change follows formal lifecycle
p3	Changes are approved by approving authority	p6	Changed programs are retested
p4	Change requests subjected to impact analysis	p7	Maintenance standards exist

Table 1: Operationalisation of Research Constructs

Premature degeneration of application system components is brought about through the organisation neglecting to put into place appropriate maintenance controls. In fact, Lientz and Swanson (1980) found that the use of controls, tools and techniques tends to keep the quality of the design specifications, programs and documentation in check. By instituting these maintenance controls there are fewer tendencies for errors to be introduced into programs during the change, for documentation to be out of step with the modified programs and for the specifications to "decay" unnecessarily. In this way, program modification is made only when justified and with due consideration for the application system quality. Thus, we make the following hypothesis:

H2: ***Maintenance Controls directly predicts Application Quality.***

Extensive prior empirical studies (e.g. Berns, 1984; Vessey and Weber, 1983; Guimaraes, 1983) have established that the extent of the maintenance burden to the organisation is heavily influenced by the level of complexity of the programs. Maintainers find it less frustrating and more motivated to work with application systems displaying higher quality in terms of the ease of change, currency of the documentation and run stability. The maintenance staff will feel less compelled to leave the company out of frustration and dissatisfaction. There will also be less corrective maintenance to undertake, thus increasing the productivity of the maintainer. This leads to the third hypothesis:

H3: ***Application Quality directly predicts Maintainer Effectiveness.***

There is higher confidence and satisfaction among users when they perceive that the application system is less prone to breakdown and maintenance personnel is receptive to their requests for change. Current and easy to understand documentation promotes co-operation from the user. Higher quality documentation and specifications will also mean that users will understand the application better and not be overly dependent on the maintainers for information about the application. Users are also aware that relevant documentation is available to aid the maintainer in the change and to pre-empt any unintended code from being introduced. Better documentation further means that there is continuity whenever there is staff turnover on the user side. With a stable system that is responsive to the business, management too is expected to give their support more readily and to invest further in new techniques and tools. This leads to the following hypothesis:

H4: ***Application Quality directly predicts User Knowledge.***

When users are knowledgeable about the application, they are better able to articulate their requirements to the maintainers. This should lead to improved rapport and co-operation between the maintainers and the users. Furthermore, when users are aware of the change process and the complexity of the application system, they will be more restrained in requesting unjustified changes and more subdued in their demands. With reduced user turnover and reduction in the attendant loss of application knowledge, the maintainers will be able to devote more time to completing the change rather than re-training the new users. It is also hypothesised that support from management for the maintenance function should also raise the maintenance personnel’s motivation as this implies that their contribution to the company is being recognised. We therefore make the following hypothesis:

H5: ***User knowledge directly predicts Maintainer Effectiveness.***

MODEL TESTING

Given that (i) our research hypotheses are conjectural, (ii) a mixture of data types (binary and interval data) with unknown distributions are involved, (iii) the sample size is relatively small, and (iv) that none of our research constructs is observed directly but are measured by several indicators, the model is tested using partial least squares (PLS).

The PLS Model

Figure 1 depicts the PLS path diagram of predictive relationships in our research model. It comprises the structural (inner) and measurement (outer) models. The measurement model is collectively made up of blocks of constructs, also known as the latent variables (oval shaped) and their respective indicators, also known as the manifest variables (square shaped), whereas the structural model is the network of interconnecting constructs denoting the hypothesised predictive relationships.

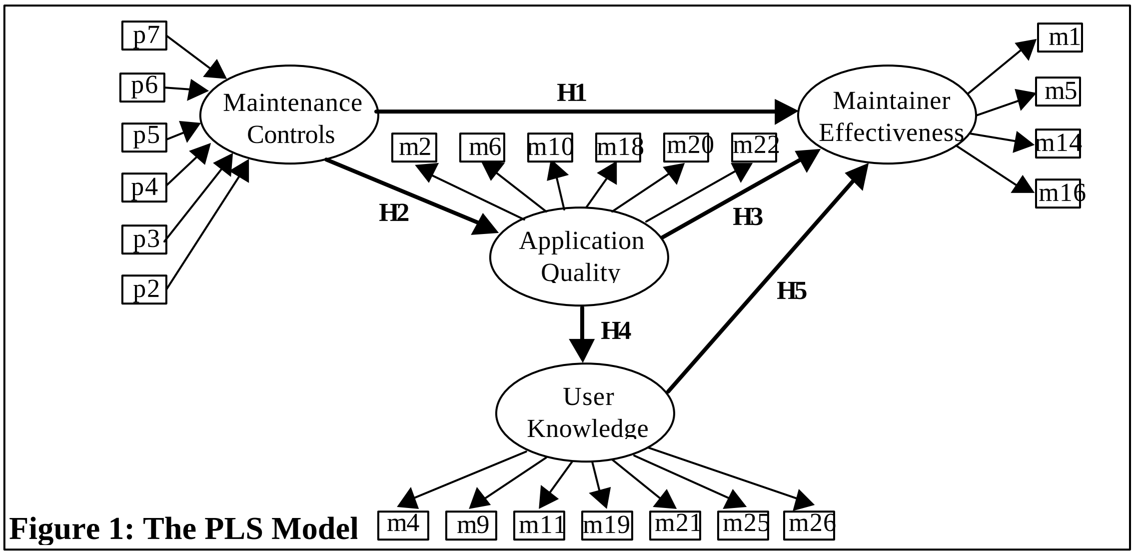

Figure 1: The PLS Model

It can be seen from Figure 1 that the constructs Maintainer Effectiveness, Application Quality and User Knowledge are modelled with reflective indicators (arrows point out of the construct), each of which is an item from the Lientz and Swanson (1980) problem set. These three constructs were first derived by Lientz and Swanson (1980) from factorial analysis of their twenty-six problem items, and were subsequently referenced by other similar studies. Reflective indicators for a given construct are expected to “reflect” the nature of the construct they posit to represent (i.e. assumed to measure the same phenomena). Consequently, scores on the reflective indicators for a given construct should tend to be similar for a given respondent in the survey.

On the other hand, Maintenance Controls is modelled with formative indicators (arrows point into the construct) each of which is associated with maintenance controls. These indicators "form" the construct as a composite. All of these indicators are dichotomous. The use of formative indicators to anchor Maintenance Controls is appropriate since (1) Maintenance Controls can be treated as an index of the level of "methods" usage, (2) scores on formative indicators tend not to be similar for any given respondent, and (3) Maintenance Controls is a less abstract construct as compared to Maintainer Effectiveness, Application Quality or User Knowledge. Thus, the respondents are expected to use a selection of "controls" that best meet the needs of their installations. In summary, Maintenance Controls is linearly measured by a set of dichotomous formative indicators.

Evaluation of the Measurement Model

Table 2 presents the loadings and cross-loadings of the indicators on the four constructs based on the measurement model. Three observations are apparent: (1) the loadings are all statistically significant at the .01 level and are higher than the acceptable 0.55 level of reliability (except for some formative indicators), (2) the sign of the loadings are all in the same direction, and (3) no indicator loaded more highly on another construct than it does on the construct it is intended to measure. It is therefore concluded that the criterion for unidimensionality is met for each of the constructs. A test of the internal consistency (i.e. of whether the indicators "hang together" for each construct) using Fornell and Larcker (1981)'s measurement[10], reveal internal coefficients of 0.860, 0.883, 0.865, and 0.690 for Maintainer Effectiveness, Application Quality, User knowledge, and Maintenance Controls respectively which are well above or close to the acceptable level of 0.7 (Nunally, 1978).

	MAINTAINER EFFECTIVENESS	APPLICATION QUALITY	USER KNOWLEDGE	MAINTENANCE CONTROLS
Reflective Indicators				
m1	**.712 ****	.320 **	.200	.182 *
m5	**.818 ****	.587 **	.379 **	.259 *
m14	**.770 ****	.423 **	.485 **	.258 *
m16	**.807 ****	.563 **	.490 **	.320 **
m2	.456 **	**.684 ****	.316 **	.315 **
m6	.580 **	**.749 ****	.477 **	.210
m10	.296 **	**.656 ****	.459 **	.314 **
m18	.387 **	**.631 ****	.399 **	.325 **
m20	.561 **	**.852 ****	.548 **	.493 **
m22	.526 **	**.887 ****	.512 **	.395 **
m4	.494 **	.450 **	**.662 ****	.179
m9	.183	.273 *	**.531 ****	.097
m11	.277 *	.327 **	**.664 ****	.157
m19	.327 **	.416 **	**.758 ****	.119
m21	.238 *	.425 **	**.590 ****	.177
m24	.478 **	.516 **	**.810 ****	.252 *
m25	.400 **	.475 **	**.799 ****	.201
Formative Indicators				
p2	.059	.181	.283 **	**.317 ****
p3	.179	.210	.159	**.480 ****
p4	-.003	.264 *	.008	**.370 ****
p5	.162	.355 **	.288 **	**.667 ****
p6	.105	.199	.017	**.389 ****
p7	.364 **	.331 **	.149	**.841 ****

Note: * - Signif. LE .05 ** - Signif. LE .01 (2-tailed)

Table 2: Loadings and Cross-loadings of Constructs

[10] Fornell and Larcker (1981)'s measure is used instead of Cronbach's Alpha because it uses the loadings estimated within the model and is commonly adopted in PLS analysis.

Evaluation of the Structural Model

The adequacy of the psychometric properties in the measurement model enables us to further evaluate the results from the structural model summarised in Figure 2.

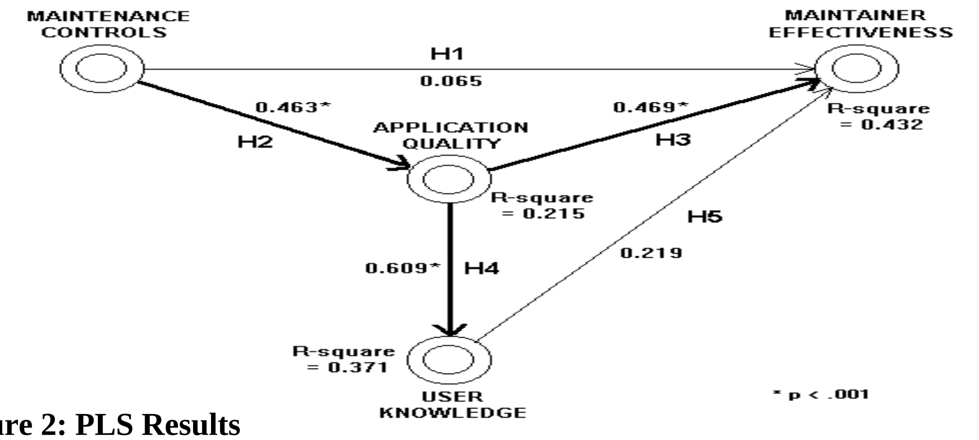

Figure 2: PLS Results

Results of the PLS analysis with Jack-knifing, as shown in Table 3, reveal that three out of the five path coefficients are statistically significant at the .001 level.[11] These path coefficients are associated with the following paths: (1) Maintenance Controls → Application Quality, (2) Application Quality → Maintainer Effectiveness, and (3) Application Quality → User Knowledge. Judging from the Jack-knifed t-values, it also appears that of the two insignificant paths, Maintenance Controls → Maintainer Effectiveness is statistically weaker. Regardless, all the paths are in the direction proposed.

Hypothesised Paths	Direct Effect (1) Path Coeff.	Direct Effect (1) t-Value	Indirect Effect (2)	Total Effect (1) + (2)
H1: Maintenance Controls → Maintainer Effectiveness	.065	.315	.279 (A)	.344
H2: Maintenance Controls → Application Quality	.463	4.194 ***	---	.463
H3: Application Quality → Maintainer Effectiveness	.469	3.640 ***	.133 (B)	.602
H4: Application Quality → User Knowledge	.609	7.739 ***	---	.609
H5: User Knowledge → Maintainer Effectiveness	.219	1.507	---	.219

Note: *** p < .001
(A) (H2→ H3) + (H2 → H4 → H5) = (.463 x .469) + (.463 x .609 x .219) = .217 + .062 = .279
(B) (H4 → H5) = .609 x .219 = .133

Table 3: Testing of Structural Model

The table also shows the direct, indirect and total (direct + indirect) effects suggesting the relative importance of antecedent constructs. Thus, with regard to Maintainer Effectiveness, which is the focus of this study, the following salient observations can be made:

1. No *direct* influence was found from Maintenance Controls to Maintainer Effectiveness.
2. The *total* influence of Maintenance Controls on Maintainer Effectiveness (.344) is mostly contributed by the mediating effect of Application Quality (.217), and to a much smaller extent by the secondary mediating effect of User Knowledge (.062) (see note (A) in Table 3).
3. The total effect of Application Quality on Maintainer Effectiveness (.602) also has an indirect contribution from the mediating effect of User Knowledge (.133) (see note (B) in Table 3) even though the path linking User Knowledge and Maintainer Effectiveness is not statistically significant.
4. Among the posited predictors of Maintainer Effectiveness, Application Quality is the most cogent, whether directly exerting its influence or indirectly as a mediating construct.

[11] For a good overview of the PLS method and how the path coefficients are derived, refer to Chin (1998).

5. Application Quality is also a strong predictor of User Knowledge.

Broadly, the evidence described and the high R^2 values in the model given the relatively small sample size in this study, suggest the usefulness of the PLS model to explain the level of influence of the antecedent constructs on Maintainer Effectiveness.

DISCUSSION

The PLS model results suggest that Maintenance Controls has little *direct* impact on Maintainer Effectiveness thereby rejecting ***H1***. These results may reflect the indifference of maintainers towards maintenance controls or other similar maintenance management practices introduced by the IS organisation to regulate the maintenance activity. It is probable that the maintainers believe that such procedures are instituted more for the benefit of the organisation than for them, or even mandated by audit.

Maintenance Controls, however, exercises a strong influence on Application Quality, and through Application Quality a moderate influence on Maintainer Effectiveness, thereby lending support to ***H2*** and ***H3.*** Indeed, our results indicate that Application Quality is the most cogent predictor of Maintainer Effectiveness in comparison to Maintenance Controls and User Knowledge. Thus, it appears that when maintenance control measures exist in the organisation, it is likely that the quality of the original documentation, data, programs and specifications will be sustained or even improved. Thus, although the maintenance staff may not be keen on adhering to controls, they will find maintenance work less frustrating and more fulfilling when they work with higher quality applications. This will lead to an increase in maintainer motivation and productivity and consequently they may be encouraged to stay on the job, thereby increasing the overall effectiveness of the maintainers.

The results further indicate that Application Quality has a strong influence on User Knowledge, giving support to ***H4***. Thus, when it is evident to the users that the IS organisation is concerned with quality in the application systems under maintenance e.g. keeping documentation current or responding promptly to reported application failures, the users are likely to display keener interest and understanding of the application systems and the maintenance process. At the same time, management will show greater approval of the work of the maintenance function, as the application systems are meeting their business objectives.

However, the results also suggest that User Knowledge has low influence on Maintainer Effectiveness, thus rejecting ***H5***. This is perhaps due to the fact that the IS organisations surveyed involved mainly centralised mainframe-based systems where it is the supervisors or managers who interact with the users, and not the maintainers themselves. It would be of interest to determine if User Knowledge will have a significant impact on Maintainer Effectiveness in a distributed, end-user environment where the maintainers interact with the users directly.

Two implications for managing software maintenance can be drawn from this study. First, IS management should recognise the overriding importance of application quality to the performance of their maintainers. IS organisations are inclined to stress quality during the development phase but to accord it lesser importance, or worse, to neglect it altogether once the new application system is handed over to the maintenance staff (Swanson and Beath, 1989). By paying attention to the quality of the maintained application systems and proactively taking suitable quality-enhancing steps, IS organisations could reduce the frustrations, raise the motivation and hence improve the overall effectiveness of the maintenance staff.

Second, these results suggest that IS management can enhance the effectiveness of the maintainers by using appropriate maintenance control methods. However, the use of these tools does not appear to improve the effectiveness of maintainers *directly*. Rather the use of these tools could potentially enhance the quality of the applications, thereby improving the effectiveness of the maintainers when they work with better applications.

LIMITATIONS

Our study has several limitations. First, the hypothesised relationships between the *a priori* constructs are not grounded in existing theory but based purely on the authors' personal experience of software maintenance in a similar environment to the organisations surveyed. Thus, there is a threat to the internal validity of the predictive model and the hypothesised relationships. However, this is not deemed a serious problem as our intention is not to make causal claims but simply to "estimate the likelihood of an event given information about other events" (Falk and Miller, 1992).

Second, the sample size is relatively small. Statistically, PLS can accommodate small samples and, in fact, this is one of its primary strengths. However, statistical conclusion validity can only be achieved with a large sample when "it is reasonable to presume co-variation (between two variables) given a specified alpha level and the obtained variances" (Cook and Campbell, 1979). Although it is observed from comparative statistics (Tan and Gable, 1997) that there are salient differences among the sector, hardware platform and language categories, because of the small sample size, it was not possible to assess whether or not the pattern of predicted relationships identified in the model would also hold true across all categories.

Third, the nature of the sample (Singapore, midrange and mainframe installations) reduces confidence in the external validity or generalisability of the findings. While insights gleaned from the study reveal the difficulties in maintaining midrange and mainframe-based application systems, their extensibility to other computer and software platforms (e.g. distributed networks) and other countries is uncertain. Further survey involving a larger sample of organizations across different computing environments and countries should be conducted to address this problem.

Finally, while the study revealed strong evidence of the validity of the three constructs derived from the Lientz and Swanson (1980) problem set i.e. Maintainer Effectiveness, Application Quality and User Knowledge, the same confidence cannot be extended to Maintenance Controls, which is modeled with dichotomous formative indicators. Improved measurement of Maintenance Controls e.g. using scaled indicators could well increase the predictive power of the model.

REFERENCES

Banker, R.D., S.M. Datar and C.F. Kemerer (1991) A Model to Evaluate Variables Impacting the Productivity of Software Maintenance Projects, *Management Science* (37:1), pp.1-18.

Banker, R.D., S.M. Datar, and D. Zweig (1989) Software complexity and maintainability, J. I. DeGross, J. C. Henderson, and B. R. Konsynski (Eds.), *Proceedings of the Tenth International Conference on Information Systems*, Boston, (December 1989), 247-255.

Banker, R.D., and S.A. Slaughter (1994) Project size and software maintenance productivity: Empirical evidence on economies of scale in software maintenance, In J. I. Degross, S. L. Huff, and M. C. Munro (Eds), *Proceedings of the Fifteenth InternationalConference on Information Systems*, Vancouver, Canada (December 1994), 279-289.

Berns, G. M. (1984) Assessing Software Maintainability, *Communications of the ACM* (27:1), pp.134-143.

Chin, W. (1998) *The Partial Least Squares Approach to SEM*. In *Modern Methods for Business Research* (Marcoulides, G.A., Ed.), Lawrence Erlbaum Associates Publishers, Mahwah, NJ.

Cook, T.D. and D.T. Campell (1979) *Quasi-Experimentation: Design and Analysis Issues for Field Settings*, Rand McNally, Chicago, IL.

Couger, J.D. and M.A. Colter (1985) *Maintenance Programming: Improved Productivity Through Motivation*, Prentice-Hall, Englewood Cliffs, NJ.

Evans, M.W.and J. Marciniak (1987) *Software Quality Assurance and Management*, John Wiley, Chichester, UK.

Falk, R. F., and N. B. Miller (1992) *A Primer for Soft Modeling*, The University of Akron Press, Akron, OH.

Fornell, C. and D.F. Larker (1981) Evaluating Structural Equation Models with Unobservable Variables and Measurement Error, *Journal of Marketing Research*, 18, February, pp.39-50.

Guimaraes, T. (1983) Managing Application Program Maintenance Expenditures, *Communications of the ACM* (26:10), pp.739 - 746.

Lientz, B. P. and E.B. Swanson (1980) *Software Maintenance Management: A Study of the Maintenance of Computer Application Software in 487 Data Processing Organisations*, Addison-Wesley, Reading, MA.

Nosek, J.T. and P. Palvia (1990) Software Maintenance Management: Changes in the Last Decade, *Journal of Software Maintenance* (2:3), pp.157-174.

Nunnally, Jum C. (1978) *Psychometric Theory*, McGraw-Hill, New York, NY, 2nd ed.

Swanson, E.B. and C.M. Beath (1989) *Maintaining Information Systems in Organisations*, John Wiley, Chichester, England.

Tan, W. G. (1999) *Maintenance Programmer Effectiveness: A Survey of Software Maintainer and Maintenance Managers' Problems, Methods, and Effectiveness with Centrally-Maintained Application Systems*, Unpublished PhD Dissertation, Queensland University of Technology.

Tan, W. G. and Guy G. Gable (1997) Maintaining Centralised Application Systems: A Cross-Country, Cross-Sector, Cross-Platform Comparison, *European Journal of Information Systems* (6: 4), pp.193-207.

Vessey, I. and Weber, R. (1983) Some Factors Affecting Program Repair Maintenance: An Empirical Study, *Communications of the ACM* (26:2), pp.128-134.

COPYRIGHT