

December 2003

A Revelatory Case Study into the Adequacy of Standard Maintenance Models in an ERP Context

Celeste Ng

Queensland University of Technology

Guy Gable

Queensland University of Technology

Taizan Chan

Queensland University of Technology

Follow this and additional works at: <http://aisel.aisnet.org/pacis2003>

Recommended Citation

Ng, Celeste; Gable, Guy; and Chan, Taizan, "A Revelatory Case Study into the Adequacy of Standard Maintenance Models in an ERP Context" (2003). *PACIS 2003 Proceedings*. 72.

<http://aisel.aisnet.org/pacis2003/72>

This material is brought to you by the Pacific Asia Conference on Information Systems (PACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in PACIS 2003 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

A Revelatory Case Study into the Adequacy of Standard Maintenance Models in an ERP Context

Celeste See Pui Ng^a, Guy Gable^b and Taizan Chan^c

^a Centre of Information Technology Innovation, Queensland University of Technology
Brisbane, Australia /
Department of Information Management, Yuan-Ze University
Chung-Li, Taiwan R.O.C.
celeste@saturn.yzu.edu.tw

^b School of Information Systems, Queensland University of Technology
Brisbane, Australia
g.gable@qut.edu.au

^c School of Information Systems, Queensland University of Technology
Brisbane, Australia
t.chan@qut.edu.au

Abstract

Standards are consensus-based documents that codify best practices. The use of standards (such as IEEE/EIA 12207, a maintenance model) is noted to assure that software projects are successfully completed on time, within budget and with the intended benefits. However, available standards for software maintenance model are mainly meant for in-house custom software. Are the published standards sufficient to provide a solution for the ERP environment? Or are activities involved in ERP maintenance and upgrade somehow unique such that a new maintenance model is warranted? This study entailed a comprehensive case study of a large Government agency to gather empirical data on their ERP maintenance model – from software maintenance preparation through software upgrade. Our findings show that the existing standard for maintenance model is generally comprehensive but does not cover all the requirements of an ERP maintenance model, such as: (1) inclusion of long-term vendor maintenance support; (2) searching and implementing maintenance support from the vendor, as well as reporting maintenance requests to the vendor; (3) deciding whether to keep prior user-enhancements, and reapplying them (if applicable) after patch maintenance or upgrade; and (4) fully assessing new functionality in each (potential) upgrade version.

Keywords

Enterprise Resource Planning, ERP, maintenance model, software standard, case study, maintenance methodology

1 Introduction

Enterprise resource planning (ERP) is off-the-shelf packaged software, which integrates most of the fundamental business processing functions and systems from different functional areas and business units into a single software system, database, unified interface and channel of communication. Unlike traditional in-house software, ERP software is unique in several ways (see (Markus & Tanis 1999, Ng, Gable & Chan 2002)). For example, it is bought from a vendor versus built in-house; with helpdesk and maintenance support available from the vendor versus completely maintained by internal maintenance personnel; and replacement of an installed version is done by choosing from readily available versions from the vendor versus re-engineering or rewriting the whole system internally, to name a few. These differences suggest that the organization, management, control and execution of ERP maintenance and upgrade are not purely an internal issue driven by internal users and IT-staff only, as with the traditional in-house software. However, neither are ERP maintenance and upgrade activities completely controlled by the vendor and outsourced to a third-party. In contrast, both the ERP software vendor and client-organization have significant influence on the maintenance and upgrade activities. The vendor plays an important role in maintenance support, such that client-organizations' maintenance management and upgrade decisions and processes have become more complex than before.

According to Pigoski (1997), a maintenance model is usually defined and used in order to reflect and capture the essence of an organization's software maintenance procedures and management issues. The main advantages of a maintenance model are that it helps to define, plan and manage maintenance activities; thus it improves maintenance processes, and facilitates modification of the software (IEEE 1998, ISO/IEC 1995). It provides clarity to foster understanding and communication among all parties involved, facilitates effective maintenance support to the system users or stakeholder in general, and therefore helps in reducing the effort and cost of maintenance (Sneed 1996).

According to Croll (2002), chair of IEEE Software Engineering Standards Committee, standards are consensus-based documents that codify best practices, and represent the collected experience of others in the same field dealing with the same issue. The use of standards (such as IEEE/EIA 12207, a maintenance model) is noted to assure that software projects are successfully completed on time, within budget and with the intended benefits (Brotbeck, Miller & Statz 1999). Card in (Ferguson & Sheard 1998) states that standards improve communication between and within organizations by defining concepts and terminology, and by setting expectations for performance. Other authors (Heineman, Botsford, Caldeira, Kaiser, Kellner & Madhavji 1994) agree that the motivations for applying models of software processes are to facilitate human understanding, support process improvement, automate processes, and support process management. Although there are standard software maintenance models, they are designed for internally maintained software. A standard maintenance model for large commercial off-the-shelf software is lacking. Neither is there any such model for ERP in particular, which is maintained by both the client-organization and software vendor in 'partnership'. Are the published standard models for (in-house) software maintenance appropriate for the ERP environment? Or, are activities involved in ERP maintenance and upgrade somehow unique such that a new maintenance model is warranted? The study reported herein entailed a comprehensive case study of a large Government agency aimed at understanding current practices in ERP maintenance and

upgrade activities, and investigating whether existing maintenance models are sufficient in the context of ERP.

Discussion herein on the maintenance model is focused on details of the activities involved in managing and executing software maintenance preparation, software maintenance procedures, and software upgrades; and the order in which these activities are performed. These software process modelling requirements are viewed as highly desirable characteristics of modelling methodology by the Software Engineering Institute (SEI) (Kellner & Hansen 1988). This paper proceeds by reviewing the literature on software maintenance models in Section 2. Activities covered by the models are elaborated, and implications for this research are drawn. In Section 3, the research method, data collection and data analysis are described. Section 4 provides an in-depth illustration of the case organization's ERP maintenance model. Deficiencies in IEEE/EIA 12207.2-1997 in the context of the ERP case are identified and discussed in Section 5. The paper concludes with discussion on possible future research.

2 Literature Review

2.1 IEEE Standard 1219-1998 and ISO/IEC 12207

Review of the literature reveals two well-recognized, standard software maintenance models. The first, IEEE Standard for Software Maintenance (IEEE Standard 1219-1998), from the Institute of Electrical and Electronics Engineers (IEEE), is a revision of IEEE Standard 1219-1992 (1998) (The last four digits of the standard number represent the year of IEEE Standards Association (IEEE-SA) Standards Board approval). The IEEE standard is recognized by the American National Standards Institute (ANSI). This standard is intended for a wide ranging audience including software development managers, maintainers, software quality assurance personnel, software configuration management personnel, programmers, and researchers. This standard proposes seven phases in the in-house software maintenance process: (1) problem/modification identification, classification and prioritization; (2) analysis; (3) design; (4) implementation; (5) regression/system testing; (6) acceptance testing; and (7) delivery.

The second standard software maintenance model is from the International Organization for Standardization and International Electrotechnical Commission (ISO/IEC) – named Information Technology – Software Life Cycle Processes (ISO/IEC 12207). ISO/IEC 12207 is a standard for software life cycle processes, covering the acquisition, supply, development, operation, and maintenance processes (ISO/IEC 1995). In the current study context, discussion on ISO/IEC 12207 is focused on the maintenance process only. It lists six main activities of the in-house software maintenance process, namely: (1) process implementation; (2) problem and modification analysis; (3) modification implementation; (4) maintenance review and acceptance; (5) migration; and (6) software retirement.

IEEE Standard 1219-1998 versus ISO/IEC 12207 - The IEEE Standard 1219-1998 mainly emphasizes the activities after software delivery. On the other hand, ISO/IEC 12207 does not only cover software post-delivery activities, but also pre-delivery and software retirement activities. Pigoski (1997) states that in order to ensure that software maintenance (i.e. post-delivery activities such as software modification, training, and operating a helpdesk) are provided in a timely and cost-effective manner, the pre-delivery (e.g. planning for post-

delivery) and software replacement activities should be incorporated. IEEE Standard 1219-1998 (basic process model) includes input, process, output and control for software maintenance; and focuses more on the measures/metrics of maintenance effort, determinants of maintenance effort, error rates, and projecting future maintenance needs. On the other hand, the ISO/IEC 12207 basic process model involves the process only, with no specific attention given to the input, output and control for software maintenance.

2.2 IEEE/EIA 12207.2-1997

In 1995, the Software Engineering Standards Committee (SESC) of the IEEE Computer Society adopted ISO/IEC 12207 and used it as a basis for life cycle processes within the IEEE Software Engineering Collection. The outcome of IEEE and Electronic Industries Association (EIA) adaptation of ISO/IEC 12207, specifically relevant to this study is the IEEE/EIA Guide for Information Technology – Software Life Cycle Processes – Implementation Considerations or IEEE/EIA 12207.2-1997. It uses the same activity-names as in ISO/IEC 12207 (IEEE/EIA 1997). This standard also makes reference to other related IEEE standards for example IEEE 1219-1998. This is illustrated in Figure 1.

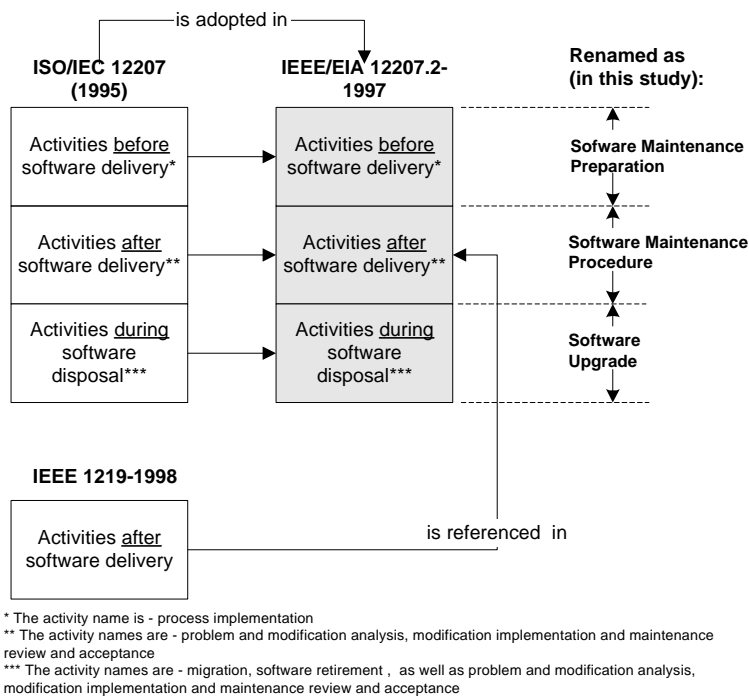


Figure 1. The standards referenced in this study.

IEEE/EIA 12207.2-1997 is used as the basis of discussion on the standard for maintenance model for the remainder of this paper. The primary reason for choosing IEEE/EIA 12207.2-1997 (instead of selecting IEEE Standard 1219-1998 or ISO/IEC 12207) is that not only does it include both the pre-delivery activities and software retirement (i.e. the ISO/IEC 12207 characteristics), but also quantification factors and metrics for measurable maintenance attributes (e.g. maintenance effort, replacement policy, etc.). The latter is generally a strength of IEEE standards. The primary tasks involved in each activity are summarized in Table 1.

	Activity	Task
1	Process implementation	Develop plans for conducting maintenance activities; establish controls, rules and methods to record and track maintenance requests; outline workflow of a maintenance request; define a maintenance organization; describe the arrangement for resource allocations and performance tracking; define anticipated future maintenance requirements; identify maintenance effort determinants; define software configuration management (SCM) processes for managing modifications to the existing system
2	Problem and modification analysis	Classify maintenance request; assign priority; verify and analyze the impact of the problem; identify alternative solutions; perform preliminary cost and benefit estimation for the modification; approve the selected modification option; carry out a detailed analysis on the modification.
3	Modification implementation	Identify the affected module; modify software module documentation; create test cases for the new design, safety and regression testing; detail documentation to be updated; define the test and evaluation criteria; do coding; conduct unit testing; integrate the modified software with the system; perform integration/regression test; carry out risk analysis; implement a test-readiness review to assess preparedness for system and acceptance test; notify user community of the product delivery schedule; develop an archival version of the system for backup; perform installation and training at customer site
4	Maintenance review and acceptance	Conduct review(s) with the modification authorizer's organization to verify the integrity, interoperability test and interface test of the modified system; conduct functional configuration audit; and perform physical configuration audit
5	Migration*	Develop, document, and execute a migration plan; notify users of the migration plans; conduct parallel operations of the old and new environment and provide the necessary training; notify the relevant parties of the migration schedule; conduct a post-operation review to assess the impact of the changes to the new environment; archive the data, documentation, logs and code used by or associated with the old environment
6	Software retirement **	Develop a software replacement policy by taking into consideration the replacement drivers; develop, document and execute a retirement plan; notify users of the retirement plans and activities; parallel operation of the retiring and new software product, and provide user training; notify those involved the retirement schedule; archive the data, documentation, logs and code used or associated with the retired software product.

* It is intended for the situation where a software product is migrated from an old to a new operational environment.

** It occurs when a software product is retired from the production (based upon a request of the owner).

Table 1. Activities and tasks covered in the IEEE/EIA 12207.2-1997.

In this study we adopt the three maintenance life cycle stage names given by the ISO/IEC (1995) and Pigoski's (1997) maintenance model, namely - *software maintenance preparation*, *software maintenance procedure*, and *software upgrade* stages (see Figure 1). These three stage-names are proposed in this study rather than using the maintenance activity-names given in the existing standard (i.e. ISO/IEC 12207) because the proposed terms are less ambiguous and are more intuitive. For instance, maintenance preparation is more intuitive than process implementation; and software upgrade is more generally used in the ERP context rather than software retirement.

2.3 IEEE/EIA 12207.2-1997 and implications for ERP packaged software maintenance

The IEEE/EIA 12207.2-1997 standard is comprehensive and detailed, covering most of the fundamental tasks in each of the salient maintenance activities. However, the standard seems lacking with regard to several fundamental aspects of the ERP maintenance environment. This is summarized in Table 2.

Issue	IEEE/EIA 12207.2-1997 – emphasizes	ERP – emphasizes
Software environment	Internally maintained software	Externally maintained software, i.e. maintenance support is readily available from the vendor (Ng et al. 2002)
Role involved	Addresses the roles of the users, maintainer, and maintenance manager	ERP vendor, and third-party vendor (Hirt & Swanson 2001)
Type of maintenance request	The modification implementation activity – focuses on software modification such as perfective, adaptive or corrective maintenance of the custom code (in-house software)	Implement user-enhancement to the vendor (standard) code; implement patch maintenance originated from the vendor (Ng 2001b)
Maintenance effort	The modification implementation – emphasizes writing software code	Research for available maintenance support from the vendor (patches and updates (Nah, Faja & Cata 2001)); Relatively more effort on impact analysis of the modification or new changes to the installed version (Collins 1999)

Table 2. Emphasis in IEEE/EIA 12207.2-1997 and ERP maintenance environment.

3 Research methodology

3.1 The case study and data collection

The case study organization is an ERP-employing Government Agency (GA) in Australia. GA was established in July 1996 and is a shared-service provider to other Government departments. GA has approximately 270 staff and its annual budget is around A\$20M. It has implemented the SAP R/3 Finance and Human Resources modules, and has more than four years experience in the management of ERP maintenance activities. These include managing and implementing patches, corrective maintenance, and enhancement maintenance. The maintenance data collected from GA was based on its previous SAP R/3 version 3.1H. Currently, GA is using a newly installed (since April 2002) SAP R/3 version 4.6C. The case study sought to gain in-depth understanding of GA's maintenance preparation, maintenance procedure (for different types of maintenance requests), and software upgrade activities. Sources of study evidence included semi-structured interviews, maintenance activity database (consisting of all change requests), user support database, upgrade business case, and upgrade planning resources report.

Semi-structured interviews were conducted with the senior maintenance managers - Systems Development Manager and Systems Operations Manager, and a Business Analyst. These interviews helped in understanding the ERP maintenance model. Issues discussed in the

interviews included the ERP maintenance preparation, maintenance procedure for different maintenance requests, and the software upgrade process. Both the ERP maintenance database and the user support database were investigated to identify types of maintenance requests implemented by GA and participants in these maintenance projects, and to gain insight into activities and tasks, which occurred along the maintenance procedure. The upgrade business case and upgrade planning resources documentation were consulted to identify procedures involved in the upgrade preparation and upgrade execution, and issues resolved along the upgrade process.

3.2 Data analysis

Data gathered from the interviews, relating to: (i) maintenance preparation and initial planning; (ii) maintenance procedure or workflow; and (iii) the upgrade process, are mapped onto the three main stages in GA’s maintenance model: maintenance preparation, maintenance procedure, and software upgrade stages respectively (see Figure 2). GA implicitly follows these stages. Data collected from the databases relating to maintenance activities associated with different request types are mapped onto the maintenance procedure stage; whereas information from the upgrade business case and upgrade planning resources documentation that are connected to the ERP maintenance preparation and upgrade process is mapped onto the maintenance preparation and software upgrade stages respectively (of GA’s maintenance model).

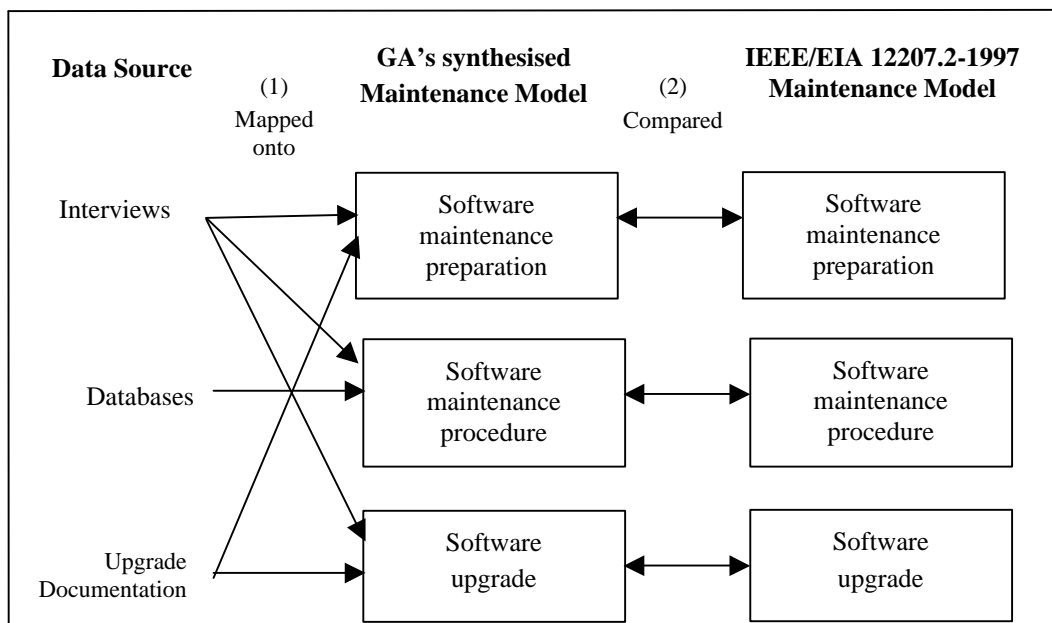


Figure 2. Data analysis – the process of synthesizing GA’s ERP maintenance model.

For comparability with GA’s maintenance model, IEEE/EIA 12207.2-1997 process implementation activity is mapped onto its maintenance model of software maintenance preparation stage; problem and modification analysis, modification implementation and maintenance review and acceptance are mapped onto its software maintenance procedure stage; and migration and software retirement, as well as problem and modification analysis, modification implementation and maintenance review and acceptance are mapped into its

software upgrade stage. GA's maintenance model is then compared with the IEEE/EIA 12207.2-1997 maintenance model stage-by-stage, activity-by-activity, and task-by-task.

4 Findings

In this section, we present GA's synthesised (ERP) maintenance model. Discussion on the activities and tasks involved in GA's maintenance model has been largely condensed due to the size limit for this paper. This is summarized in Table 3.

Software maintenance preparation – Senior management of GA participate actively in the maintenance preparation stage. They pay considerable attention to vendor support issues, benefit-realization (from the ERP system), maintenance expenditures, maintenance services provided to its clients, and other maintenance management issues. Information on the basic activities involved in this stage has been distilled from data collected in the interviews and GA's upgrade documentation (see Figure 2).

Software maintenance procedure – GA's maintenance activities are initiated from essentially two sources: the system users and IT-staff, and the software vendor. The former source introduces requests such as user support, corrective requests and enhancement requests. The latter introduces patches and new versions for upgrade. User support requests are related to: simple security issues such as modifying or setting-up user profiles and changing authorization profiles; consultation on system usage and software functionality; and training. All other (system users and IT-staff) requests are called change requests by GA. Activities detailed at this stage (in Table 3) are synthesised from interviews conducted with senior management and/or substantiated by information in GA's maintenance and user support databases (as illustrated in Figure 2). Maintenance requests are usually reported to GA's help desk. Once a request has arrived, all its maintenance details will be recorded in the relevant database. Steps involved in processing the system user and IT-staff maintenance requests are as follows, depending on the type of request. Note that for corrective requests, regardless of the originator (either the system users or IT-staff) the same maintenance procedure is followed. The main difference between the procedure for a system-user-initiated enhancement, and an IT-staff-initiated enhancement, is that the latter will not incur a cost to the system users¹. Figure 3 (derived from the interviews with GA) shows the flowchart of maintenance procedure for requests initiated by system user and IT-staff in GA.

Software upgrade – According to GA, the upgrade process is very similar to the patch maintenance procedure, the main differences being that upgrade requires more thorough planning and business justification, more effort for impact analysis and re-application of previous modifications or user-enhancements (if the new version has not incorporated the required functionality), longer time to complete, more money and resources to implement,

¹This is an idiosyncrasy of the service provider role of GA, and the particular charge-back arrangements it has with the client agencies. This situation is not unique to ERP. However, increased research attention to maintenance implications of outsourcing and shared services is warranted.

and serious consideration of potential system downtime. The basic activities involved at this stage (see Table 3) are distilled from GA's upgrade documentation and/or further substantiated by the interview transcripts (as shown in Figure 2).

Stage	Activity / task involved
Maintenance preparation	<p>Define the core objectives of maintenance and/or the objectives of the ERP implementation.</p> <p>Identify the scope, benefits, costs, and risks of the system; estimate resources required and/or outsourcing needs; outline maintenance support from the vendor, such as the support window² for the software, conditions to remain eligible for maintenance support, the types of maintenance support available from the vendor, how and where to get them; define maintenance organization, for instance maintenance unit(s) or team(s), and maintenance team(s) responsibilities and job specifications; define the maintenance management issues – all the environments needed for maintenance, a mechanism to identify and classify maintenance requests; establish maintenance strategies, including how each of the maintenance request-types is serviced (for example batch, and on-the-fly); define maintenance service for the system users, for example the types of maintenance support available to the users, how and where to access them, types of maintenance requests required to be charged back to the user’s organization, and what criteria the fees will be based on; establish a software configuration management (SCM) plan, including configuration identification, configuration control, configuration status accounting, configuration audits and review, interface control, and vendor control; develop training and help desk policies; define the maintenance procedure</p>
Maintenance procedure	<p><u>User support request</u></p> <p>Create and issue a user support form; classify and prioritize request (based on existing workload); study the root of the problem (e.g. inadequate training, needs for consultation on software functionality, incorrect user-access rights, and security issues); resolve the problem/request directly and/or direct the request to the right person for solution</p> <p><u>Change request</u></p> <p>Classify the request; obtain approval for the request; create a change request form; prioritize based on the request type</p> <p><u>Corrective – bugs found in vendor’s code</u></p> <p>Search for maintenance support through the Online Support System (OSS) notes and (if the bug fix is not available from the vendor) report the bugs to the vendor; apply vendor’s code and conduct impact analysis in the Development System (DEV); perform modification adjustment, system testing, and user acceptance test in Quality Assurance System (QAS); deliver the new system to the Production System (PRD)</p> <p><u>Corrective – bugs found in custom code</u></p> <p>Design solution for the problem and update the relevant documentation; implement changes in the DEV system; transport changes to the QAS system for testing and verification; transport the new system into the PRD system</p> <p><u>Enhancement – functionality not provided by the vendor</u></p> <p>If vendor-support is still not available, report the request to the vendor; propose and approve solution; conduct cost-estimation for the maintenance; issue a quotation form to the user, and obtain approval from the user; approve by the maintenance manager and prioritize request; design solution and update the relevant documentation; implement changes; transport changes to the QAS for testing and verification; transport the new system into the PRD</p> <p><u>Enhancement – available from the vendor</u></p> <p>Conduct cost-estimation for the maintenance; issue a quotation to the user; obtain approval from the user and maintenance manager, and prioritize</p>

² Support window is a time period, during which a client-organization is eligible for help desk support, bug fixes, and new and/or improved features from the vendor. Typically a vendor will support a given version of its software for 2-3 years, though the length of this period varies greatly.

	<p>request; apply the vendor’s code and conduct impact analysis in the DEV; perform modification adjustment, system testing and user acceptance test in QAS; deliver the new system into the production</p> <p>Patch maintenance</p> <p>Create and issue a change request form; implement the patch by applying or copying the patch into the existing ERP system using a program (supplied by the software vendor) specific for this purpose; conduct detailed impact analysis of the LCP on each of the previous user-enhancements or modifications; make modification adjustments – reapply the previous modifications (if necessary); transport changes to the QAS system, and perform complete system re-testing of performance and integration; transport the new system to the production</p>
<p>Software upgrade</p>	<p>Design a project methodology for the upgrade; research for upgrade options available, determine their availability dates, pros and cons, stability, and the support window (i.e. vendor maintenance support completion dates) of each option; decide on the type of upgrade (technical or functional); develop a business case to justify the upgrade decision, and identify the factors influencing this decision; plan for the upgrade date, which will minimize work disruptions and downtime; evaluate costs for the whole upgrade, and develop a detailed plan for budget allocations (including the hardware and training costs), and personnel requirements; assess project risks; make full assessment of the new features or functionality in each option and for each module of interest, consider of how a new functionality benefits the organization, and draft a plan for benefit realizations for the new business improvements; Make recommendation for the upgrade release or version; conduct impact analysis between the new upgrade version and the existing version; examine the impacts of the upgrade on user training, interfaces and desktop, reporting capability, supporting documentations, change management, testing and security; study the impacts of the upgrade on hardware sizing, database and application server capacity, and network loading requirements; install the new version onto the DEV; construction (or development) – reapplying previous modification and re-develop previous reporting capability, etc. (if necessary); conduct a thorough testing of the upgrade system, and user acceptance testing; carry out the trial upgrades between the DEV system and QAS system; conversion (or go live) – deliver the well-tested system into the production system</p>

Table 3. Summary of GA’s synthesised ERP maintenance model.

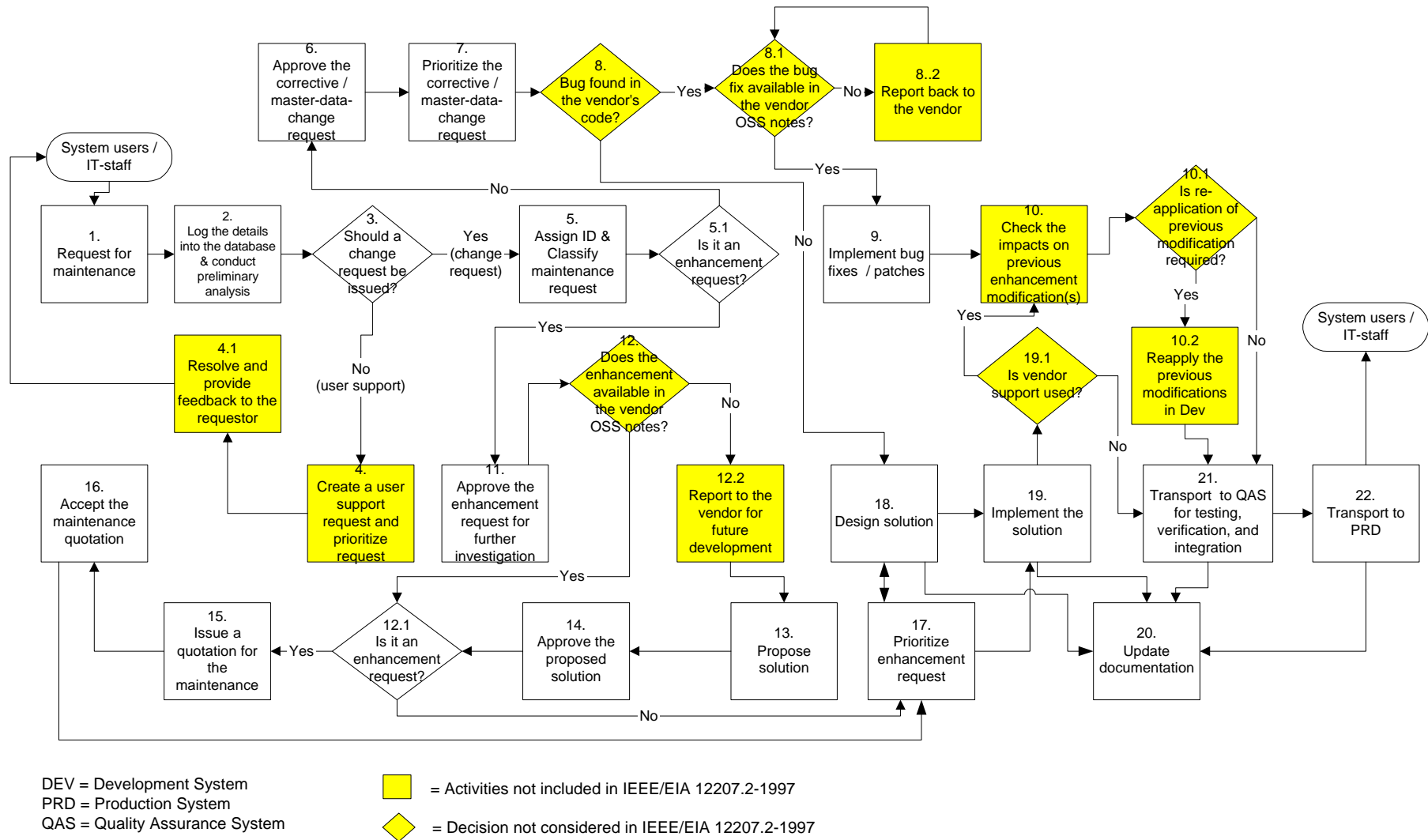


Figure 3: Flowchart of GA's maintenance procedure for requests that originate from the system user and IT-staff.

5 Discussion on Deficiencies in IEEE/EIA 12207.2-1997

Having reviewed in detail the activities involved in GA's ERP maintenance model, with some substantiation of these activities from related ERP literature, we now discuss deficiencies in IEEE/EIA Guide for Information Technology – Software Life Cycle Processes – Implementation Considerations (IEEE/EIA 12207.2-1997). These are identified by comparing IEEE/EIA 12207.2-1997 software maintenance preparation, software maintenance procedure and software upgrade stage with GA's software maintenance preparation, software maintenance procedure and software upgrade stage respectively. Note that the following discussion has relevance beyond the GA-specific case, and it supports our propositions in Section 2.3.

- Software maintenance preparation

Unlike traditional in-house software, ERP is off-the-shelf packaged software that is not only maintained by the ERP-employing organization but also by the vendor. This observation is consistent with the study by Hirt and Swanson (2001). The software vendor plays a significant role in an adopting-organization's maintenance activities. The vendor introduces maintenance activities (e.g. patches and new versions), and is responsible for continuous research and development of the software. This clearly influences an ERP-using organization's maintenance and upgrade decisions, strategies and policies. Thus, in planning for ERP maintenance, issues such as the magnitude and frequency of vendor maintenance activity and support, and the projected timing of withdrawal of vendor support for a given version, must be considered. Yet, the issue of vendor maintenance support is not considered in IEEE/EIA 12207.2-1997 standard for software maintenance model. Moreover, the maintenance effort determinant for example the system age discussed in the standard may not be applicable in the context of ERP. This is because system age is most likely an unknown factor (or transparent) to the ERP-adopting organizations. In contrast, the number of previous user-enhancements is found to affect the patch maintenance effort (Ng 2001a), yet is not discussed in the standard.

- Software maintenance procedure

From GA, we observe that ERP maintenance procedure not only involves maintenance activities originated from internal sources but also external requests from the vendor. Consistent with results reported in (Nah et al. 2001), searching for vendor bug fixes, bug reporting to the vendor, and enquiries for new functionality from the vendor are a few of the main maintenance activities in an ERP environment. An additional activity found in ERP maintenance is reapplying previous modifications (whenever necessary) every time patch maintenance (or upgrade) is implemented. This is because the custom code could be overwritten by a patch (or a new version) if the vendor makes changes to the same software code and incorporates these in the patch or new version. Thus, in order to retain the customized functionality, re-application of the previous modification(s) is required. (Otherwise, no coding is required at all for maintenance support from the vendor or using the vendor's code.) However, these main activities are not included in the IEEE/EIA 12207.2-1997 standard. Additionally, the standard does not incorporate the maintenance activity of resolving user support requests but merely addresses software change requests. According to more recent software maintenance taxonomies for in-house software (Chapin, 2000) and ERP software (Ng et al., 2002), user support requests are also

considered a software maintenance activity. The existing standard for software maintenance model needs to reconsider this factor in order to reflect the practical reality that these support requests entail substantial maintenance resources.

- Software upgrade

When upgrading an ERP system, the existing system is replaced by a new version readily available from the vendor. The user-organization must study and understand all new versions as they become available in order to decide which to implement. This entails fully assessing new functionality available from each version in order to identify the right version for the organization (note that this implicitly assumes that not all organizations will realize sufficient benefits to implement all new versions); and to prepare for the process of functionality comparison between a new version and the installed version during impact analysis. This research and selection activity is not considered in software migration or software retirement in IEEE/EIA 12207.2-1997. On the contrary, that standard focuses on the effort and procedures needed to develop, re-engineer and/or rewrite the existing system - activities that are typically, relatively less important, if required at all, with the implementation of a new package software version.

The installation of a new version will overwrite the existing system code. As a result, impact analysis between the new version and the installed version must be done, where the installed version, in addition to the 'vanilla' vendor code, also includes custom user-enhancements and modifications. This analysis and comparison process, will facilitate deciding which of the previous modifications or user-enhancements to keep (and re-apply to the new version once installed), and which are no longer needed. Here again, the existing standard does not include this activity of deciding whether to retain the installed user-enhancements. This exclusion from the standard is understandable because in-house software, for which the standard caters, is fully tailored to an organization's business requirements. This is different from the generic ERP application software that most likely does not fit perfectly with the adopting-organization's culture and business processes.

Table 4 below provides a summary of our findings and discussions so far, specifying deficiencies in IEEE/EIA 12207.2-1997, and flagging where these deficiencies (in terms of maintenance activities) are ERP-specific. Note that the symbol '—' means that the deficiency is irrelevant to the particular software environment (designated by the column), whereas 'X' indicates that the deficiency is a concern.

Stage		Deficiency	In-house	ERP
Maintenance Preparation	1	Inclusion of vendor maintenance support	—	X
Maintenance Procedure	2	Search availability of maintenance support	—	X
	3	Report maintenance requests to the vendor	—	X
	4	Make decisions on prior user-enhancements - keep, replace or abort; and reapply previous user-enhancements (after patch maintenance)	—	X
	5	User support activities	X	X
Replacement /upgrade	6	Make research on available upgrade options	—	X
	7	Fully assess new functionality in each (potential) new version	—	X
	8	Make decisions on prior user-enhancements; reapply previous user-enhancements (after upgrade)	—	X

Table 4. Summary of deficiencies in IEEE/EIA 12207.2-1997 and its relevancy to different software environments.

6 Conclusion and future research work

The key objective of this paper has been to investigate whether the widely cited standard for software maintenance model (IEEE/EIA 12207) is sufficient in the ERP context. Our findings show that IEEE/EIA 12207.2-1997 does not adequately cater for the ERP maintenance environment. Key differences in an ERP environment are: long-term vendor maintenance support; searching and implementing maintenance support from the vendor, as well as reporting maintenance requests to the vendor; deciding whether to keep prior user-enhancements, and reapplying them (if applicable) after patch maintenance or upgrade; and fully assessing new functionality in each (potential) upgrade version.

This study has several limitations. Firstly, the findings on the ERP upgrade process at GA are mainly based on a technical upgrade only, and on upgrade to a new version from the same vendor (as the installed version). Secondly, the results from this study are based on a single-case in the public sector. Thus, the extensibility of the results may be limited (Baskerville & Lee 1999, Yin 1994). However, the findings here are believed to be applicable and extendable to ERP organizations having the similar characteristics as GA.

Follow-on research aimed at building and validating a standard for ERP maintenance model, through multiple case studies, surveys, and Delphi studies involving practitioners from different sectors, industries, and research institutions is a worthwhile endeavour. This work should identify the commonalities and major differences in maintenance models employed across different ERP-employing organizations based in different sectors and industries. Hopefully, this will produce a generalizable ERP maintenance model, that can be used by new or existing ERP-employing organizations to learn and make use of best practices in ERP maintenance processes in order to save these organizations time, cost, and human capital.

References

- Baskerville, R. and Lee, A. S. (1999), "Distinctions Among Different Types of Generalizing in Information Systems Research" in *International Working Conference on New Information Technologies in Organizational Processes: Field Studies and Theoretical Reflections on the Future of Work*, Kluwer Academic, pp. 49-65.
- Brotbeck, G., Miller, T. and Statz, J. (1999), "A Survey of Current Best Practices and Utilization of Standards In the Public and Private Sectors", TeraQuest Metrics, Inc., viewed 22 January 2003, <<http://www.dir.state.tx.us/eod/qa/bestprac.htm>>.
- Chapin, N. (2000), "Software Maintenance Types--A Fresh View" in *International Conference on Software Maintenance*, IEEE Computer Society: Los Alamitos CA, pp. 247-252.
- Collins, K. (1999), "Strategy and Execution of ERP Upgrades", *Government Finance Review*, vol. 15, no. 4, pp. 43-47.

- Croll, P. R. (2002), In *14th Annual DoD Software Technology Conference (IEEE-sponsored track)* presentation slide.
- Ferguson, J. and Sheard, S. (1998), "Leveraging Your CMM Efforts for IEEE/EIA 12207", *IEEE Software*, vol. September/October, no. 23-28.
- Heineman, G. T., Botsford, J. E., Caldiera, G., Kaiser, G. E., Kellner, M. I. and Madhavji, N. H. (1994), "Emerging Technologies That Support A Software Process Life Cycle", *IBM Systems Journal*, vol. 33, no. 3, pp. 501-520.
- Hirt, S. G. and Swanson, E. B. (2001), "Emergent Maintenance of ERP: New Roles and Relationships", *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 13, no. 6, pp. 373-397.
- IEEE (1998), *IEEE Standard for Software Maintenance, IEEE Std 1219-1998*, Institute of Electrical and Electronics Engineers, New York.
- IEEE/EIA (1997), *IEEE/EIA 12207.2-1997 Guide for Information Technology - Software life cycle processes - Implementation considerations*, Institute of Electrical and Electronics Engineers, Inc., New York NY.
- ISO/IEC (1995), *ISO 12207: Information Technology - Software Life Cycle Processes*, ISO/IEC, Geneva, Switzerland.
- Kellner, M. I. and Hansen, G. A. (1988), Software Engineering Institute (SEI), Carnegie Mellon University, Pittsburgh, Pennsylvania.
- Markus, L. and Tanis, C. (1999), In *Framing the Domains of IT Management: Projecting the Future Through the Past* (Eds, Zmud, R. W. and Price, M. F.) Pinnaflex Educational Resources, Cincinnati, OH, pp. 173-207.
- Nah, F., Faja, S. and Cata, T. (2001), "Characteristics of ERP Software Maintenance: A Multiple Case Study", *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 13, no. 6, pp. 399-414.
- Ng, C. S. P. (2001a), "A Decision Framework for Enterprise Resource Planning Maintenance and Upgrade: A Client Perspective", *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 13, no. 6, pp. 431-468.
- Ng, C. S. P. (2001b), "A Framework for Enterprise Resource Planning Maintenance and Upgrade Decisions" in *Americas Conference on Information Systems*, Association for Information Systems: Atlanta, GA (on CD), pp. 1026-1029.
- Ng, C. S. P., Gable, G. G. and Chan, T. (2002), "An ERP-client Benefits-oriented Maintenance Taxonomy", *Journal of Systems and Software*, vol. 64, no. 2, pp. 87-109.
- Pigoski, T. M. (1997), *Practical Software Maintenance: Best Practices for Managing Your Software Investment*, John Wiley & Sons, Inc., New York.
- Sneed, H. M. (1996), "Modeling the Maintenance Process at Zurich Life Insurance" in *International Conference on Software Maintenance*, IEEE Computer Society: Los Alamitos CA, pp. 217-226.
- Yin, R. K. (1994), *Case Study Research: Design and Methods*, Sage Publications, Thousand Oaks, California.