Maintaining ERP Packaged Software
– A Revelatory Case Study

Abstract

For many organizations, maintaining and upgrading enterprise resource planning (ERP) systems (large packaged application software) is often far more costly than the initial implementation. Systematic planning and knowledge of the fundamental maintenance processes and maintenance-related management data are required in order to effectively and efficiently administer maintenance activities. This paper reports a revelatory case study of GSP, a high-performing ERP service provider to government agencies in Australia. GSP ERP maintenance-process and maintenance-data standards are compared with the IEEE/EIA 12207 software engineering standard for custom software, also drawing upon published research, to identify how practices in the ERP context diverge from the IEEE standard. While the results show that many best practices reflected in the IEEE standard have broad relevance to software generally, divergent practices in the ERP context necessitate a shift in management focus, additional responsibilities, and different maintenance decision criteria. Study findings may provide useful guidance to practitioners, as well as input to the IEEE and other related standards.

INTRODUCTION

Enterprise resource planning systems (ERP), also known as enterprise systems, are configurable, and cross-functional standard software packages (Kumar and van Hillegersberg, 2000). ERP provides integrated functionality for most fundamental business functions and processes (e.g. financials, human resources management, sales and distribution, manufacturing and logistics). For some industries with com-

plex operational characteristics (such as metal, food and chemical), ERP has become a competitive necessity (Ragowsky and Gefen, 2008). Despite ERP market maturity and diminished sales opportunities from large enterprises since 2000 (due in example to saturation), ERP vendors generated USD86 billion in revenues in 2005 and expect to reach USD137 billion in 2010 mainly through maintenance fee growth (Whiting, 2006).

Like traditional in-house software, ERP packaged software requires substantial maintenance. Unlike traditional in-house software, ERP maintenance can be divided into vendor-side maintenance and client-side maintenance. While software maintenance performed by the vendor applies across their customer base, software maintenance by the client (including implementing maintenance support provided by the vendor) applies to the client's installed version only. A survey by Glass and Vessey (1999) suggests that total annual client-side ERP maintenance costs, inclusive of annual vendor maintenance charges of approximately 17%[1] of original software costs (Wailgum, 2008), total approximately 25% (5-50%) of original implementation costs. Assuming that the original software costs are 15% of original implementation costs (O'Brien and Marakas, 2008), then based on the Glass and Vessey study, total annual client-side maintenance costs could represent up to 167% of original software costs (25%/15%). The focus in the current paper is on these client-side maintenance activities. During the 20 months period for which related records were available, GSP, the case firm studied, spent 12,480 hours on client-side maintenance activities in relation to their ERP system (equivalent to annual costs of USD873,600 at an average full-cost hourly rate of USD70).

The majority of large organizations have already implemented ERP, with adoption over the past decade (Reilly, 2005) spreading to medium-sized then small-sized enterprises (SMEs) (Jacobson et al., 2007). The large and growing installed base of ERP means market expenditure on ERP maintenance related services is escalating. As with software generally, maintenance activity is the longest and most costly phase in the lifecycle for ERP software (Glass, 2003). Inappropriate management of ERP maintenance can be

costly and may lead to - failure in meeting user and customer expectations, failure to realize business benefits, failure in daily business transactions, and even failure in ERP implementation as a whole. However, very little is known about the processes and data involved in ERP maintenance. Nidumolu and Subramani (2003) emphasized the value of centrally devised and mandated standards (i.e. methods and performance criteria) as an effective means of controlling the software process, which can ensure a better level of performance from the maintenance team. Achieving such control requires a predefined, standardized, and well-informed maintenance process. A survey of ERP users found that most (i.e. almost 93%) would consider adopting a 'de jure' standard (i.e. a standard that is supported by the international standardization bodies) if one existed for ERP maintenance management (Ng et al., 2006).

The current study thus addresses the following research questions; "How is ERP maintenance conducted at GSP (i.e. the case firm examined in this case study)? What are the activities and corresponding required maintenance management data?" and "How do these maintenance activities and maintenance management data diverge from the widely available in-house software maintenance-process and maintenance-data standards described in IEEE/EIA 12207?" Each of these questions is addressed through the case study, which aims to describe the activities and data involved in ERP packaged software maintenance management, and to identify extensions required to the de jure standards IEEE/EIA 12207.0 and 12207.1 for ERP maintenance-management (including upgrades). The case study is considered revelatory in that the researchers had access to inordinately detailed, relevant and complete data, and opportunity to analyze a phenomenon that was previously inaccessible.

In the following section, the literature on differences between packaged software and in-house software is reviewed, as well as that on maintenance-processes and maintenance-data standards. Section 3 describes the case study data collection and data analysis, while Section 4 synthesizes the case organization's ERP maintenance-process and maintenance-data. Section 5 discusses the unique activities and maintenance

data involved in ERP maintenance and compares these to the IEEE/EIA 12207 standard. Lastly, Section 6 presents study conclusions and suggests potential future research directions.

LITERATURE REVIEW

An Overview of Packaged Software and In-House Software Differences

Packaged software and in-house software differ in terms of aspects such as the rate of software change, source of costs, risks, control and relationship management, time-to-market, and maintenance responsibility. In response to market competition, packaged software tends to change more frequently through product updates or patches[2] (Hybertson et al., 1997; Carney et al., 2000) than in-house software. Packaged software using-organizations incur costs for both annual maintenance fees payable to the vendor, as well as for typical in-house software corrective, adaptive and enhancement maintenance (Hybertson et al., 1997).

Packaged software entails vendor viability and maintenance support risks (Davis, 1988), as well as software product reliability and fit with user requirements risks (McDermid, 1997). Additionally, packaged software using-organizations must also manage the relationship with the vendor (Oberndorf et al., 2000). On the other hand, developing in-house software may involve development risks, cost/budget risks, system performance risks, and insufficient documentation risks (Davis, 1988). In general, the time-to-market of packaged software is relatively shorter than that of in-house software, and has lower resource costs to the client-organizations (Tran and Liu, 1997; Voas, 1998). Further, packaged software maintenance is shared with, and can be outsourced to the vendor (Hybertson et al., 1997), whereas in-house software is generally entirely maintained internally. Appendix 1 summarizes the main differences between packaged and in-house software as identified from the literature.

Packaged Software Maintenance

Reifer et al. (2003) state that packaged (commercial off-the-shelf) software maintenance is relatively immature, and that few software lifecycle models address the packaged software maintenance process. Hybertson et al. (1997) point out that packaged software products require different kinds of maintenance activities than those required for a mostly custom system. Carney et al. (2000) agree, suggesting additional difficulties not typically encountered with in-house software systems due to the packaged software maintenance environment not being completely under the client maintainer's control.

A review of the packaged software literature suggests several maintenance activities unique to the packaged software environment. Packaged software such as ERP is purchased from a vendor, and continuously improved and maintained by the vendor. Hirt and Swanson (2001) note that ERP maintenance requires sharing of the maintenance tasks between the client-organization, the vendor and possibly third-parties (e.g. ASP, consulting firms). The ERP client-organization reports maintenance problems to the vendor (and consultant) and thereafter tracks problem resolution by the vendor (Nah et al., 2001). At a technical level, the implementation of ERP software "patches" and the upgrading to new versions of the package entail a thorough impact analysis to identify the object to be modified and estimate the potential consequences of carrying out the change (Ajila, 1995). This in turn involves making adjustments to previous modifications if they have been overwritten (Shi and Qian, 2001).

Carney et al. (2000) suggest that with packaged software, additional effort is most often required to negotiate with the vendor respective responsibilities for ongoing maintenance, determine the types of changes that can be made to the packaged software, and estimate the costs of retesting and recertifying the software. Additional effort is also required to manage licensing issues (Oberndorf et al., 2000) and contractual issues (Carney et al., 2000).

Further, packaged software upgrades require that client-organizations monitor the future development capability and financial situation of the vendor while continuously evaluating the upgrade options (Reifer et al., 2003). Oberndorf et al. (2000) suggest that the client must also maintain current knowledge of the available and emerging marketplace relevant to the installed packaged software.

The distinctive characteristics of packaged software maintenance activities as discussed above suggest a need to investigate lifecycle-data within the packaged software context. For example, Carney, Hissam and Plakosh (2000), based in the Software Engineering Institute (SEI) at Carnegie Melon University, suggest that the modification of packaged software code is an important decision, requiring that certain specific data items be captured (see Appendix 2). To that end, we now turn to an investigation of the current IEEE/EIA 12207.0 maintenance-process standard.

IEEE/EIA 12207.0 Maintenance-process Standard

IEEE/EIA[3] 12207 is known as the Industry Implementation of the International Standard ISO/IEC[4] 12207, which is a software engineering standard that comprises maintenance-process and maintenance-data standards. It is comprised of three parts, i.e. 12207.0, 12207.1 and 12207.2. IEEE/EIA 12207.0 (the base standard) contains ISO/IEC 12207 - Information Technology Software Lifecycle Processes, in its original form. The second part (i.e. 12207.1) is designed to provide more in-depth guidance on the data to be recorded, and the third part (i.e. 12207.2) describes the orderly activities to be implemented in software lifecycle processes. In principle, aside from the amount of documentation and detail, the key processes covered in IEEE/EIA 12207.0 and IEEE/EIA 12207.2 are fundamentally similar. Thus, in this paper we hereafter refer to IEEE/EIA 12207.0 as the process-standard, and IEEE/EIA 12207.1 as the data-standard. Also, 'IEEE/EIA 12207' refers to both standards unless otherwise specified. In the current study, we define a process standard as a standard that describes a set of predefined activities or tasks organized in an

interconnected sequence in order to achieve specific objectives, while a data standard is defined as a standard that informs the set of data items and/or information items to be collected during the activities undertaken in a software process.

According to Croll (2002), standards codify best practice and represent the collective experience of others in the same field dealing with the same issue. Card (1998) in (Ferguson and Sheard, 1998) states that "standards improve communication between and within organizations by defining concepts and terminology, and by setting expectations for performance" (p. 24). A process-standard such as IEEE/EIA 12207.0 provides the guiding principles, strategies, framework, and visibility for practitioners to plan and conduct maintenance procedures and manage software replacement in a systematic manner. Usually the de jure process-standard or methodology is adapted based on the organization's requirements and environment; the resultant visibility of maintenance activities allows deficiencies or bottlenecks in the maintenance process to be detected faster, and corrected by improving and/or reengineering the maintenance process.

IEEE/EIA 12207.0 covers five main processes in the software lifecycle – acquisition, supply, development, operation, and maintenance. The maintenance-process is comprised of six main activities- process implementation, problem and modification analysis, modification implementation, maintenance review and acceptance, migration, and software replacement.

Although this study is particularly interested in the maintenance-process, the acquisition-process is also relevant to packaged software maintenance. The acquisition-process manages activities associated with acquiring a system, software service or software products (including off-the-shelf software) from the supplier, and thus is pertinent to the acquisition of ERP version upgrades. According to the IEEE/EIA 12207.0 standard (1997), the acquisition-process includes the following activities- initiation, request-for-proposal preparation, contract preparation and update, supplier monitoring, and acceptance and completion. The IEEE/EIA 12207.0 standard is comprehensive and detailed, covering most of the fundamental

tasks related to each of the salient maintenance activities for software in general. Appendix 3 describes in detail each activity covered in the maintenance- and acquisition-processes of the IEEE/EIA 12207.0 standard, and is also relevant to our subsequent data analysis.

IEEE/EIA 12207.1 Maintenance-data Standard

In this study, maintenance-data is defined as a collection of facts from which conclusions may be drawn about maintenance tasks or activities within a maintenance process. Data can be collected using forms (e.g., maintenance request form, change report, software engineering report), interviews (with users, testers, programmers, analysts, managers), and via computerized systems (e.g. built-in change management system in ERP, maintenance request database); while raw data alone is often insufficient for managers, useful information or metrics can be derived from valid data.

The information and data items prescribed in IEEE/EIA 12207.1 and required as part of the maintenance- and acquisition-processes are summarized in Appendix 4. Note that 'data item' is defined as a single set of facts (regarding an event, activity, person, or object), whereas 'information item' (such as a report, plan, form, or record) consists of a group of data items.

Though the data and information items covered by IEEE/EIA 12207.1 appear comprehensive in terms of software generally, given the unique characteristics of packaged software (e.g. see Appendix 1), there is merit in evaluating the possible need for unique ERP packaged software maintenance-data.

STUDY METHODOLOGY

A qualitative case study research strategy was employed to explore and describe the maintenance-process and maintenance-data within the ERP context. As our primary interest centers on investigating the process involved, or the "how" research questions regarding ERP maintenance (a contemporary set of events over which we have no control during the study), the case study research strategy is appropriate

(Yin, 1994). Further, a 'single' case study was conducted, which is justified on the grounds that the case is revelatory; according to Yin (1994), a single revelatory case study is justified where there is "an opportunity to observe and analyze a phenomenon previously inaccessible to scientific investigation" (p. 40). Although ERP maintenance issues exist in all ERP client-organizations, few researchers have previously gained access to the detailed data required to carefully compare ERP maintenance-process and maintenance-data practice with those of IEEE/EIA 12207.

Research Procedures

According to Maxwell (1996), threats to the validity of qualitative research stem from personal biases of the researcher, as well as participant reactions during the study. These threats were minimized in this study by - (1) giving the study participants a chance to confirm and validate the study data, the researchers' interpretations of the data, and the study conclusions; (2) collecting multiple sources of rich, descriptive data to lessen the likelihood of important omissions; and (3) recording interviews and producing verbatim transcriptions (to minimize the threat of inaccuracies). These measures helped to control against potential subjectivity in accordance with the recommendations of Maxwell (1996) and Yin (1994). To increase the reliability of the case study research, we employed a detailed case protocol and maintained a case study database. The case protocol contains an overview of the research topic, case study questions and field study procedures that guided evidence collection and analysis. The case study database serves as a readily accessible central store of data and information references collected; it is comprised of all the interview-transcripts, interview audiotapes, databases and documentation, case study notes, emails, and survey questions and responses collected throughout the case study, as well as an annotated bibliography describing the contents of each data source collected.

Case Organization

This case study was based on the ERP maintenance activities of a medium-sized State government agency in Australia. The agency, GSP (Government Services Provider - a fictitious name for the purposes of this study), was a corporate information system service provider to other state government agencies and departments (see Figure 1). A key objective of the case study was to carefully describe GSP's implicit (undocumented) ERP maintenance-process and maintenance-data. The GSP case study is revelatory given the fortuitous availability of complete and detailed relevant historical data; GSP had retained several years of detailed ERP related maintenance activity records, which were made accessible to the researchers.

GSP was established in 1996; its business mission to provide integrated business support solutions to government departments and agencies. At the time of the study, GSP was a service provider for two government departments and three government agencies (see Figure 1) that, combined, had approximately 8000 employees. GSP had 260 staff members and its revenue base was approximately USD15.4 million per annum. GSP's mandate was to facilitate corporate resource sharing. The provision of corporate information system services by GSP was in accord with a Service Level Agreement (SLA) between GSP and its aforementioned clients. In addition to supporting several legacy systems, GSP's major role was providing SAP R/3 application services. Seeking Y2K compliance, GSP had implemented two fundamental SAP R/3 modules – Financials and Human Resources – across its five clients. Data for this case study pertains to the version first installed – SAP R/3 3.1H. The case study sought to gain in-depth understanding of GSP's maintenance-preparation, maintenance-procedures (for different types of maintenance requests), and software-upgrade activities.

<<Insert Fig 1 Here>>

Data collection and analysis

Data were collected from multiple sources to ensure richness. These included semi-structured interviews with several high-level managers, various maintenance related databases (such as the user-support database, the change-request database, the patch-support database, and the SAP system modifications database), and documentation (for example, maintenance request forms, the upgrade business case, and the upgrade planning resources report). As depicted in Figure 2, the two major steps involved in the analysis of GSP's ERP maintenance-process and maintenance-data model involved (1) identification and (2) comparison. **'**Identification' involved mapping the relevant activities and data pertaining to maintenance and upgrade activities into the three main stages of GSP's ERP maintenance-process in order to specify GSP's implicit (undocumented) ERP maintenance-process and maintenance-data (see the Identify step in Figure 2). 'Comparison' involved detailed analysis of the similarities and differences between the synthesized GSP (ERP) maintenance-process and IEEE/EIA 12207.0 (the common software maintenance-process standard), and between the synthesized GSP (ERP) maintenance-data and IEEE/EIA12207.1 (the common software maintenance-data standard).

<<Insert Fig 2 Here>>

GSP's Maintenance-process

Identification – Semi-structured interviews were conducted with senior maintenance managers, including the Systems Development Manager, the Systems Operations Manager, and a Business Analyst. Issues discussed in the interviews pertained to ERP maintenance preparation, maintenance procedures for different maintenance requests, and ERP upgrade activities. These interviews helped to better understand the GSP (ERP) maintenance-process, as well as their ERP upgrades. Next, the data was mapped onto the three main stages of the GSP (ERP) maintenance-process (GSP implicitly followed these stages) - maintenance-preparation, maintenance-procedures, and software-upgrades (see Figure 2). Software maintenance-preparation defines the orderly planning activities for maintenance management and related main-

tenance procedures. Software maintenance-procedures describe maintenance execution from the request initiation to the delivery of maintenance work on the production system. Also, this stage includes activities related to managing patch introductions and installations to the production system. Finally, software-upgrades define the activities involved in upgrading a software system, starting from the upgrade decision initiation to the completion of the new system for system users.

Further, the ERP change-request database, user-support database, patch support database, and SAP system modification database were investigated to identify types of maintenance requests implemented by GSP, and to gain insight into the overall activities and tasks performed across the maintenance-procedure (covering activities from the initiation of the maintenance request to the delivery and user acceptance of a maintenance solution based on the request). These activities and tasks were mapped onto GSP's maintenance-procedure stage. Also, the upgrade business case and upgrade planning resources documentation were consulted to identify procedures involved in upgrade preparation and execution, as well as to examine the issues resolved during the upgrade process. These procedures were then mapped onto the maintenance-preparation and software-upgrade stages of the GSP maintenance-process.

Comparison – To allow for comparisons with the GSP (ERP) maintenance-process, the IEEE/EIA 12207.0 process implementation activity was mapped onto their software maintenance-preparation stage (as shown in the lower section of Figure 2); problem and modification analysis, modification implementation and maintenance review and acceptance were mapped onto their software maintenance-procedure stage; and initiation, request-for-proposal preparation, contract preparation and update, supplier monitoring, modification implementation, review and acceptance, migration and software retirement were mapped onto their software-upgrade stage (see Figure 2). The GSP maintenance-process was then compared with the IEEE/EIA 12207.0 maintenance-process task-by-task, activity-by-activity, and stage-by-stage, with the objective of determining whether any improvements were required of the IEEE/EIA 12207.0's maintenance-process in the context of ERP.

The GSP Maintenance-data

Identification – Identification of the GSP (ERP) maintenance-data involved a review of all maintenance attributes in GSP's maintenance request forms, change-request database, patch support database, SAP system modification database, user-support database, and maintenance and upgrade documents. Follow-up interviews were also conducted in order to validate the GSP (ERP) maintenance-data.

Maintenance forms were used by GSP to record data items pertaining to maintenance requests. GSP have two types of maintenance forms, namely the System-Investigation-Request Form (or simply the maintenance request form), and the SAP Transport Request Form. The latter is a form signed by high-level management authorizing the delivery or transport of new changes or updates (for maintenance requests) from the development system (DEV) to the quality assurance system (QAS) and then to the production system (PRD). Although most of the data items in the maintenance forms are recorded in the change-request database, some of the maintenance-data are not stored as part of the electronic database. Further, GSP maintained three other databases, for instance (1) a patch support database to record details of newly introduced patches by the vendor, patches already installed to the ERP system, and patches yet to be installed; (2) a SAP system modification database to keep data on previous modifications to the existing ERP system and the associated complexity level in reapplying them; and (3) a user-support database of details regarding service desk requests. These maintenance forms and other databases consisted of the basic data sources used to identify the data items for the maintenance-procedure stage.

Data and information items on maintenance preparation and upgrade projects were identified mainly from maintenance and upgrade documentation. The semi-structured interviews conducted with the General Manager, Systems Development Manager and Systems Operations Manager were also used to confirm the objective of each data item and information item found in GSP's databases and documentation.

Comparison – All the maintenance data items in GSP's maintenance forms (i.e. System-Investigation-Request Form and SAP Transport Request Form) and databases (change-request, user-support, patch support, SAP system modification) were mapped onto GSP's maintenance-data. Each data item or information item in GSP's maintenance-data was compared and cross-referenced with each of the IEEE/EIA 12207.1 data and information items. The mapping of the GSP maintenance items onto the IEEE/EIA 12207.1 standard was based on the item objective as provided in the GSP documentation and confirmed with GSP management. To enhance the accuracy and reliability, mappings were validated through iterative-feedback and further interviews with GSP senior managers.

FINDINGS

ERP Maintenance-process

In this section, we present the GSP (ERP) maintenance-process, as derived from the data sources shown in Figure 2. Discussion of the activities and tasks involved in the GSP maintenance-process has been condensed, and focuses on the three stages, namely maintenance-preparation, maintenance-procedures and software-upgrades (see Appendix 5 for detailed activities and tasks). Appendix 5 was used to make comparisons with IEEE/EIA 12207.0 as detailed in Appendix 3.

Software maintenance-preparation – Senior GSP managers participate actively in the maintenance-preparation stage. They pay considerable attention to the relationship with the vendor and vendor support issues, benefit-realization (from the ERP system), maintenance expenditures, maintenance services provided to GSP clients, and other maintenance management issues including maintenance classification and maintenance strategy. One high-level manager stated "We have a maintenance classification system to categorize all maintenance requests from the clients." This is further evidenced from data sources such as the maintenance request forms and the change-request database.

Software maintenance-procedures – GSP maintenance activities are initiated from essentially two sources, i.e. internal (from system users and IT-staff), and external (from the software vendor). The former source introduces requests such as user support, corrective requests and enhancement requests. The latter introduces patches and new upgrade versions. In resolving a maintenance request, the Systems Development Manager said that they first examine whether the problem has already been resolved by the vendor with a patch, and otherwise they require a custom-made solution. Requests that are resolved using a vendor patch are recorded in the change-request database using the patch number. Also, GSP records the purpose/objective/intention of each patch obtained from the vendor in the patch-support database.

Software-upgrades – The upgrade process is somewhat similar to the patch maintenance procedure with some minor differences. Both apply the standard code provided by the vendor, which results in many similarities (Note that patch maintenance is different from a modification or traditional software maintenance performed by developing custom code.). According to the Systems Development Manager, the main differences between the two procedures are that upgrades require more thorough planning, business justification, money, resources to implement, serious consideration of potential system downtime, effort for impact analysis and re-application of previous modifications or user-enhancements (if the new version has not incorporated the required functionality), and a longer time to complete. This is evident from the contents in the upgrade business case and upgrade planning resources report, as well as from estimates of upgrade effort for each modification inside the SAP system modifications database.

The GSP ERP Maintenance-data

Appendix 6 summarizes the GSP ERP maintenance data. From our comparison between the IEEE/EIA 12207.1 maintenance-data (Appendix 4) and the derived GSP maintenance-data (Appendix 6), we find that the former includes most of the fundamental ERP maintenance-data; however, it lacks the following more specific ERP maintenance data and information items such as functional area, previous modifica-

tions and custom development report, patch progress report, vendor maintenance support report, vendor support request, upgrade option assessment and recommendation report, functional impact analysis report, and non-functional impact analysis report. The objective and rationale behind each GSP data or information item listed in Appendix 6 is provided in Appendix 7.

The functional area data item derives from GSP's maintenance request form. It is required to ease identification of which functional area (such as accounts payable, inventory management, procurement, financial performance, assets management, payroll, employment, leave management; etc.) requires maintenance services; this is particularly important as the ERP spans various functional areas.

ERP packaged software is meant to be configurable and modifiable to serve a large customer base. Therefore, GSP configures and modifies the system to meet its clients' unique business requirements. In order to facilitate tracking and controlling of all modifications to and customizations of the ERP system, and to allow impact analysis to be easily conducted during patch implementation and new version upgrades, GSP maintains the details of all modifications and custom developments in its previous modifications and custom development report (this report comes from the GSP SAP system modification database – see Figure 2).

GSP also possesses other information items specific to packaged software maintenance. The patch progress report (from GSP's patch support database) describes patches that have already been implemented and those yet to be implemented in the existing system. The vendor maintenance support report (derived from initial implementation and upgrade documents) defines the support provided by the vendor. Vendor support requests (derived from GSP's change-request database and interview-transcripts) report the need for maintenance support from the vendor. Also, the upgrade option assessment and recommendation report (from GSP's upgrade documentation – see Figure 2) gives a detailed assessment and reasons for a selected upgrade option.

Finally, the functional impact analysis report describes the functional discrepancies between the previously installed version and the newly upgraded version. Although a small portion of the data items in this report is similar to those covered in the system requirement specification information item of IEEE/EIA 12207.1 in Appendix 4, the functional impact analysis report focuses more on describing how the newly upgraded version affects existing modifications and custom developments, as well as previous functionality in=use. The non-functional impact analysis report, on the other hand, outlines any new and/or additional technical requirements for the newly upgraded version, and the effect of the newly upgraded version on user training and existing supporting documentation. Both of these two impact analysis reports are archived as part of GSP's upgrade documentation (see Figure 2).

UNIQUE ACTIVITIES AND DATA FOR COMMON ERP PACKAGED SOFTWARE MAINTENANCE

In conducting the comparative analysis, we identified several instances within the study ERP maintenance context (or packaged software generally) that diverge from the IEEE/EIA 12207 standard, and that require additional data to be recorded and reported when using the latter. The specific divergent practices associated with each of the three stages of GSP maintenance are more fully discussed below.

ERP Software Maintenance-preparation

The major activities and data/information items in the GSP ERP software maintenance-preparation stage are summarized in Table 1. There is one major activity (P.3) and information item (i.e. Vendor Maintenance Support Report) unique to ERP packaged software.

<<Insert Table 1 Here>>

P.3. Manage relationship with vendor (<u>rationale</u>- reduce/minimize communication breakdown risk with the vendor and control internal maintenance costs) — Unlike traditional in-house software, ERP packaged software is maintained not only by GSP, but also by the vendor. The ERP software vendor plays an integral role in the client 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, thereby influencing the ERP client-organization's maintenance and upgrade decisions, strategies and policies. In order to cost-effectively maintain and make the right upgrade decisions, GSP must continuously communicate with and manage vendor relationships. Also, managing vendor and/or consulting firm relationships is important in terms of addressing the client-organization's knowledge barriers associated with the ERP package configuration, and the inclusion of new business processes (Robey et al., 2002). The actions covered in this major activity include negotiating with the vendor (for specific maintenance support and new functional requirements), managing licensing and contract issues, and determining the types of changes that can be made to the ERP system and the appropriate new version for future upgrades. These activities should be included when applying IEEE/EIA 12207 within an ERP context.

Further, these additional activities produce an added information item (i.e. 'vendor's maintenance support report') to help manage and obtain guaranteed maintenance support from the vendor. The data items in this report are provided in Appendix 6. The data is more or less the same as the contractual issues (as in Appendix 2) suggested by Carney et al. (2000) for packaged software in general, other than regarding how and where to get maintenance support (as the GSP maintenance process covers not only support from the vendor but also internal maintenance activities), and the types of tailoring options (as it is the intention of the vendor to provide flexibility to the client-organization in terms of making changes to the system). That is to say, the general principles for managing vendor relationships are much the same regardless of whether the software package in question is ERP or another package.

Also, note that the 'vendor's maintenance support report' is appropriate to consider and include in the IEEE/EIA 12207.1 maintenance-process when applied in an ERP context; however, the 'acquisition plan' (information item) in the standard's acquisition-process (see Appendix 3) has some similarities with this particular information item as observed in the GSP ERP maintenance-preparation stage.

ERP Software Maintenance-procedures

The major activities and data/information items found in GSP's ERP software maintenance-procedure stage are presented in Table 2, where we find three major activities (i.e. M.1.1, M.3, and M.7.2) and their associated information items that are unique to ERP packaged software maintenance.

<<Insert Table 2 Here>>

M.1.1. Monitor and record patch introduction (rationale- avoid instances of installed system becoming out-of-date; facilitate future system updates; avoid unnecessary internal maintenance costs) — GSP must monitor patch development regularly, record patches as they are implemented and those yet to be implemented, and evaluate the relevancy of each patch to the installed version. The output from these activities culminates in a 'patch progress report', and the details of the associated data items are given in Appendix 6. This additional major activity and information item may be needed to supplement the IEEE/EIA 12207 standard within an ERP maintenance environment.

M.3. Identify and request vendor maintenance support (rationale- reduce future internal maintenance costs by minimizing the number of custom developments) — Consistent with results reported by Nah et al. (2001), fundamental maintenance activities in the ERP environment include client-organizations researching and requesting necessary maintenance support from the vendor before developing any solution internally for a maintenance request. The activities covered in this major activity are explained in Appendix 5

and are essential ERP maintenance activities that need to be considered when using the IEEE/EIA 12207 standard for ERP client-organizations.

In association with this major activity (i.e. reporting bugs or maintenance requests to the vendor), a 'vendor support request' is commonly submitted to the vendor. The data items in this request are illustrated in Appendix 6. This information item is very similar to the 'modification proposal' suggested by Carney et al. (2000) (see Appendix 2), but 'modification proposal' includes more detailed items such as the cost of the modification, projected costs for duplicating in the future, available resources to perform the modification, and fallback plans. Both practices mentioned in the ERP packaged software context diverge from the IEEE/EIA 12207 standard.

M.7.2. Make decisions on whether to reapply previous modifications and/or custom developments (rationale- avoid unnecessary custom development) — Implementing a patch may overwrite custom code. In order to retain the customized functionality, an impact analysis is conducted to identify the previous modifications or custom developments that could be affected or overwritten. Discrepancies between a patch and previous modifications to interfaces, screens, program modules and reporting capabilities must also be examined, and decisions must be made whether to reapply previous modifications or custom developments. The activities covered in this major activity are essential and are reasonable to incorporate into the modification implementation or migration activities of IEEE/EIA 12207.0 (in Appendix 3) for ERP packaged software.

The information item associated with these additional activities is a 'previous modification and custom development' report. This report needs to be updated after each patch implementation project in order to provide up-to-date information regarding the modifications and custom developments done to the installed system. All the data items in this report are given in Appendix 6. This report has the same purpose as the pre- and post-modification reports suggested by Carney et al. (2000) (as shown in Appendix 2);

however, the two reports contain more details regarding vendor's approval for technical or functional behavior and contractual effect of the intended modification, as well as vendor's predictions regarding the potential for the modified version to become its standard commercial offering. Large ERP vendors usually provide an automated patch system for clients to update new and/or critical changes to their systems, such as "Solution Manager" provided by the SAP vendor. When applying IEEE/EIA 12207.1 within the ERP context, this additional report is necessary.

ERP Software-upgrades

Interestingly, although both the request-for-proposal preparation and the contract preparation and update activities in IEEE/EIA 12207.0 (see Appendix 3) appear relevant, the detailed tasks covered in these two activities are not widely covered in the GSP ERP software-upgrade stage. (In contrast, the acquisition-process (both its activities and information items) in IEEE/EIA 12207 would be very useful for organizations planning to upgrade to a new version offered by a different ERP vendor.) This is because GSP acquires the new upgraded version from the same vendor; therefore, no vendor selection task is needed. Neither is a detailed system specification required, as the incumbent vendor is familiar with the GSP installed version system specifications. Finally, no complicated contract preparation is undertaken since GSP have established trust in the vendor based on previous contract experience and a long-term relationship with the vendor.

Table 3 provides a summary of the major activities, and information and/or data items required in the GSP's ERP software-upgrade stage. The table highlights two major activities (i.e. U.2 and U.4) and the associated information items unique to the ERP software-upgrade.

<<Insert Table 3 Here>>

U.2. Research upgrade options available (rationale- ensure the selected upgrade option is the best choice possible) — An ERP upgrade involves planning, selecting, implementing, and installing a new version readily available from the vendor. This is similar to the initial ERP implementation project, see (Robey et al., 2002). ERP upgrade projects require thorough research on the upgrade options available from the vendor(s). Details of the activities carried out during this major activity are provided in Appendix 5. The information item for these additional activities is 'upgrade option assessment and recommendation report', while the data items in this report can be found in Appendix 6. Thus, this major activity and the associated information item are appropriate considerations for the IEEE/EIA 12207 standard for ERP client-organizations.

U.4. Conduct functional and non-functional impact analysis between the new upgraded version and the installed version (rationale- minimize re-development costs, future maintenance costs, system downtime, run-time errors and computational errors) — This major activity begins by examining the functionality in-use, followed by an impact analysis between the new upgrade option and the installed ERP system, including a decision regarding which previous modifications to keep (assuming that some modifications and/or custom development features were unavoidable with the installed version.) These additional tasks need to be considered in the IEEE/EIA 12207.0 standard within the ERP context. Further, the outputs from this major activity include the 'functional impact analysis report', and the 'non-functional impact analysis report'; the contents of both reports are described in detail in Appendix 6. These two additional information items are necessary for ERP maintenance management and when applying the IEEE/EIA 12207.1 standard to accommodate the specific requirements of ERP maintenance.

Gaps found in IEEE/EIA 12207 in the ERP packaged software maintenance context

Table 4 summarizes the gaps identified between the IEEE/EIA 12207 maintenance-process and maintenance-data and our analysis of the ERP packaged software maintenance experience of GSP. According to Grünbacher et al. (2004), in order to enforce quality assurance, the purpose and the desired results of the activities in the process need to be clearly defined. Table 4 contains such a definition of the purpose and the desired results of the additional activities recommended to ensure the IEEE/EIA 12207 standard can accommodate ERP maintenance.

<<Insert Table 4 Here>>

CONCLUSION

In attention to the study research questions 'How is ERP maintenance conducted at GSP?' and 'What are the activities and correspondent required maintenance management data?', the revelatory case study identified, documented and synthesized GSP's explicit ERP maintenance activities and correspondent required maintenance management data. These insights enabled detailed comparison with IEEE/EIA 12207, thereby addressing the study question 'How do these maintenance activities and maintenance management data diverge from the widely available in-house software maintenance-process and maintenance-data described in IEEE/EIA 12207?' Though many best practices reflected in the standard have broad relevance to software generally, several important deviations and gaps were identified, suggesting these are worthy of careful consideration by practitioners and for possible inclusion in ERP standards. We believe that practices identified are important and relevant across ERP maintenance in general.

<u>Implications for practice</u> The additional activities identified in this study and deemed necessary for the ERP maintenance process require a different type of management focus, different emphasis in terms of effort, additional responsibilities, and different factors to be considered in decision making. The ERP client-organization must extend the traditional in-house maintenance boundaries to include licensing and negotiation with vendors, vendor maintenance support management, and market watch. With packaged software, more effort is required in terms of conducting impact analyses, retesting, and system integration, especially when interoperability with other packaged software and/or legacy systems is required (cf. (Oberndorf et al., 2000)). ERP client-organizations generally do not maintain the system alone; the ERP vendor also plays a role in this aspect. Thus, extra effort is called for in terms of managing vendor maintenance support, as well as coordinating maintenance work between internal maintenance requests and patches and/or new versions from the vendor. When making ERP maintenance and upgrade decisions, an ERP client-organization must consider not only its internal organizational needs, but also future vendor maintenance support, upgrade compatibility, and business vision.

The results also suggest that identification and collection of relevant maintenance data and information items can help to retain maintenance knowledge within the organization, improve management productivity, reduce the time and cost involved in making maintenance decisions, and allow the system to be more easily maintained. Moreover, the findings provide initial guidelines for the adaptation of other software process assessment and maturity standards (e.g. ISO/IEC 15504 –and CMMI), in particular to application service providers (ASPs) who provide access to ERP applications and maintenance services to ERP client-organizations.

There are two limitations associated with this study. First, our examination of ERP software-upgrades focuses on the activities involved in replacing an installed ERP version with a new release/version from the same vendor. It does not include an examination of the activities undertaken when replacing an installed ERP version with a custom system or with an ERP system from a different vendor. Second, the

single case study approach brings attention to the 'completeness' of the practices identified at GST; it is impossible to be certain that GST have not overlooked some further practice that might be usefully introduced. Nonetheless, the utility of unique practices reported is we believe apparent, and given the general, non-module specific processes observed, it can be claimed that the divergent practices identified are neither time-dependent nor ERP module-dependent.

Implications for research Continued research in this area is required in order to further develop a common framework of best practices in ERP maintenance. With the aim of producing generic and comprehensive standards for ERP maintenance-processes and maintenance-data, surveys as well as longitudinal and multiple case studies across large, medium and small government agencies and private sector organizations in a range of industries, and employing a variety of ERP modules deriving from various ERP vendors, are warranted. Even given the existence of such generic and comprehensive standards for ERP maintenance, their adoption within ERP-client organizations is still a costly and complex process. Further research might examine the critical success factors for effective adoption of such standards. There is currently a lack of studies into process standard adoption, yet this too is a fundamental issue for the ultimate process and/or product quality assurance.

Since the case study, SAP has introduced the concepts of 'customer competency centres' and offered a comprehensive maintenance support tool 'solution manager'. These are likely to become the mainstays of future SAP-related maintenance activities by client-organizations. Competency centres are centralized SAP knowledge-bases consisting of people within an organization who are knowledgeable functional and technical support staff, with competencies that range from managing the support desk to development and maintenance of releases and patches (Miller, 2004). The purpose of the competency centre is to "provide coordinated support for enterprise-level business applications … that align with an enterprise's organizational and political constraints" (Phelan, 2008). Competency centres with 'solution manager', an online

tool, provide client-organizations with a wide range of functions for solution deployment, ongoing system operation, and effective internal customer support service management consistent with the ITIL standard (SAP AG, 2009b). Solution manager also serves as a central repository (of databases) for testing material, test results, and system users support messages or change requests details (SAP AG, 2009a). Further study is warranted into the extent to which solution manager and the competency centre concepts address some of the issues reported herein. Such study too might suggest further useful changes to the IEEE standards.

Endnotes

1   A recent announcement suggests this figure will increase to 22% over the next four years from 2008, see (Prior and Shepherd, 2008).

2   A 'patch' is a small update to software code that incorporates the latest changes made to the software program. It is distributed by the software vendor to its client-organizations.

3   IEEE/EIA stands for Institute of Electrical and Electronics Engineers and Electronic Industries Association.

4   ISO/IEC stands for International Organization for Standardization and International Electrotechnical Commission.

Acknowledgements

REFERENCES

Ajila, S. (1995). Software Maintenance: An Approach to Impact Analysis of Objects Change, *Software: Practice and Experience* 25(10): 1155-1181.

Carney, D., S. A. Hissam and D. Plakosh (2000). Complex COTS-based Software System: Practical Steps for Their Maintenance, *Journal of Software Maintenance: Research and Practice* 12(6): 357-376.

Croll, P. R. (2002). How to Use Standards as Best Practice Information Aids for CMMI[SM]-Compliant Process Engineering. 14th Annual DoD Software Technology Conference (IEEE-sponsored track), IEEE Computer Society: Los Alamitos CA**:** presentation slide.

Davis, G. B. (1988). Information Systems to Buy, Build, or Customize?, *Accounting Horizons* 2(1): 101-103.

Ferguson, J. and S. Sheard (1998). Leveraging Your CMM Efforts for IEEE/EIA 12207, *IEEE Software* September/October: 23-28.

Glass, R. L. (2003). *Facts and Fallacies of Software Engineering*. Boston, MA: Addison Wesley Professional.

Glass, R. L. and I. Vessey (1999). Enterprise Resource Planning Systems: Can They Handle the Enhancement Changes Most Enterprises Require? *The Software Practitione*r. 9**:** 1-12.

Grünbacher, P., M. Halling, S. Biffl, H. Kitapci and B. W. Boehm (2004). Integrating Collaborative Processes and Quality Assurance Techniques: Experiences from Requirements Negotiation., J*ournal of Management Information Systems* 20(4): 9-29.

Hirt, S. G. and E. B. Swanson (2001). Emergent Maintenance of ERP: New Roles and Relationships, J*ournal of Software Maintenance and Evolution: Research and Practice* 13(6): 373-397.

Hybertson, D. W., A. D. Ta and W. M. Thomas (1997). Maintenance of COTS-intensive Software Systems, J*ournal of Software Maintenance and Evolution: Research and Practice* 9: 203-216.

IEEE/EIA (1997). IEEE/EIA 12207.2-1997 Guide for Information Technology - Software life cycle processes - Implementation considerations. New York NY: Institute of Electrical and Electronics Engineers, Inc.

Jacobson, S., J. Shepherd, M. D'Aquila and K. Carter (2007). The ERP Market Report, 2006-2011. Boston, AMR Research, Inc.: **3**6.

Kumar, K. and J. van Hillegersberg (2000). ERP Experience and Evolutions, C*ommunications of the ACM* 43(4): 22-26.

Maxwell, J. A. (1996). Qualitative Research Design: An Interactive Approach. Thousand Oaks, California: Sage Publications.

McDermid, J. A. (1997). COTS: The Expensive Solution?, in IEE Colloquium on COTS and Safety Critical Systems: IEE: Savoy Place, UK. 1/1-1/4.

Miller, B. (2004). SAP Customer Competence Centre [WWW document], Forrester. h[ttp://www.sap.com/turkey/services/pdf/Misc_Forrester_SAP_Competence_Center.pdf](http://www.sap.com/turkey/services/pdf/Misc_Forrester_SAP_Competence_Center.pdf) (accessed 5 March 2009).

Nah, F., S. Faja and T. Cata (2001). Characteristics of ERP Software Maintenance: A Multiple Case Study, J*ournal of Software Maintenance and Evolution: Research and Practice* 13(6): 399-414.

Ng, C. S. P., G. G. Gable and T. Chan (2003a). An ERP Maintenance Model, in 36th Hawaii International Conference on Systems Sciences; Hawaii, USA: IEEE Computer Society: Los Alamitos, CA. on CD-ROM.

Ng, C. S. P., G. G. Gable and T. Chan (2003b). A Revelatory Case Study into the Adequacy of Standard Maintenance Models in an ERP Context, in Pacific Asia Conference on Information Systems; Adelaide, Australia. 1039-1054.

Ng, C. S. P., P.-Y. Hsu and W.-H. Tsai (2006). Salient Factors for Maintenance Standard Adoption in Enterprise Resource Planning Context: An Exploratory Study, in 39th Hawaii International Conference on Systems Sciences; Hawaii, USA: IEEE Computer Society: Los Alamitos, CA. on CD-ROM.

Nidumolu, S. R. and M. R. Subramani (2003). The Matrix of Control: Combining Process and Structure Approaches to Managing Software Development., J*ournal of Management Information Systems* 20(3): 159-196.

Oberndorf, T., L. Brownsword and C. A. Sledge (2000). An Activity Framework for COTS-Based Systems. Pittsburgh, PA, Software Engineering Institute, Carnegie Mellon University: **9**4.

O'Brien, J. A. and G. M. Marakas (2008). Management Information Systems. Boston: McGraw-Hill.

Phelan, P. (2008). ERP Competence Center Model for Providing Business Application Support: A Retrospective on 2007's Research [WWW document], Gartner Group. http://www.gartner.com (accessed 5th March 2009).

Prior, D. and J. Shepherd (2008). SAP Enterprise Support Contracts: Counting the Real Costs [WWW document], AMR Research. http://www.amrresearch.com/Content/View.asp?pmillid=21709 (accessed 18 Nov 2008).

Ragowsky, A. and D. Gefen (2008). What Makes the Competitive Contribution of ERP Strategic, T*he DATA BASE for Advances in Information Systems* 39(2): 33-49.

Reifer, D. J., V. R. Basili, B. W. Boehm and B. Clark (2003). Eight Lessons Learned during COTS-Based Systems Maintenance, I*EEE Software* 20(5): 94-96.

Reilly, K. (2005). Over 20% of Small Manufacturing and Services Companies are Evaluating ERP for the First Time in the Nest 12 Months. [WWW document], AMR Research, Inc. www.amrresearch.com/Content/View.asp?pmillid=18291&docid=12568 (accessed 13 March 2009).

Robey, D., J. W. Ross and M.-C. Boudreau (2002). Learning to Implement Enterprise Systems: An Exploratory Study of the Dialectics of Change, J*ournal of Management Information Systems* 19(1): 17-46.

SAP AG (2009a). SAP - Components & Tools of SAP NetWeaver: SAP Solution Manager [WWW document], SAP AG. http://www.sap.com/platform/netweaver/components/solutionmanager/index.epx (accessed 4th March 2009).

SAP AG (2009b). SAP NetWeaver - Solution Manager [WWW document], SAP AG. https://websmp207.sap-ag.de/~sapidp/011000358700003127162006 (accessed 4th March 2009).

Shi, S. and J. Qian (2001). Upgrading to R/3 Release 4.5 and Beyond: An ABAP Developer's Guide, S*AP Professional Journal:* 3-26.

Tran, V. and D.-B. Liu (1997). A Risk-Mitigating Model for the Development of Reliable and Maintainable Large-Scale Commercial-Off-The-Shelf Integrated Software Systems, in Annual Reliability and Maintainability Symposium: IEEE Computer Society: Los Alamitos, CA. 361-367.

Voas, J. (1998). Are COTS Products and Component Packaging Killing Software Malleability?, in International Conference on Software Maintenance; Maryland: IEEE Computer Society: Los Alamitos CA. 156-157.

Wailgum, T. (2008). U.K. SAP Users Angry Over SAP Support Price Increases [WWW document], CXO Media Inc. http://www.cio.com/article/print/439491 (accessed 8 Jan 2009).

Whiting, R. (2006). Lower-Cost Options Free IT From Software Maintenance Fees [WWW document], InformationWeek. http://www.informationweek.com/shared/printableArticle.jhtml?articleID=192300361 (accessed 15 Jan 2008).

Yin, R. K. (1994). *Case Study Research: Design and Methods*. Thousand Oaks, California: Sage Publications.

List of tables

Table 1 GSP's ERP Software Maintenance-Preparation Stage

Major activity identifier	Major activity name		Data or information item	
	Name	Correspond to Appendix 3	Name	Correspond to Appendix 4
P.1.	Define maintenance, maintenance organization, and associated policies	PI -(A)	Maintenance objective, software system characteristics, maintainer's roles and responsibilities, training course, service desk services	Not given but should be common across all software
P.2.	Estimate maintenance resources	PI -(B)	Resources required	Not given but should be common across all software
P.3.	*Manage relationship with vendor*	*Not available*	*Vendor's maintenance support report*	*Not available*
P.4.	Define all maintenance management issues	PI -(C)	Maintenance classification, maintenance strategies	Not given but should be common across all software

Table 2 GSP's ERP Software Maintenance-Procedure Stage

Major activity identifier	Major activity name		Data or information item	
	Name	Correspond to Appendix 3	Name	Correspond to Appendix 4
M.1.1. *OR*	*Monitor and record patch introduction*	*Not available*	*Patch progress report*	*Not available*
M.1.2.	Receive maintenance request	PA -(A)	Service desk record	Not given but should be common across all software
M.2.	Classify and approve request	PA-(B)	Maintenance request form	Change-request or problem & resolution report *
M.3.	*Identify and request vendor maintenance support*	*Not available*	*Vendor support request*	*Not available*
M.4.	Analyze the problem	PA -(C)	(Updated) Maintenance request form	Change-request or problem & resolution report *
M.5.	Design solution	PA -(D)	(Updated) Maintenance request form	Change-request or problem & resolution report *
M.6.1. OR	Implement solution in the Development System	MO -(A)	Previous modification & custom dev. report	Change-request or problem & resolution report *
M.6.2.	Apply vendor code or patch in the DEV	MO -(A)	(Updated) Patch progress report	Change-request or problem & resolution report *
M.7.1. OR	Conduct impact analysis of custom development or patches	MO -(B)	Impact analysis results	Change-request or problem & resolution report *
M.7.2.	*Make decision on whether to reapply previous modification or custom development*	*Not available*	*(Updated) Previous modification & custom dev. report*	*Not available*
M.8.	Deliver/transport changes to QAS for testing and verification	MR -(A)	Transport request form, transport approval form	Change-request or problem & resolution report *
M.9.	Deliver/transport new system to the PRD	MR -(B)	(Updated) Transport request form, transport	Change-request or problem & resolution report *

			approval form	

* Change request or problem & resolution report – depends on the type of maintenance request.

Table 3 GSP's ERP Software-Upgrade Stage

Major activity identifier	Major activity name		Data or information item	
	Name	Correspond to Appendix 3	Name	Correspond to Appendix 4
U.1.	Design a project methodology for the upgrade	MI – (A)	Upgrade work breakdown structure	Mig-plan
U.2.	*Research upgrade options available*	*Not available*	*Upgrade option assessment and recommendation report*	*Not available*
U.3.	Develop a business case	IN	Upgrade business case	Co-desc, sys- spec
U.4.	*Conduct functional and non-functional impact analysis between the new upgraded version and the installed version*	*Not available*	*Functional and non-functional impact analysis report*	*Not available*
U.5.	Update contract with the vendor	RP, CP, SM	(Updated) Vendor's maintenance support report	Acq-plan
U.6.	Install and construct the new version in the DEV	AC	(Updated) Functional and non-functional impact analysis report	TV- report
U.7.	Conduct testing and trial upgrades between the DEV system and QAS system	AC	Testing plan, trial upgrade results	TV-report
U.8.	Conversion (or go live) in PRD	AC	Go-live results	TV- report

Table 4 Gaps in the IEEE/EIA 12207 in the context of ERP maintenance

		Additional Activity(ies) and Data Item(s) Recommended	
Stage	Major activity	Activity (the *purpose/rationale*)	Data Item (*desired result*)
Maintenance-preparation	Manage relationship with vendor and vendor's maintenance support	o Negotiate software contract, functional requirement and support window o Keep vendor informed of user requirements and expectations o Determine vendor maintenance support (Reduce/ minimize communication breakdown risk with vendor and control internal maintenance costs)	o Vendor maintenance support report
Maintenance-procedure	Monitor and record vendor maintenance support	o Monitor patches introduced by the vendor o Download patches o Determine the goal, criticality, and importance of each patch o Record all patches already and yet to be implemented to the existing system o Estimate patch maintenance costs (Avoid having installed system become out-of-date and avoid unnecessary internal maintenance costs)	o Patch progress report
	Identify the need for and request vendor maintenance support	o Determine whether vendor provides solution for maintenance request o Report the change request back to vendor (if necessary) (Minimize unnecessary internal maintenance costs)	o Vendor support request
	Make decision on whether to reapply previous modifications and custom developments	o Identify modifications that have been overwritten o Determine customer business processes that are affected but no longer needed after patch implementation o Analyze impacts and discrepancies of a patch on current modifications o Make decision whether to reapply previous modifications or custom development o Reapply previous modifications or custom development overwritten during patch implementation (Reduce future maintenance costs and avoid unnecessary cus-	o Previous modification and custom development report

|				
		tom development)		
Upgrade	Research upgrade options available	o Conduct upgrade option search and gather related information for each option o Decide on type of upgrade o Make full assessment of new features or functionality for each option o Draft plan of how a new functionality benefits the organization, and plan for benefit realizations for the new business improvements o Make recommendation for the upgrade release or version (Ensure selected upgrade option is the best choice possible)	o	Upgrade option assessment and recommendation report
	Conduct functional and non-functional impact analysis between the new upgraded version and the installed version	o Examine the functionality in-use for installed version o Conduct functional impact analysis between new upgraded version and existing version o Identify and determine technical requirements and impacts of new upgraded version (Minimize re-development costs, future maintenance costs, system downtime, run-time errors and computational errors)	o o	Functional impact analysis report Non-functional impact analysis report

List of figures
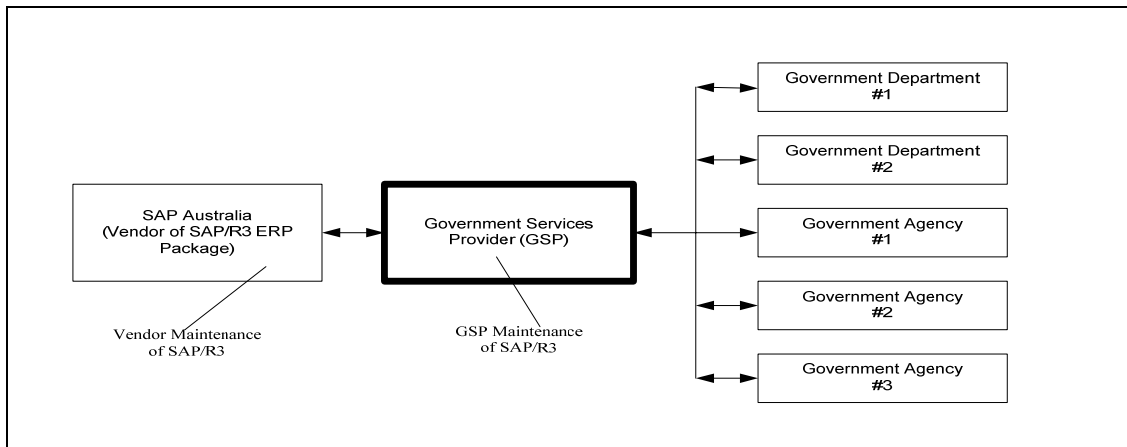
Figure 1 The Case Organization and Relationship with other Organizations
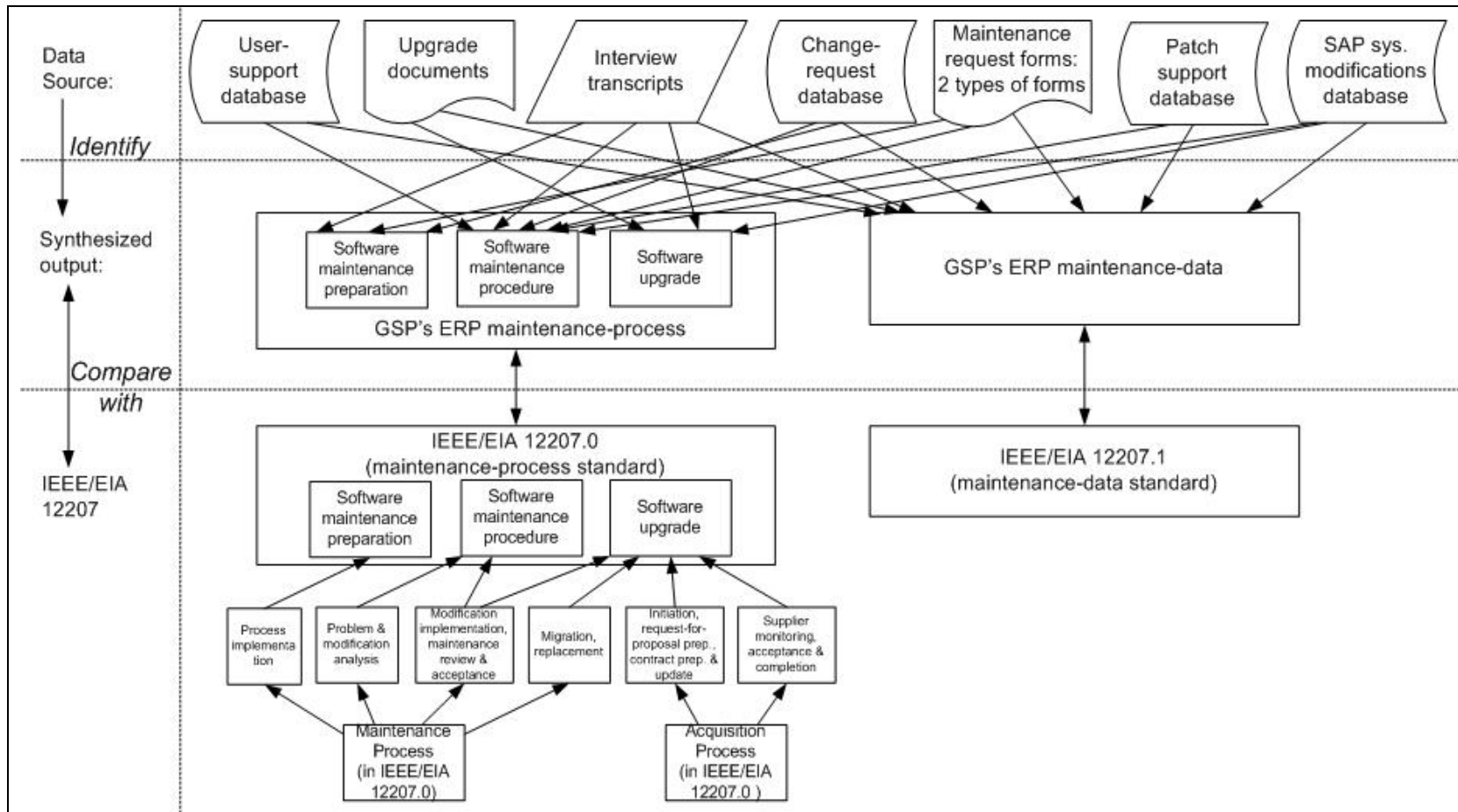
Figure 2 Data Analysis Process For GSP's ERP Maintenance-Process And Maintenance-Data

Appendix 1 Comparison Between Packaged Software And In-House Software Contexts

Context	Factor	Packaged software	In-house software
Software	Keeping pace with advances in technology	Vendor strives to be current	Typically lags, at least following initial implementation
	Requirements volatility (rate of software change)	Frequency of product updates (or patches) is greater	Relatively slower (fewer users to satisfy, costs cannot be justified due to lesser economies of scale)
	Main (client) costs	License fees, maintenance charges (patches, upgrades, system integration), consultant fees, internal maintenance costs	Development and in-house maintenance (corrective, adaptive and enhancement)
	Risks	Vendor viability, technical and cost risks	Development, control, performance and documentation risks
User	User community	A large community of users, e.g., user-groups or all organizations using the product	Within one organization
	User control over changes and improvements	Less control	Relatively more control
Vendor	Vendor relationship	Very important to manage the relationship	No relationship with package vendor
Acquisition and development	Phases involved	Requirements→identification of package candidates→configuration and modification→integration→test→installation	Requirements→design →code→test→installation
	Market research	Extensive	Mostly not required
	Time-to-market	Shorter time-to-market with lower resource costs	Much longer time-to-market with higher resource costs
Maintenance	Maintenance function	Shared maintenance – partly outsourced from the vendor	Mostly in-house

Appendix 2 Fundamental Data For Modifying Packaged Software Code

Information item	Data item [a]
Contractual issue	(1) Term of the existing licenses; (2) expected frequency of releases; (3) vendor policies concerning emergency updates (e.g., patches required to repair vulnerabilities); (4) vendor prediction of the expected upward compatibility of future releases of the software (for maintenance of the modified form of the software); (5) validity of licenses; (6) liability in case of component failure; and (7) availability of source code
Modification proposal (To be sent to the vendor for feedbacks)	(1) Functional effect of the modification; (2) technical factors (performance, safety, reliability) affected by the requested modification; (3) business effect of the modification (license, ongoing cost, and liability issues); (4) upward compatibility with future releases of the component (projected costs for duplicating the modification in future releases, and the potential for the modification to become part of the standard product); (5) cost; (6) available schedule; (7) available resources to perform the modification; (8) risks; and (9) fallback plan
Pre-modification report	(1) Primary cause for the modification; (2) reason that no other solution is possible; (3) description of the planned modification; (4) approval from the Control Board; (5) vendor approval for any resulting technical or functional behavior of the intended modification; (6) vendor approval for the contractual effect of the modification (changes in license fees and changes in any guaranteed short-term maintenance and support); and (7) vendor predictions on the probability that the modified version will become part of its standard commercial offering, intended release date and licensing information, as well as ongoing costs of additional modifications
Post-modification report	(1) Technical description of the modification; (2) all applicable test data, including verification that the modified configuration-item(s) passed all tests satisfactorily; (3) working versions of any tools used to make the modification; and (4) identity of the person(s)/organization(s) that actually performed the modification

[a] Source: (Carney et al., 2000)

	Appendix 3 Activities In IEEE/EIA 12207.0 – Maintenance-Process And Acquisition-Process*
	Major activities
Maintenance-process**	Process implementation (PI) – (A) Define a maintenance organization; (B) describe the arrangement for resource allocation and performance tracking; develop formal statement of anticipated future maintenance requirements; identify maintenance effort determinants; (C) develop plan and workflow for conducting maintenance activities from arrival to delivery; classify maintenance requests; prioritize and assign requests; set up a charge-back system; establish controls, rules and methods to record and track maintenance requests; and establish the software configuration management (SCM) process
	Problem and modification analysis (PA) – (A) Assign ID to the request; (B) classify maintenance requests; determine the criticality and priority of requests; replicate or verify the problem and analyze the impact of the problem report or the modification request on the organization, existing system and interfacing system; determine whether to accept the request; (C) identify alternative solutions; conduct analysis of conversion requirements, safety and security implications; conduct preliminary estimation for the modification's size, scope, short-term and long-term costs, value of undertaking the maintenance and time to modify the request; document the request, analysis results, and implementation options; obtain approval for the selected modification option; (D) carry out a detailed analysis – define firm requirement(s) for the modification, identify the element to be modified, identify safety and security issues, devise a test strategy, and develop an implementation plan and schedule for implementation
	Modification implementation (MI) – (A) Identify the affected module; modify software module documentation; create test cases for the new design, safety and security issues, and regression test; describe documentation to be updated; update modification list; define and document the criteria for testing and evaluating the modified and unmodified parts of the system; write coding; (B) conduct unit testing; conduct regression tests; carry out risk analysis; and implement a test-readiness review to assess preparedness for a system test
	Maintenance review and acceptance (MR) – (A) Conduct review(s) with the modification authorizer's organization, system functional test, interface test, and regression test; conduct acceptance tests at the functional level, interoperability test, functional configuration audit (FCA) and physical configuration audit (PCA); (B) notify the user community of the product delivery schedule; make an archival version of the system for backup; and conduct installation and training at customer site
	Migration (MI) – (A) Develop, and document a migration plan; (B) execute a migration plan; notify the relevant parties of the migration schedule; notify users of the migration plan; (if necessary) conduct parallel operations in the old and new environments, and provide the necessary training; conduct a post-operation review, and send review results to the appropriate authorities for information, guidance and action; and archive the data, documentation, logs and code used by or associated with the old environment
Acquisition-process	Initiation (IN) – Describe a concept or a need to define and analyze the system requirements; perform system requirements analysis, perform the definition and analysis of software requirements; consider options for acquisition; consider the risks, costs, and benefits for each option; ensure the requirements for the software product and the documentation are available; ensure proprietary, usage, ownership, warranty and licensing rights are satisfied; plan for future support for the software product; prepare, document and execute an acquisition plan; define and document the acceptance strategy and conditions (criteria)
	Request-for-proposal preparation (RP) – Document the acquisition requirements (i.e. system requirements, scope statement, instructions for bidders, list of software products, terms and conditions, control of subcontracts, technical constraints); define the contract milestones at which the supplier's progress will be reviewed and audited as part of monitoring the acquisition
	Contract preparation and update (CP) – Establish a procedure for supplier selection; prepare and negotiate a contract that addresses the acquisition requirement; control changes (project plans, costs, benefits, quality, and schedule) to the contract through negotiation
	Supplier monitoring (SM) – Cooperate with the supplier to provide all necessary information in a timely manner and resolve all pending items
	Acceptance and completion (AC) – Prepare for acceptance based on the defined acceptance strategy and criteria; prepare for test cases, test data, test procedures, and test environment; conduct acceptance review and acceptance testing of the deliverable; take the responsibility for the configuration management of the delivered software product

*This Appendix is an expanded version of Appendix 1 in (Ng et al., 2003b)

** The replacement activity in the maintenance process is excluded here as it is not applicable to this study

Appendix 4 IEEE/EIA 12207.1 – Data and Information Items

Information item		All data items in this information item				
Maintenance-process	Change request or modification request	Purpose	date of initiation	scope	Subject	originator of request
		change to correct software error	justification for change	nature of change	originator of change request	priority of change
		assumptions/constraints that affect change	analysis and evaluation of change	impact to schedule, cost, products, and test	approval or disapproval of change request	detailed description of requested item, or service
		(configuration control of change request)	identification and recording of change request	new or modified capability/function or other changes needed	verification of the implementation and release of modified system	information regarding receipt, recording, and tracking changes from the users
	Problem report and resolution report	date of issue and status	purpose	scope	status of problem	change history
		description of the problem to enable problem resolution	description of the corrective action taken	issuing organization	date problem discovered	originator, originator's assessment of urgency
		identification of the software item or software configuration item and/or the software lifecycle process in which the problem was observed				
	Migration plan (Mig-pan)	requirement analysis and definition of migration	development of migration tools	conversion of software product and data	migration execution and verification	support for the old environment in the future
Acquisition-process	Acquisition plan (Acq-plan)	planned employment of the system	Requirements for the system	type of contract to be employed	support concept to be used	responsibilities of the organizations involved
		risks considered as well as methods to manage the risks				
	Concept of operations description (Co-desc)	description of current situation or system	summary of impacts	concepts for the proposed system	operational scenarios	justification for and nature of changes
		analysis of the proposed system	priorities, assumptions, constraints, advantages, limitations, alternatives, and trade-offs considered			
	System requirements specification (Sys-spec)	system identification and overview	required states and modes	rationale	packaging requirements	computer resource requirements
		business, organizational, and user requirements	operations and maintenance requirements	system external interface requirements	system environmental requirements	design constraints and qualification requirements
		computer software requirements	physical requirements	system quality characteristics	internal data requirements	installation-dependent data requirements
		personnel, training, and logistics requirements	computer communications requirements	precedence and criticality of requirements	safety, security, and privacy protection requirements	
		requirements for the functions and performance of the system			human-factors engineering requirements	
	Test or validation results report (TV-report)	System identification and overview		test log	rationale for decisions	
		overview of test results, including overall assessment of the software tested, and impact of test environment		detailed test results, including test identifier, test summary, problems encountered, and deviations from test cases/procedures		

Appendix 5 Summary Of GSP's ERP Maintenance-Process*			
Stage	**Major activity – task involved**	**Rationale**	**Data source**
Maintenance-preparation	P.1. Define maintenance, maintenance organization, and associated policies – Define the core objectives of maintenance and their alignments to the initial objectives of the ERP implementation; identify the scope, benefits, costs, and risks of the system; define maintenance unit(s) or role(s), and maintenance team(s) responsibilities and job specifications; determine the types of training; match training classes to different user groups; define the help desk policies	Avoid an maintenance project that may deviate from the organization's business goals; ensure the team members recognize their roles and responsibilities; ensure the users know how to use the system and where to look for help	Interviews Maintenance request forms Change-request database
	P.2. Estimate maintenance resources requirement – Estimate all resources required and/or outsourcing needs	Facilitate organization's resource allocation and budgeting;	
	P.3. Manage relationship with vendor – Negotiate software contract, functional requirement and support window; keep the vendor informed of user requirements and expectations; provide relevant feedback (on the ERP system) to the vendor; invite the vendor to participate in important meeting or decision related the ERP system such as making modifications to the ERP system; outline maintenance support from the vendor, such as the support window[1] for the software, conditions to remain eligible for maintenance support, the types of maintenance support available from the vendor, and how and where to get them	Reduce/ minimize communication breakdown risk with the vendor and controlling internal maintenance costs	
	P.4. Define all maintenance management issues – Define all the environments needed for maintenance; develop a mechanism to identify and classify maintenance requests; establish maintenance strategies, including how each type of maintenance request 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; define the maintenance procedure	Improve the ease of future maintenance management	
Maintenance-procedure	M.1.1. Monitor and record patch introduction – Monitor and track the patches introduced by the vendor; download patches; record the patches; study the patches relevancy to the installed system	Avoid the installed system from being out-of-date and avoid unnecessary internal maintenance costs	Interviews User-support database Change-request database Patch-support database SAP system modifications database Maintenance request forms
	M.1.2. Receive maintenance request – Determine if a request is a user-support or change-request; study the root of the problem (e.g. inadequate training, needs for consultation on software functionality, incorrect user-access rights, and security issues); determine whether to resolve the problem/request directly and/or direct the request to the right person for solution	Provide a venue for users to report maintenance requests	
	M.2. Classify and approve the request – Create and issue a user-support or change-request form; classify and prioritize request (based on existing workload or request type); obtain approval for the request	Facilitate maintenance request processing	
	M.3. Identify and request for vendor's maintenance supports – Identify whether the existing patches are useful for the request; if support is not available from the vendor, report the request to the vendor and propose solution/recommendation	Minimize unnecessary internal maintenance costs	
	M.4. Analyze the problem – Conduct cost-estimation for the maintenance; issue a quotation to the user; obtain approval from the user and maintenance manager	Produce a suitable solution to a problem	
	M.5. Design solution – Design solution for the problem	Provide customized solution to meet specific need	

---

[1] 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.

	M.6.1. Implement the solution in the Development System (DEV) – Implement changes in the DEV system; update the relevant documentation	Avoid disruptions to the production system	
	M.6.2. Apply vendor's code or patch in the Development System (DEV) – 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 in DEV system; update the relevant documentation	Avoid disruptions to the production system; minimize unnecessary internal maintenance costs	
	M.7.1. Conduct impact analysis of the custom development or patches – Conduct detailed impact analysis of the custom development on the standard code, or impact analysis of the patch on each previous custom developments	Minimize possible system downtime and run-time or computational errors	
	M.7.2. Make decision on whether to reapply previous modification or custom development – Decide whether to reapply previous developments or modifications; make modification adjustments – reapply the previous modifications (if necessary)	Reduce future maintenance costs; avoid unnecessary custom development	
	M.8. Deliver/transport changes to the Quality Assurance System (QAS) for testing and verification – Conduct system testing, and user acceptance test; perform complete system re-testing of performance and integration	Ensure the system can perform well	
	M.9. Deliver/transport the new system to the Production System (PRD)	Utilize the updated system	
Software-upgrade	U.1. Design a project methodology for the upgrade – Develop a project methodology to initiate, plan, execute, monitor, control and close the upgrade project	Provide well-planned guidelines for upgrade execution	Interviews SAP system modifications database Upgrade business case Upgrade planning resources report
	U.2. Research upgrade options available – Identify upgrade options available; determine their availability dates, pros and cons, stability, and the support window (i.e. vendor maintenance support completion date) of each option; decide on the type of upgrade (technical or functional); make full assessment of the new features or functionality in each option (for each module of interest); decide how a new functionality benefits the organization; draft a plan for benefit realization of the new business improvements; make recommendation for the upgraded version	Ensure the selected upgrade option is the best choice possible	
	U.3. Develop a business case – Define the scope of the upgrade; identify the factors influencing the upgrade 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; justify the upgrade decision	Ensure the upgrade project meets the organization strategic goals	
	U.4. Conduct functional and non-functional impact analysis between the new upgrade version and the installed version – Examine the functionality in-use of the installed version; conduct functional impact analysis between the new upgraded version and the existing version; examine the technical impacts of the upgrade on user training, interfaces and desktop, reporting capability, supporting documentation, change management, testing and security; study the impacts of the upgrade on hardware sizing, database and application server capacity, and network loading requirements	Minimize re-development costs, future maintenance costs, system downtime, run-time errors and computational errors	
	U.5. Update contract with the vendor – Prepare, negotiate and update existing contract	Ensure the quality of future maintenance services from the vendor	
	U.6. Install and construct the new version in the DEV – Install the new version onto the DEV; construct the new system with custom development and/or reapply previous modifications and re-develop previous reporting capability, etc. (if necessary); conduct testing	Avoid possible disruptions to the production system	
	U.7. Conduct testing and trial upgrades between the DEV system and QAS system – Conduct thorough testing of the upgrade system and user acceptance testing; carry out trial upgrades between the DEV system and QAS system	Ensure the new system functions and performs as expected	
	U.8. Conversion (or go live) in PRD – Deliver the well-tested system into the PRD system	Realize more business benefits	

*This Appendix is an expanded version of Appendix 3 in (Ng et al., 2003b)

Appendix 6	GSP's ERP Maintenance-Data	
	Information item	**Data item**
Software maintenance-preparation	System information	Maintenance objectives, software system characteristics, resources required to maintain the system, maintainer's roles and responsibilities, maintenance classification, maintenance strategies, training course, service desk services
	Vendor's maintenance support report	Contract warranty, license, support window, eligibility for maintenance support, doable modifications and custom developments, type of tailoring options and associated risks, types of support, support frequency and magnitude, how and where to get support
Software maintenance-procedure	Service desk record	Record ID, requestor, functional area, priority, status, request date, committed date, completed date, requestor department, problem description, and comments
	Maintenance request form	Request ID, product ID, raised by, cross-functional area, priority, problem description, date raised, time of occurrence, test phase, work type, problem type, action to be taken, date of action, issues to consider problem status, description of changes, related changes, approved by, quotation, resolution, resolution impact, patch number, training updated, online documentation updated, completed date, approval to migrate, transported on, accepted by
	Work sheet	Estimated time (in hours) to complete the maintenance request, projected resolution availability, maintenance effort (in hours) spent in servicing the maintenance request, analyst and programmer involved in analyzing the solution(s) for the maintenance request
	Transport request form	Requester, request ID, description of changes, technical objects included in the transport, source system, target system, type of transport, transport results, date and time completed, and completed by
	Transport approval form	Request ID, transport description, transport type, target system, target clients, authorizer, date and time required, date transport completed, signature
	Vendor support request	Background of the requesting organization, objectives of the request, description/background of the problem or request, module involved, suggested resolution(s), impacts on future maintenance supports, projected completion date, requester's perception of the affects of the change
	Patch progress report	Patch number, patch objective, patch description, patch date, patch status, patch benefits, implementation decision, impacts of the patch, number of notes (i.e. bug fixes or minor enhancements), implementation effort, impact analysis results
	Previous modification and custom development report	Modification description, business reason, application area, validation, program name, object type (e.g. interface, report), modification still-required, criticality, the user-department(s) requiring the modification, remarks and effort required to reapply (including impact analysis, implementing and testing efforts)
Software-upgrade	Upgrade methodology	Upgrade work breakdown structure (or simply upgrade activities)
	Upgrade option assessment and recommendation report	Upgrade option availability date, pros and cons, stability, support window, module(s) of interest, functionalities provided, new functionalities and their benefits, technical requirements, functionalities required, final upgrade decision
	Upgrade business case	Upgrade objectives, business drivers, nature of the upgrade, upgrade justification, factors influencing upgrade, upgrade date, upgrade timeline, upgrade costs and benefits, opportunity costs of not upgrading, upgrade project risks, upgrade option assessment and recommendation, budget allocation, personnel requirements
	Functional impact analysis report	Module(s) affected, previous functionalities, previous functionalities still-required, previous functionalities affected, new required functionalities, previous modifications and custom developments, previous modifications affected, new required modifications, reapplication decision for previous modifications, estimated effort, complexity level
	Non-functional impact analysis report	User training, user interfaces and desktop, supporting documentation, change management, testing needs, security, hardware sizing, database, application server capacity, network loading requirements
	(Updated) vendor maintenance support report	Contract warranty, license, support window, eligibility for maintenance supports, doable modification and custom development, type of tailoring options and associated risks, type of supports, supports frequency and magnitude, how and where to get support
	Testing plan	Module(s) involved, required functionalities, previous modifications, new modifications, new functionalities, unit test, integration test, system tests
	Trial upgrade results	Trial upgrade success or failure, reasons, acceptance and sign-off, unresolved issues
	Go-live results	Go-live success or failure, reasons, acceptance and sign-off, unresolved issues

Appendix 7 GSP's ERP maintenance-data description*	
	Data/ Information item – Objective
Software maintenance-preparation	{System information} – Define the characteristics of the software system
	Maintenance objectives – Define the objectives of performing ERP software maintenance; ensure that each maintenance project is conformed to business objectives and initial implementation objectives
	Resources required – Identify the resources required to maintain the system; indicate if outsourcing is required
	Maintenance team plan – Define the roles and responsibilities of the maintenance team members
	Maintenance classification – Use to categorize maintenance requests
	Maintenance strategies – Describe how each request type is serviced; what, how and where the users are serviced; type of request requiring charge-back to the users and the fee schedules
	Training policy – Describe the type of training, courses provided, training stages involved, course requirements and certifications
	Service desk policy – Describe the procedures used to access and provide helpdesk service to users
	{Vendor's maintenance support report}*** – Serve as a maintenance-support-reference and communication tool with the vendor
Software maintenance-procedure	{Service desk record} – Record each maintenance request reported at the service desk
	Request ID – Uniquely identify each change request or maintenance request
	Product ID – Identify the software product to which problems refer; also used by *GSP* for billing purposes (to client agencies).
	Raised by – Determine the originating person; helpful in identifying the environment-specific and source-specific problem
	Functional area – Represents the business application area(s) associated with a maintenance request
	Priority - Measure of the importance of a request to the system users
	Problem description – Refer to the activity, process, or operation taking place when the problem was encountered
	Date raised – Describe the date at which a problem occurred
	Time of occurrence – Describe the relative time at which a problem occurred
	Test phase – Determine the specific functional category that requires specific testing; identify if a particular functional category tends to generate abnormally large number of maintenance requests (i.e. correction or enhancement)
	Work type – Identify the categories of maintenance requests (e.g. corrective, enhancement)
	Problem type – Classify the problem into several categories to facilitate problem resolution
	Action to be taken – Show whether a request is approved by the systems manager; and allow identification of the number of requests being rejected or deferred
	Date of action – Record the date of a request when its state (e.g. opened, closed, assigned for evaluation) changes; it is important to track the time spent on analyzing and resolving the request, and to identify delays incurred
	Issues of consideration – Identify future issues related to a change request that is deferred and/or related to future maintenance supports
	Problem status – Indicate the job-status of a maintenance request (such as in-progress, user-testing, on-hold, awaiting client quote, closed)
	Description of changes – Describe the objective(s) of request and the software object(s) changed to resolve the discovered problem; identify software units prone to change due to correction and/or enhancement; discover software volatility
	Related changes – Indicate the list of software object(s) required to be changed in resolving the problem in question (including training needs and documentation); useful to estimate time required to resolve a request
	Approved by – Indicate that the maintenance solution has been approved by the fixer
	Quotation – Indicate the estimated cost of implementing the maintenance request (in the GSP case, this attribute is used for user-initiated enhancement requests only)
	Resolution – Describe how the maintenance problem is resolved, and contact person for resolving the problem
	Resolution impact – Identify whether online help needs to be changed, or user training is needed as a result of the maintenance solution
	Patch number – Identify whether a maintenance request is satisfied using the readily available patches distributed by the vendor
	{Work sheet} – Record the time spent in servicing and the personnel who service the maintenance request
	Training updated – Indicate that training material has been prepared and updated in relation to a change request
	Online documentation updated – Indicate that the online documentation has been updated in relation to a change request
	Completed – Indicate the date when the maintenance solution is released; useful to identify the efficiency of the maintenance project-management in meeting the projected deadlines
	Approval to migrate – Identify whether a maintenance solution is approved to migrate to the Quality Assurance System and/or the Production System
	Transported on – Show the date the maintenance solution was applied to the problem-originating site
	Accepted by – Indicate that the maintenance solution has been accepted by the system users
	{Transport request form} – Request for the changes to be transported to the Quality Assurance System and/or the Production System
	{Transport approval form} – Authorize the changes to be transported to the Quality Assurance System and/or the Production System
	{Vendor support request}*** – Indicate a possibility that the requirement could be incorporated into a new version in the future
	{Patch progress report}*** – Track how up-to-date the installed version is compared to the vendor's standard code

	Impact analysis results – Show the objects to modify or update in order to accomplish a change
	{Previous modification and custom development report}*** – Facilitate the process of tracking the modifications done and allow more effective impact analysis to be conducted
Software-upgrade	{Upgrade methodology} – Contain the plan and activities involved in carrying the upgraded project
	{Upgrade option assessment and recommendation report}*** – Select the most suitable upgrade option
	{Upgrade business case} – Make a feasibility assessment for the upgraded project, and justify the upgraded project from business perspective
	{Functional impact analysis report}*** – Identify functionalities and business processes required in the new version
	{Non-functional impact analysis report}*** – Identify technical and other requirements in the new version
	{(Updated) vendor's maintenance support report}*** – Provide up-to-date information regarding the vendor's maintenance support for the new upgraded version
	{Testing plan} – Allow comprehensive testing to be conducted
	{Trial upgrade results} – Track the success or failure rate of trial upgrade
	{Go-live results} – Record and approve the persons who have accepted the new system

Note: (1) {} is an informational item. (2) *** is new item identified from the case study.

*This is an extension of (Ng et al., 2003b; Ng et al., 2003a).