An ERP-client benefit-oriented maintenance taxonomy

Celeste See Pui Ng *, Guy G. Gable, Taizan Chan

Information Systems Management Research Group (ISMRG), Queensland University of Technology, GPO Box 2434, Brisbane, QLD 4001, Australia

Received 17 August 2001; accepted 28 August 2001

Abstract

The worldwide installed base of enterprise resource planning (ERP) systems has increased rapidly over the past 10 years now comprising tens of thousands of installations in large- and medium-sized organizations and millions of licensed users. Similar to traditional information systems (IS), ERP systems must be maintained and upgraded. It is therefore not surprising that ERP maintenance activities have become the largest budget provision in the IS departments of many ERP-using organizations. Yet, there has been limited study of ERP maintenance activities. Are they simply instances of traditional software maintenance activities to which traditional software maintenance research findings can be generalized? Or are they fundamentally different, such that new research, specific to ERP maintenance, is required to help alleviate the ERP maintenance burden? This paper reports a case study of a large organization that implemented ERP (an SAP system) more than three years ago. From the case study and data collected, we observe the following distinctions of ERP maintenance: (1) the ERP-using organization, in addition to addressing internally originated change-requests, also implements maintenance introduced by the vendor; (2) requests for user-support concerning the ERP system behavior, function and training constitute a main part of ERP maintenance activity; and (3) similar to the in-house software environment, enhancement is the major maintenance activity in the ERP environment, encompassing almost 64% of the total change-request effort. In light of these and other findings, we ultimately: (1) propose a clear and precise definition of ERP maintenance; (2) conclude that ERP maintenance cannot be sufficiently described by existing software maintenance taxonomies; and (3) propose a benefits-oriented taxonomy, that better represents ERP maintenance activities. Three salient dimensions (for characterizing requests) incorporated in the proposed ERP maintenance taxonomy are: (1) who is the maintenance source? (2) why is it important to service the request? and (3) what—whether there is any impact of implementing the request on the installed module(s)?

© 2002 Elsevier Science Inc. All rights reserved.

Keywords: Case study; ERP characteristics; Maintenance taxonomy; Business benefit; Patch maintenance; Upgrade; User-support; ERP maintenance definition

1. Introduction

An enterprise resource planning (ERP) system is enterprise-wide packaged application software with fully integrated business processes for enterprise management. ERP sales saw dramatic growth in the nineties. Today ERP has become a basic business information-processing requirement for many organizations. Over the last decade, organizations worldwide have spent around $300 billion on ERP implementations (James and Wolf, 2000). Despite their huge price tag, many organizations introduced ERP systems seeking best business practices, tightly coupled system integration, economies-of-scale for maintenance support, competitive advantage, and globalization potential. The ERP installed base now comprises tens of thousands of organizations worldwide and millions of licensed users (Girard, 2000).

Similar to in-house software, it has been observed that costs of maintaining ERP can exceed those of initial software implementation, with average annual ERP maintenance costs estimated at 25% of original implementation costs (Glass and Vessey, 1999). This observation is akin to that made in the early seventies, when it was realized that the rapid adoption of IS had led to an
“iceberg” of software maintenance costs, most of which were hidden below the surface. Organizations often focus on the initial IS investment, paying relatively little attention to total cost of ownership and lifecycle costs of maintenance and upgrades. Glass and Vessey suggest that these post-implementation costs can be 70% of the total software cost (Glass and Vessey, 1999). Furthermore, the large ERP installed-base, together with a potentially large number of future ERP adoptions by small- and medium-sized enterprises (SME), is expected to result in a growing “iceberg” of costs associated with ERP maintenance and upgrade.

Despite costly lessons learned from paying insufficient attention to maintenance issues related to in-house developed software, little is known of the nature of maintenance performed on ERP. This paper reviews literature on in-house and ERP software to identify differences between ERP and in-house software maintenance activities and issues (‘Is ERP maintenance simply an instance of traditional software maintenance?’); describes an exploratory, descriptive case study aimed at better understanding the ERP maintenance environment (‘What are the maintenance activities involved in an ERP environment?’); examines the dimensions for characterizing ERP maintenance requests; and investigates the sufficiency and suitability for ERP of existing in-house software maintenance typologies (‘Is there a need for a new ERP maintenance taxonomy?’).

Discussion of ERP maintenance activities in this paper is from the client or ERP-using organization’s perspective, where maintenance activities pertain to the client’s installed version. ERP maintenance activities from the vendor’s perspective deserve separate discussion. It is important to note that prior literature has focused on the two extreme examples illustrated in Table 1 (though little on the left-hand side). In reality, these extremes are rare. ERP are implemented to integrate many existing in-house systems. Further, in-house development often extensively employs consultants and contractors, and facilities management and maintenance of in-house systems can be outsourced. Maintenance issues associated with integrated ERP and existing legacy systems, too, is worthy of separate discussion.

IEEE standard for software maintenance defines software maintenance as the “modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a modified environment” (IEEE, 1998). In this study, we define ERP maintenance as “post-implementation activities related to the packaged application software undertaken by the client-organization from the time the system goes live (i.e. successfully implemented and transported to the production environment) until it is retired from an organization’s production system. A clear and precise definition of ERP maintenance is given by the end of this paper.

We define customization (or configuration) as effort to configure the ERP system using switches/tables provided by the vendor, in order to adapt part of the system to support an organization’s preferred business processes, practices and requirements. (Note that the typical ERP contains 800–1000 business processes, and 8000 or more configuration tables (Glass, 1998). Customization is done by choosing among these business processes and setting the configuration-table values.) On the other hand, the term “modification” refers to changes made to the existing ERP standard code, software properties and functionality.

The paper is organized as follows. In Section 2 we review the literature, highlighting differences between the characteristics of ERP and typical in-house software. In light of these differences, suitability of traditional software maintenance taxonomies for describing ERP maintenance activities is also discussed. In Section 3, the case study is presented, and data collection and analysis are described. In Section 4, maintenance activities in the case organization are identified, and differences and similarities with in-house software maintenance are examined. A clear and precise definition of ERP maintenance is presented and the salient dimensions for characterizing ERP maintenance requests are determined. ERP maintenance activities of the case organization are then mapped onto existing in-house software maintenance classifications. Based on findings from the literature and the case study, Section 5 describes a new taxonomy of ERP maintenance activities based on client-benefits. The paper concludes with brief discussion of contributions from this study and possible future research directions.

### 2. Literature review

With the objectives of identifying differences between ERP and in-house software characteristics, and assessing the sufficiency for ERP of existing in-house software maintenance typologies, two main streams of literature are reviewed—i.e. literature on the characteristics and nature of ERP, and literature on in-house software maintenance.

#### 2.1. ERP and in-house software characteristics

It is useful to contrast ERP and in-house software along the software lifecycle phases, i.e. from selection

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Comparison of two extremes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERP</td>
<td>In-house</td>
</tr>
<tr>
<td>Only</td>
<td>100% Custom-built</td>
</tr>
<tr>
<td>Fully integrated</td>
<td>Loosely coupled with other systems</td>
</tr>
<tr>
<td>Wall-to-wall, cross function</td>
<td>Single function</td>
</tr>
<tr>
<td>Most outsourcing of support</td>
<td>100% In-house support</td>
</tr>
</tbody>
</table>
through replacement. Selection of an ERP system is found to involve not only choosing the software system (i.e., software functionality, and its technological environment) that fits an organization's business practices or requirements, but also selecting a vendor who is reliable and stable with a reasonable installed base of customers and a good record of customer service and ongoing maintenance support (Wilson, 1999; Jakovljevic, 2000a,b,c). Conversely, no such effort (or system selection stage) is required for in-house software. Indeed, custom development involves system specification and physical design. At the acquisition stage, ERP is considered relatively easier to acquire (i.e., bought off-the-shelf) than in-house software (i.e., involves software development). For vanilla (as-is) implementations of ERP (and packaged software generally), little if any system analysis, design or coding are required (Davis, 1988). However, like packaged software in general, ERP is a generic enterprise solution, often requiring customization and (some) unavoidable modifications. Alternatively, organizations will accept some degree of misfit (Soh et al., 2000; Stedmam, 2000; Martin et al., 2002), and adapt their business processes or operations to the software (Greenbaum, 1996; Hecht, 1997; Bingi et al., 1999; Davenport, 1999; Glover et al., 1999). In-house software, on the other hand, is tailor-made and developed based on the organization-specific structures, culture and business processes (Brehm et al., 2001).

Five main benefits sought from ERP implementation are competitive advantage (Weston and Stedmam, 1998; Deloitte Touche Tohmatsu, 1999a,b; Heald and Kelly, 1999; Travis, 1999; Shang and Seddon, 2000; Irani and Love, 2001), globalization (Vernadat, 1996; Gable, 1998; Holland et al., 1998; Freedman, 1999), integrated system (Davenport, 1999; Jakovljevic, 2000a,b,c; Markus, 2001), best practice/business processes (Bingi et al., 1999; Carlino and Kelly, 1999a,b; Deloitte Touche Tohmatsu, 1999a,b; Sieber et al., 1999; Jakovljevic, 2000a,b,c; Markus, 2000), and cost effectiveness or reductions (Hicks and Stecke, 1995; Butler, 1999; Carlino and Kelly, 1999a,b; Stein 1999a,b; Norris et al., 2000; Shanks et al., 2000; Markus, 2001). These have been key motivations for the rapid deployment of ERP over the past ten years (Jakovljevic, 2000a,b,c); and, are based on the business perspective (Markus and Tanis, 1999).

ERP provides the opportunity, infrastructure and advanced technologies that allow organization to (1) gain competitive business advantage by preparing the organization for future challenges (Heald and Kelly, 1999; Travis, 1999) and the need to remain competitive, according to Advanced Manufacturing Research, Inc., Boston in (Weston and Stedmam, 1998), (2) dynamically adapt, grow and extend businesses (Irani and Love, 2001) and adopt new business strategies or develop new partnerships (Deloitte Touche Tohmatsu, 1999a,b) by having an open system and the capacity to operate worldwide, (3) increase customer responsiveness, data access and satisfaction by reducing customer service time and facilitating worldwide access and distribution of information about company, product and sales (Jakovljevic, 2000a,b,c) by using the customer relationship management (CRM) application, and (4) build cost leadership by achieving economies-of-scale through streamlined processes or shared services (Shang and Seddon, 2000).

The functionality in ERP allows organizations to do business in multiple-currencies and languages, and the uniform interface of ERP-product across national borders have helped facilitate progress towards globalization and global market development (Gable, 1998; Holland et al., 1998; Freedman, 1999). Globalization enables organizations to operate in the worldwide market (see Vernadat, 1996), provide equivalent levels of service to customers worldwide (Jakovljevic, 2000a,b,c), overcome the problem of information fragmentation, and improve information flow to and from customers, suppliers and business partners. Once achieved, e-commerce and e-business become possible through second-wave applications such as supply-chain management (SCM), online procurement, and CRM (Jakovljevic, 2000a,b,c).

ERP brings the benefit of full business processes integration and standardization (Deloitte Touche Tohmatsu, 1999a,b; Stein 1999a,b; Whearyl, 1999; Jakovljevic, 2000a,b,c; Markus, 2000; Markus, 2001). The availability of a wide range of applications and modules in ERP packages has meant that user-organizations can satisfy most of their application needs with the single ERP. This has eliminated integration complexities associated with applications from multiple-vendors and has enhanced information flow among internal processes. An integrated and centralized system has provided support for complete data visibility for all levels of organizational management, thereby facilitating corporate and strategic decision-making (Hicks and Stecke, 1995; King, 2000; Ross and Vitale, 2000).

According to Davenport and Short (1990), business process is defined as “a set of logically related tasks performed to achieve a defined business outcome” (p. 12). ERP serves as a more controlled and more coherent contact point among the internal business units, regardless of geographical separation, and has improved the overall business processes and practices (see Hammer and Champy, 1996; Barnes, 1999; Bingi et al., 1999; Davenport, 1999; Sieber et al., 1999). This has also facilitated improved business performance (Deloitte Touche Tohmatsu, 1999a,b) and shortened product cycle times (Minahan, 1998).

The characteristics of real-time, single and centralized database, and the availability of maintenance support from the vendor have allowed: (1) operational cost reduction, for example reduction in time and cost
associated with order re-entry errors, data entry, wrong shipment and administrative burden for the sales force (SAP, 2002), (2) savings in integration expenses for different applications from different vendors (Stein 1999a,b), (3) shortened cycle times and reduced inventories (Minahan, 1998), and (4) lower maintenance cost, as the cost is spread over many other users (Hicks and Stecke, 1995; Butler, 1999; Whearly, 1999; Markus, 2001).

In contrast, in the past, traditional in-house software development focused more on technical aspects such as programming and advanced software engineering techniques (Gibson et al., 1999). Although benefits are considered in the feasibility study during the in-house software development, they are primarily meant to get expenditure approval. Once the expenditure is approved, little further attention is paid to benefits, and most effort is expended on technical implementation (Ward et al., 1999). In some cases, technology was utilized to speed up existing processes but performance deficiencies of these processes were not addressed (Hammer and Champy, 1996). Some organizations use technology (i.e. the system) as administrators’ tool to establish control (Dhillon, 2000) as opposed to delivering benefits. Most of these in-house applications are designed to support an individual operational or tactical area. Hence, integration among these individual applications and business processes is not only loosely coupled but also prohibits and complicates centralized decision making processes.

Implementation of a large ERP system requires not only substantial time and effort, but also a wide range of expertise and knowledge (besides knowledge of existing business processes) on the functional aspects of the package, system configuration and system integration, technical knowledge of the related hardware and software, project management and change management, managing knowledge transfer, and organizing system users’ training. ERP-adopting organizations typically lack this expertise and usually outsource these activities to the ERP vendor, hardware vendor, and consulting firms (Simon, 1997; Holland et al., 1998; Sumner, 1999). Implementing in-house software may not demand such a wide range of expertise, as the software is usually constrained to the expertise and knowledge of the in-house system developers and system users. Also, while having a balanced project team with both internal business people and IT staff is crucial for ERP implementation projects (Shanks et al., 2000), traditional in-house software has been largely IT driven.

In addition to commonly understood critical success factors (CSFs) such as obtaining and maintaining top management support (Bingi et al., 1999; Sumner, 1999), well-defined implementation scope, and project goal management (Clemons, 1998), two of the most important CSFs in ERP implementation are: (1) choosing the right business processes (Ross and Vitale, 2000), and (2) ensuring minimal customization and system modification (Davis, 1998; Martin et al., 2002). In contrast, the latter two factors are less pivotal for in-house software.

In the maintenance phase, there are several obvious differences between ERP and in-house applications. While ERP change-requests may come from either the software vendor or system users and IT-staff (these are discussed in detail in Section 4), in-house software maintenance requests generally originate internally (i.e. system users and IT-staff) only (except as regards other packaged system software e.g. the operating system and database management system). With ERP, maintenance support for bug fixes (Songini, 2000) and regular functionality enhancements (Gray, 1998; Stein 1999a,b) are available from the ERP vendor. Though this support may not cover all (internal) user maintenance requests, it distinguishes ERP from in-house software that must be fully supported internally. Traditionally, the process of in-house software maintenance involves maintenance problem area analysis (including identification of the causes, solutions, and trade-offs among different solutions), designing the modifications, writing and implementing modifications, system and user acceptance testing, and final delivery of the changes into production (IEEE, 1998). With ERP, neither system design nor coding is required if the maintenance originates from the vendor. However, extensive impact analysis may be required for implementing patches introduced by the vendor, in order to determine whether re-application of prior modifications are needed. These must be re-tested and re-verified to ensure that the system is working properly following the patches. Finally, the changes are transported to the production environment (Collins, 1999). The primary motivations for ERP enhancement maintenance among others are to realize more business benefits. In contrast, in-house software system enhancement has mainly focused on: (1) cost effectiveness of the programs (see Swanson, 1976), for instance improving software code quality, maintainability and enhancing software performance, (2) providing new reports and adding data to existing reports (Lientz and Swanson, 1980), and (3) improving quality of documentation (Lientz et al., 1978). (This is not surprising, as the implementation of traditional in-house software is more technical-focused (as discussed earlier) and therefore organizations tend to forget about benefits in maintenance phase!)

ERP is comprised of tightly integrated application modules. Given its breadth and integration, it is argued that each change to ERP can cause more extensive ripple effects and impacts on its existing code (or other integrated application modules) than typical standalone in-house application software. On the other hand, if maintenance patches supplied by the vendor have been thoroughly tested prior to release to clients, user organizations should be largely shielded from these ripple
effects (our experience on the case study suggests that this is not always the case; this is an interesting area of further research). The complexity of in-house software maintenance is reported to be a function of the system size (Banker et al., 1993; Kemerer and Slaughter, 1997; Banker et al., 1998), system age (Lientz and Swanson, 1980; Martin and McClure, 1983; Banker et al., 1989), quality of the original system (Krogstie and Solvberg, 1994), the amount of maintenance done previously (Swanson and Beath, 1989; Gode et al., 1990), and software development practices (such as the use of a code generator and structured techniques) (Slaughter and Banker, 1996). While these may also be the drivers of ERP maintenance complexity for the vendor, the complexity of ERP maintenance for the client organization (when implementing the vendor’s patches) is largely dependent on the amount of modification done during initial implementation (400-Group, 1998) and post-implementation. This is because the application of patches may cause previous modifications or changes done to the standard code to be overwritten or to no longer function correctly. As a result, each modification may cause extra effort in impact analysis, re-applying previous code, and re-testing the whole system.

Software replacement is one of the major activities involved in software maintenance. In-house software replacement is driven by increasing deterioration of the code structure resulting from cumulative maintenance over time (Lehman and Belady, 1980; Lyons, 1981; Swanson and Beath, 1989). As the code structure deteriorates over time, it becomes increasingly difficult and requires more effort, to maintain. Thus, maintenance becomes more costly over time. In light of this, it is argued in the existing literature that over time it becomes increasingly more economic to replace the old code by rewriting the software from scratch (Gupta and Ragunathan, 1988; Gode et al., 1990; Chan et al., 1996); that is, when the projected lifecycle costs of maintaining the old system are greater than the costs of rewriting and maintaining a new system. In contrast, the decision to upgrade ERP is usually not driven by code deterioration or anticipated reduction in maintenance costs alone, but by the desire to: (1) realize business benefits of a new version (Deloitte Touche Tohmatsu, 1999a,b; Davenport 2000a,b), and (2) avoid vendor support termination (Craig, 1999; Stedman, 1999). The ERP-adopting organization does not have to develop and re-write the ERP system itself but rather it replaces (or upgrades) the old version with a readily available new version from the ERP vendor. A summary of differences between ERP and in-house software is given in Table 2.

2.2. ERP and in-house software maintenance classifications

In light of differences discussed in the previous section, here we explore the suitability of traditional software maintenance taxonomies for describing ERP maintenance activities. A widely accepted and applied classification of software maintenance differentiates corrective, adaptive, and perfective maintenance activities (Swanson, 1976). The main criterion that differentiates these three categories is the intention of the maintenance request (which is usually to remedy the cause of the request). The corrective and adaptive maintenance categories are respectively meant to keep the system up and functioning correctly, and to be able to operate well in a new environment. Perfective maintenance seeks to improve the cost effectiveness of the program, and to better serve the system users’ needs. Detailed descriptions of each of these three categories are shown in Table 3.

It is noted that Swanson’s classical maintenance classification does not consider user-support maintenance requests, which are later found, in a study by Abran and Nguyen Kim (1991) (on in-house developed application), to account for 22% of the total software maintenance effort. User-support requests are associated with consulting with, and supporting (user) requests relating to the system’s behavior, rules and functions. Taking into consideration that ERP is an enterprise-wide system entailing fully integrated application modules and business processes, we believe that responding to user-support requests may constitute a significant ERP maintenance activity, possibly even more significant for ERP than in-house software. (This proposition is supported in Section 4.2.1.)

Burch and Grupe (1993) propose an extension to Swanson’s classical maintenance categorization a new dimension called “preventive”, which is defined as “proactive maintenance—[which] consists of a periodic inspection and review of the system to uncover and anticipate problems” (p. 25). Preventive maintenance is believed to be less substantial in the ERP maintenance environment. Firstly, the nature of maintenance in an ERP environment is relatively more reactive than typical in-house software. While in-house maintainers are familiar enough with homegrown software systems (which are relatively smaller in size) to do periodic inspections, familiarizing and gaining full understanding of an ERP system represents a daunting challenge (if not

---

1 This is an extreme case. In practice, this option assumes that all the relevant documentation and people, who are familiar with the system (in relation to what and how the software system functions), are available in an organization (Glass, 1991).

2 Vendors withdraw support for older versions in order to contain and minimize their own maintenance costs, and to guarantee availability of human resources, skills and services support for their clients. Hence, they must focus their maintenance support resources on one or few version(s). While this is of interest, this paper focuses on maintenance of ERP from the client perspective.
Table 2
Summary of differences between ERP and in-house software

<table>
<thead>
<tr>
<th>Phase, acquisition, and implementation</th>
<th>Issue</th>
<th>ERP</th>
<th>In-house</th>
</tr>
</thead>
<tbody>
<tr>
<td>System selection</td>
<td>Require detailed selection process for the right ERP system from the right vendor</td>
<td>Bought off-the-shelf; there is some misfit; and most of the time customization and/or modification are required</td>
<td>System is developed internally and therefore no vendor or software selection is required 100% Tailor-made with no misfit</td>
</tr>
<tr>
<td>Acquisition method</td>
<td>Bought off-the-shelf; there is some misfit; and most of the time customization and/or modification are required</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deployment purpose</td>
<td>Gain competitive advantage; achieve globalization; adopt best business practices; benefit from an integrated system; outsource maintenance support</td>
<td>Speed up business processes; utilize advanced software engineering techniques; and facilitate management support in each individual functional area (loose integration among business processes)</td>
<td></td>
</tr>
<tr>
<td>Level of decision support</td>
<td>Facilitate top management/executive corporate and strategic decision-making</td>
<td>Mainly to support lower- and middle-level management to carry out operational and tactical planning and tasks</td>
<td></td>
</tr>
<tr>
<td>Implementation staff</td>
<td>Involve outsourcing to the ERP vendor, hardware vendor, and consulting firms; and require a balanced project team of business and IT staff</td>
<td>Involve mainly internal (both IT and system users) staff</td>
<td></td>
</tr>
<tr>
<td>Implementation success factors</td>
<td>Choosing the right business processes; ensuring minimal system modification</td>
<td>Relatively lesser effort is required to choose business processes or to conduct system modifications; system modifications are not an issue</td>
<td></td>
</tr>
<tr>
<td>Maintenance</td>
<td>Vendor, and system users and IT-staff</td>
<td>System users and IT-staff</td>
<td></td>
</tr>
<tr>
<td>Maintenance activities</td>
<td>No coding or system design is required if the request is from the vendor; may involve extensive impact analysis, reconfiguration, re-applying the previous modifications, and re-testing the whole system</td>
<td>System design, program design, coding, unit-testing</td>
<td></td>
</tr>
<tr>
<td>Maintenance motivation</td>
<td>Realize more business benefits: competitive advantage, globalization, best business practices, improved system integration, cost reduction</td>
<td>Focused on the technical aspects—program maintainability and performance, documentation quality</td>
<td></td>
</tr>
<tr>
<td>Impact on other application modules</td>
<td>Ripple effects across tightly integrated modules can be extensive and complex; however, for patches these are usually borne by the vendor in advance of release of the patch</td>
<td>Each maintenance task has relatively lesser impact on the other application modules</td>
<td></td>
</tr>
<tr>
<td>Drivers of maintenance complexity</td>
<td>Amount of previous modifications or changes made to the software—including add-ons/extensions done during the initial implementation, and post-implementation</td>
<td>System’s age, size, amount of maintenance done previously, and the original software quality, development practice</td>
<td></td>
</tr>
<tr>
<td>Replacement</td>
<td>Driven by benefit realization, and vendor support termination</td>
<td>Driven by increasing code structure deterioration, and maintenance and rewrite costs</td>
<td></td>
</tr>
<tr>
<td>Disposal method</td>
<td>Upgrade to readily available new version from the market</td>
<td>Rewrite the software from scratch with fresh code</td>
<td></td>
</tr>
</tbody>
</table>

Table 3
Swanson’s categories of maintenance activities (Swanson, 1976, pp. 492–493)

<table>
<thead>
<tr>
<th>Maintenance category</th>
<th>Objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corrective</td>
<td>Correct errors in a program (i.e. processing failure); fix program that does not perform satisfactorily (i.e. performance failure); and repair violations (if any) of the predetermined programming-standards (i.e. implementation failure)</td>
</tr>
<tr>
<td>Adaptive</td>
<td>Adapt a program to a changed environment (data and/or processing environment)—for example logical restructuring of a database, installation of a new hardware or operating system version, recoding existing program</td>
</tr>
<tr>
<td>Perfective</td>
<td>Eliminate program processing inefficiency such as inferior computational algorithm, inappropriate language features or poor use of computer operator time; enhance program performance; reformat report to improve the readability; add new data element in a report; add new functionality, insert comments in a program; rewrite program documentation; improve program maintainability—to make the program to be more easily modified (when it must be modified)</td>
</tr>
</tbody>
</table>
impossibility) to ERP maintainers because of the system’s size, and complicated application logic. Secondly, this type of system inspection is more likely to be initiated and monitored by the vendor rather than the ERP-using organization partly because of limited access to the system code and the availability of maintenance-support from the software vendor, and because of apprehensions about the ripple effects that may result should this maintenance be implemented by the client organization. Brehm et al. (2001) too suggest that preventive maintenance is a task for the vendor, not clients.

A more recent maintenance classification proposed by Chapin (2000) differentiates 12 types of software maintenance based on evidence of: (1) activities involved in interface support, (2) usage and changes to the documentation, (3) changes to the software properties, and (4) changes to the software functionality. The 12 types are training, consultive, evaluative, reformative, updative, groomsative, preventive, performance, adaptive, reductive, corrective, and enhancive. Table 4 provides characteristics of these classes of maintenance activities.

Chapin’s maintenance classification, while valuable, is believed to be deficient in an ERP context because it does not consider the benefit of doing maintenance, particularly business benefits. ERP software implementation is a business-project (or business-driven, not simply an IT initiative) (Davenport 2000a,b), having huge impact on an organization’s processes and strategies (Bingi et al., 1999; Davenport, 1999; Deloitte Touche Tohmatsu, 1999a,b). ERP implementation is led by the business people with the IT people playing a much smaller role than in traditional in-house software projects. Both Swanson and Chapin’s maintenance taxonomies place much emphasis on the technical aspects of doing maintenance. As a consequence, the Swanson and Chapin taxonomies are not intuitive to business people who are interested in business benefits (if any) of a request. Most organizations implementing ERP systems aim to derive benefits from the systems, and this factor is undoubtedly the driver for continual maintenance of these systems. However, the traditional taxonomies do not facilitate cost and benefit justification of doing ERP maintenance. The strengths and weaknesses of these two classifications are summarized in Table 5.

Thus far the literature review indicates that ERP is not simply an instance of traditional (in-house) software. ERP and in-house software differ in fundamental ways. This suggests that it is important to further investigate the activities involved in the ERP maintenance environment, and to collect empirical data to reveal

<table>
<thead>
<tr>
<th>Maintenance category</th>
<th>Evidence of activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support interface:</td>
<td>Providing training to stakeholders about the system implemented</td>
</tr>
<tr>
<td>Training Consultive</td>
<td>Making time and cost estimates for proposed maintenance work, serving on a help desk, assisting a customer in preparing a maintenance work request, and making specialized knowledge about the software resources available</td>
</tr>
<tr>
<td>Evaluative</td>
<td>Studying the source code and other documentation, tracing how a proposed change might ripple, preparing and running tests, examining the interactions between operating system features and the software to be maintained, searching for needed data, and debugging</td>
</tr>
<tr>
<td>Documentation:</td>
<td>Improving the readability of the documentation, modifying the documentation to incorporate the effects of changes in the local standards manual, preparing training materials, and adding entries to a data dictionary</td>
</tr>
<tr>
<td>Reformativ</td>
<td>Replacing obsolete documentation with accurate current documentation, preparing Unified Modeling Language (UML) models to document existing source code, and incorporating test plans into the documentation</td>
</tr>
<tr>
<td>Updativ</td>
<td>Replacing algorithms or components with more elegant ones, changing data naming conventions, doing backups, modifying access authorizations, and recompiling source code</td>
</tr>
<tr>
<td>Groomativ</td>
<td>Restructuring the code to improve maintainability and put in place a base for making a future change to some new technology</td>
</tr>
<tr>
<td>Preventive</td>
<td>Activities that affect the user (but not the functionality) such as improving system up time, replacing algorithms or components with faster ones</td>
</tr>
<tr>
<td>Performance</td>
<td>Porting/adapting the custom-built software to a new platform, increasing Commercial off-the-shelf (COTS) utilization, and moving to object-oriented technologies</td>
</tr>
<tr>
<td>Adaptive</td>
<td>Eliminating data from output received by the customer, reducing data flows into or out of the software, and reflecting a customer’s narrowed business plan</td>
</tr>
<tr>
<td>Reductiv</td>
<td>Fixing detected bugs, adding more defensive programming, and changing the handling of exceptions</td>
</tr>
<tr>
<td>Correctiv</td>
<td>Adding or replacing business rules to extend or expand the system’s functionality accessible to the customer, and adding data flows into or out of the software</td>
</tr>
</tbody>
</table>
deficiencies of existing software maintenance taxonomies.

3. The case study

Given the literature reviewed in the previous section revealed no guiding theory on ERP maintenance, yet there is an urgent need to understand the phenomenon, this study employs the case study research method (Benbasat et al., 1987; Gable, 1994; Yin, 1994). The case study method is also chosen because it offers opportunity for the researchers to develop a holistic view (Gummesson, 1988) of ERP maintenance activities.

3.1. Background

We studied a State Government Agency (GA) in Australia. GA was created in 1996 to provide corporate services (such as ERP services) to other government departments. This agency had several years of experience in maintaining its ERP system, a SAP R/3 system. GA is a recognized SAP service provider. The agency was selected for several reasons: (1) GA’s comprehensive ERP maintenance records; (2) the Information Systems Management Research Group’s (ISMRG) strong relationship and involvement with State Government in collaborative research; and (3) the ISMRG’s ongoing, existing collaborative projects with GA. GA currently runs and maintains two fundamental SAP modules: Finance (FI) and Human Resources (HR). The FI module has been in operation since November 1998 and HR since April 1999. The initial motivation for GA’s move from its previous finance and human resources management systems to SAP was Year 2000 (Y2K) compliance. At the time of this writing, GA has successfully upgraded its SAP version 3.1H to the latest SAP version 4.6C.

In managing and maintaining the SAP system, GA runs three SAP environments for development, testing, and production. This is a common arrangement in ERP-using organizations (Parkinson et al., 1999). All maintenance is done on the development environment first, and then tested in testing environment for system correctness and user acceptance, before being transferred into the production environment. In some cases, the development and testing environments must be iterated through several times before the change reaches the production system.

3.2. Data collection

In addition to various supporting documents, there were three main sources of case evidence. A total of six 1-h interviews were conducted and tape-recorded. Interviews were conducted with the General Manager, Systems Development Manager, and Systems Operations Manager to obtain background information on GA and to understand their ERP maintenance activities (including reasons for doing the maintenance). These people were invited to participate in this study because of their high level of involvement and responsibility in relation to this research problem. The General Manager holds the seniormost position in the GA organization and is in-charge of executive matters. The main responsibility of the Systems Development Manager is the maintenance and enhancement of the SAP system. The Systems Operations Manager is responsible for ensuring the continuing operation of the SAP system, system administration, and liaising with the service bureau that provides facilities management support (hardware operations and maintenance).

A second source of evidence in the case study was the change-request database. At the time of the study, the change-request database contained 1616 data points or change-requests, which were documented from 2

---

### Table 5

<table>
<thead>
<tr>
<th>Existing maintenance classification</th>
<th>Strengths</th>
<th>Weaknesses (in the context of ERP)</th>
</tr>
</thead>
</table>
| Classical (Swanson, 1976)          | • Based on the objective of the maintenance activity  
• Provides fundamental classes of maintenance activities: corrective, adaptive, and perfective | Does not include maintenance activities that are intended to (see Section 4.1 for details):  
• Provide user-support to system users  
• Keep the installed system up to the vendor’s standard version  
• Maximize business-benefit of doing the maintenance |
| Modern (Chapin, 2000)              | • Based on evidence of activities on interface support, documentation, software properties, and/or software functionality  
• Provides comprehensive classification of maintenance activities: enhance, corrective, consultive, training, groomative, evaluative, reformative, updative, preventive, performance, adaptive, and reductive | Does not consider (see Section 4.1 for details) the evidence of:  
• Maximizing the business-benefit of doing the maintenance activities |
November 1998 to 27 June 2000. The database contains information about the types of maintenance requests, maintenance description (i.e. what was modified and activities involved to resolve the problem), time spent on each request and other details for both completed and in-progress maintenance requests.

The third source of evidence was the user-support database containing only requests pertaining to user-support of the SAP system; it captures information on the types of support-request, support-problem descriptions (i.e. activities undertaken to solve the problem), and details of the requestors and maintainers involved.

3.3. Data analysis

Tape-recorded interviews were transcribed following the interviews. The transcripts were referenced to provide a reliable description of the GA maintenance environment, including the objectives of doing each maintenance activity (or request type) and benefits of maintaining the ERP system. The transcripts were re-examined throughout the process of data analysis to ensure valid data interpretation. The change-request database and user-support database were analyzed in detail for evidence of activities involved in each maintenance request type, and to obtain the frequency and effort distributions across GA’s maintenance activities.

From these observations and our better understanding of GA’s ERP maintenance objectives and activities: we next defined ERP maintenance, then identified salient dimensions for characterizing maintenance requests, and mapped the maintenance activities onto the Swanson (1976) and Chapin (2000) in-house software maintenance classifications (see Fig. 1). The objective here is to investigate whether existing maintenance taxonomies are sufficient and appropriate to classify all ERP maintenance activities.

4. Findings and interpretation

4.1. GA’s ERP maintenance requests: description

GA differentiates ERP maintenance activities into seven categories: GA-enhancement, user-enhancement, corrective, master-data-change, user-support, patch, and upgrade. GA-enhancement, user-enhancement, corrective, master-data change, and user-support are the internally originated maintenance activities. Internally originated maintenance requests are introduced by two groups: GA’s IT-staff, and the Government Departments (i.e. GA’s clients, including GA itself) that outsource SAP to GA. These Government Departments are referred to as the system users. GA-enhancements are initiated by GA IT-staff only, whereas user-enhancements and user-support requests are introduced by the system users only. Corrective and master-data-change can be initiated by either group.

Patch and upgrade activities are introduced by the software vendor. While system users incur no incremental cost (from GA) for maintenance requests related to master-data-change, corrective, GA-enhancement, user-support, patch or upgrade, costs associated with user-enhancement does flow back to the system users. A change-request in this study refers to either master-data-change, corrective, GA-enhancement, patch or upgrade. GA maintenance activities are summarized in Table 6.

The SAP patches, implemented by GA, are known as legal-change-patches (LCPs) or simply patches (see Section 4.2.4 for more details). According to GA’s senior manager, in version 3.1H LCPs contain not only
corrections and adjustments for legal changes in the Human Resources module, but also for the Finance module.

[...] These bug fixes and enhancements could be meant for any module in SAP. Hence, an LCP may comprise some fixes and enhancements for manufacturing, some for accounting, others for payroll, etc. Systems Development Manager, March 2001.

Though the vendor distributes patches on a regular basis, it is the responsibility of the client organization to implement them into their ERP system. Interviews with GA senior managers found that GA implements these patches for three main reasons:

(i) Fix bugs in the existing system,
(ii) Adapt to external environmental changes such as changed Government regulations, and
(iii) Keep the installed system up to the standard required by the vendor.

As regards point (iii), patches are sometimes implemented even though they may not be relevant, useful or have impact on the functionality, programs or sub-modules (or the module installed) used by the client-organization.

Some of these LCPs cover modules that are not used [in GA]. Overall, the patches have impact on 30% of our [GA] implementation. However, for LCPs that cover the modules used [by GA], in particular modules such as HR where extensive changes to the system have been made, then extensive impact analysis has to be done. Any overwritten custom code will be carefully evaluated and tested. Irrespective of the extent of impact an LCP introduced, only a small fraction (1-5% (max)) of the LCP is deemed really useful [to GA]. Despite the cost associated with incorporating LCP, [GA] has to incorporate all the LCPs to ensure that their implementation remain a “standard SAP”—Systems Development Manager, June 2000.

Keeping an existing system up to the vendor’s requirements by applying the patches is important. One of the senior managers notes:

“This is because of two main reasons: (1) a later LCP works only if earlier LCPs have been incorporated. Thus, to ensure that [GA] is able to exploit an LCP that may be useful, it must also incorporate those that are not; and (2) upgrade cannot be done unless all the LCPs prior to the new version have been incorporated”—Systems Development Manager, May 2000.

On the other hand, GA upgrades to a new version for two main reasons:

(i) Keep the system a supported-version—i.e. technical upgrade, and
(ii) Obtain new software functionality and more business improvement—i.e. functional upgrade.

For point (i), GA’s Systems Development Manager notes:

Given the huge cost associated with upgrade [...], GA would not have decided to upgrade if not for the fact that the vendor is withdrawing support for the version we are currently using—Systems Development Manager, June 2000.

Table 7 includes detailed descriptions of the objectives and sample of activities of each GA’s maintenance request category. (The objectives and evidences of GA’s ERP maintenance activities are studied in detail here because they are the fundamental characteristics addressed in Swanson and Chapin’s maintenance classifications. The objectives of these maintenance activities/requests (i.e. in the second-column of Table 7) have been gathered from interviews with GA’s senior managers, whereas the evidence of maintenance activities (i.e. in

---

3 Note that no data analysis on upgrade requests is possible in this paper because GA was just considering its first upgrade decision during the initial data collection for this research project.
the third-column) are from data captured in the change-request and user-support databases. The activities listed in Table 7 may not be comprehensive, but they represent the majority of activities involved in each maintenance request category undertaken by GA. GA patch-maintenance is divided in this study into three sub-categories based on distinct maintenance objectives. Similarly, upgrade activity is usefully divided into two sub-categories of activities that are different in nature and meant for different purposes. This table will be intensively referenced in: (1) Section 4.4 where we identifying the dimensions for characterizing ERP maintenance requests, and (2) Section 4.5 where we map GA’s maintenance requests into Swanson and Chapin’s maintenance taxonomies.

4.2. GA’s ERP maintenance requests: basic descriptive statistics

This section presents the basic descriptive statistics, i.e. frequency and effort distributions, of GA’s ERP maintenance requests derived from the user-support and change-request databases. We first focus on GA’s internally originated user-support requests, and change-requests. This is followed by discussion on ERP vendor-initiated maintenance activities. Note that GA batches
patch-maintenance and therefore substantially reducing their frequency (counts) making comparison (of frequency counts) with the other maintenance categories questionable. Basic descriptive statistics for GA’s ERP upgrade is not discussed here since there is only one upgrade project done so far.

4.2.1. Internally originated maintenance requests: user-support

User-support requests are introduced by GA’s SAP system users. These requests are for consultation or assistance with the system functionality and usage, or training in the SAP system. They are recorded in the user-support database. The data analyzed in this section is based upon the maintenance data recorded by GA from 2 November 1998 to 27 June 2000. The number of completed requests and effort spent on this category of requests are shown in Table 8.

A study by Abran and Nguyenkim of in-house software maintenance showed that user-support requests accounted for 22% of annual total maintenance effort (Abran and Nguyenkim, 1991). In the current study, user-support requests constitute 32% of total maintenance effort, over the study period of 20 months. (Note that this percentage is the most conservative, as the total maintenance time for 1438 change-requests is computed using the mean value (=15.5) from 593 completed maintenance projects—more details is given later. If the median statistic (=5.24) is used, user-support requests will constitute 59% of total maintenance effort. On the other hand, if the mode statistic (=1) is used, user-support requests will account for 88% of total maintenance time.) It is noted that the external validity of this finding is limited because both cases compared here are based on a single organization.

In general, this finding supports the earlier suggestion that user-support in an ERP environment may be more important than for in-house software. This can be explained that for in-house software, the system users participate in the software development process (such as prototyping system development), and the system is built based on their business processing requirements and in a way akin to how they always do business. Consequently, the system users do not require as much user-support from the IT-staff as is required in an ERP (packaged software) environment. Nevertheless, longitudinal study of the distribution of ERP user-support requests would be valuable to further validate the findings in this section.

4.2.2. Internally originated maintenance requests: change-requests

Analysis of GA’s internally originated change-requests (total of 1616 requests) shows that 64% \((n = 1031)\) are GA-enhancement and 5% \((n = 81)\) are user-enhancement requests (see Table 9). Reasons for the relatively smaller amount of system user-enhancements are: first, a desire by requesting users to minimize incremental costs of GA services; and second, GA tends to restrict user-enhancements, which usually involve writing custom code or making changes to SAP code. Though the users pay for their enhancement work, GA absorbs future, ongoing costs of maintaining these enhancements. Corrective maintenance requests account for 21% \((n = 347)\) of total requests, whereas master-data-change constitutes about 10% of the total. (Of note is that the corrective requests, collected from GA, comprise bugs found in both customized code and in the standard ERP/SAP code.)

Although there are 1616 maintenance requests in the GA change-request database, only 1438 of these have been completed. From that figure, only 593 requests have recorded the effort/time spent in completing the requests. Table 10 shows the effort distribution among GA-enhancement, user-enhancement, corrective and master-data-change maintenance requests for the 593 completed and valid maintenance requests.

It is observed that 64% of GA’s total maintenance effort on internally originated change-requests is spent on enhancement requests (Table 10); this is inclusive of both GA-enhancements and user-enhancements. Cor-

<table>
<thead>
<tr>
<th>Internal change-requests</th>
<th>Frequency count</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA-enhancement</td>
<td>1031</td>
</tr>
<tr>
<td>User-enhancement</td>
<td>81</td>
</tr>
<tr>
<td>Corrective</td>
<td>347</td>
</tr>
<tr>
<td>Master-data-change</td>
<td>157</td>
</tr>
<tr>
<td>Total</td>
<td>1616</td>
</tr>
</tbody>
</table>

*Based on the mean value for 593 completed maintenance projects.
rective requests command only 22% of the total effort, whereas master-data-change requests take 8% of the total effort. In general, this study confirms the findings by Glass and Vessey (1999), that enhancement maintenance is a major ERP maintenance activity.

4.2.3. Distribution of change-requests versus user-support requests

A time series is plotted in Fig. 2; it shows the number of change-requests and user-support requests arriving over the period from November 1998 to June 2000. (The readers need to be careful, since frequency count here is not representative of the amount of maintenance effort.) Fig. 2 shows that the number of user-support requests is relatively larger than change-requests during the first 6–7 months after the SAP system was introduced at GA. This is because the ERP system is: (1) a new system with which the users are not familiar, and (2) a large system with substantially different interfaces from their earlier system. As a result, they tend to seek advice, training and assistance during the first few months of using and interacting with the system. It is further noted that during the first six months both the change-requests and user-support requests are continually decreasing.

In the seventh month (May 1999), both change-requests and user-support requests are found to spike suddenly over 1580% and 100% respectively. This sharp increase in demand for maintenance coincides with the introduction of SAP Human Resources module (in late April 1999). (The first peak occurred in November 1998 is in conjunction with introduction of the SAP Finance module.)

After the first seven months—i.e. when the users have become more familiar and confident with the ERP system, the frequency of change-requests and user-support requests begins to decrease slowly over the subsequent 13-month period (Fig. 2). In fact, after using the ERP system for (the first) seven months, user-support requests become relatively fewer than change-requests.

Alternatively, from the installation per module perspective, it is observed that (from Fig. 2) the number of user-support requests will stabilize after 2–3 months of a successful module implementation. It takes two months for the demand for user-support for Finance module to settle down and three months for both the Finance (FI) and Human Resources (HR) modules. In general, it makes sense to expect large a number of user-support requests during the first month after a new module is implemented (e.g. in November 1998 and May 1999). These drop dramatically in the second (in December 1998 and June 1999) and third months (of July 1999), and eventually the number of requests becomes more or less the same thereafter. This finding provides some insights into human resources requirements for helpdesk and other maintenance support, to management. As for change-request, consistent pattern between the FI and HR module, in terms of changes in demand, is also observed. However, the demand for change-request seems to become stable five months after both FI and HR go live.

4.2.4. Externally originated ERP maintenance requests: patches

There are only three patch-maintenance records for version 3.1H—in the change-request database: two were completed, and one was still in progress during our first data collection period. GA batches the patches for maintenance purposes (refer to Table 11).

| Table 10 |
| --- | --- |
| Effort distribution among the internally originated change-requests | |
| Internal change-requests | Effort\(^a\) (h) (%) |
| --- | --- | --- |
| GA-enhancement | 3857 | 42 |
| User-enhancement | 2031 | 22 |
| Corrective | 2585 | 28 |
| Master-data-change | 695 | 8 |
| Total | 9168 | 100 |

\(^a\) The statistics is based on 593 valid data of the requests.

Fig. 2. Number of change-requests and user-support requests over time.
We have chosen to batch LCPs and apply the LCP based on business need; this becomes a problem when there is an LCP addressing a legislative requirement that we require urgently as we then have to apply a backlog of LCP. SAP seems to assume that you are up to date—General Manager, May 2001.

For example, the first patch-maintenance consists of 34 LCPs numbered 29–62. Note that the implementation of LCPs numbered 1–28 was not recorded in the change-request database as they were done during the initial SAP system implementation. The second patch-maintenance consists of a batch of ten LCPs, numbered 63–72, and the third is a batch of eight LCPs, numbered 73–80. Essentially GA batches LCPs in this way to achieve economies-of-scale. GA’s General Manager suggests:

[...] Implementing each patch individually would require several times the substantial effort already expended here, and would take up 80% of GA’s maintenance resources. [...] How one manages LCPs can have a huge impact on the costs of maintenance, and, that one’s management strategy is to batch these periodically, and to monitor patches for cumulative benefits and wait until the perceived benefits justify the effort—General Manager, May 2001.

It is found that the first patch-maintenance request was primarily meant to keep the system up to the standard SAP, in order that the second patch-maintenance could be implemented. The second patch-maintenance was stimulated by the need to become fully Y2K compliant, whereas the third patch-maintenance is motivated by the need to deal with new tax regulations (i.e. Goods and Services Taxation—GST) in Australia.

To summarize findings so far, the empirical data analysis on GA’s ERP maintenance activities supports the earlier suggestion that requests for user-support (concerning the ERP system behavior, function and training) constitute a main part of GA’s ERP maintenance activities. It is observed that the number of user-support requests will increase dramatically in the first 2–3 months of a successful module implementation and become stable thereafter. Similar to in-house software environment, enhancement is the major maintenance activity in GA’s ERP environment, encompassing almost 64% of total change-requests effort.

4.3. ERP maintenance definition

Based on the empirical data, discussed in Sections 4.1 and 4.2, this study defines ERP maintenance as:

Post-implementation activities related to the packaged application software undertaken by the client-organization from the time the system goes live (i.e. successfully implemented and transported to the production system) until it is retired from an organization’s production system:

- To keep the system running,
- To adapt to a changed environment in order to operate well,
- To provide help to the system users in using the system,
- To realize benefits from the system (best business processes or practices, improved system integration, operational cost effectiveness), and
- To keep the system a supported-version and meet the vendor’s requirements for standard code.

These activities include:

- Implementing internal change-requests (initiated by an ERP-using organization’s system users and IT-staff),
- Responding or handling user-support requests (initiated by an ERP-using organization’s system users),
- Upgrading to new versions/releases (introduced by the vendor), and
- Performing patches (support provided by the vendor).

The first two objectives of ERP maintenance are similar to the objectives of in-house software maintenance given by Swanson (1976).

Both internal and external change-requests would entail changes to the system properties such as data dictionary, programs, screens, user interfaces, and/or documentation. These comprise bug fixes, adaptation to the external environment, enhancement to existing functionality, additional functionality, and maintaining the system up to the vendor’s requirements (patches and new releases or upgrade versions). User-support is usually related to software system training, and consultation on the system usage and functionality.

<table>
<thead>
<tr>
<th>Maintenance</th>
<th>Number of LCPs (version 3.1H)</th>
<th>Main objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request-1</td>
<td>34</td>
<td>Keep the system up to the standard SAP</td>
</tr>
<tr>
<td>Request-2</td>
<td>10</td>
<td>Keep the system fully Y2K compliant (for adaptation)</td>
</tr>
<tr>
<td>Request-3a</td>
<td>8</td>
<td>Adapt the system to the new taxation regulation—GST</td>
</tr>
<tr>
<td>Total</td>
<td>52\textsuperscript{a,b}</td>
<td></td>
</tr>
</tbody>
</table>

\textsuperscript{a} This was not completed at the point of our initial data collection.  
\textsuperscript{b} This does not necessarily represent the exact number of LCPs introduced by the vendor during the study period.
4.4. Salient dimensions for characterizing ERP maintenance requests

Having gone through the analysis and discussions in previous sections, we observe that ERP maintenance requests can be usefully characterized along three salient dimensions: (1) source of the maintenance request, (2) the existence of any business-benefit, and (3) the existence of any impact of the maintenance on the client’s installed ERP module(s).

4.4.1. Maintenance source

As demonstrated from GA’s maintenance activities, both internally originated and vendor originated maintenance are substantial. Thus, this dimension “maintenance source” comprises the two main players: ERP-client and ERP-vendor.

4.4.2. Business-benefit

An observation from Table 7 is that, some of the maintenance requests (such as GA-enhancements and user-enhancements) are meant for realizing more business-benefits from the system. Some of the GA-enhancements are meant for obtaining benefits for example best practice/business processes, integrated system and cost reduction from the SAP system, and some of the user-enhancements are to improve the requesting system user’s business processes. GA’s Systems Operations Manager states:

[...] In addition to cost reduction, a main driver of ERP maintenance and upgrade is to maximize business benefit realization—Systems Operations Manager, June 2001.

This dimension “business benefit” distinguishes maintenance requests, which bring business-benefits from those not. The classic maintenance-intentions for keeping an installed version up and running correctly and adapting the software to a changed environment, as well as user-support, are subsumed to be the mandatory and basic requirements before an organization is considering about business benefit. A maintenance request can contribute to one or more of the business benefits such as best practice or business processes, integrated system and operational cost reduction. A request is perceived to bring business benefit if it:

(1) Improves or enhances the way an organization does business—to streamline best practice or business processes and enhance system integration;
(2) Improves or enhances the existing ERP functionality; and/or
(3) Could keep an existing version away from vendor-support termination—cost effective.

4.4.3. Impact on installed module(s)

Based on the earlier discussion in Section 4.1 on the reasons, given by GA’s senior manager, for doing the patch-maintenance and upgrade, it is found that while some of the maintenance (support) introduced by the vendor are useful, some are not. A maintenance support is perceived to be useful if it is related to and has impact on the module(s) installed on GA’s ERP system. Patches incorporating bug fixes (or improved functionality) for module, which is not used by GA, are perceived as having no impact on GA’s ERP system. Besides external requests, internal maintenance requests such as user-support requests also do not affect the installed ERP module(s), when they are serviced or handled. Therefore, they have no impact on the installed ERP module(s).

This dimension “impact on installed module(s)” indicates whether a maintenance request has any impact of:

(1) Repairing, improving or enhancing, adding and/or deleting the functionality in the client’s ERP module(s); and/or
(2) Making changes to the ERP software properties, such as configuration files, reports, table, screen, interfaces, program documentation, documentation and so forth in the client’s ERP module(s) installed.

A request is said to have no impact if its implementation does not result in any changes at all to the client’s ERP system’s functionality and/or ERP software properties.

4.4.4. An illustration using GA as an example

Fig. 3 illustrates how different categories of GA’s maintenance requests (presented in Table 7) appear on a 3-D illustration of the mentioned three dimensions.

Area-A in Fig. 3: Maintenance categories (1) GA-enhancement and (2) user-enhancement (refer to Table 7) are more likely located around Area-A, in Table 7 because:

- They are initiated by an ERP-client, i.e. GA’s system users and IT-staff.
- Some of them, which enhance business processes/practices, system integration and existing functionality, or add new functionality to the system, will bring business-benefit(s) to GA and system users.
- These activities (obviously) have impact on GA’s (the client’s) ERP module(s), since they are requested to make changes to the system.

Area-B in Fig. 3: On the other hand, category (3) corrective (from Table 7) is more likely to be found in Area-B, in Table 7 because it is an internal originated request. It does not bring any additional business benefit but has impact on GA’s ERP system installed module(s).
Area-C in Fig. 3: Maintenance category (4) master-data-change (from Table 7)—pertinent to updating the master data file and documentation falls into Area-C of Fig. 3 as it is introduced internally. This activity is mandatory for adapting to a changed environment; however, it is unlikely to increase any business benefit to GA. This category has an impact on the installed module(s) as it involves making changes to the software properties such as tables, data fields and data definition.

Area-D in Fig. 3: Category (5) user-support—associated with the system training, behavior, operating, and functioning (refer to Table 7) is likely to fall in Area-D, in Fig. 3 since it is internally originated, but does not bring any business benefit since they do not improve or change the installed module(s). Therefore, it also does not have any impact on GA installed ERP module(s).

Area-E in Fig. 3: Categories (6(i)) patch-maintenance (for bug fixes) and (6(ii)) patch-maintenance (for adaptation to the client’s changing environment, refer to Table 7) are likely to fall in Area-E, in Fig. 3.

- They are introduced by the vendor.
- These requests are for correcting bugs, and/or adapting the system to a changed environment, where GA is operating. However, they would not contribute to any business benefit (although if there are not implemented, result in barriers to business benefits).
- They are perceived to have high impact on GA’s installed module(s) if the patches are relevant to its module(s).

Area-F in Fig. 3: The maintenance categories (6(iii)) patch-maintenance (for standard code, and (7(i)) technical upgrade (from Table 7) are more likely to fall in Area-F, in Fig. 3.

- They are vendor initiated.
- They are likely to bring some business-benefit to GA because they keep an existing version a standard version and avoid vendor-support termination.
- They have impacts on GA’s installed ERP module(s) as they would replace some (or all) of the custom code in the client’s installed version with the vendor’s standard code. In this case, reapplying previous enhancements may result, depending on GA’s system users’ requirements.

Area-G in Fig. 3: Maintenance category (7(ii)) functional upgrade (refer to Table 7) is likely located Area-G, in Fig. 3. This is because

- It is a vendor-introduced maintenance activity.
- A functional upgrade (involving substantially improved and new functionality) would allow GA to realize significant business benefits from the new system.
- It is perceived highly relevant to an ERP-client’s system functionality as it provides new and improved functionality required by the client.

4.5. Mapping GA’s ERP maintenance requests onto existing maintenance classifications

With further observations made in the previous section (precisely, the three salient dimensions for characterizing ERP maintenance request) and better under-
standing of GA’s ERP maintenance activities, in this section, GA’s ERP maintenance activities are mapped into the existing in-house software maintenance classifications. The main objective is to investigate whether the existing maintenance classifications are sufficient in the context of ERP. GA’s ERP maintenance activities in Table 7 are compared with Table 3 (Swanson’s software maintenance classification), and with Table 4 (Chapin’s maintenance classification) in Section 2.2.

4.5.1. Results from Swanson’s classification

Each of the GA maintenance request categories is mapped onto Swanson’s classification (refer to Table 3) based on the objectives or intentions of doing the ERP maintenance (by referring to column-2 of Table 7). The objective is to discover which of Swanson’s maintenance-category(s) are relevant to describe GA’s ERP maintenance activities. The results are shown in Table 12.

Note that these mappings are approximations only; they are not one-to-one nor are they exact-matches. For instance, we find that GA-enhancement is under both the adaptive and perfective maintenance category defined in Swanson’s maintenance taxonomy. This does not necessarily imply that GA-enhancement covers all activities described (in adaptive and perfective) for the in-house software environment. For instance, GA-enhancement does not include the effort to improve the maintainability of the R/3 programs (used by GA).

Results in Table 12 indicate that user-support, patches (for standard code), and technical upgrade are not covered by Swanson’s maintenance classification. None of Swanson’s maintenance categories are intended to encompass help desk or user-support nor intended to keep the installed system a vendor supported-version.

4.5.2. Results from Chapin’s classification

The evidence of GA’s maintenance activities (i.e. in column-3 of Table 7) is mapped onto Chapin’s maintenance classification (refer to Table 4), which are defined based on the evidence of the maintenance activities, to identify which of Chapin’s maintenance category(s) are appropriate to explain GA’s ERP maintenance activities. The results are shown in Table 13.

From Table 13, it is noted that, technically, all GA’s ERP maintenance activities could be described by one or more categories of maintenance activities in Chapin’s maintenance classification. However, it is observed that Chapin classification does not take into account an important ERP maintenance characteristic, i.e. the business benefit of doing the maintenance (see the characteristics of some of ERP maintenance requests in Fig. 3). Neither is this factor considered in Swanson’s maintenance classification.

This factor (i.e. business benefit of doing maintenance) is neglected most likely because enhancement in in-house software maintenance is mainly targeted at technical aspect such as enhancing program maintainability, improving the performance and efficiency of programs (Swanson, 1976), operational efficiency (Vessey and Weber, 1983), and system maintainability (Martin and McClure, 1983, p. 85). On the contrary, ERP system is unlike the traditional information system (IS) that is used for transaction processing system and/or administrators’ tool to establish control, as reported in Hammer and Champy (1996), Davenport (2000a,b), Dhillon (2000). ERP system maintenance brings also the business benefits (see Irani and Love, 2001; Murphy and Simon, 2002).

5. The proposed ERP maintenance taxonomy

The analysis so far shows that the existing in-house software maintenance classifications are lacking in the context of ERP. Thus, a new ERP maintenance taxonomy is required. The proposed maintenance taxonomy consists of two levels. The level-1 of this taxonomy helps in determining whether a maintenance request requires the level-2 of benefit categorization. The reason for this two-level maintenance taxonomy is that some ERP maintenance requests bring business benefit but the mandatory maintenance requests do not. Requests that could bring business benefits need to be identified and carefully treated in order to quantify the benefits or opportunities of the requests, which can be subsequently used to justify maintenance decisions. The process involved in classifying an ERP maintenance request in the proposed maintenance taxonomy is demonstrated in Fig. 4.

5.1. Fundamental ERP maintenance category

Level-1 includes classifying a maintenance request based on the three salient dimensions (for characterizing
ERP maintenance requests) identified and discussed in Section 4.4. This is done by determining: (1) who is the maintenance source? (2) why is it important to service the request? and (3) what—whether there is any impact of implementing the request on the installed module(s)? Fig. 5 shows how the nine categories of ERP maintenance requests are classified. The nine categories are enhancement, adaptive, corrective, user-support, functional upgrade, technical upgrade, patch-maintenance-adaptive, patch-maintenance-corrective, and patch-maintenance-standard.

5.2. Client-benefits oriented taxonomy of ERP maintenance

Level-2 involves classifying those requests with business benefit into the respective category(s) of business benefits. It is assumed that all mandatory requests such as all corrective and adaptive requests, and user-support requests do not require business benefit categorization. From Fig. 5, the four categories of ERP maintenance requests that require further benefit categorization are enhancement, functional upgrade, minor enhancement, technical upgrade, and patch-maintenance—for keeping the installed system up to the standard required by the vendor.

Based on literature 4 in Section 2.1, this study proposes five main categories of benefits for ERP maintenance activities. These five categories are competitive advantage, globalization, integrated system, best practice/business processes, and cost reduction. These benefit categories are also partially substantiated by the empirical findings from GA. The categories of integrated system, best practice/business processes, and cost re-

---

X = The maintenance activity falls under the proposed maintenance category.

---

Table 13

<table>
<thead>
<tr>
<th>GA maintenance request</th>
<th>GA’s category of maintenance activities</th>
<th>Chapin’s category of maintenance classification</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Support interface</td>
<td>Documentation</td>
</tr>
<tr>
<td></td>
<td>Training</td>
<td>Refor-</td>
</tr>
<tr>
<td></td>
<td>ConsultiVe</td>
<td>mative</td>
</tr>
<tr>
<td></td>
<td>Evaluative</td>
<td>Update</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Groo-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>mative</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Preven-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>tive</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Performance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Adaptive</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reduc-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>tive</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Correct-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Enhan-</td>
</tr>
<tr>
<td>GA-enhancement</td>
<td></td>
<td>- X X</td>
</tr>
<tr>
<td>User-enhancement</td>
<td></td>
<td>- X X X</td>
</tr>
<tr>
<td>Corrective</td>
<td></td>
<td>X - X</td>
</tr>
<tr>
<td>Master-data-change</td>
<td></td>
<td>- X X X</td>
</tr>
<tr>
<td>User-support</td>
<td></td>
<td>X - X</td>
</tr>
<tr>
<td>Patch (bug fix)</td>
<td></td>
<td>X - X</td>
</tr>
<tr>
<td>Patch (adaptation)</td>
<td></td>
<td>X - X</td>
</tr>
<tr>
<td>Patch (standard code)</td>
<td></td>
<td>X - X</td>
</tr>
<tr>
<td>Technical upgrade</td>
<td></td>
<td>X - X</td>
</tr>
<tr>
<td>Functional upgrade</td>
<td></td>
<td>X - X</td>
</tr>
</tbody>
</table>

---

Fig. 4. The process of identifying client-benefits oriented taxonomy of ERP maintenance.
duction have been mentioned by GA’s senior managers as the benefits for maintaining their ERP system but not the competitive advantage, and globalization categories. This is not surprising due to the nature of a Government organization. The definition and description of the five dimensions of business-benefits are distilled (from the prior studies and trade literature on ERP benefits) and summarized in Table 14. Also shown in Table 14 are the characteristics of ERP that drive these five categories of business benefits.

The viewpoint of a senior manager is assessed in categorizing maintenance requests from the business benefit perspective. This is determined based on the business reason, objective, or intention of doing the maintenance request. Table 15 shows the primary objectives or intentions of the five business-benefit categories; the types

![Table 14](image-url)

<table>
<thead>
<tr>
<th>Category</th>
<th>Distilled definition or description from the literature</th>
<th>Characteristics of ERP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Competitive advantage</td>
<td>Increase and improve the capabilities and power to compete with other competitors</td>
<td>Advanced technologies, best practice, integrated system, worldwide interoperability</td>
</tr>
<tr>
<td>Globalization</td>
<td>Enhance information flow to and from customers, suppliers, and other business partners outside the enterprise in a tightly coupled mode, and flexibility to operate in worldwide market</td>
<td>Unified interface, multi-currency and language, integrated system</td>
</tr>
<tr>
<td>Integrated system</td>
<td>Improve the flow of information through centralized system, better system integration and communication among internal business processes</td>
<td>Cross-functional, a single vendor application</td>
</tr>
<tr>
<td>Best practice/business processes</td>
<td>Improve business processes and practices, and business performance</td>
<td>Standardized processes, integrated system, off-the-shelf software</td>
</tr>
<tr>
<td>Cost reduction</td>
<td>Cost cut in activities related to business administration and processing, and system maintenance, and ensure ongoing system support from the vendor</td>
<td>Real-time, single and centralized database, integrated system, best practice, maintenance support</td>
</tr>
</tbody>
</table>
of changes (maintenance) that could be done to the system; and examples of ERP maintenance categories, which can deliver the categories of benefits proposed in this study. It is argued that, depending on the maintenance objective(s) or intention(s) and scope, a maintenance request is not necessary contributing to a single benefit only. For instance, a request which primary is to provide better integrated system could also trigger, lead to and result in other benefits such as cost reduction in producing and manufacturing a product, facilitating the implementation of a best practice that is not possible before, enhancing the potential for operating in worldwide market, and/or providing the opportunities for new market. This example can be traced using Fig. 6. Fig. 6 provides the snapshot of the possible inter-connectivity among the business benefit categories. In sum, it is important for a manager to first determine the primary objective of a maintenance request in order to precisely

Table 15
Benefit categorization in ERP maintenance request

<table>
<thead>
<tr>
<th>Category</th>
<th>Primary objective or intention</th>
<th>Type of changes (maintenance) that could be done to the system</th>
<th>Potentially delivered by the following ERP maintenance category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Competitive advantage</td>
<td>Increase market share, business opportunity (thus, enhance revenue generation), customer loyalty, customer satisfaction, service quality</td>
<td>Adopt advanced technology/Add new functionality</td>
<td>• Enhancement</td>
</tr>
<tr>
<td>Globalization</td>
<td>Facilitate global access to customer, supplier and business partners, and vice versa</td>
<td>Enhance information flow and system integration with customers, suppliers and business partners</td>
<td>• Functional upgrade</td>
</tr>
<tr>
<td>Integrated system</td>
<td>Improve integration among internal business processes, enhance data management and accuracy of decisions</td>
<td>Improve integration among application modules and processes in the system</td>
<td>• Functional upgrade</td>
</tr>
<tr>
<td>Best practice/business process</td>
<td>Enhance efficiency and effectiveness in business processes and business performance/revenue, and shorten cycle times</td>
<td>Reengineer or redesign processes</td>
<td>• Enhancement</td>
</tr>
<tr>
<td>Cost reduction</td>
<td>Minimize cost (to the company)</td>
<td>Upgrade to supported version Replace custom code with standard code</td>
<td>• Functional upgrade</td>
</tr>
</tbody>
</table>

and unambiguously identified the benefit category(s) that it is contributing. Details of a method of how to quantify the benefit(s) delivered in a maintenance request can be found in (Ng, 2001).

6. Conclusions and future research

The study makes three main contributions. Firstly, it provides a detailed description of activities involved in an ERP maintenance environment. From there, the observations made are as follows: (1) the ERP-using organization does not only maintain internally originated change-requests, but also implements maintenance introduced by the vendor; (2) requests for user-support concerning the ERP system behavior, function and training constitute a main part of ERP maintenance activity; and (3) similar to the in-house software environment, enhancement is the major maintenance activity in the ERP environment, encompassing almost 64% of total change-request effort. Based on these differences and other findings, a clear and concise definition of ERP maintenance is proposed.

Secondly, the literature review and our findings show that ERP maintenance is not an instance of in-house software maintenance. Thus, ultimately, we conclude that existing in-house maintenance taxonomies are not sufficient to capture the characteristics of ERP maintenance activities.

Thirdly, the salient dimensions for characterizing ERP maintenance requests are identified, and an ERP-
specific maintenance taxonomy is developed. The three salient dimensions are the maintenance source, existence of any business-benefits and existence of any impact on the installed module(s). The proposed ERP maintenance taxonomy represents an extension beyond the modern-view of maintenance-activity typology, in two ways: (1) it covers vendor-initiated maintenance activities, and (2) it classifies the relevant maintenance activities based on the benefit-perspective. Although enhancement activities in in-house software may also have significant impact on the way the employing-organizations do business, (to the knowledge of the authors) there is no maintenance taxonomy proposing to classify maintenance categories based on the benefit-perspective. This taxonomy is not only important in order to manage, record and track all maintenance activities in an ERP-client organization, but also to facilitate cost and benefit justification of the maintenance activities and assist practitioner in quantifying user-opportunity cost of not-implementing maintenance requests. The benefit-perspective of the classification ultimately allows ERP-using organizations to (i) prioritize the maintenance requests based on the importance of the benefit to the organization’s business objectives, and (ii) choose the most appropriate version of ERP to upgrade (using the same concept as in point (i)), which can eventually reduce the total maintenance cost in the future, and the total cost of ownership for ERP software.

We acknowledge that the findings in this study are based on a single case study. As a result, the completeness of the proposed ERP maintenance taxonomy, and its generalizability warrant further research and discussion. There is still a need to conduct further field studies from different organizations, module(s) installed, organization size, vendor and hardware infrastructure to validate the reliability of the proposed ERP maintenance taxonomy; and thorough data coding is needed in order to produce a comprehensive list of (all possible) maintenance activities involved in each of the ERP maintenance categories identified in this study. However, we feel that GA is a typical Government organization and analogous to a maintenance department servicing a number of user departments, who aims to maximize benefits such cost reductions, best practice and integrated system from an ERP system. Moreover, other related and more general ERP literature is referenced and incorporated in the proposed ERP taxonomy. Therefore, to some degree some of the findings reported in this paper are generalizable and some are applicable to organizations having the same characteristics (Baskerville and Lee, 1999).

Besides this, for future research, it would be useful to further sub-categorize the main ERP maintenance categories into a more detailed and refined level. For instance, ERP enhancement could be further categorized into enhancement involving: (1) system properties belonged to the vendor, (2) system properties belonged to the ERP-client, (3) functionality that can be implemented within the standard ERP code, (4) functionality that has to be custom-made, (5) different maintenance/tailoring options, and so forth. The main objective of this endeavor is to provide a set of comprehensive and pragmatic guidelines to the ERP maintenance managers so that this can facilitate more accurate estimations of costs and benefits for the maintenance request or activity. Longitudinal study would be required to identify whether maintenance patterns or characteristics exist over a time period such that it would allow the ERP maintenance-managers to make predictions and plan for their future maintenance activities.

Acknowledgements

The authors would like to thank the reviewers for their encouraging and motivating feedbacks.

References


Barnes, M., 1999. Customization of ERP apps requires development skills. Information Week (722), 9A–9D.


Hicks, D.A., Stecke, K.E., 1995. The ERP maze: enterprise resource planning and other production and inventory control software. IIE Solutions 27 (8), 12–16.


Stein, T., 1999a. ERP Overhaul for Lawson—Strategic Ledger Among New Features, InformationWeek, 276/16/11.


Celeste S.P. Ng is a Ph.D. candidate in the Information Systems Management Research Group (ISMRG) at the Queensland University of Technology in Australia. Celeste received her B.Sc. from Campbell University in North Carolina, USA in 1997, and MIT from Queensland University of Technology, Australia in 1998. Her major research interests include ERP maintenance and upgrade decision, ERP maintenance management and methodology, maintenance request scheduling, maintenance performance, benefit realization, and economic modeling. Her email is celesteng@hotmail.com.

Guy G. Gable Professor Guy Gable, Queensland University of Technology, has published widely (e.g. Management Science, Journal of Strategic Information Systems, European Journal of Information Systems, Journal of Software Maintenance and Evolution, Information Systems Frontiers, ICIS) and is on the editorial boards of eight journals. He is Chief Investigator in collaboration with SAP on the ARC funded grant “Cooperative ERP Lifecycle Knowledge Management”. His research interests include: enterprise systems, maintenance management, knowledge management, and research methods. He has a particular interest in knowledge management practices of large consulting companies and in the client-consultant relationship. His 1991 doctoral thesis (Bradford) titled “Consultant Engagement Success Factors” won the inaugural ICIS thesis award.

Taizan Chan is a Lecturer in the Faculty of Information Technology, Queensland University of Technology. He received his PhD in Computing and Information Sciences from the National University of Singapore in 1998. His major research interests include the economics of software maintenance, ERP life-cycle management, and electronic markets. He has published in international conference proceedings such as the International Conference on Information Systems and the International Conference on Software Maintenance as well as in journals, such as IEEE Transactions on Software Engineering and the Journal of Software Maintenance and Evolution: Research and Practice.