JOURNAL OF SOFTWARE MAINTENANCE AND EVOLUTION: RESEARCH AND PRACTICE
J. Softw. Maint. Evol.: Res. Pract. 2001; **13**:431–468 (DOI: 10.1002/smr.241)

Research

A decision framework for enterprise resource planning maintenance and upgrade: A client perspective

Celeste See Pui Ng*,†

Information Systems Management Research Centre, Queensland University of Technology, 4001 QLD, Australia

SUMMARY

Over the last decade, many large organizations have been shifting from developing their own information system (IS) to licensing and installing large software packages known as enterprise resource planning (ERP) systems. These organizations now face the challenge of maintaining these costly systems and many are about to make their first upgrade decisions. The study reported herein aims to address the following research questions from the ERP-client perspective: (1) What are the fundamental factors driving ERP maintenance and upgrade decisions? (2) How do these factors differ from those for traditional custom, in-house software? and (3) Do existing software and hardware replacement models suffice for ERP maintenance and upgrade decision modeling? A single case study method and empirical data analysis were conducted and are presented here. We observe that ERP maintenance and upgrade characteristics are indeed unique in three ways: (1) most organizations maintain and upgrade their ERP systems in order to realize increased business benefits from the systems; (2) a new version upgrade reduces the number of ERP-client-done enhancement modifications; and (3) a new version upgrade potentially reduces future legal change patch (LCP) maintenance distributed by the ERP vendor. It was also found that on average LCP maintenance is almost as costly as *user enhancements*. Based on these findings, we conclude that the existing in-house software and hardware replacement models are insufficient for ERP situations. We propose a preliminary ERP decision model that overcomes the observed insufficiencies. Copyright © 2001 John Wiley & Sons, Ltd.

KEY WORDS: package software; user opportunity cost; software maintenance; software upgrade; decision support; case study

*Correspondence to: Celeste See Pui Ng, Information Systems Management Research Centre, Queensland University of Technology, 4001 QLD, Australia.
†E-mail: celesteng@hotmail.com

1. INTRODUCTION

Over the last decade, many large organizations have been shifting from developing their business application software to licensing and installing large commercial off-the-shelf software known as enterprise resource planning (ERP) systems. ERP systems are fully integrated, enterprise-wide business applications with not only a complete set of traditional modules such as accounting, human resources management, sales and distribution, and manufacturing, but they also provide extensions such as supply chain management, data warehouse, and customer relationship management. AMR, a well-known research and technology analysis center, claims that in the past decade approximately $300 billion has been invested in ERP worldwide [1]. Trade reports indicate that there are tens of thousands of organizations using ERP and millions of licensed users [2].

Many ERP-using organizations now have several years experience of maintaining their ERP systems and are facing their first ERP upgrades. Glass and Vessey [3] found that annual ERP client-initiated maintenance costs on average approximately 25% of the initial ERP investment. AMR Research states that an ERP upgrade costs approximately a further 25–33% of the initial investment [4], see also [5]. ERP maintenance and upgrade are thus costly decisions. The meager trade reports on ERP upgrade policies [6–9] provide mostly anecdotal evidence of practitioners' experiences, pitfalls to avoid, and factors to consider prior to upgrading. However, little has been written on the ERP maintenance and upgrade decisions. In order to bridge this gap, this paper critically reviews the existing literature on ERP maintenance and upgrade decision drivers and synthesizes a decision framework based on the literature and new findings from our case study. This decision framework is important to ERP managers in making well-informed ERP maintenance and upgrade decisions.

In an ERP environment, maintenance activities originate from two main sources, the ERP-client organization and the vendor [10], and they are illustrated in Figure 1. The former includes requests for enhancement, bug fixes, ongoing system support, and helpdesk. The latter comprises support package or patch and upgrade, which are distributed by the vendor but implemented by an ERP client on its ERP system. A support package or patch contains corrections and further adjustments due to legal changes, and minor corrections for errors in the repository or data dictionary enhancements for an installed version. On the other hand, the upgrade version contains substantial enhancements and new functionality.

The size of an ERP upgrade could be classified based on two main dimensions: upgrade-version and upgrade-scope. Basically, there are two categories of upgrade-version: minor and major upgrade. A minor upgrade is described as an upgrade involving the same version series such as from 3.1H to 3.1I or 4.0B to 4.6C whereas a major upgrade is a migration to a different series, for example from three series to four series or higher. Upgrade-scope dimension consists of technical upgrade and functional upgrade. A technical upgrade is a 'like to like functionality replacement'. On the other hand, a functional upgrade delivers major business improvements and enhancement benefits. However, a detailed discussion on this distinction is not the focus of this paper.

In this study, ERP maintenance and upgrade (or simply MU) decisions involve deciding the amount of ERP client-initiated maintenance-requests, vendor-introduced patches to implement and/or the timing to upgrade an installed ERP system with a new and readily available version. MU decisions will be used together because they are inextricably inter-related, as the upgrade can be postponed by continuing to maintain the existing system. The decision framework proposed is limited to judgement

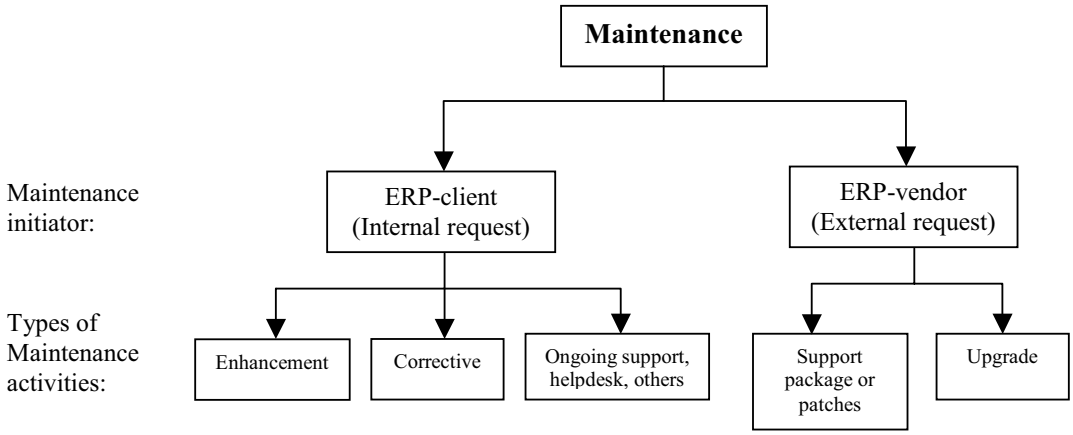

Figure 1. ERP maintenance activities.

based on cost and benefit perspectives. The discussion on ERP MU decisions in this paper focuses mainly on the client perspective. Although vendors too have ERP maintenance and upgrade problems, this is not the main concern of this paper.

The organization of this paper is as follows. In Section 2, background information on factors that influence ERP MU decisions such as ERP maintenance, new version upgrades, and benefit realization are presented. Section 3 describes the case study, data collection, and data analysis. In Section 4, findings from the case study are compared with the fundamental ERP MU decision factors synthesized from the literature; and new findings from the empirical evidence are discussed. Following this (in Section 5), these factors (or characteristics driving ERP MU decisions) are evaluated by considering the characteristics considered in the existing in-house software and hardware replacement models. In light of the differences identified, an ERP MU decision framework is proposed in Section 6. A numerical example, illustrating how the proposed model is to work, is provided. In the last section, conclusions and directions for future research are provided.

2. LITERATURE REVIEW

2.1. ERP maintenance

It is argued in the trade press that ERP vendors drive much of their clients' ERP maintenance and upgrade activities. (This differs from the traditional in-house software in which user-organization has full control over all the maintenance activities.) Vendors provide two main types of maintenance: support packages or legal change patches (LCPs) to the installed version, and new versions of the ERP system for upgrades. Both support package and new version upgrades are supplied by the vendor and implemented by the client, with/without assistance from external consultants, in order to fix bugs,

enhance existing functionality, and/or add new functionality to the installed ERP. The main differences between support package/LCPs and new version upgrades are: (i) LCPs have relatively smaller impact on an installed ERP code whereas new version upgrades will overwrite an installed version with a new one; (ii) LCPs are introduced more frequently than new version upgrades; (iii) while LCPs are mandatory (in order to keep an installed version up to date), new version upgrades are optional as long as an installed version is still supported by the vendor; (iv) a new version upgrade is much more complex, expensive, and lengthy to implement than a LCP implementation; and (v) functional upgrade provides extensive new functionality than LCP maintenance.

ERP systems and industry solutions are generic solutions. More often than not, modifications which refer to the changes made to the standard ERP code and/or development of custom code are required during the implementation of an ERP system [11,12]. The number of modifications is dependent on the degree of fit between the ERP system and the organization's existing (or desired) business processes, and the willingness of the organization to adapt its way of doing business to the package [13]. Anecdotal evidence suggests that the more modifications are done to the system at implementation time, the higher will be the billable hours or cost per LCP maintenance job [6]. This is because implementing a LCP maintenance will overwrite some of the custom code and previous modifications. Therefore, more effort is required to conduct impact analysis to verify the effects of each LCP (or support package) on each of the previous modifications [8]. Sometimes, re-application and re-testing of some of the previous modifications are required if they have been overwritten by a LCP. In fact, this modification does not end after the implementation phase. The modification done during the system implementation is the first maintenance project on an ERP system. These modification requests could also be introduced in the post-implementation or maintenance phase. They are called enhancement modifications in this study. Hence, modification should be a function of what is done during system implementation and also an accumulation of the related activities in the post-implementation phase.

2.2. New version upgrade

Unlike in-house software replacement, which requires much software development effort and extensive details of what and how the software system functions [14], the ERP-client could upgrade the installed version with a version readily available on the market. Upgrading an installed version to a new version is part of ERP post-implementation activities. In contrast to LCP maintenance, organizations typically upgrade to a new ERP version in order to realize the benefits of substantial new functionality [15], and new technologies or business opportunities (such as enterprise portal, data warehouse, and e-commerce) [16]. On the other hand, sometimes organizations feel compelled to upgrade, as the vendor withdraws support for old versions. ERP vendors impose a fixed time period for maintenance support for each version of the system [8,17,18]. A supported version of an ERP system is eligible for helpdesk support, and LCP support. After the support window is reached, ERP-clients will be entirely on their own in maintaining their systems. (However, in some cases, the vendors offer a service of monitoring the unsupported systems for their clients but this support is very limited and incurs a fee.)

Nevertheless, most of the trade press has cited that cost is prohibitive in considering an upgrade to an ERP system [4]. The upgrade cost could be as high as 25–33% of the initial ERP implementation [5]. Similar to the initial ERP implementation and acquisition cost, upgrade cost (including technical and functional upgrade) consists of the software cost, hardware cost, user training cost, consultancy fees,

and the upgrade implementation cost. A new version of an ERP system, which has more functionality, flexibility, and extensibility, will generally cost more to upgrade. It is perceived that an organization is more likely to upgrade to the latest version of an ERP system. Some of the advantages for upgrading to the latest version of an ERP system are: (i) a minimal number of future upgrades; (ii) a longer period of maintenance support from the vendor; and (iii) the gaining of a competitive advantage through early utilization of new technologies. Upgrading to a new version of an ERP system often entails some purchase of additional and more powerful hardware [19], or new hardware that is compatible with the new system.

User training costs, associated with re-training the ERP system's users, are driven by changes in the user interface and/or new functionality in a new system [5]. An increase in costs is anticipated if these features are dramatically different from an existing system. An ERP upgrade is a complex and large project requiring a wide range of knowledge and expertise, for example, in the area of software functionality, system configuration, system integration, business processes, project management, change management, and other technical aspects of new software and hardware. This knowledge and expertise is necessary for a successful upgrade of an ERP system [9,20,21]. However, not all of this knowledge and expertise is available internally, particularly for the first ERP upgrade project. Hence, most of the time, external consultants are required.

On the other hand, upgrade implementation costs involve the cost of data conversion, system analysis, system integration and testing, and post-implementation turmoil [22,23]. Post-implementation turmoil here refers to the potential system disruptions (e.g. a dip in operational efficiency and effectiveness) [24] or downtime in relation to the implementation of an upgrade or installation of a new system. It includes change management [7,25,26]. In general, upgrade implementation costs are driven by the complexity of the upgrade project, which is determined by the number of modules to be implemented and the number of business units involved.

2.3. Benefit realization

The third driver influencing an ERP MU decision is benefit realization from an installed ERP system, which is also known as ERP's *second wave*. ERP is a real-time and centralized system, and supports complete data visibility for all levels of organizational management. Hence, ERP facilitates corporate and strategic decision making [27–29], enhances information flow among internal processes, and improves information flow to and from customers, suppliers, and business partners. In contrast, most of the traditional in-house applications are designed to support an individual operational or tactical area. Therefore, integration among these individual applications and business processes is not only loosely coupled but is also restrictive, if not prohibiting, to benefit realization from the disparate systems.

Benefit realization in this study means achieving full capabilities and benefits (or business values) from ERP systems through continuous systems maintenance. Besides internal maintenance requests from the system users, ERP-clients also receive maintenance 'requests' (as mentioned earlier in Section 2.1) from the vendor. An ERP vendor continuously updates, improves, and develops the ERP software. These business improvements or enhancements and latest technologies are delivered to clients in the forms of support packages or LCPs, and new version upgrades. In order to realize benefits from ERP systems, ERP-employing organizations are required to incorporate support packages, and upgrade to the newest version of an ERP system. (This is also important to the vendor in order to contain its maintenance costs and focus its development efforts on one or a few versions.) These

Table I. Examples of benefit realization from ERP systems.

Benefit categories	Quantification examples
Best business practice	Amherst Corporate Computer Sales and Solutions, an online seller of IT products and services, has experienced a 20% (or equivalent to $71 000) productivity increase in terms of sales per employee through best business practices in the ERP system [38]
Competitive advantage	Autodesk, a leading maker of computer-aided design software, uses an enterprise system and is now able to deliver 98% of its orders to customers within 24 hours as compared to two weeks with the legacy system [40]. This makes Autodesk competitive among its competitors
	Fujitsu Microelectronics achieved a reduction in the cycle time for filling orders from 18 days to a day and a half [40]
Globalization	In the petrochemical industry, an enterprise system that facilitates information flow through the supply chain and electronic information sharing is of paramount importance for survival [40]
Integrated system	Owens Corning, an international goods supplier, has had its spare-parts inventory reduced by 50% (an estimated saving of $65 million) by its integrated enterprise system [40], which links financial, supply chain, and order management processes seamlessly
	Integrated systems are utilized in high-tech companies to inject more discipline into an organization, exert more management control, and impose more uniform business processes [40]
Operational cost reduction	Elf Atochem North America, a $2 billion regional chemical subsidiary, was expecting reductions in annual operating costs of tens of millions of dollars from using its enterprise system [40]

maintenance activities could be beneficial to ERP-clients for operational cost reduction, and generation of business opportunities [29–34]. It is reported that most organizations implement ERP systems because of some fundamental benefits that these organizations could realize from the enterprise systems. These benefits include best business practices [35–38], competitive advantage [39–42], globalization [43–45], integrated systems [46–48], and operational cost reduction [48–50]. It is assumed that these benefits will remain as the drivers for continuous ERP MU activities.

Best business practices are associated with improvements in business processes and business performance such as cost, quality, service, speed, and productivity, whereas competitive advantage relates to expansion of the capabilities and power to compete with other competitors. On the other hand, globalization is being engaged in activities which enhance the flow of information to and from customers, suppliers, and other business partners outside the enterprise in a tightly coupled mode. An integrated system benefits from an enhanced flow of information, better-informed corporate decision-making processes, and seamless system integration and communication among internal business processes. Operational cost reduction is concerned with minimal system operational and maintenance costs.

Table I provides some illustrations of how these benefits have been realized and quantified by some ERP-clients.

3. RESEARCH METHOD

3.1. Case study

In order to gain first-hand insights into ERP maintenance activities and upgrade issues, a case study was conducted [51–53]. The firm involved was a government agency (GA) in Australia. This organization was chosen for several reasons: (1) GA's comprehensive ERP maintenance records; (2) the Information System Management Research Centre (ISMRC) support from GA and involvement with State Government in collaborative research; and (3) the ISMRC's ongoing, existing collaborative projects with GA. GA was set up in 1996 and it is a corporate services provider to other Queensland government departments, all of whom use the SAP R/3 ERP system. These Queensland government departments are clients to GA. (SAP, an abbreviation for Systems, Applications, and Products, is a Germany company and one of the leading ERP vendors; the R/3 system is a client–server version of SAP's software.)

In 1994, the Queensland government planned to replace the non-year-2000 compliant financial and human resources management systems. The SAP R/3 financial (FI) and human resources (HR) modules were selected. The FI module was implemented in November 1998 and HR in late April 1999. The initial version was 3.1H, and GA is upgrading to version 4.6C.

3.2. Data collection

There were five main sources of data collection for the purpose of this study: GA's maintenance database, a series of semi-structured interviews, maintenance and upgrade documentations, GA's LCP support database, and GA's SAP system modifications database. GA's maintenance database sufficiently records most of the necessary data pertaining to maintenance (i.e. change requests). This includes details on the request type, a description of the problem(s) associated with the request, the time spent on servicing the request, and the respective maintainer(s) and system analyst(s) involved in the maintenance request. It also contains information about the LCP maintenance projects that have been done. The semi-structured interviews were tape-recorded and then transcribed verbatim for context validation by the interviewees. These interviews were conducted with the General Manager, Systems Development Manager, Systems Operations Manager, and System/Business Analyst. Some of the interview questions were: What are the activities involved in the ERP maintenance environment? How does the firm manage its maintenance activities? What are the benefits and costs associated with maintenance activities and upgrade? What are the factors driving ERP maintenance and upgrade decisions, and how could these decisions be optimized? Relevant documents such as (i) GA's Upgrade Business Case, detailing why GA upgrades to a new version, and all the cost components related to its upgrade exercise, and (ii) SAP R/3 enhancement-maintenance documentation (obtained from the SAP website), describing different options available for implementing enhancement to the SAP system, were consulted. GA's LCP support database, which is provided by the SAP vendor, contains information about all the LCPs for different versions of the SAP software. On the other hand, the SAP system modifications database consists of details of GA's existing enhancement modifications, which were done to its installed version of the SAP system, and information about those enhancements that would be required and those that would not be needed after the new upgrade to version 4.6C.

3.3. Data analysis

GA's maintenance database was examined to study maintenance activities done by GA, and determine factors influencing the ERP/SAP maintenance effort. Documentations on SAP maintenance, obtained from the SAP website, were consulted to investigate the options available for carrying out enhancement to an SAP system. Findings from the interview transcripts and evidence from the Upgrade Business Case were utilized to identify the factors driving ERP MU decisions. GA's LCP support database was explored to study the LCPs, introduced in a series of SAP software versions, distributed by the vendor. On the other hand, GA's SAP system modifications database was analyzed to research the impact of an upgrade on its existing or *previous* enhancement modifications.

Descriptive data analysis was conducted on: (i) the maintenance request type, maintenance-project complexity, and LCP-project maintenance effort, from the maintenance database; (ii) the number of enhancement modifications required before and after the upgrade, from the modification database; and (iii) the number of LCPs in each version and the number of notes in each LCP, from the data collected from GA's LCP support database. Non-parametric techniques such as the Kruskal–Wallis test (for several independent samples), and the Mann–Whitney U test (for two independent samples) were used to examine possible differences in the mean between two or more groups [54].

The empirical results were applied to extend the existing knowledge of the ERP MU, and propose a better representation or model of ERP MU decisions. For instance, a reduction in the number of previous enhancement modifications after an upgrade is incorporated into the decision framework proposed later on. The interview transcripts and Upgrade Business Case were used to determine the reliability of the ERP MU framework synthesized from the existing literature. For example, it is found that the two sources are consistent in that benefit realization is one of the main drivers for ERP MU decisions.

Note that the following discussion is focussed on GA's ERP/SAP change requests, which were initiated from November 1998 to June 2000. Although maintenance requests related to the system usage, user training, and helpdesk is an interesting topic (for more details see [10]), it is not explored in this paper.

4. FINDINGS

4.1. ERP-client internal maintenance activities and effort determinant

Our study shows that maintenance requests from the ERP-client itself make up a major part of the overall ERP maintenance activities. In this study, it is observed that GA manages ERP maintenance activities that come from two main (internal) sources: (i) GA, the corporate service provider, and (ii) system users, GA's client organizations. GA-initiated maintenance activities are mainly targeted at improving the quality of ongoing support to its clients and containing its operational cost (i.e. called *GA enhancement*). System users' requests are aimed at correcting faults in the system and ongoing system support (i.e. *corrective* and *master-data change*), and requests for new functionality (*user enhancement*), which, while important to the business and operating environments, are not currently available from the installed version. An observation made from GA is that in addressing its system users' maintenance requests, GA usually tries to fulfill the requests within the standard ERP code. All

Table II. GA and system users: maintenance effort by request type.

Source	Request type	Effort (h)	Effort (%)	Request No.	Request (%)	h/request
GA	GA enhancement	3857	42	228	38	17
System user	User enhancement	2031	22	43	7	47
	Corrective	2585	28	223	38	12
	Master-data change	695	8	99	17	7
Total		9168	100	593	100	15

corrective maintenance, except that associated with the modified SAP code, will be reported back to SAP and the relevant code or LCP (from the vendor) will then be applied.

Note that there are a number of options available in implementing enhancement requests [13,55]: (i) configuration by setting of the system parameters via SAP's own interface; (ii) a bolt-on, a third-party package designed to work with the ERP system; (iii) enhancement to the ABAP/4 Dictionary elements such as table append, text enhancement, and field enhancement; (iv) add-on functionality using SAP customer exits; (v) the creation of customer-specific objects, for instance screens, and reports; (vi) workflow programming to create non-standard workflows; (vii) ERP programming, writing additional applications using the vendor's ERP programming language, without changing the standard code; (viii) interface development to interface the ERP system with legacy systems; and (ix) modification of the SAP standard objects. Customer exit is the SAP R/3 Enhancement Concept that allows clients to add their own functionality to SAP without modifying the standard code. There are four basic types of customer exits: menu exits, screen exits, function module exits, and keyword exits. Brehm *et al.* [13] argue that enhancement-maintenance efforts depend on the type of these tailoring options. Note that option (i) is generally more favorable than options (ii)–(ix) because it will not be overwritten and affected by the SAP standard code during an upgrade or the implementation of a support package (or LCP). Thus, unlike other options, no ongoing maintenance effort is required for enhancement option (i).

While LCP maintenance efforts are claimed to be dependent on the number of enhancement modifications, this study investigates and provides empirical analysis of the factors which affect the amount of effort required to complete the ERP-client internal maintenance activities/requests. (Note that the ERP-client maintenance activities in GA's case include those initiated by itself and its own clients.)

4.1.1. Maintenance request type

Effort analysis of the completed maintenance requests (a total of 593 valid data points) shows that *user enhancement* maintenance requests, which are initiated by GA's clients, are the most demanding (i.e. on time and effort). On average, it takes 47 (maintainer and analyst) hours to service a user-enhancement maintenance request (refer to Table II). In general, far more effort is required to implement a *user*

Table III. The Kruskal–Wallis test on four categories of request type.

	Maintenance effort (h)
Chi-square	56.648
df	3
Asymp. Sig.	0.000

enhancement request than a *GA enhancement* request (i.e. 17 hours per request). That is, on average, the implementation of a *GA enhancement* request consumes only 36% of effort required in servicing a user-enhancement request. This is explained as follows. *GA enhancement* is usually made around the standard SAP code whereas most of the *user enhancement* requests involve developing custom code or making changes to the standard SAP code.

By comparison, the average amount of effort to service a *corrective* maintenance request (12 h/request) and *master-data change* (7 h/request) is relatively less than the average effort required in a *GA enhancement* maintenance request. This is related to the actual task involved in *corrective* and *master-data change* maintenance. *Corrective* maintenance involves activities such as correcting page formats, reports, error messages, and user interfaces. Depending on the source of the bug or error, if it originates from the standard SAP code, usually SAP's patches will be applied. However, for bugs found in one of the customizations or modifications, GA will resolve the bugs for the users. *Master-data change* maintenance comprises simpler tasks such as updating the master file and database.

An empirical test, the Kruskal–Wallis test [54], was conducted to determine if the means of the request types were statistically different. The results in Table III indicate that significant differences did exist in the maintenance effort across the four maintenance request types ($p < 0.01$). Hence, we can say with some confidence that the maintenance request type can be used to predict the amount of effort required for completion.

4.1.2. Maintenance complexity

The scope of a maintenance project is used as a surrogate for maintenance complexity in this study. A maintenance project is defined as *small* (by GA) if not more than one staff member is required in order to accomplish the request (e.g. requests related to security and system configuration). On the other hand, a request which requires more than one staff member for maintenance tasks such as system analysis and programming is classified as a *large* project. (Due to restricted details of these data, another intermediate cluster of maintenance complexity is not possible in our data analysis.) It is observed that more than one-third of the small scope maintenance projects are involved in the request type of *master-data change* (i.e. work related to ongoing system support). On the other hand, more than half of the large scope projects are associated with *user enhancement*. Analysis on maintenance complexity (for a total of 564 valid data points) shows that small scope maintenance projects take,

Table IV. GA and system users: the maintenance effort of two categories of maintenance scope.

	Small	Large	Total
Sum (h)	3055	5658	8713
Mean (h)	10	21	15
Standard deviation (h)	19	47	36
N (h)	297	267	564
Percentage of total request	53	47	100
Percentage of total effort	35	65	100

Table V. The Mann–Whitney test on maintenance complexity.

	Maintenance effort (h)
Mann–Whitney U	31066.500
Wilcoxon W	75319.500
Z	−4.442
Asymp. Sig. (2-tailed)	0.000[a]

[a] Similarly, this result is also significant with the Kruskal–Wallis test: chi-squared value = 19.736, $p < 0.01$.

on average, 10 h to complete as compared to 21 h for large scope ones (see Table IV). However, the standard deviation for the former is 19 h and that for the latter is 47 h. We note that other factors such as the effectiveness of project team communications, familiarity of the maintainer(s) with the application system, and expertise and programming knowledge of the maintainer(s) in the related area may have some impact on the overall maintenance effort.

The output from the Mann–Whitney test [54] on maintenance complexity indicates that the result was significant ($p < 0.01$), thus significant differences in maintenance effort exist between small and large scope maintenance projects (see Table V). Therefore, we are confident in saying that the scope of a maintenance project can be used to determine the amount of maintenance effort required.

4.2. LCP maintenance

Data analysis on GA's two large LCP maintenance projects shows that the average effort per LCP is 46 h (see Table VI). The average number of hours per LCP for the first project (57 h/LCP) is observed to be much larger than the second (10 h/LCP). This could be explained by the following factors: (i) familiarities with the LCP maintenance processes and procedures, the average effort required per

Table VI. LCP maintenance effort.

LCP project	No. of LCPs	No. of notes	Notes/LCP	Effort (h)	h/note	h/LCP
Project-1[a] (LCP # 29-62)	34	4406	130	1918	0.44	57
Project-2 (LCP # 63-72)	10	1212	121	99	0.08	10
Total	44[b]	5618	128	2017	0.36	46

[a] The project for LCP # 1-28 had been implemented earlier but was not recorded in the maintenance database.
[b] Does not necessarily represent the exact number of LCPs introduced by the vendor during the study period.

LCP drops significantly after the first LCP project; (ii) the batch size of 34 may not be as productive as a batch size of 10, 57 h/LCP versus 10 h/LCP; and (iii) the number and characteristics of the modules involved in an LCP may have an impact on the total effort required for maintenance. Based on our data in Table VI, the number of notes per LCP (in this case) does not seem to be a critical factor influencing the LCP maintenance effort, as they are almost the same in both projects and yet the average number of hours per LCP in these projects are considerably different. Note that a 'note' is either a bug fix or an enhancement in an LCP. However, the characteristics of these notes are worth further investigation. More empirical data are required in order to validate the significance of these factors.

LCP maintenance accounts for approximately 18% (or 2017 h) of GA's total ERP maintenance effort. GA's General Manager emphasized that if they were to implement LCPs each time they arrived, as many do, instead of batching them, this activity would take up 80% of their maintenance effort or three times the effort now expended. He suggested that how one manages LCPs can have a huge impact on the costs of maintenance; a management strategy might be to batch enter these periodically, e.g. monthly or quarterly, or to monitor LCPs for cumulative benefits and wait until the perceived benefits justify the effort (as does GA). Consistent with the literature, interviews with GA's Systems Development Manager confirmed that the LCP maintenance effort is driven by the number of enhancement modifications done to the standard ERP code.

4.3. New version upgrade

Interviews with GA's senior managers and a review of the Upgrade Business Case show that upgrade implementation costs also depend on the version gap or migration path between an existing and a new upgrade version. For instance, upgrade from 3.1H to 3.1I, which are both in the same series, would be relatively cheaper and less complex than from a three series to a four series. The latter is more costly because of the changes in system structure and architecture, and also input screens. Following the changes in input screens completely new supporting documentation and extensive re-training of staff are required. Consistent with the trade press, upgrade cost is a major issue and is prohibitive for GA when considering its ERP system upgrade.

Table VII. A comparison of the total number of enhancement modifications before and after a new version upgrade.

	Before upgrade (from implementation date until the system development is frozen)	Immediately after the upgrade
Number of previous enhancement modifications	730	498

4.3.1. Impact on previous enhancement modifications

GA is an example of an ERP-client that has upgraded its ERP system because of the withdrawal of the vendor's support for its current version of 3.1H. An upgrade process will reinstate the installed ERP system by the standard/*vanilla* ERP version of the new version. However, the previous enhancement modifications (including both *GA enhancement* and *user enhancement*) that were set by using system parameters, customer objects, or customer exits, which adhered to SAP strict naming conventions, will not be affected by the upgrade process (nor LCP implementation). In contrast, organization-specific enhancements involving modifications of SAP code will be overwritten. Basic steps involved in an upgrade (or LCP implementation) to identify its impact on *previous* enhancement modifications are: (i) analyzing if the previous developments (e.g. those built using system parameters, customer exits, or customer objects) *are working properly* in the light of the changes to the SAP standard code or objects; and (ii) investigating if all the previous enhancement modifications, which were developed by using the enhancement options from (ii)–(ix) discussed in Section 4.1, *are affected or overwritten*. However, this is usually signaled in the form of warning/error messages to inform the client-organization of these changes. Depending on the new functionality in the new version, some of the overwritten enhancement modifications may need to be re-applied. Re-testing will begin after re-entering the previous enhancement modifications. Therefore, the upgrade costs also depend on this endeavor.

This study finds an interesting result concerning the impact of an upgrade on previous enhancement modifications. Analysis of the number of enhancements done on the installed version of the SAP system, from the system-implementation date until the system development is frozen for upgrade, was conducted for GA by a consulting firm. Based on the consulting firm's investigation of GA's installed version 3.1H, and the new capabilities and functionality in version 4.6C, it was found that after the upgrade the number of enhancement modifications decreased by one-third of the original total (see Table VII). Note that these enhancement modifications have an impact on the total effort required to upgrade the SAP system (or to implement an LCP maintenance project in the future). These enhancement modifications could be one of the enhancement options from (ii)–(ix), discussed earlier in Section 4.1.

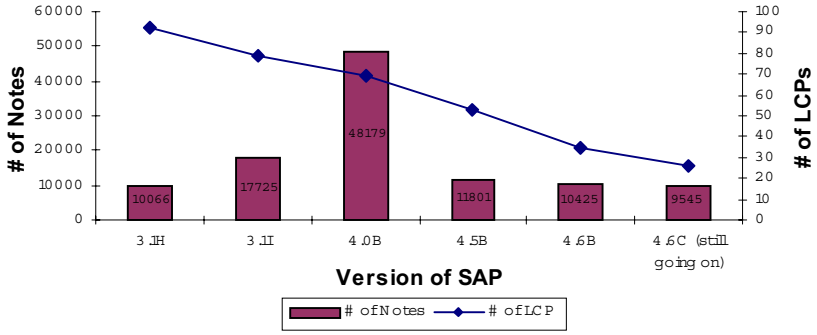

Figure 2. SAP versions with the respective number of LCPs and notes.

The existing literature (see [6,8]) claims that support package or patch maintenance effort is driven by the number of enhancement modifications. In the light of this, together with the finding of a decrease in the total number of enhancement modifications after upgrade, it is expected that maintenance costs and effort for a new upgrade version would be reduced as compared to continuing to maintain an existing installed version. Thus, this could be one of the main advantages which an ERP manager should look for in considering an upgrade option. This decrease is mainly because these enhancement modifications are either now available in the new version or some of them are no longer required. Therefore, this benefit factor should be considered as one of the advantages when making an ERP upgrade decision.

4.3.2. Impact on future LCP maintenance

Analysis of the number of notes, where each note represents a fix or an enhancement, introduced in each of the LCPs initiated in the SAP R/3 versions 3.1H, 3.1I, 4.0B, 4.5B, 4.6B, and 4.6C (up to the end of June 2001 only), shows that the total number of notes and LCPs decreases from an older version to a newer version (Figure 2). It is found that, on average, the number of notes decreases by about 44% (from one version to another) for the last three versions (from 4.0B to 4.6B). On the other hand, the number of LCPs, on average, decreases by about 21% from an older version to a later version. Hence, with this incentive it is expected that ERP-employing organizations would be able to set off this cost reduction (besides the cost component mentioned in previous section) against the upgrade cost in order to better justify their upgrade decisions (instead of delaying the upgrade decision, which also leads to delays in benefit realization and higher maintenance cost for the installed version).

It is noted that factors such as the characteristics of each version, the number of users associated with each version, and external technology changes are also worth further investigation in order to refine our findings. For example, a larger number of users will lead to more frequent usage of the system. When this happens more information about the system's capabilities, functionality, and limitations will

emerge and be reported to the vendor. Thus, more enhancements and bug fixes (or LCP) may occur. On the other hand, as the system matures and stabilizes, fewer LCPs or notes are expected.

4.4. Benefit realization

The existing trade reports argue that maintenance and upgrades are fundamentally driven by benefit realization from the ERP system. Consistent with these reports, GA maintains its existing system and upgrades to a new version of ERP mainly with the aim of realizing benefits from the system; essentially, it looks for operational cost reduction, integration between financial and human resources systems, and the adoption of best business practices. With benefit realization in mind, Ng *et al.* [10] propose a framework to classify the ERP maintenance activities based on the ERP-client's benefit perspectives. In the proposed framework, the authors adopt a two-step process in classifying ERP maintenance activities. The first step involves an initial categorization using Chapin's maintenance activities taxonomy [56] to distinguish enhancement maintenance from the mandatory maintenance activities (such as corrective and ongoing maintenance support); the second step suggests a further classification of enhancement maintenance based on five primary categories of benefit, which are identified as 'integrated system', 'best business practice', 'competitive advantage', 'globalization', and 'operational cost reduction' (see Section 2.3 for details). These are the prevailing benefits of ERP systems to the clients. Recognizing this benefit component is crucial in order to justify costs and benefits for the dominant and expensive enhancement maintenance on ERP systems.

While continual maintenance and regular upgrades may allow organizations to achieve some of the mentioned benefits (although sometimes at high prices), delaying these requests (regardless of whether they are enhancement, corrective or LCP) and upgrades will incur user opportunity cost to the organizations (because of foregoing an earlier benefit realization from the requests and upgrades) (see also Nellemann [57] in his discussion of opportunity cost associated with 'doing nothing'). Thus, ERP-using organizations should consider this factor as one of the tradeoffs in justifying MU decisions.

4.4.1. User opportunity cost quantification: A simple method

While Table I in Section 2.3 shows some examples of benefit realization in some organizations from their ERP systems after the maintenance or upgrade was done, some ERP clients may pay very little attention to quantifying user opportunity costs of not implementing the maintenance requests and upgrades. Omitting this quantification step, an organization could be uncertain if it has made an accurate or a *best* decision on maintenance and upgrade. Hence, in this section, an illustration of a simple method to quantify user opportunity cost, which is basically translated from the unrealized benefit(s), is provided.

It is assumed that all maintenance requests and new version upgrades can be mapped onto one or more of the benefit categories identified in the previous section. This is particularly true with a new version upgrade as it introduces a substantial number of new enhancements and additional functionality. This assumption is also reasonable because ERP/SAP clients are informed of the objectives and benefits of each LCP and upgrade version once they are made available to them. In quantifying user opportunity cost, we propose a two-step 'weighting' approach. This is done by first 'weighting' the benefit categories based on their degree of importance or criticality to an ERP-client's business objectives, using a scale of 1 to 3, where '1' indicates least important and '3' is the most

important. For an example, 'operational cost reduction' is highly important to GA's business mission, hence its importance level is assigned to '3'. On the other hand, 'integrated system' and 'best business practices' have intermediate impact on GA's business mission. Therefore, these benefit categories are given value '2'. However, due to the nature and background of GA, 'competitive position' and 'globalization' are the least important. The objective of this step is to allow strategic evaluation to be incorporated in the user opportunity cost quantification (see also [58]).

The second step involves identifying the benefit(s) delivered by a request and quantifying them (the unrealized benefits) in terms of dollar values, such as those found in the literature cited in Section 2.3. Hence, we define the amount of user opportunity cost, for an unfulfilled request, as

$$\text{Opportunity cost} = \sum_{j=1}^{5} W_j x_j \tag{1}$$

The subscript j, taking values from 1 to 5, represents one of the five benefit categories. W_j denotes a degree of importance/criticality of benefit j to an ERP-client's business mission, and x_j indicates the unrealized benefit value of the request evaluated under benefit j. The judgement of which benefit categories a request falls under and its value is assessed by senior managers and based on the objective(s) of a request.

4.4.2. *Benefit realization quantification: examples*

For illustration purposes, assume that there are three (non-mandatory) maintenance requests received by GA. Note that the requests' objectives and benefit descriptions are taken from the benefit realization instances published in the Queensland Government Financial Management (QGFMS) Benefit Realization Guidelines [59] and are as illustrated in Table VIII. Hypothetical benefit value quantifications for these benefits are also given in this table. Maintenance request 1 is perceived to provide the benefit of 'operational cost reduction' only, and the monetary value for this benefit is computed to be $25 050. On the other hand, maintenance request 2 is found to deliver 'competitive advantage' and 'globalization' benefits only. Their monetary values are evaluated as $10 000 and $1500, respectively. Maintenance request 3 contributes to 'integrated system' and 'best business practices' benefits. They are assessed to be $10 000 and $3000, respectively.

It is noted that while some benefits are tangible and quantifiable, others are intangible and difficult to measure. However, in assessing intangible benefits, the quantification technique proposed by Hares and Royle [60] is adopted and suggested.

4.4.3. *User opportunity cost quantification: an example*

In contrast to the above, if these three requests are delayed, the benefit values computed in Table VIII become the unrealized benefit values. Using the data in Table VIII, Table IX is tabulated. The total basic user opportunity cost per month is computed using Equation (1), and request 1 is found to be the most costly to be foregone, followed by request 3. Hence, in order to minimize total user opportunity costs (incurred now and in the future), requests 1 and 3 should be implemented first prior to request 2, assuming that all requests are independent.

Table VIII. Examples of maintenance requests and benefit quantifications.

Objective of the maintenance request	Benefit descriptions	Benefit quantifications
Maintenance-request-1 Electronically matched payments against open invoices	• Reduced time and cost spent manually matching receipts and invoices 'Operational cost reduction' category	No. of staff involved × reduction in time taken to complete a match transaction × No. of match transactions per month × staff's hourly rate, e.g. 3 staffs × 0.25 h × 334 transactions × \$100 = \$25 050
Maintenance-request-2 Easy access to and reports of current and historical sales information	• More accurate sales planning 'Competitive advantage' category	Additional unit of sales per month × net revenue per unit sales, e.g. 50 sales unit × \$200 = \$10 000
	• Better information to inform manager of customer history when negotiating payment or contract terms 'Globalization' category	Amount of time saved in dealing with a customer × No. of customers per month × manager's hourly rate, e.g. 0.5 hour × 20 customers × \$150 = \$1500
Maintenance-request-3 Integrated Accounts Payable and Accounts Receivable modules; providing business units with up-to-date cash flow information	• Reduction in the number of third-party and legacy systems maintained 'Integrated system' category	No. of legacy systems × maintenance cost per month, e.g. 2 systems × \$5000 = \$10 000
	• Reduction in lead time for the production of monthly reports 'Best business practices' category	Amount of reduced lead time (days) × No. of reports × cost of lead time, e.g. 3 days × 50 reports × \$20 = \$3000

4.5. Summary of the findings

The key findings of this section can be summarized as follows.

- Request-type and maintenance-complexity variables can be used to determine the amount of effort required to complete an ERP-client-initiated maintenance request.
- On average, carrying out LCP maintenance was almost as effort demanding as carrying out a *user enhancement* request.
- New version upgrades reduce the number of enhancement modifications performed by the ERP client.
- New version upgrades potentially reduce future LCP or support package maintenance distributed by the ERP vendor.

Table IX. User opportunity costs (in $1000) and maintenance request priorities.

	Benefit categories					Total basic user opportunity cost (per month)	Maintenance request priority rank
	Operational cost reduction ($W_1 = 3$)	Integrated system ($W_2 = 2$)	Best business practices ($W_3 = 2$)	Competitive advantage ($W_4 = 1$)	Globalization ($W_5 = 1$)		
Request 1	25.05	0	0	0	0	75.15	1st
Request 2	0	0	0	10	1.5	11.5	3rd
Request 3	0	10	3	0	0	26	2nd

Table X. A comparison of characteristics considered in existing replacement models and in the proposed ERP MU model.

Characteristics considered	In-house software model	Hardware model	ERP model
Software technology, system size, and age	X	NA	NA
Task complexity	X	NA	X (client maintenance)
Number of previous enhancements	X	NA	X (LCP maintenance)
Hardware capacity	NA	X	NA
Hardware utilization	NA	X	NA
Fix revenue (or output) generated	NA	X	NA
Hardware acquisition size	NA	X	NA
User opportunity (benefit realization)	NC	NC	X
Reduction in previous (user) enhancements	NA	NA	X
Reduction in amount of (LCP) maintenance	NC	NC	X

X = considered, NA = not applicable, NC = applicable but not considered.

5. IMPLICATION FROM THE FINDINGS

The findings obtained from the previous section (on factors driving ERP MU decisions) are compared with the existing in-house software and hardware replacement models to determine if they (the existing models) are adequate in the context of ERP. The results are summarized in Table X.

The existing in-house software replacement models (see [61,62]) consider characteristics such as software technology, source code size, and age as the main drivers influencing software replacement decisions. Software technology refers to the programming language and the environment used to develop and maintain a system; it has impact on program structure [63]. Thus, it affects the software maintainability and maintenance effort. A better technology, for example 4GL (Fourth Generation Language), is argued to contribute to a better program structure. Therefore, it is expected to have a higher maintainability and require less maintenance effort than 3GL or lesser programs. A large system generally implies more source code, and for this reason more effort is required to comprehend the program before debugging [64]. As the system ages it tends to be less well ordered [64] and also becomes more complex [65]. As a consequence, it will require more effort to maintain. These factors are unlikely to be applicable in package software because: (1) it is most likely that organizations do not have any choice as to the software technology that is used in a particular package system that they want to purchase; (2) source code size plays a limited impact on ERP-clients' maintenance effort because these organizations make a relatively small number of changes/modifications to the system, and most of the ERP-software maintenance is shouldered by the vendor; and (3) ERP-using organizations have no control over the system age at all as the software code is managed by the vendor.

In contrast, factors such as task complexity and number of previous enhancement modifications are found to have an impact on both in-house developed and ERP package software maintenance decisions. However, the factor 'previous enhancement modifications' has slightly different impact and

Table XI. Impact of enhancement[a] maintenance on software code and maintenance effort.

	Request type influenced by this request	Impact on software code	Impact on maintenance effort
In-house software	All types of maintenance requests— corrective, adaptive, perfective/enhancement, software replacement	Source code structure and quality deterioration Software size may increase by increased custom code	Increased effort for source code comprehension Increased effort for programming
ERP package software	LCP or support package maintenance, and software upgrade only	Standard code is changed/affected, and/or custom code increases	Increased effort for impact analysis before and after the LCP or support package implementation, and the new version upgrade exercise

[a] Enhancement is referred to activities which involve making changes to the (standard) software code; it is associated with enhancement initiated by the ERP client but not those from the vendor.

implications on the software code and maintenance effort requirements in in-house and ERP software. This is summarized in Table XI. In the context of in-house software, it is argued that the greater the modifications done to the system, the greater the deterioration in the software code structure and quality [66,67]. This increases the difficulties in understanding the existing code and making changes to it. As a result, more effort is necessary in both the source-code comprehension and programming phases. In contrast, the larger the number of ERP enhancement modifications (that are initiated by the ERP client and which involve changing the standard code or writing custom code), the higher the amount of custom code and/or standard code being changed and affected. This increases the amount of effort required to analyze the impact of a support package or LCP, and new version upgrade on these enhancements.

The hardware literature (see [68,69]) argues that hardware capacity and utilization deteriorates as it ages. However, software does not deteriorate by itself or when it is used. In contrast, software deterioration is caused by the maintenance work done on the source code over time [70]. This is the opposite for hardware as maintenance work repairs hardware deterioration [71]. Hopp and Nair [72] have relaxed this assumption by assuming that hardware does not deteriorate, and consider factors such as costs and revenues (or output) generated by the new hardware or technology. They presume that costs and revenues for technologies are known, and revenues are constant over time. The latter assumption is not realistic in software systems because their revenues are believed to change depending on software functionality and the timeliness of this software functionality. Other models such as the one proposed in [73] are also not applicable for software as they consider (hardware) acquisition size a determinant of the replacement decision.

Results from this study, on the ERP maintenance and upgrade (MU) environment, show that user opportunity (or benefit realization), a reduction in the number of previous enhancements modifications, and a reduction in the amount of future LCP maintenance are important characteristics for an ERP MU decision model. Although user opportunity and a reduction in the amount of future maintenance may also be applicable in in-house software and hardware, they are not considered in the existing literature. On the other hand, a reduction in the number of previous enhancement modifications is unlikely to be an in-house software replacement driver because its internal-system enhancements are unlikely to be available from external sources and therefore could not be replaced or reduced. It is also logically impossible to apply this factor in a hardware replacement model, as hardware is less malleable and usually more enhanced than its predecessors. In the light of these differences, a specific ERP MU decision framework is needed, and one is proposed in the next section.

6. A FRAMEWORK FOR ERP MAINTENANCE AND UPGRADE DECISIONS

6.1. Decision alternatives

The proposed decision framework (for GA) takes into account four fundamental decision alternatives, besides the typical option of 'doing nothing'. These options include: (i) the system user maintenance requests; (ii) GA's enhancement maintenance, (iii) LCP or support package maintenance; and (iv) a new version upgrade.

6.2. Tradeoffs involved in decision alternatives

The discussion of the preliminary decision framework is structured around the initiators and decision alternatives of GA's maintenance activities and it is summarized in Figure 3. Descriptions of the variables used are given in Table XII.

6.2.1. System users' maintenance

As discussed earlier, there are three types of system users' change request: enhancement, corrective, and master-data change. The decision to implement a *user enhancement* request is based on the tradeoffs between the ongoing maintenance cost and the user opportunity cost of not implementing it. This enhancement implementation cost (in GA's case) is charged back to the user, and does not incur a cost to GA. As a result, it is not included in the total *user enhancement* maintenance cost in the framework. In contrast, the ongoing maintenance cost for this enhancement, which follows each LCP maintenance project (see Section 4.2 for details) and upgrade (see Section 4.3.1), is borne by GA. Hence, for each *user enhancement* modification, we define

$$\begin{aligned}\text{Ongoing maintenance cost} = {} & (\text{number of LCP projects} + \text{number of upgrades}) \\ & \times (\text{a fraction of its implementation cost}) \end{aligned} \tag{2}$$

On the other hand, user opportunity cost will be incurred if this enhancement request is ignored. However, there is a possibility that this enhancement would be introduced in the next version by the

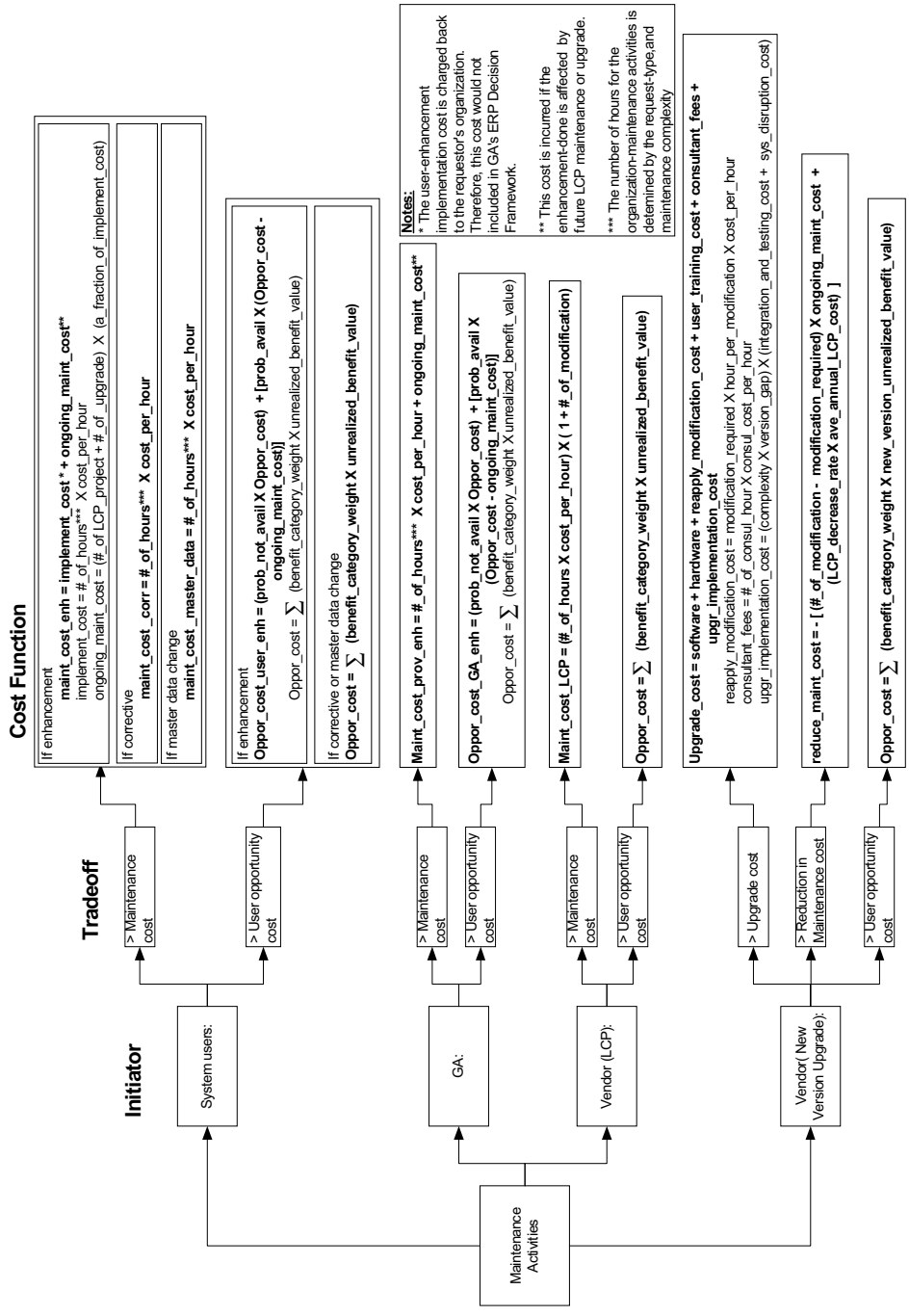

Figure 3. ERP decision framework: decision alternatives, tradeoffs, and cost functions.

Table XII. A description of the variables considered in the decision framework.

Cost component, etc.	Maintenance initiator	Variable involved	Description
Maintenance cost	System users (GA's clients)	Maint_cost_enh	Cost for maintaining a *user enhancement* request
		Implement_cost	Cost for implementing a *user enhancement* request (i.e. the initial cost). Its cost factor depends on complexity and request types
		Maint_cost_corr	Cost for maintaining a user *corrective* request
		Maint_cost_master_data	Cost for maintaining a user *master-data change* request
	System users, GA	Ongoing_maint_cost	Ongoing maintenance cost of the enhancement modification
		#_of_modification	The total number of enhancement modifications in the existing version, including those done during system implementation
		A_fraction_of_ implement_cost	A certain small portion of an enhancement-modification's implementation cost
	GA	Maint_cost_prov_enh	Cost for implementing GA's enhancement
	Vendor	Maint_cost_LCP	LCP maintenance cost (driven by the number of enhancement modifications)
User opportunity cost	Vendor, GA, system user	Oppor_cost	User opportunity cost incurred when the respective LCP maintenance, new version upgrade, user *corrective* or *master-data change* maintenance is not satisfied
	System user-enhancement	Oppor_cost_user_enh	User opportunity cost incurred when a *user enhancement* modification request is not satisfied, taking the uncertainty factor into consideration
	GA enhancement	Oppor_cost_GA_enh	User opportunity cost incurred when *GA enhancement* request is not satisfied, taking the uncertainty factor into consideration

Table XII. Continued.

Cost component, etc.	Maintenance initiator	Variable involved	Description
	System users, GA, vendor	Benefit_category_weight	Importance of a benefit category to an ERP client's business objectives (i.e. measured on a scale of 1–3)
		Unrealized_benefit_value	Unrealized benefit value of a request evaluated on a benefit category
	Vendor	New_version_unrealized_benefit_value	Unrealized benefit value of a new version evaluated on a benefit category
New version upgrade	Vendor	Upgrade_cost	Total cost incurred in an upgrade exercise
		Software	Cost of licensing a new upgrade version
		Hardware	Cost associated with purchasing additional hardware
		Reapply_modification_cost	Cost incurred in order to re-apply/implement previous enhancement modifications
		Modification_required	Number of previous enhancement modifications still required after an upgrade
		Hour_per_modification	Average hours required in order to re-implement one unit of a previous enhancement modification
		User_training_cost	Cost of educating and training the system users
		Consultant_fees	Cost of hiring consultants for the upgrade
		#_of_consul_hour	Total number of consultant-hours required for an upgrade
		Consul_cost_per_hour	Consultant's rate per hour
		Upgr_implementation_cost	Upgrade implementation cost
		Complexity	Complexity of an upgrade project (i.e. high, medium, low). It is determined by the number of modules to be implemented and the number of business units involved
		Version_gap	Version gap between the existing and new version upgrade (i.e. functionality, upgrade path, technical aspect of upgrade)

Table XII. Continued.

Cost component, etc.	Maintenance initiator	Variable involved	Description
		Integration_and_testing_cost	Cost involved in integrating and testing a new system
		Sys_disruption_cost	Cost of system downtime and disruption during the implementation and testing time and change management following a new system upgrade
		Reduce_maint_cost	Reduction in maintenance cost following a new version upgrade
		LCP_decrease_rate	Average rate at which the number of LCPs decreases from one version to another
		Ave_annual_LCP_cost	Average annual LCP maintenance cost
Maintenance request	System users, GA, vendor	#_of_hour	Total number of hours, an estimation of the number of staff required and project duration—based on request type and project complexity for completing a request (as assessed by senior manager)
Organizational factor	GA	Cost_per_hour	Maintainer and analyst's hourly rate
LCP project	GA	#_of_LCP_project	Estimation of total LCP projects over a time horizon
Upgrade project	GA	#_of_upgrade	Estimation of total upgrade projects over a time horizon
External uncertainties	System user, GA	Prob_avail	Probability that an enhancement requested by a system user/GA will be available in the next upgrade version
		Prob_not_avail	Probability that an enhancement requested by a system user/GA will not be available in the next upgrade version

vendor. With this, the user opportunity is formulated as:

$$\begin{aligned}\text{opportunity cost for } \textit{user enhancement} = {} & [(\text{probability of being not available in the next version}) \\ & \times (\text{opportunity cost of the request})] \\ & + \{(\text{probability of being available in the next version}) \\ & \times [(\text{opportunity cost of the request}) \\ & - (\text{ongoing maintenance cost})]\}, \end{aligned} \tag{3}$$

where the opportunity cost of the request is the total basic user opportunity cost, and it is equal to the sum of each benefit-category weight multiplied by the unrealized benefit value for that benefit category (see equation 1 in Section 4.4 for details). Also, note that the relationship between the probability of an enhancement being available and not available in the next version is as follows:

$$\begin{aligned}&\text{probability of being available in the next version} \\ &\quad = 1 - (\text{probability of being not available in the next version}) \end{aligned} \tag{4}$$

For other types of requests such as *corrective* and *master-data change*, the maintenance cost is simply computed as follows:

$$\text{maintenance cost for } \textit{corrective} = (\text{estimated number of hours}) \times (\text{cost per labor hour}) \tag{5}$$

$$\text{maintenance cost for } \textit{master-data change} = (\text{estimated number of hours}) \times (\text{cost per labor hour}) \tag{6}$$

where the estimated number of hours is determined by the request type and maintenance complexity (refer to the empirical evidence in Sections 4.1.1 and 4.1.2).

Likewise, the cost of forgoing this request(s) is calculated as

$$\begin{aligned}\text{opportunity cost for } \textit{corrective} = {} & \sum [(\text{benefit-category weight}) \\ & \times (\text{unrealized benefit value for that benefit category})] \end{aligned} \tag{7}$$

$$\begin{aligned}\text{opportunity cost for } \textit{master-data change} = {} & \sum [(\text{benefit-category weight}) \\ & \times (\text{unrealized benefit value for that benefit category})] \end{aligned} \tag{8}$$

6.2.2. *GA enhancement*

As mentioned earlier, GA, the service provider, introduces enhancement maintenance for ongoing support of the SAP system, maintenance cost reduction, and improvements to the performance of the system. The cost of performing this request is equal to

$$\begin{aligned}\text{maintenance cost for } \textit{GA enhancement} = {} & (\text{estimated number of hours}) \\ & \times (\text{cost per labor hour}) + \text{ongoing maintenance cost} \end{aligned} \tag{9}$$

Note that the ongoing maintenance cost in the above equation is similar to that for *user enhancement*. It is incurred if *GA enhancement* falls under either one of the enhancement options from (ii)–(ix) as detailed in Section 4.1.

On the other hand, the effect of not doing anything with this request is the user opportunity cost. Similar to the user opportunity cost formulation for user enhancement discussed earlier, there are possibilities and uncertainties involved in relation to whether this enhancement would be introduced in the next version. Thus, user opportunity cost for GA enhancement is as follows:

$$\begin{aligned}\text{opportunity cost for } \textit{GA enhancement} = {} & [(\text{probability of being not available in the next version}) \\ & \times (\text{opportunity cost of the request})] \\ & + \{(\text{probability of being available in the next version}) \\ & \times [(\text{opportunity cost of the request}) \\ & - (\text{ongoing maintenance cost})]\} \end{aligned} \tag{10}$$

Note that for *GA enhancement* that has no impact on future LCP maintenance effort or upgrade effort, the formulation for this opportunity cost is the same as for *corrective* or *master-data change* maintenance requests.

6.2.3. Vendor's LCP maintenance

In contrast to the system user- and GA-enhancement maintenance, the LCP maintenance cost is not only driven by the estimated number of hours and cost per labor hour but also the number of enhancement modifications done during the implementation project (see the last paragraph in Section 2.1) and the number of enhancement modifications done to the existing version during post-implementation (supported by evidence in Section 4.2.) That is,

$$\begin{aligned}\text{Maintenance cost for LCP} = {} & (\text{estimated number of hours}) \times (\text{cost per labor hour}) \\ & \times (1 + \text{number of enhancement modifications done during} \\ & \quad \text{implementation project} \\ & + \text{number of enhancement modifications done during} \\ & \quad \text{post-implementation}) \end{aligned} \tag{11}$$

We assume that each enhancement modification has a linear relationship with the LCP maintenance cost, such that if there are two enhancement modifications then the total LCP maintenance costs are twice as costly.

While costs are incurred when the LCP is implemented, user opportunity costs are incurred if it is delayed. That is,

$$\begin{aligned}\text{opportunity cost for LCP} = {} & \sum [(\text{benefit-category weight}) \\ & \times (\text{unrealized benefit value for that benefit category})] \end{aligned} \tag{12}$$

6.2.4. *Upgrade*

Upgrade cost, on the other hand, is the sum of the following elements:

$$\begin{aligned}\text{upgrade cost} = {} & \text{software license cost} + \text{hardware cost} + \text{user training cost} + \text{consultancy fees} \\ & + \text{previous enhancement modification re-application cost} \\ & + \text{upgrade implementation cost}\end{aligned} \tag{13}$$

The previous enhancement modification re-application cost refers to the cost of re-applying previous enhancement modifications onto a new upgrade version (as discussed in Section 4.3.1), and it is equal to

$$\begin{aligned}&\text{previous enhancement modification reapplication cost} \\ &\quad = (\text{total number of enhancement modifications still required after upgrade}) \\ &\qquad \times (\text{number of hours per enhancement modification}) \times (\text{cost per labor hour})\end{aligned} \tag{14}$$

The upgrade implementation cost depends on the complexity or scope of the upgrade (which is a subjective evaluation based on the number of modules to be implemented and the number of business units involved), the version gap between the existing and new (upgrade) version, system integration and testing, and the system disruption cost. That is,

$$\begin{aligned}\text{upgrade implementation cost} = {} & [\text{complexity} \times \text{version gap}] \\ & \times [(\text{system integration and testing cost}) + (\text{system disruption cost})]\end{aligned} \tag{15}$$

Here, we assume that both the system integration and testing cost, and the system disruption cost are increased by the compound factors of both upgrade complexity and version gap.

While it may be cheaper not to do an upgrade, the upgrade has a positive impact on future maintenance costs (refer to Sections 4.3.1 and 4.3.2). The reduction in maintenance costs comprises two components: the number of previous enhancement modifications (for ongoing maintenance) and the amount of LCP maintenance. It is as follows:

$$\begin{aligned}&\text{reduction in maintenance cost} \\ &\quad = -[(\text{total number of enhancement modifications in the existing version} \\ &\qquad - \text{total number of enhancement modifications still required after upgrade}) \\ &\qquad \times (\text{ongoing maintenance cost})] + [(\text{decrease in number of LCPs}) \\ &\qquad \times (\text{average annual LCP maintenance cost})]\end{aligned} \tag{16}$$

Besides this, if the upgrade option is forgone it is expected to contribute to a larger amount of user opportunity costs. Therefore, for the upgrade option, we define

$$\begin{aligned}\text{opportunity cost for upgrade} = {} & \sum [(\text{benefit-category weight}) \\ & \times (\text{unrealized benefit value of the new version} \\ & \quad \text{for that benefit category})]\end{aligned} \tag{17}$$

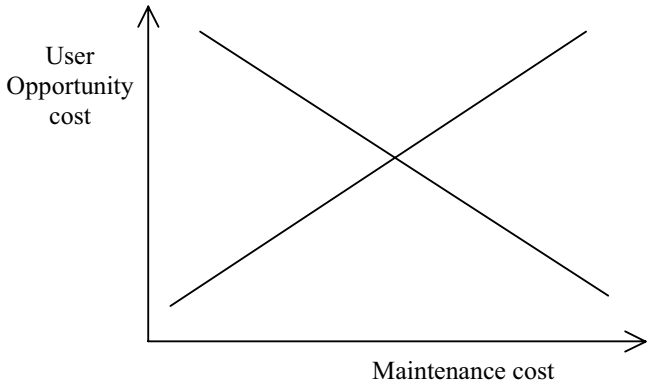

Figure 4. Tradeoff between opportunity cost and maintenance cost.

(In an extreme case, where the vendor withdraws supports for an installed version, the total user opportunity cost would be extremely high, under the 'operational cost reduction' benefit category. In such a situation, an upgrade decision is obviously the most feasible option.)

A summary of the ERP maintenance and upgrade decision framework is illustrated in Figure 3 (note that the variable names have been abbreviated in the framework). A detailed description of the variables used in the decision framework is given in Table XII.

6.3. An illustration of an ERP decision model

Using the three hypothetical maintenance requests discussed in section 4.4, further information about these requests are described as follows. Requests 1 and 2 are *user enhancements* whereas request 3 is a *GA enhancement*. They arrive at the same time. The decision problem is to determine which maintenance request(s) to maintain at the beginning of the first month. (Assuming that the required resources would be available, more than one maintenance project can be started at the same time.) The maintenance decision is assessed over a three-year period, and is based on tradeoffs between the total user opportunity costs and the total maintenance costs. The relationship between these two cost components is as shown in Figure 4. That is, the total user opportunity costs can be reduced by satisfying a maintenance request(s) but at the expense of maintenance costs. Conversely, the total maintenance costs could be contained by delaying the maintenance request(s) or doing nothing but at the expense of total user opportunity costs. The objective is to minimize the total costs of ERP software. Thus, the course of action is determined by considering the minimum costs.

Project requirements for the three maintenance requests, such as number of staff required, and project duration, are as described in Table XIII. Although no maintenance-implementation cost is incurred by GA for requests related to user enhancements, ongoing maintenance costs are incurred (as they would be affected by an LCP project or an upgrade project in the future). This ongoing maintenance cost is calculated using Equation (2). Assuming that the ongoing maintenance cost for request 1 is 10% of its

Table XIII. Maintenance requests: total maintenance costs.

Maintenance request	Category	Maintenance implementation cost			Ongoing implementation cost			Total ongoing maintenance cost for 3 years ($)
		Implementation attribute	Implementation quantifier ($)	Implementation cost ($)	Ongoing maintenance attribute (%)	Ongoing maintenance quantifier (%)	Ongoing maintenance cost (1st month) ($)	
Request 1	User enhancement	No. of staff required	3		Percentage of implementation cost	10	9600.00	394 768.03
		Project duration (months)	2					
		Staff hourly rate	100.00	96 000.00				
Request 2	User enhancement	No. of staff required	4					
		Project duration (days)	21					
		Staff hourly rate	100.00	67 200.00	Percentage of implementation cost	15	10 080.00	424 670.43
Request 3	GA enhancement	No. of staff required	5					
		Project duration (months)	1					
		Staff hourly rate	120.00	96 000.00				
No. of hours per man-month	160							
No. of hours per man-day	8							
Inflation rate (per annum)	10%							
Inflation rate (per month)	0.8%							
Total number of months	36							

Table XIII. Continued.

Ongoing maintenance cost:

Item details	Monetary value (PV) ($)	Actual cumulated ongoing maintenance cost after 3 years ($)
Request 1	9 600.00	394 768.03
Request 2	10 080.00	424 670.43

	Request 1		Request 2	
Month	Increment cost ($)	Cumulated cost ($)	Incremental cost ($)	Cumulated cost ($)
1	9600.00	9600.00	10 080.00	10 080.00
2	9680.00	19 280.00	10 164.00	20 244.00
3	9760.67	29 040.67	10 248.70	30 492.70
4	9842.01	38 882.67	10 334.11	40 826.81
5	9924.02	48 806.69	10 420.22	51 247.03
6	10 006.72	58 813.42	10 507.06	61 754.09
7	10 090.11	68 903.53	10 594.62	72 348.71
8	10 174.20	79 077.72	10 682.91	83 031.61
9	10 258.98	89 336.71	10 771.93	93 803.54
10	10 344.47	99 681.18	10 861.70	104 665.24
11	10 430.68	110 111.85	10 952.21	115 617.45
12	10 517.60	120 629.45	11 043.48	126 660.93
13	10 605.25	131 234.70	11 135.51	137 796.43
14	10 693.62	141 928.32	11 228.30	149 024.74
15	10 782.74	152 711.06	11 321.87	160 346.61
16	10 872.59	163 583.65	11 416.22	171 762.83
17	10 963.20	174 546.85	11 511.36	183 274.19
18	11 054.56	185 601.40	11 607.28	194 881.47
19	11 146.68	196 748.08	11 704.01	206 585.49
20	11 239.57	207 987.65	11 801.55	218 387.03
21	11 333.23	219 320.88	11 899.89	230 286.92
22	11 427.67	230 748.55	11 999.06	242 285.98
23	11 522.90	242 271.46	12 099.05	254 385.03
24	11 618.93	253 890.39	12 199.88	266 584.91
25	11 715.75	265 606.14	12 301.54	278 886.45
26	11 813.38	277 419.53	12 404.05	291 290.50
27	11 911.83	289 331.35	12 507.42	303 797.92
28	12 011.09	301 342.45	12 611.65	316 409.57
29	12 111.19	313 453.64	12 716.75	329 126.32
30	12 212.11	325 665.75	12 822.72	341 949.04
31	12 313.88	337 979.63	12 929.58	354 878.61
32	12 416.50	350 396.13	13 037.32	367 915.93
33	12 519.97	362 916.10	13 145.97	381 061.90
34	12 624.30	375 540.40	13 255.52	394 317.42
35	12 729.50	388 269.90	13 365.98	407 683.39
36	12 835.58	401 105.48	13 477.36	421 160.76
Followed by an upgrade	12 942.55	414 048.03	13 589.67	434 750.43

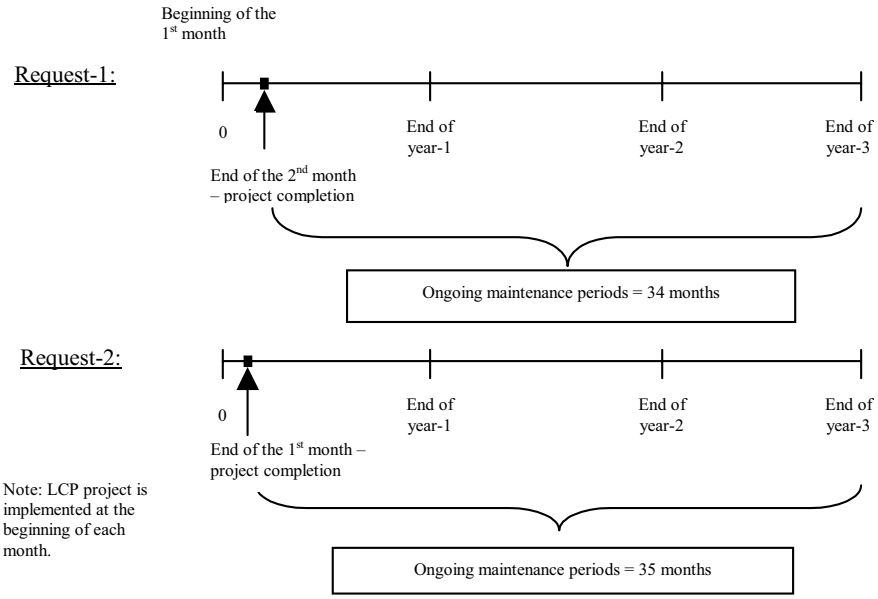

Figure 5. Request-1 and request-2: ongoing maintenance cost payment.

maintenance implementation cost and that for request 2 is 15%, their ongoing maintenance costs (for the first month) are evaluated as \$9600 and \$10 080, respectively. Note that the ongoing maintenance costs for requests 1 and 2 are incurred each time an LCP project or an upgrade project is implemented. It is assumed that GA would implement an LCP project once a month and there would be an upgrade at the end of the third year. With adjustment to the expected project completion for requests 1 and 2, there would be only 34 ongoing-maintenance-cost payments (or 34 LCP projects) for request 1 and 35 ongoing-maintenance-cost payments for request 2 in the three-year period. This is illustrated in Figure 5.

We assume that the inflation rate is 10% per annum. It is constant for the three-year time horizon. The maintenance cost per month is expected to increase with this inflation rate. The maintenance implementation cost for *GA enhancement* is computed using Equation (9). It is assumed that the enhancement in request 3 has no impact on future maintenance or upgrade effort. Thus, its ongoing maintenance cost is basically zero. The total cumulated maintenance costs, for the three requests, over the three-year period are shown in Table XIII.

The total basic user opportunity costs incurred in the first month for the three requests are as computed in Table IX and appear in Table XIV. The user opportunity costs per month are assumed to decrease with the inflation rate. The total cumulated opportunity costs by the end of year 3 for the three requests are as illustrated in Table XIV. While the total user opportunity costs for *user enhancements*

Table XIV. Maintenance requests: total opportunity costs.

		Opportunity cost		User enhancement opportunity cost		
Maintenance request	Category	Total basic opportunity cost (first month) ($)	Total opportunity cost by the end of year 3 ($)	User enhancement attribute	User enhancement quantifier	Total opportunity cost for user enhancement ($)
Request 1	User enhancement	75 150.00	2345 693.57	Probability available in upgrade version	50%	
				Probability not available in upgraded version	50%	
				Total opportunity costs for 3 years	$2345 693.57	
				Total ongoing maintenance costs for 3 years	$394 768.03	
				Difference between total opportunity and ongoing maintenance costs	$1950 925.54	2148 309.56
Request 2	User enhancement	11 500.00	358 955.10	Probability available in upgraded version	75%	
				Probability not available in upgraded version	25%	
				Total opportunity costs for 3 years	$358 955.10	
				Total ongoing maintenance costs for 3 years	$424 670.43	
				Difference between total opportunity and ongoing maintenance costs	$(65 715.33)	252 787.50
Request 3	GA enhancement	26 000.00	811 550.67			
Inflation rate (per annum)	10%					
Inflation rate (per month)	0.8%					
Total number of months	36					

Table XIV. Continued.

Opportunity costs:

Item details	Monetary value (PV) ($)	Cumulated monetary value after 3 years ($)
Request 1	75 150.00	2345 693.57
Request 2	11 500.00	358 955.10
Request 3	26 000.00	811 550.67

	Request 1		Request 2		Request 3	
Month	Value after depreciation ($)	Cumulated value ($)	Value after depreciation ($)	Cumulated value ($)	Value after depreciation ($)	Cumulated value ($)
1	75 150.00	75 150.00	11 500.00	11 500.00	26 000.00	26 000.00
2	74 523.75	149 673.75	11 404.17	22 904.17	25 783.33	51 783.33
3	73 902.72	223 576.47	11 309.13	34 213.30	25 568.47	77 351.81
4	73 286.86	296 863.33	11 214.89	45 428.19	25 355.40	102 707.21
5	72 676.14	369 539.47	11 121.43	56 549.62	25 144.11	127 851.31
6	72 070.50	441 609.97	11 028.75	67 578.37	24 934.57	152 785.89
7	71 469.92	513 079.89	10 936.85	78 515.22	24 726.78	177 512.67
8	70 874.33	583 954.23	10 845.71	89 360.93	24 520.73	202 033.40
9	70 283.71	654 237.94	10 755.33	100 116.25	24 316.39	226 349.79
10	69 698.02	723 935.96	10 665.70	110 781.95	24 113.75	250 463.54
11	69 117.20	793 053.16	10 576.82	121 358.77	23 912.80	274 376.34
12	68 541.22	861 594.38	10 488.68	131 847.44	23 713.53	298 089.87
13	67 970.05	929 564.43	10 401.27	142 248.71	23 515.92	321 605.79
14	67 403.63	996 968.06	10 314.59	152 563.31	23 319.95	344 925.74
15	66 841.93	1063 809.99	10 228.64	162 791.95	23 125.62	368 051.36
16	66 284.92	1130 094.91	10 143.40	172 935.35	22 932.91	390 984.27
17	65 732.54	1195 827.45	10 058.87	182 994.22	22 741.80	413 726.06
18	65 184.77	1261 012.22	9975.05	192 969.27	22 552.28	436 278.35
19	64 641.56	1325 653.79	9891.92	202 861.19	22 364.35	458 642.69
20	64 102.89	1389 756.67	9809.49	212 670.68	22 177.98	480 820.67
21	63 568.69	1453 325.37	9727.74	222 398.43	21 993.16	502 813.83
22	63 038.96	1516 364.32	9646.68	232 045.11	21 809.88	524 623.72
23	62 513.63	1578 877.95	9566.29	241 611.40	21 628.14	546 251.85
24	61 992.68	1640 870.64	9486.57	251 097.97	21 447.90	567 699.75
25	61 476.08	1702 346.71	9407.52	260 505.49	21 269.17	588 968.92
26	60 963.78	1763 310.49	9329.12	269 834.61	21 091.93	610 060.85
27	60 455.75	1823 766.24	9251.38	279 085.98	20 916.16	630 977.01
28	59 951.95	1883 718.19	9174.28	288 260.27	20 741.86	651 718.87
29	59 452.35	1943 170.53	9097.83	297 358.10	20 569.01	672 287.88
30	58 956.91	2002 127.45	9022.02	306 380.11	20 397.60	692 685.48
31	58 465.60	2060 593.05	8946.83	315 326.95	20 227.62	712 913.10
32	57 978.39	2118 571.44	8872.28	324 199.22	20 059.06	732 972.16
33	57 495.24	2176 066.68	8798.34	332 997.56	19 891.90	752 864.05
34	57 016.11	2233 082.79	8725.02	341 722.58	19 726.13	772 590.19
35	56 540.98	2289 623.77	8652.31	350 374.89	19 561.75	792 151.94
36	56 069.80	2345 693.57	8580.21	358 955.10	19 398.73	811 550.67

Table XV. Maintenance requests: decision policy.

Enhancement Category	Total ongoing maintenance cost ($)	Total maintenance implementation cost ($)	Total basic opportunity cost (in the first month) ($)	Total opportunity cost for user enhancement ($)	Total opportunity cost for GA enhancement ($)	Decision
User enhancement	394 768.03	—	75 150.00	2148 309.56	—	Maintain
User enhancement	424 670.43	—	11 500.00	252 787.50	—	Do nothing
GA enhancement	—	96 000.00	26 000.00	—	811 550.67	Maintain

are calculated using Equation (3), Equation (10) is used for *GA enhancement*. The probabilities of enhancements in requests 1 and 2 being available in the next version are 50% and 75%, respectively.

The maintenance decision policy with respect to these assumed data is to maintain requests 1 and 3, but disregard request 2 for the time being (see Table XV).

7. CONCLUDING REMARKS

The existing trade press has effectively captured the fundamental decision factors for ERP MU but generally has overlooked some details such as: (i) ERP-client's internal maintenance request costs, and its maintenance effort drivers ([13] is an exception); (ii) the impact of new functionality in a new version on the number of previous enhancement modifications still required after an upgrade; and (iii) the reduction in future LCP maintenance after an upgrade process. Hence, the synthesized ERP maintenance and upgrade (MU) decision framework has been extended based on these new findings and observations. This study concludes by suggesting that in making an ERP MU decision, an organization can at best take into account three fundamental factors: the maintenance of the ERP system, the new version upgrade, and benefit realization. Each decision alternative (for example, the number of LCPs, the amount of ERP-client-initiated maintenance, upgrade or do nothing) has its own tradeoffs between MU costs and user opportunity costs. The objective function is to minimize these (total) costs. Hence, one should aim at minimizing the current and future MU costs, and maximizing the current and future benefit realization from the system.

The contribution of this study in practice is that the proposed framework can be: (i) used as a guideline for ERP managers to justify the cost and benefit of choosing the decision alternatives; (ii) referenced as the critical success factors (CSFs) in order to reduce the total ERP software cost; and (iii) utilized to (help ERP managers to) better manage their maintenance activities and control the frequencies of upgrades (provided historical data is available to predict the monthly (or yearly) average number of user requests, vendor LCPs, and upgrade strategies (i.e. the frequency of the introduction of new version upgrades) adopted by the vendor).

However, the proposed framework represents a simplistic model of the ERP MU decision domain. This framework may not be complete. It deals with the basic ERP MU decision factors only. It

requires further validation and testing in other ERP-using organizations from different industries and backgrounds. This framework also poses further questions: (i) How could the user opportunity cost be evaluated systematically in ERP MU context and to what extent? (ii) How would the proposed framework be implemented and used to generate solutions (i.e. ERP maintenance and upgrade policies) for ERP-employing organizations? (iii) What are the factors driving support package/LCP maintenance efforts? and (iv) What are the underlying factors influencing the reduction rate in maintenance costs after an upgrade?

ACKNOWLEDGEMENTS

The author would like to thank the reviewers for their helpful and stimulating comments.

REFERENCES

1. Carlino J, Nelson S, Smith N. AMR research predicts enterprise applications market will reach $78 billion by 2004. *AMR Research*, 2000. http://www.amrresearch.com/pressroom/files/000727.asp [29 August 2000].
2. Girard K. Back-office rebirth. *Business 2.0*, 2000. http://www.business2.com/articles/mag/0,1640,13785,FF.html [12 August 2000].
3. Glass RL, Vessey I. Enterprise resource planning systems: Can they handle the enhancement changes most enterprises require? *The Software Practitioner* 1999; **9**(5):1–12.
4. Carlino J, Nelson S, Smith N. AMR research study reveals SAP R/3 upgrade cost users 25- to 33 percent of initial investment. *AMR Research*, 2000. http://www.amrresearch.com/pressroom/files/00426.asp [29 August 2000].
5. Songini ML. Users vent frustration over Oracle CRM/ERP upgrades. *Computerworld* 2000; **34**(44):105.
6. 400-Group. Mitsubishi electric outlaws internal SAP customization endeavors after struggling early. *I/S Analyzer* 1998; **37**(4):2–6.
7. 400-Group. Boston Beer Co. minimizes coding changes, foregoes to keep SAP running smoothly. *I/S Analyzer* 1998; **37**(4):13–15.
8. Collins K. Strategy and execution of ERP upgrades. *Government Finance Review* 1999; **15**(4):43–47.
9. Radosevich L. Cheek-ups critical for ERP apps. *Infoworld* 1998; **20**(43):67.
10. Ng CSP, Chan T, Gable GG. A client-benefits oriented taxonomy of ERP maintenance. *Proceedings International Conference on Software Maintenance*. IEEE Computer Society: Los Alamitos CA, 2001; 528–537.
11. Stedman C. Order entry flexibility an ERP issue. *Computerworld* 2000; **34**(4):10.
12. Soh C, Sia SK, Tay-Yap J. Cultural fits and misfits: Is ERP a universal solution? *Communications of the ACM* 2000; **43**(4):47–51.
13. Brehm L, Heinzl A, Markus ML. Tailoring ERP Systems: A spectrum of choices and their implications. *Proceedings 34th Hawaii International Conference on Systems Sciences*. IEEE Computer Society: Los Alamitos CA, 2001 (on CD).
14. Glass RL. The re-engineering decision: Lessons from the best of practice. *The Software Practitioner* 1991; (May):5–7.
15. Stein T. ERP overhaul for Lawson—strategic ledger among new features. *InformationWeek* 1999; (726):42.
16. Callaway E. ERP—the next generation: ERP is web enabled for E-business (1st edn). Computer Technology Research Corp.: Charleston, 2000; 166 pp.
17. Craig R. Laurier enterprise system upgrade. *Proceedings of the International Conference of Information Systems*. Association for Computing Machinery: New York NY, 1999; 654–662. http://www.acm.org/pubs/contents/proceedings/misc/352925/ [20 December 2000].
18. Stedman C. Peoplesoft acts on dip in user satisfaction. *Computerworld* 1999; **33**(36):6.
19. Jakovljevic PJ. Essential ERP—its underpinning technology. *TechnologyEvaluation.Com*, 2000. http://www.technologyevaluation.com/research/researchhighlights/businessapplications/2000/03/research_notes/tu_ba_pj_03_00_2.asp [14 February 2001].
20. Norris G, Hurley JR, Hartley KM, Dunleavy JR, Balls JD. *E-Business and ERP: Transforming the Enterprise*. John Wiley & Sons: New York NY, 2000; 194 pp.
21. Pearson D. Complex compromises—outsourcing ERP. *CIO Magazine*, 1998. http://www.cio.com/archive/060198_erp.html [20 October 1999].

22. Jakovljevic PJ. ERP beginner's guide in so many words. *TechnologyEvaluation.Com*, 2001. http://www.technologyevaluation.com/research/researchhighlights/businessapplications/2001/02/research_notes/tu_ba_pj_02_05_01_1.asp [5 February 2001].
23. Slater D. The hidden cost of enterprise software. *CIO Magazine*, 1998. http://www.cio.com/archive/enterprise/011598_erp.html [20 October 1999].
24. Thompson O. The "Old ERP" dilemma: Replace or add-on. *TechnologyEvaluation.Com*, 2001. http://www.technologyevaluation.com/research/researchhighlights/businessapplications/2001/06/research_notes/tu_ba_xot_06_08_01_1.asp [14 June 2001].
25. Clemons C. Successful implementation of an enterprise system: A case study. *Proceedings of the Americas Conference on Information Systems*. Association for Information Systems: Atlanta GA, 1998; 109–191.
26. Glover SM, Prawitt DF, Romney MB. Implementing ERP. *Internal Auditor* 1999; **61**(1):40–45.
27. Hicks DA, Stecke KE. The ERP maze: Enterprise resource planning and other production and inventory control software. *IIE Solutions* 1995; **27**(8):12–16.
28. Gray L. ERP: Decisions, decisions. *PC Week* 1998; **15**(28):6.
29. Ross JW, Vitale MR. The ERP revolution: Surviving vs. thriving. *Information Systems Frontiers* 2000; **2**(2):233–241.
30. Mullin R. ERP software: The next generation. *Chemical Week* 1997; **159**(20):40–41.
31. Fontana J. Breaking the ERP barrier. *Internetweek* 1999; (697):19.
32. Morphy E. When ERP meets ROI. *Export Today* 1999; **15**(5):42–47.
33. Weston R. ERP users find competitive advantages. *Computerworld* 1998; **32**(15):24.
34. Davenport TH. The future of enterprise system-enabled organizations. *Information Systems Frontiers* 2000; **2**(2):163–180.
35. Hammer M, Champy J. *Reengineering the Corporation: A Manifesto for Business Revolution*. Allen and Unwin: St Leonards, NSW, 1996; 232 pp.
36. Bingi P, Sharma MK, Godla JK. Critical issues affecting an ERP implementation. *Information Systems Management* 1999; **16**(5):7–14.
37. Sieber T, Siau K, Nah F, Sieber M. Implementing SAP R/3 at the University of Nebraska. *Proceedings of the International Conference on Information Systems*. Association for Computing Machinery: New York NY, 1999; 629–649. http://www.acm.org/pubs/contents/proceedings/misc/352925/ [20 December 2000].
38. McDonnell S. Squeezing more out of ERP. *Computerworld* 2000; **34**(40):56–57.
39. Davenport TH. Holistic management of megapackage change: The case of SAP R/3. *Proceedings of the Americas Conference on Information Systems*. Association for Information Systems: Atlanta GA, 1996; 51a–51c (on CD).
40. Davenport TH. Putting the enterprise into the enterprise system. *Harvard Business Review* 1998; **76**(4):121–131.
41. Heald K, Kelly J. AMR research foresees that implementing technologies will lead to competitive advantage. *AMR Research*, 1999. http://www.amrresearch.com/pressroom/files/99430.asp [29 August 2000].
42. Wilson T. Oracle suite integrates ERP, E-business apps. *InternetWeek* 1999; (783):1.
43. Jakovljevic PJ. The essential ERP—it's genesis and future. *TechnologyEvaluation.Com*, 2000. http://www.technologyevaluation.com/research_notes/02-00/prn_TU_BA_PJ_02_00_1.htm [28 February 2000].
44. Markus L, Tanis C. The enterprise systems—from adoption to success. Zmud RW, Price MF (eds.) *Framing the domains of IT management: Projecting the future through the past*. Pinnaflex Educational Resources: Cincinnati OH, 1999; 173–207.
45. Vernadat FB. *Enterprise Modeling and Integration: Principles and Applications* (1st edn). Chapman and Hall: London, 1996; 513 pp.
46. Deloitte-Touche-Tohmatsu. ERP's second wave—maximizing the value of ERP enabled processes (Part 1). *Deloitte Touche Tohmatsu*, 1999. http://www.deloitte.com/publications/m0499_1.html [30 June 2000].
47. Shanks G, Parr A, Hu B, Corbitt B, Thanasankit T, Seddon P. Differences in critical success factors in ERP implementation in Australia and China: A cultural analysis. *Proceedings European Conference on Information Systems*. Vienna University of Economics and Business Administration: Vienna Austria, 2000; 537–544.
48. Markus ML. Viewpoint: Reflections on the systems integration enterprise. *Business Process Management Journal* 2001; **7**(3):171–180.
49. Butler J. Risk management skills needed in a packaged software environment. *Information System Management* 1999; **16**(5):15.
50. Stein T. Making ERP add up. *InformationWeek* 1999; (735):59–63.
51. Benbasat I, Goldstein DK, Mead M. The case research strategy in studies of information systems. *MIS Quarterly* 1987; **11**(3):369–386.
52. Gable GG. The case study methodology: An example in information systems. *Proceedings of the South East Asia Regional Computer Confederation*. Gabungan Komputer Nasional Malaysia: Kuala Lumpur Malaysia, 1992; 65.1–65.21.
53. Yin RK. *Case Study Research: Design and Methods* (2nd edn), vol. 5. Sage Publications: Thousand Oaks CA, 1994; 171 pp.
54. Coakes SJ. *SPSS: Analysis Without Anguish: Version 7.0, 7.5,8.0 for Windows*. Jacaranda Wiley: Brisbane, 1999; 283 pp.
55. SAP AG. *BC Enhancements to the SAP Standard*. SAP AG: Waldorf Germany, 1998; 45 pp.

56. Chapin N. Software maintenance types—a fresh view. *Proceedings International Conference on Software Maintenance*. IEEE Computer Society: Los Alamitos CA, 2000; 247–252.
57. Nellemann DO. World class cost management and CIM justification: The way of the 90s. Muglia VO (ed.) *Enterprise Information Exchange: A Roadmap for Electronic Data Interchange for the Manufacturing Company*. Computer and Automated Systems Association of the Society of Manufacturing Engineers: Dearborn MI, 1993; 163–171.
58. Sarkis J, Sundarraj RP. A decision model for strategic evaluation of enterprise information technologies. *Information Systems Management* 2001; **18**(3):62–72.
59. Queensland Treasury. *QGFMS Benefit Realization Guidelines*. The State of Queensland (Queensland Treasury): Brisbane, 2000; 34 pp.
60. Hares J, Royle D. *Measuring the Value of Information Technology*. Wiley: New York, 1994; 268 pp.
61. Gode DK, Barua A, Mukhopadhyay T. On the economics of the software replacement problem. *Proceedings of the Eleventh International Conference on Information Systems*. Association for Computing Machinery: New York NY, 1990; 159–170.
62. Chan T, Chung SL, Ho TH. An economic model to estimate software rewriting and replacement times. *IEEE Transactions on Software Engineering* 1996; **22**:580–598.
63. Bergland GD. Structured design methodologies. Bergland GD, Gordon RD (eds.) *Tutorial: Software Design Strategies*. IEEE Computer Society: Los Angeles CA, 1981; 475–493.
64. Lientz BP, Swanson EB. *Software Maintenance Management: A Study of the Maintenance of Computer Application Software in 487 Data Processing Organizations*. Addison-Wesley: Reading MA, 1980; 214 pp.
65. Banker RD, Datar SM, Zweig D. Software complexity and maintainability. *Proceedings of the International Conference on Information Systems*. Association for Computing Machinery: New York NY, 1989; 247–255.
66. Lehman MM, Parr FN. Program evolution and its impact on software engineering. *Proceedings 2nd International Conference on Software Engineering*. IEEE Computer Society: Long Beach CA, 1976; 350–355.
67. Lehman MM, Belady LA (eds.). *Program Evolution Processes of Software Change*. Academic Press: London, 1980; 538 pp.
68. Rajagopalan S. Deterministic capacity expansion under deterioration. *Management Science* 1992; **38**(4):525–539.
69. Rajagopalan S, Singh MR, Morton TE. Capacity expansion and replacement in growing markets with uncertain technological breakthroughs. *Management Science* 1998; **44**(1):12–30.
70. Lyons MJ. Salvaging your software asset (tool based maintenance). *AFIPS Conference Proceedings of the 1981 National Computer Conference*, vol. 50. AFIPS Press: Reston VA, 1981; 337–341.
71. Brooks FP. *The Mythical Man-Month: Essays on Software Engineering*. Addison-Wesley: Reading MA, 1979; 195 pp.
72. Hopp WJ, Nair SK. Timing replacement decisions under discontinuous technological change. *Naval Research Logistics* 1991; **38**:203–220.
73. Scarf PA, Bouamra O. A capital equipment replacement model for a fleet with variable size. *Journal of Quality in Maintenance* 1999; **5**(1):40–49.

AUTHOR'S BIOGRAPHY

Celeste S. P. Ng is a PhD candidate in the Information Systems Management Research Centre (ISMRC) at the Queensland University of Technology in Australia. Celeste received her BSc from Campbell University in North Carolina, U.S.A. in 1997, and MIT from Queensland University of Technology, Australia in 1998. Celeste's major research interests include ERP maintenance and upgrade, decision support system, and ERP software maintenance management system.