A Maintenance-data-model of Enterprise Resource Planning

Celeste See Pui Ng
Guy Gable
Taizan Chan

Information Systems Management Research Centre (ISMRC)
Queensland University of Technology
Brisbane, Australia
Email: c0.ng@qut.edu.au, g.gable@qut.edu.au, t.chan@qut.edu.au

Abstract

Two distinct maintenance-data-models are studied: a government Enterprise Resource Planning (ERP) maintenance-data-model, and the Software Engineering Industries (SEI) maintenance-data-model. The objective is to: (i) determine whether the SEI maintenance-data-model is sufficient in the context of ERP (by comparing with an ERP case), (ii) identify whether the ERP maintenance-data-model in this study has adequately captured the essential and common maintenance attributes (by comparing with the SEI), and (iii) proposed a new ERP maintenance-data-model as necessary. Our findings suggest that: (i) there are variations to the SEI model in an ERP-context, and (ii) there are rooms for improvements in our ERP case's maintenance-data-model. Thus, a new ERP maintenance-data-model capturing the fundamental ERP maintenance attributes is proposed. This model is imperative for: (i) enhancing the reporting and visibility of maintenance activities, (ii) monitoring of the maintenance problems, resolutions and performance, and (iii) helping maintenance manager to better manage maintenance activities and make well-informed maintenance decisions.

Keywords

UF Software Maintenance, EI0206.02 Information quality, EI0208 IS utilization, EI0211.01 Organizational impacts, EL0302 Measuring IS success, EI0107 Information evaluation, DA06 Organizational Effectiveness

INTRODUCTION

Enterprise Resource Planning (ERP) is large, integrated, commercial off-the-shelf software that supports most traditional business processes. The new-generation of ERP includes supply chain management, customer relationship management, e-commerce, business-to-business procurement and data warehouse. Over the last decade, organisations worldwide have invested more than $300 billion in ERP implementations (James and Wolf 2000). The customer base of SAP, the largest ERP vendor, includes over ten thousand customers and millions of licensed users (Girard 2000).

Like traditional in-house software, ERP too requires ongoing maintenance. And, though the software vendor is responsible for much traditional maintenance of ERP packages, maintenance activities and costs for user-organisations are substantial (Glass and Vessey 1999). In order for user-organisations to contain these ERP maintenance costs, and to insure a well-functioning ERP system that continues to satisfy the organisation's needs, an effective ERP maintenance model is crucial to manage maintenance activities, to monitor maintenance problems, and to control and manage maintenance efforts. Traditionally, maintenance model is used to reflect and captures the essence of an organization's software maintenance process (Pigoski 1997). It helps to define the activities of the maintenance, and improves maintenance processes (IEEE 1998). Therefore, it is used to plan and manage maintenance, modify the software, and helps to reduce the effort and cost of maintenance. Central to the maintenance model is the maintenance database for capturing the fundamental details of ERP maintenance attributes, from the first maintenance request, through new version upgrades, until the ERP system is retired from production.

The importance of a well-defined and well-designed maintenance-data-model is well understood (Florac 1992, Kajko-Mattsson 1998). Though existing, general-purpose maintenance-data-models may be sufficient to meet the basic needs for the management of ERP maintenance activities, they may not be robust enough to allow detailed monitoring of ERP-specific maintenance problems, resources and conditions. A well-organized maintenance-data-model of ERP maintenance activities is important for: improving the reporting-quality and management of maintenance activities, allowing effective monitoring of maintenance activities, and making better informed maintenance decisions (i.e. prioritising ERP maintenance jobs, evaluating tradeoffs among different maintenance alternatives, and deciding when to proceed with upgrades to a new version).

This paper proposes a maintenance-data model to capture essential ERP maintenance attributes. For instance, maintenance attributes here may include: what maintenance request is reported, what is software object(s) assessed and diagnosed, how the request is resolved and delivered to the system-users, and what is the cost and benefits for the maintenance request. The unit of analysis is the 'change or maintenance request'. Some of the common measurable attributes of a change request include: timing, request-objective, request-type, solution method, software components affected, impacts on other software properties, number of staffs involved, maintenance effort, request-benefits, request-costs, etc. This study focuses on the attributes and management of maintenance requests that originate from system users only; it does not address maintenance requests from the vendor (often called patches).

Our proposed ERP maintenance-data-model is a refinement and extension of the Software Engineering Institute (SEI) software quality measurement framework (Florac 1992) and an ERP maintenance-data-model. The SEI maintenance-data-model is referenced in this study as a standard model for reporting software maintenance problems and defects. With this model, Florac proposes the measurable attributes that are common to traditional software maintenance activities. SEI is recognized by software professionals from the industry, academia, and government, who have participated in its development. It has also been used in prior studies. For example, Kajko-Mattsson (1998) validated the SEI model using three European industrial maintenance-systems. Findings from our case study of a Queensland Government Agency suggest useful variations to the SEI model in an ERP-context.

This paper proceeds by first introducing the research method, subjects involved in this study, data collection and data analysis approaches. Findings are then reviewed and discussed, and a new ERP maintenance-data-model is proposed for the ERP maintenance environment. Finally, implications for practice are drawn and future research directions are discussed.

RESEARCH METHODOLOGY

Case Study

The study employed the case study method (Yin 1994, Gable 1994). Due to limited resources (i.e. time and cost) a single case study was conducted. The case firm is a Queensland Government Agency (identified as *QGA*), a corporate service provider to five Queensland Government Departments. *QGA* have more than two years' experience maintaining and managing the SAP finance and human resources modules, version 3.1H. We sought to study maintenance activities within *QGA* primarily because of our ready access to their detailed maintenance records.

Data Collection

Main sources of case study evidence collected include: (i) *System-Investigation-Request* (SIR) Form (maintenance request form), (ii) SAP Transport Request Form, (iii) the SIR maintenance database, and (iv) a series of tape-recorded semi-structured interviews. The SIR form is used by *QGA* to record details (or attributes) of a maintenance request from the moment it is made until it is completed and delivered to the system users. Each request requires a separate SIR form. The SAP Transport Request Form is primarily used to document information on who approves the delivery of the change request and into which system(s). Although most of the attributes in the SIR form are recorded in the maintenance database, some of the maintenance attributes in this form are not stored as part of the electronic database (the SIR database). Detailed, semi-structured interviews were conducted with the General Manager, Systems Development Manager and Systems Operations Manager for detailed explanations of their ERP maintenance activities, procedures, policies, and management issues and to confirm (amongst other things) the objectives and meanings of each of the data attributes found in their SIR maintenance database. Tapes of the interviews were transcribed verbatim and returned to the case firm for confirmation.

Data Analysis

All attributes in the SEI maintenance-data model were mapped into the SEI local model (see Figure 1 below). All the maintenance attributes in *QGA* maintenance forms (i.e. *System-Investigation-Request* (SIR) Form, and SAP Transport Request Form), and SIR maintenance database were mapped into the *QGA* local maintenance-data model. The SEI is used as the base, standard model; all of its attributes are applied to the global maintenance-data-model. Each of these attributes is compared and cross-referenced with all the attributes in the *QGA* local maintenance-data model. Information obtained from the interviews on both maintenance forms, and database are consolidated to facilitate the attributes mapping-process between the *QGA* local maintenance-data

model and the SEI local model. The mapping of *QGA* attributes onto SEI is based on the key objectives of these attributes, and is validated through an iterative-feedback process with the *QGA* senior managers to enhance the accuracy and reliability in mapping the attributes. Attributes in the *QGA*-maintenance-model that are found to be similar (i.e. in terms of their objectives) with the SEI are identified and marked. The remaining attributes in the *QGA* maintenance-data model, which are dissimilar to the SEI model, are then integrated into the global maintenance-data-model. Following this, related literature was consulted to further improve the quality and relevance of the proposed maintenance-data-model.

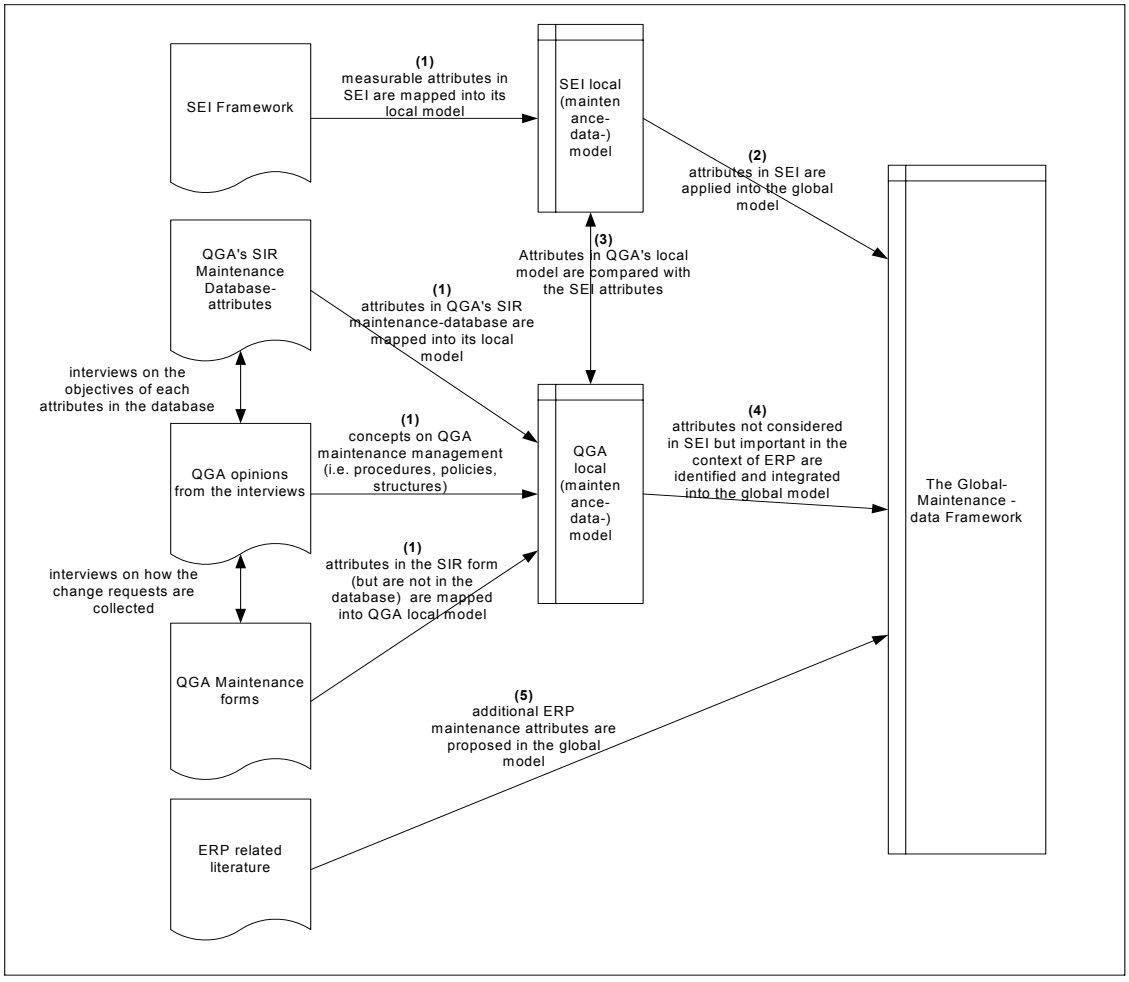

Figure 1: The process of mapping maintenance attributes into the proposed maintenance-data-model of ERP maintenance

Note that the following discussion is focussed on maintenance request that could cause some changes to be made to an ERP system. Maintenance request for user-support that is associated with user training, ERP system usage and software functionality enquiries is not covered here. Thus, maintenance request and change request is used interchangeably in this text.

FINDINGS

From our analysis, we found that the SEI maintenance-data-model includes most of the fundamental ERP maintenance attributes. Nonetheless, it is quite general, and insufficient to capture: (i) more specific ERP maintenance attributes such as the 'vendor change request number' (to indicate that a request was satisfied using readily available patches from the vendor), and 'functional area' (representing the business application area(s) involved in the maintenance), and (ii) other relevant maintenance attributes such as 'resolution impact' (i.e. impact on user training and online documentation), 'work type' (to identify different types of requests), 'approval to migrate' (to transport a change request from the Development System to the Quality Assurance System and to the Production System), 'service desk reference' (an identification number assigned to each maintenance request that arrives at the service desk (or help desk), 'training updated' (to indicate that training material has been updated), and 'online documentation updated' (to show that online documentation has been updated).

Deficiencies found in the *QGA* maintenance-data-model are: (i) inexhaustive values for some attributes such as 'problem status' (e.g. does not capture common-states like *assign-for-evaluation*, *assign-for-resolution-design*, *assign-for-resolution*, *assign-for-transport*), and (ii) lack of the attribute 'uniqueness' to systematically record maintenance activities.

These observations suggest a need to derive a new maintenance-data-model and provide a more comprehensive ERP maintenance-data-model. Table 1 is a detailed cross-reference of the attributes between the SEI and *QGA* maintenance-data-models. The first column of the table displays the fundamental maintenance attributes common to these activities. The second and the third columns show the attributes observed (present indicated by the symbol "X" and not present by the symbol "–") in the SEI and the *QGA* maintenance-data-models respectively. Where *QGA* uses a different descriptor for an SEI attribute, the *QGA* term is displayed in the *QGA* column in parentheses. The rightmost column in Table 1 describes the objective of each of the attributes.

Table 1: Cross-reference of the SEI and the *QGA* maintenance model attributes

Attribute		SEI	*QGA*	Objective
Investigation Stage	Problem ID	X	X (SIR[a] #)	Uniquely identify each change request or maintenance request
	Product ID	X	X	Identify the software product to which problems refer; also used by *QGA* for billing purpose (to client agencies).
	Problem type	X	X	Classify the problem into several categories to facilitate problem resolution
	Criticality	X	X (Priority)	Measure of the importance of a request to the system users
	Urgency	X	–[b]	Priority of a request assessed by the system-maintenance managers
	Finding activity	X	X (Problem Descrip-tion)	Refer to the activity, process, or operation taking place when the problem was encountered
	Finding mode	X	–	Identify whether a maintenance problem was discovered in an operational or in a non-operational environment
	Date of occurrence	X	X (Date Raised)	Date at which a problem occurred
	Time of occurrence	X	X	Relative time at which a problem occurred
	Originator	X	X (Raised By)	Determine the originating person; helpful in identifying the environment-specific and source-specific problem
	Environment	X	X (Test Phase)	Determine if problem is uniquely related to a specific functional category; identify if a particular functional category tends to generate abnormally large number of maintenance request (i.e. correction or enhancement)
	Projected availability	X	–[b]	The date when the request resolution is committed to be available
	Work Type	–	X	Identify the categories of maintenance requests (e.g. corrective, enhancement)
	Functional Area	–	X	Represents the business application area(s) associated with a maintenance request
	Service desk reference	–	X	Reference number issued when a system user initiates a maintenance request at the service desk
	Action to be taken	–	X	Show whether a request is approved by the systems manager; and allow identification of the # of requests being rejected or deferred
	Issues of consideration	–	X	Identify future issues related to a change request that is deferred
Resolution Stage	Uniqueness	X	–	Differentiate between a unique and a duplicate maintenance problem
	Problem status	X	X[b] (Status)	Indicate the job-status of a maintenance request (such as in-progress, user-testing, on-hold, awaiting client quote, closed)
	Problem status date	X	X[b] (Date Actioned)	Record the date of a request when its states (e.g. opened, closed, assigned for evaluation) changes; it is important to track the time spent on analysing and resolving the request, and to identify delay incurred
	Defect found in	X	X (Description of changes)	Identify the software object(s) containing defects, which cause a problem; identify software units that are prone to errors from one release to another
	Changes made to	X	X (Description of changes)	Identify the software object(s) changed to resolve the discovered problem; identify software units prone to change due to correction and/or enhancement; to discover software volatility

	Related changes	X	X	List of software object(s) required to be changed in resolving the problem in question (including training needs and documentation); useful to estimate time required to resolve a request
	Approved by	X	X	Indicate that the maintenance solution has been approved by the fixer
	Quotation	–	X	Indicate the estimated cost of implementing the maintenance request (in the QGA case, this attribute is used for the user-initiated enhancement request only)
	Resolution	–	X	Describe how the maintenance problem is resolved, and contact person for resolving the problem
	Estimate time	–	X	Estimate of time (in hours) to complete the maintenance request
	Resolution impact	–	X	Identify whether online help needs to be changed; and user training is needed as a result of the maintenance solution
	Vendor change request number	–	X	Identify whether a maintenance request is satisfied by using the readily available patches distributed by the vendor
	Training updated	–	X	Indicate that training material has been prepared and updated in relation to a change request
	Online documentation updated	–	X	Indicate that the online documentation has been updated in relation to a change request
	Released	X	X (Completed)	The date when the maintenance solution is released; useful to identify the efficiency of the maintenance project-management in meeting the projected deadlines
Delivery Stage	Approval to migrate	–	X	Identify whether a maintenance solution is approved to migrate to the Quality Assurance System and/or the Production System
	Applied	X	X (Transported on)	Show the date the maintenance solution was applied to the problem-originating site
	Accepted by	X	X	Indicate that the maintenance solution has been accepted by the system users

[a] An abbreviation for System Investigation Request.
[b] Partially followed by *QGA*.

This paper seeks to rationale from two data-models employed in two different but substantially overlapping software maintenance environments - (1) custom in-house, and (2) large application packages, or more specifically Enterprise Systems. We assume that those attributes that are common to both data models are valid, and thus we focus discussion on those found to be unique to either SEI or QGA.

Observations are next made on differences examined between the SEI and *QGA* maintenance-data-models, and they are summarized in Table 2. Discussion is organised around the following three sub-headings: (a) investigation stage – the initial stage where a maintenance request is defined, categorized, analysed, and approved; (b) resolution stage – the cause(s) of a change request are examined, strategy to satisfy a request is designed, the available source of information is consulted and the newly acquired knowledge pertinent to a request-solution is documented, and the change request is implemented and approved to be transported from the Development System to the Quality Assurance System; and (c) delivery stage – the change is tested in the Quality Assurance System and transported to the Production System, and the change is accepted by the system users (or the raisers).

Investigation Stage

Table 2: Summary of main differences

Attribute	SEI	*QGA*	ERP-Specific
Investigation			
Urgency	X	–*	
Finding mode	X	–	
Projected availability	X	–*	
Work type	–	X	
Functional area	–	X	X
Service desk reference	–	X	
Action to be taken	–	X	
Issues of consideration	–	X	
Resolution			
Uniqueness	X	–	
Quotation	–	X	
Resolution	–	X	
Estimate time	–	X	
Resolution impact	–	X	
Vendor change request number	–	X	X
Training updated	–	X	
Online documentation updated	–	X	
Delivery			
Approval to migrate	–	X	

* Partially followed by *QGA*.

Attributes in SEI but not *QGA*

The SEI model incorporates most of the attributes of the change request at the Investigation Stage. The attributes 'criticality', and 'urgency' allow both system users and system managers to indicate the importance of the maintenance request from their respective perspectives. These attributes provide information required to prioritise maintenance requests. It is observed that the *QGA* maintenance-data-model does not explicitly capture the attribute of 'urgency', which can be an important factor in cost/benefit justification of a maintenance request. In contrast, *QGA* assigns the 'urgency' of a request dynamically based on current-basis (or day-to-day) maintenance demand.

Neither does *QGA* record the 'finding mode' that describes whether a problem was found in an operational environment. This attribute is important for understanding the problem behaviour with in-house software in order to reproduce the error. However, this attribute is perceived to be less useful by *QGA* in their ERP/SAP environment as all the change requests reported by users are expected to occur in the operational mode (i.e. during system execution). ERP-employing organizations do not usually perform formal reviews of the ERP software or conduct software inspections. Therefore, problems or defects are unlikely to be found in the non-operational environment.

QGA captures the attribute of 'projected availability'. It is recorded for user-enhancements requests only. This attribute is not recorded for any other requests such as corrective/bug fix.

Attributes in *QGA* but not SEI

The *QGA* model suggests that the SEI maintenance-data-model may be improved by capturing the business application area(s) involved (the 'functional area'-attribute) in a change request. Unlike single functional-area in-house software or small application package software, ERP software comprises multiple functional-areas or business applications. This attribute is useful in an ERP context to identify the business area(s) that is more volatile or error-prone.

We further note that the SEI model mainly concentrates on 'corrective' maintenance. An attribute capturing the maintenance-classification ('work-type') is highly relevant and critical. A typical maintenance system (regardless of in-house or ERP software) receives various types of maintenance requests. These are usefully categorized using an appropriate taxonomy (see (Ng et al. 2002) for recent work on identifying an ERP-specific taxonomy) for ongoing monitoring and analysis purposes. This step is important in any maintenance model in order to evaluate the frequency distribution of different types of maintenance requests over time, and to identify ongoing maintenance resource requirements. The attribute of a 'service desk reference' is also not considered in the SEI maintenance-data-model. Service/help desk is employed in the *QGA* maintenance-data-model to effectively help and liase with the system users regarding the ERP system usage, training, software functionality and maintenance problem, and to report maintenance requests to the ERP maintenance teams.

The final two attributes found in the QGA model but not SEI in this stage are 'action to be taken', and 'issues of consideration'. These are used to record whether a maintenance request is qualified for 'go-ahead' and is implementable. For instance, if a request is deferred the latter attribute would describe the issue(s) that need to be addressed before maintenance would be carried out for the request.

Resolution Stage

Attributes in SEI but not QGA

The SEI model includes several attributes to aid in resolving a maintenance request. These include identifying whether a request/defect is unique. While 'uniqueness' is perceived to be important in SEI as a warning mechanism to identify duplicate request problems, the *QGA*-model does not capture this attribute.

Attributes in *QGA* but not SEI

In contrast, deficiencies of the SEI maintenance-data-model – observed in the *QGA* context include: (i) 'quotation' for the enhancement maintenance request, (ii) how changes are made and who (the maintainer) is in-charge of resolving a particular maintenance area ('resolution'-attribute), (iii) impact of the maintenance changes on user-training material and online documentation (the 'resolution impact'), (iv) estimation of the amount of effort required for a maintenance request (i.e. the 'estimate time'-attribute), and (v) whether maintenance support is available from the vendor (i.e. 'vendor change request number'). The 'quotation'-attribute is particularly crucial in the ERP context in general, and *QGA* in particular as a service provider, in order to: (i) confirm the user willingness to pay for the maintenance, (ii) keep track of maintenance-charges for respective users, and (iii) issue an invoice to the users. In general, the attributes of 'resolution', 'training updated', and 'online documentation updated' are essential information because they: (i) facilitate the identification of appropriately experienced maintainers who can be recruited for similar maintenance problem-areas, and (ii) ensure that documentations of all tasks involved in satisfying maintenance requests are updated and verified accordingly. Although the attributes such as 'resolution', 'resolution impact', and 'estimate time' are also valuable in the in-house software maintenance environment, the attribute of 'vendor change request number' is unique to the ERP maintenance environment. The 'vendor change request number' is necessary to identify whether custom code was developed or standard code (distributed by the vendor) was previously applied to satisfy a maintenance request. This piece of information is highly critical to determine whether the maintenance job would be affected by the future maintenance or upgrade exercise. (If the standard code had been used, it would have no impact on future patch-maintenance or new version upgrade.) This attribute is also fundamental to avoid any unnecessary maintenance efforts and shorten the turnaround time in servicing some maintenance requests, particularly the corrective maintenance.

Delivery Stage

SEI suggests the good-practice of documenting activities involved in delivering maintenance-solutions to system-users. These include capturing the attributes of 'applied' (that indicates when a change request is implemented at the users' site), and 'accepted by' (to show that the problem-resolution has gone through the user-acceptance test). These attributes are also present in the *QGA* maintenance-data model. However, the SEI lacks the attribute of 'approval to migrate'. This attribute is used by *QGA* to indicate that the ERP system managers have approved the maintenance resolution to be transported from the Development System to the Quality Assurance System, and to the Production System.

Additional Attributes for Consideration in ERP Maintenance-data-model

In the previous discussions, we found that both SEI model and *QGA* maintenance-data models have their strengths and limitations. In light of this, this section proposes a new ERP maintenance-data-model by first integrating the essential attributes seen in the former two models, and then incorporating additional fundamental ERP maintenance attributes that are perceived to be highly relevant in the ERP-context.

According to the earlier studies, ERP enhancement is the major maintenance activity and the most effort demanding (Glass and Vessey 1999, Ng et al. 2002). Thus, Ng, Guy and Chan (2002) propose a categorisation for classifying the ERP enhancement maintenance based in the ERP benefit-taxonomy. This approach is important in order to justify the value of the benefit(s) of doing the enhancement request, and the cost to implement the request (see (Ng 2001) for more details). As reported by Brehm, Heinzl and Markus (Brehm et al. 2001), there is a range of options available in order to accomplish the enhancement maintenance request. This includes using the system configuration switches, user-exits or add-ons, workflow programming, ERP programming, interface development, writing reports and screens, and writing forms (e.g. using the SAPscript for the SAP forms), to name a few. The authors note that different types of enhancement methods / options pose different impacts on the future ERP maintenance effort. This is because most of the system users' enhancements involve making (some) changes to the standard ERP code. As a result, each enhancement has its associated ongoing maintenance cost or effort. Based on these literatures, the additional attributes suggested in the new ERP maintenance-data-model are as summarised in Table 3 below.

Table 3: The additional attributes for ERP maintenance

Additional attribute	Rationale
Benefit category	Identify the categories of benefits delivered by the enhancement request (e.g. operational cost reduction, integrated system, competitive-advantages)
Cost benefit	Estimate the value of the benefit delivered by the enhancement request
Implementation cost	Estimate the cost of implementing the enhancement request
Enhancement type	Indicate the enhancement method used to accomplished the enhancement request
Enhancement area	List the software object(s) and/or business application area(s) affected by the enhancement request
Ongoing maintenance cost	Justify the subsequent maintenance cost

These attributes are aimed at the Investigation Stage. They are included because they are important in order to: (i) justify whether to approve for the enhancement requests for the ERP system, (ii) reduce the total maintenance cost or total ERP software costs, (iii) facilitate making the optimal decisions on ERP maintenance, and (iv) reduce the time required to plan for the future maintenance and upgrade project (– because the information of number of enhancements, area of enhancement, method used in the enhancement activity, and the impacts of the enhancements on future maintenance would be readily available). (It is noted that more considerations are given to the enhancement maintenance requests here.)

As for illustration, Figure 2 shows the proposed ERP maintenance-data model at the lower part of the diagram; it incorporates the general-purpose SEI maintenance attributes, *QGA* (i.e. an ERP-context) maintenance-data model, and the additional ERP maintenance attributes. On the other hand, the upper part of the diagram depicts the preliminary ERP maintenance model from the *QGA*; it briefly describes *QGA*'s maintenance process starting from the point when a maintenance request is initiated (at the Investigation Stage) until the request-solution is transported by *QGA*'s Technical Team (in the Delivery Stage) to the system users in the Production System.

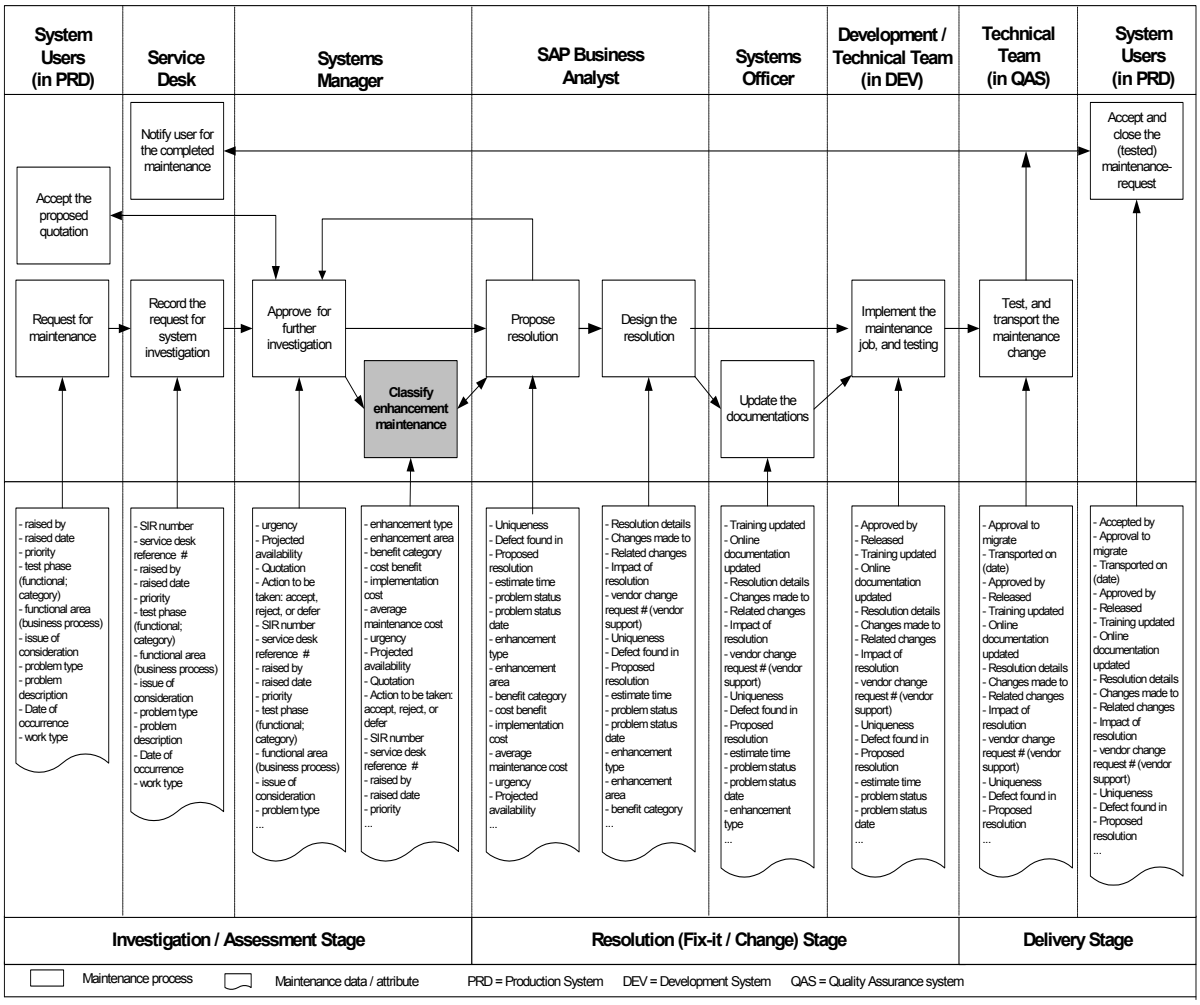

Figure 2: The proposed maintenance-data-model for ERP maintenance

CONCLUDING REMARKS

Implication For Practice

Practically, the proposed ERP maintenance-data-model is useful to ERP-using organizations for three main purposes: a repository of maintenance activities and maintenance knowledge, a maintenance controlling and monitoring system, and a maintenance improvement and decision-support tool.

A repository of maintenance activities and maintenance knowledge

The proposed model consists of well-defined and organized attributes for measuring maintenance activities. All the knowledge regarding the maintenance activities, for instance, the causes and effects of maintenance problems, remedies for similar maintenance issues, status / progress of maintenance jobs, up-to-date documentation of maintenance changes, and staff involved in all maintenance activities, are captured in the model. This can facilitate subsequent monitoring of maintenance progress, ensuring that the maintenance staffs are communicating in the same language and reporting the maintenance activities accurately; and, allowing quality-data for future data-mining and to forecast future maintenance activities, for example in predicting the upcoming maintenance requests (as in the study done by Burch and Kung (1997)).

Attributes such as 'work type', and 'problem type' allow maintenance organizations to quantify the amount of different types of maintenance requests, and types of maintenance problems initiated per month and/or per year. A well-organized maintenance-system allows the maintenance manager to be well informed about maintenance activities in his organization, and enables an ERP-using organization to maintain all the maintenance knowledge (current, past and future) within its organization.

A maintenance controlling and monitoring system

With the maintenance-data-model, the maintenance managers can be alerted when an abnormally large amounts of certain maintenance request-types ('work type') increases over time. They can then decide on appropriate action(s) by justifying whether it is aligned with the organization's business objectives; otherwise, action can be taken to alleviate and reduce the amount of the request-type(s).

On the other hand, if there are a large number of duplicate maintenance requests (by referencing the 'uniqueness'-attribute), the maintenance manager will be warned so that the right steps can be performed such as revising the testing procedures, rewriting the error-prone code or re-assessing the procedures involved in capturing the users' requirements. By checking both attributes of 'changes made to' and 'related changes', the maintenance managers can easily judge the impacts of a change request, and also justify whether user training is required after the maintenance is implemented.

Comparison can be made between the 'date raised' and the actual 'applied' date of the maintenance-solutions to identify whether the system-users are facing any serious delays; and then using the 'problem status' and 'problem status date' attributes to determine the maintenance bottleneck. Nevertheless, the maintenance backlog can be easily computed. With these indicators, appropriate control can be carried out before an unexpected situation deteriorates.

A maintenance improvement and decision-support tool

With the core maintenance information readily available, ERP managers are better informed about the status of their ERP systems and maintenance conditions. For instance, with both attributes of 'urgency' and 'criticality', requests with highest priority can be easily identified. With attributes such as 'enhancement type', 'benefit category', 'cost benefit', 'implementation cost', and 'maintenance cost', managers can quantify the costs and benefits (or tradeoffs) of implementing an enhancement maintenance request. Hence, maintenance managers can make better-informed maintenance decisions. Making the right decision is crucial to reducing the user opportunity cost of selecting other options.

ERP maintainers can serve the change request more efficiently as the user's requirements are collected in an organized and systematic manner. All maintenance data are easily accessible from the proposed maintenance-data-model. This can increase maintenance productivity – no time is wasted looking for or collecting the required information. Thus, more maintenance requests can be completed in a given time period.

By comparing the maintenance priorities (i.e. both 'urgency' and 'criticality') of each maintenance-request with its expected time for completing the request (i.e. using the attribute of 'estimated time'), managers can easily allocate and schedule maintenance staff to each of the maintenance requests. With the historical data on maintenance request-types ('work type' and 'problem type') over a time period, a manager can forecast the amount of different types of maintenance requests and can estimate whether its existing maintenance task force is sufficient to meet future maintenance-request demands.

Limitations and Future Directions

Extensibility and generality of these study findings is limited due to our studying only a single case organisation (Baskerville and Lee 1999). The proposed maintenance-data-model may not be complete. Longitudinal and multiple case studies, and possibly a survey across other government agencies and private sector organisations of all sizes and industries, are warranted in order to further validate and extend the maintenance-data-model proposed in this study.

A future research objective of the attributes is to extend the existing attributes by considering maintenance activities for patches introduced by the vendor, and designing and developing a comprehensive set of paperless maintenance forms to manage and record all maintenance activities.

REFERENCES

Baskerville, R. and Lee, A. S. (1999) *Proceedings of the International Working Conference on New Information Technologies in Organizational Processes: Field Studies and Theoretical Reflections on the Future of Work*, St. Louis, Missouri, Kluwer Academic, August 21-22.

Brehm, L., Heinzl, A. and Markus, M. L. (2001) *Proceedings of the 34th Hawaii International Conference on Systems Sciences*, Hawaii, USA, IEEE Computer Society: Los Alamitos, CA,

Burch, E. and Kung, H. J. (1997) *Proceedings of the International Conference on Software Maintenance*, Bari, Italy, IEEE Computer Society: Los Alamitos CA, October 1-3.

Florac, W. A. (1992) *Software Quality Measurement: A Framework for Counting Problems and Defects,* Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania.

Gable, G. G. (1994) Integrating Case Study and Survey Research Methods: An Example in Information Systems, *European Journal of Information Systems,* **3,** 112-126.

Girard, K. (2000) Back-office Rebirth, URL: http://www.business2.com/ http://www.business2.com/.

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

IEEE (1998) *IEEE Standard for Software Maintenance*, *IEEE Std 1219-1998,* Institute of Electrical and Electronic Engineers, New York.

James, D. and Wolf, M. L. (2000) A Second Wind for ERP, *The McKinsey Quarterly,* 100-107.

Kajko-Mattsson, M. (1998) *Proceedings of the International Conference on Software Engineering*, Kyoto, Japan, IEEE Computer Society: Long Beach CA, April 19 - 25.

Ng, C. S. P. (2001) A Decision Framework for Enterprise Resource Planning Maintenance and Upgrade: A Client Perspective, *Journal of Software Maintenance and Evolution: Research and Practice,* [Accepted for publication].

Ng, C. S. P., Gable, G. G. and Chan, T. (2002) A Client-benefits Oriented Taxonomy of ERP Maintenance., *Journal of Systems and Software,* [Accepted for publication].

Pigoski, T. M. (1997) *Practical Software Maintenance: Best Practices for Managing Your Software Investment,* John Wiley & Sons, Inc., New York.

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

COPYRIGHT