Towards an Understanding of Risk Management in Legacy Information Systems Modernisations

David Warrell
Kenneth J Stevens

SEAR: Security, E-Business, Assurance Research Group
School of Information Systems Technology and Management
University of New South Wales
e-mail: k.stevens@unsw.edu.au

Abstract

The modernisation[1] of legacy information systems is a problematic and risky process that many organisations will undertake at some stage during their lifetime. When modernising a legacy information system a number of different approaches can be used, ranging from a simple 'updating' of the existing system to its wholesale replacement. Little appears to have been written about the problems and risks that relate specifically to these different legacy information system modernisation approaches. Understanding these risks would seem important for project managers when managing modernisation projects and also for decision makers when deciding which approach is most appropriate for a particular project. This paper outlines the initial stages of a research project that to seeks identify, understand and assess the risks involved in both the modernisation process itself and the ongoing use of the modernised system across these various types of legacy system modernisation.

Keywords

Risk management, software project risk, software risk, legacy system modernisation.

INTRODUCTION

Risk management is a current concern in the Information Systems discipline. The development, implementation and operation of information systems are risky undertakings (Jiang and Klein, 2000). Particular attention has been focused on risk management as a result of recent events and trends in the Information Systems practice:

- Y2K saw organisations the world over scanning through millions of lines of poorly understood and often undocumented code written in obsolete languages. Although we now know that the doomsayers were being overly pessimistic with their predictions of widespread system malfunctions and failures, the risks involved in the hasty repair efforts were nevertheless considerable, and the consequences of failure potentially catastrophic (Chorafas, 1999; Li et al., 1999; Crawford, 2001).
- Enterprise Resource Planning (ERP) system implementations required the integration of numerous disparate systems in order to allow for the provision of the elusive enterprise-wide view. A large amount of risk was involved as systems that were never intended to talk to each other were integrated via a series of cobbled-together interfaces (Gamble, 2000).
- The advent of e-commerce saw many organisations hurrying to create a web presence. The development of e-commerce systems entailed a range of risks due to their heterogeneous user environment and high degree of interaction with other systems (Gruhn and Schöpe, 2002).

The recent development of several country-specific and universal standards pertaining to risk management has further contributed to the prominence of the topic. Australian and New Zealand Standard AS/NZS 4360:1999 provides a generic risk management framework, encompassing the context, identification, analysis, evaluation, treatment, monitoring and communication of risk (Standards Australia, 1999).

[1] "Modernisation" as used throughout this paper has a slightly different meaning to that assigned to it in Comella-Dorda et al. (2000) and Seacord et al. (2003). For the purposes of this paper, "modernisation" will be defined as any activity undertaken in an effort to improve the usefulness, functionality and/or other quality attribute of a legacy system. This may include anything from "wrapping" the existing UI with a new one using "screen-scraping" software, through to a complete replacement of the legacy system.

The rapid rate at which businesses have adopted and developed information systems over the years has had many and varied effects. Workforces have been rationalised, business processes automated or reengineered for greater efficiency and new alliances and product and service offerings have been made possible. Information systems were adopted first in organisations' core business processes, where automation could give the greatest potential return. However, the push for rapid development and implementation in order to secure the valuable 'first to market' advantage led developers to neglect system documentation in an effort to accelerate the development and implementation process. Meanwhile, management have traditionally been loath to look back and take stock of their existing IS infrastructure, preferring to engage in new initiatives. As a result of the combination of these factors, we have 'legacy' information systems: brittle, poorly understood systems that enable core business processes, and may have been doing so for many years.

Legacy information systems are a phenomenon experienced by virtually every enterprise. These systems, which may have been present in the organisation for as long as 30 or 40 years, are stable, but inflexible, and made brittle by years of ad-hoc maintenance and enhancements. This brittleness and inflexibility makes legacy systems difficult and expensive to maintain. It has been reported that, on average, 60 to 80 per cent of IT budgets are spent on maintaining legacy applications and the mainframe systems they run on (Kaplan, 2002). Previous studies put the figure at between 50 and 70 per cent (Lientz and Swanson, 1980; Nosek and Palvia, 1990), suggesting that the expense of maintaining these systems is growing as they continue to age. Recently, the advent of e-business saw businesses scrambling to become 'web-enabled'. In the process, many organisations were finding that their legacy systems were hindering their ability to move into the online space, and to cope with the new products, services and transactions demanded by this new way of doing business.

Making changes to legacy systems, in terms of integrating them into other systems, or modernising them, can be a complex and difficult process, as has been acknowledged by practitioners (Kuipers, 1995; Schneider and Feffer, 1999; Gamble, 2000; Jaklevic, 2001; Bass, 2003) and researchers (Song, 1996; Bergey et al., 1999; Bisbal et al., 1999; Canfora et al, 2000) alike. These changes, like any change in information systems, must represent a source of risk both during and subsequent from the change. However, leaving legacy systems as they are represents another source of risk to the organisation. When a legacy system is to be modernised, that modernisation can take various forms, from simple refurbishment to wholesale replacement. Various factors can be seen to influence the decision as to which modernisation approach to take. By understanding the risks associated with each of the approaches, those risks can be considered in the decision as to which modernisation approach should be adopted.

This paper provides an overview of a research project that seeks to identify the key risks associated with legacy systems modernisations and to form an understanding of how the risks are being considered and managed in these projects.

BACKGROUND

Legacy information systems are the brittle, inflexible and poorly understood, yet stable and mission-critical systems that exist in the vast majority of established organisations. They present risks to their host organisations in their current state (primarily strategic business risks associated with their expense and inflexibility), but attempts to modernise them may also be fraught with range of difficulties.

Risk management is becoming an increasingly important area of concern as organisations become more open and accountable to their stakeholders. Corporate governance requirements and international and country-based standards are adding to the weight of the risk management agenda. Many approaches have been suggested for understanding and managing organisational risks both generally and in relation to information systems. However, many of these are complex and have failed to gain wide practitioner acceptance.

Legacy Systems

Examining the literature pertaining to legacy information systems reveals a wide variety of definitions of what exactly a 'legacy system' is. Zou and Kontogiannis (2002) suggest some of the reasons why legacy systems are undesirable when they describe legacy systems as "mission critical software systems that are still in operation, but their quality and expected operational life is constantly deteriorating due to prolonged maintenance and technology updates" (p. 530). Seacord et al. (2003) claim "Software systems become legacy systems when they begin to resist evolution and modification" (p. xiii). Comella-Dorda et al. (2000) neatly sum up the legacy system dilemma by drawing an analogy between the legacy information system and the brain (p. 1):

> *"In many ways, these information systems are to an enterprise what a brain is to the higher species – a complex, poorly understood mass upon which the organism relies for its very existence."*

Common to most discussions on legacy systems is an acknowledgement of their inflexibility in terms of functionality and integration with other systems, and a 'brittleness' introduced by years of maintenance and enhancements. Also identified are the high levels of expense incurred in order to maintain the legacy applications and the obsolete hardware required to run them. As the systems age, finding individuals with an understanding of the systems and experience with the technologies involved becomes increasingly difficult.

Despite these problems, legacy systems have two very important advantages:

- They have been in operation for so long that they are very stable. For many systems, years of operation have exercised all paths of execution within the system, and have exposed bugs that even the most comprehensive testing programme could not hope to detect. As these bugs have been detected and fixed over the years, these systems have reached a level of functional stability that no new system could hope to match initially. Even where problems still exist in the system, there are usually well-established, if occasionally cumbersome, business processes in place to accommodate these shortcomings.
- They are often crucial to the on-going operations of the business. The high cost of IT in the early days of its adoption for organisational automation dictated that it be introduced into the most important parts of the business, where productivity gains due to automation would show the greatest returns (Liu and Sharp, 1994).

Legacy Information Systems Modernisation Approaches

Legacy information systems pose a dilemma for their host organisations. One the one hand, they are brittle, inflexible and expensive to maintain. On the other, they form the stable core of the organisation's operations. Undertaking a replacement or large scale reengineering initiative has large implementation risks associated with it. On the other hand, making minimal changes/adaptations to the legacy system in order to mitigate implementation risk means that much of the inherent inflexibility of the system remains. In response to this trade-off between shot-term and long-term risk, a number of different strategies have been developed for coping with legacy information systems.

Many authors in the legacy systems field have described taxonomies of modernisation approaches. Most taxonomies include maintenance as a modernisation activity. However, this is often only included for completeness, and its inclusion is occasionally problematic given the definitions adopted (e.g., Bisbal et al., 1999). At a basic level, many authors (e.g., Liu and Sharp, 1994; Edwards, 2002) identify two approaches to legacy system modernisation:

- Wrapping, or using 'screen-scraping' software to create a new user interface for the legacy system.
- Replacement of the legacy system with a new commercial off-the-shelf (COTS) or in-house developed system[2].

More complete taxonomies of legacy information system modernisation approaches demonstrate that these two approaches lie towards the extremes of a continuum of approaches. This continuum has been elaborated with varying numbers of intermediate points by a number of authors (Weiderman, 1997; Bisbal et al, 1999; Comella-Dorda et al, 2002; McNurlin and Sprague, 2002). Seacord et al. (2003) present one of the most comprehensive taxonomies of modernisation approaches, outlined in Figure 1.

[2] It could be argued that the complete replacement of a legacy system is not in fact modernising that system, as no part of the original system remains. However, replacement is included in this consideration of modernisation approaches as it still represents an attempt to expand the functionality provided by a legacy system.

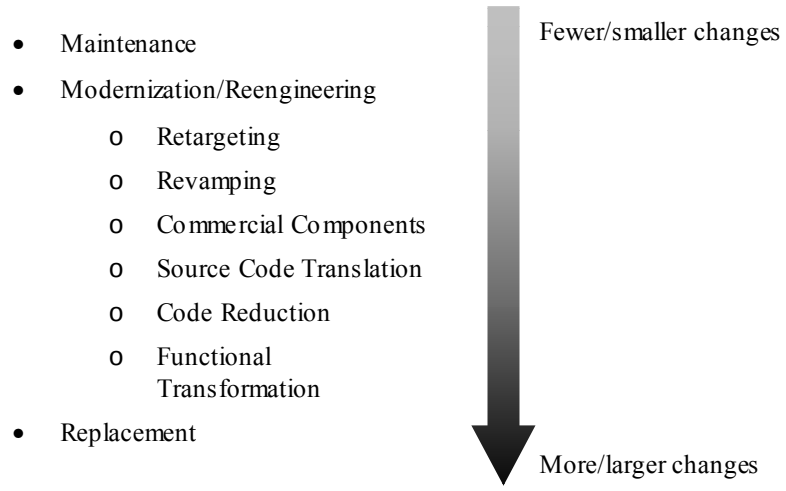

Figure 1. A taxonomy of legacy systems modernisation approaches (adapted from Seacord et al., 2003)

In addition to these taxonomies of modernisation approaches, much work has been done developing tools and/or methodologies applicable to specific modernisation approaches or parts thereof. Several working groups (Battaglia et al, 1998; Wu et al, 1997) are also working towards developing an end-to-end methodology and/or toolkit to assist in the modernisation or migration process.

Despite this large body of work supporting the modernisation process, little literature exists that seeks to examine the risks inherent in the various modernisation strategies and in the modernisation process in general. In addition, little has been written regarding the choice between competing modernisation strategies for a particular project. Ransom et al. (1998) present a method for assessing legacy systems, and suggest some modernisation options based on system characteristics determined in the assessment. However, the modernisation options suggested form a far from comprehensive list. In addition, the competing approaches' unique risk profiles are not considered as part of the assessment process.

RISK MANAGEMENT

Management of risks pertaining to information systems is becoming an increasingly important area of concern for organisations. Incorporation of IS risks into corporate governance reporting requirements and emerging risk management standards are forcing organisations to consider carefully how they are managing the risks involved in their technology. Accordingly, IS risk management has also become an important research area. Many authors have developed models of risk factors associated with IS, particularly IS implementation (e.g. Scott and Vessey, 2002). The risks associated with the use of IS are typically divided between software risk and software project risk.

Software Project Risk

Software project risks are the risks associated with IS development and implementation. Schmidt et al. (2001) define software project risk as the product of uncertainty associated with project risk factors and the magnitude of potential loss due to project failure, where a risk factor is defined as a condition that can present a serious threat to the successful completion of a software development project. The scope of software project risk is typically limited to the software development project only, and only those factors that can impact on the outcomes of the development process are considered. This constraint distinguishes software project risk from the broader concept of software risk, although numerous other definitions do exist.

The appropriate management of the risks associated with the development of a project have been found to be a consistent factor influencing project success (Jiang and Klein, 2001). As such, the management of these risks is considered to be of critical concern to project managers, as unmanaged or unmitigated risks are one of the primary causes of project failure.

Implicit in the management of these risks is the identification and understanding of the sources of these risks. The sources are those factors that can have a detrimental impact upon one or more of the success criteria of a project, such that they cause the project to run over time, cost more than anticipated or result in the application not having the functionality or usefulness required. The factors that apply to a specific project are seen to constitute that project's risk profile. Numerous various studies have derived lists of these factors, usually from

surveys of software project managers (e.g. Boehm, 1991; Barki et al., 1993; Moynihan, 1997; Ropponen and Lyytinen, 2001). Keil et al.'s (1998) list of the "Top 11" risk factors, as set out in table 1, is reasonably representative of these studies.

Rank	Risk Factor
1	Lack of top management commitment to the project
2	Failure to gain user commitment to the project
3	Misunderstanding of requirements by the developers
4	Lack of adequate user involvement in the project
5	Failure to manage end user expectations in regard to the project's outcomes
6	Changing scope and / or the objectives of the project
7	Lack of required knowledge / skills in the project personnel
8	Lack of frozen requirements, such that the requirements continue to change throughout the development project
9	Introduction of new technology
10	Insufficient / inappropriate staffing
11	Conflict between user departments

Table 1: Top 11 Software Project Risk Factors (Keil et al., 1998)

Software Risk

Software risk includes those risks, both short-term and long-term, faced by a business through the use of a particular application (or suite of applications). Numerous risks fall into this broad category, but software risk is essentially concerned with risks causing the compromise or loss of operation of the system, leading to the interruption of a business process. Software risk also incorporates the risks associated with adopting a particular type of system, in terms of the business' competitiveness both in the short- and long-term.

The impacts from the compromise or loss of operation of the system are both internal and external and arise because organisations are reliant upon the proper operation of the system to undertake business on a day-to-day basis. The inability to properly process transactions or access and provide information compromises this ability, hence exposing the business to numerous risks related to the inability to carry on business.

The sources of the compromise to the system's operation include the poor reliability of the application itself and the infrastructure on which it operates, the poor security of the system (and infrastructure), disasters and so on. These sources are quite often considered risk themselves.

The impacts include increased costs to manage the situation, lost business, lost customers, lost reputation, and an inability to comply with various standards and regulations. Many of these impacts have flow-on impacts, such as a legal liability that arises from the inability to complete a contract because of compromised operation of the system.

The risks associated with the adoption of a particular system are diverse, including:

- The strategic direction of the organisation, in terms of the contribution of the application to an organisation's competitiveness (both now and in the future). A risk arises where a system hinders the organisation's pursuit of strategic objectives.
- The technological direction of the organisation, given that the adoption of one class of technology is often at the exclusion of other types. These decisions may set the organisation on a particular technological path, which represents a risk, especially where that technology fails to develop.
- The fit of the application to the business' environment, such that businesses that operate in volatile environments may require systems that can readily accommodate environmental changes. A risk arises when the system cannot adequately adapt to changes in the environment.
- The organisational knowledge base is also impacted by the adoption of information systems. Risks arise where systems choices cause organisational knowledge to be lost. Loss of organisational knowledge may have ramifications for an organisation's ability to undertake its business processes or to retreat from a particular action, as has been the case in some instances of outsourcing.
- Other risks that are also considered within the scope of software risk include the political (both internal and external) risks associated with the development and roll-out of particular systems

Many of these software risks have traditionally been considered 'business risks' and hence the responsibility of the business unit that uses the particular information system to which the risk relate, rather than the domain of the provider of the information system. In light of the trends towards a closer relationship between information systems provision and the organisation using the systems, such as the distribution of the IS function into the business units and the close involvement of business unit in e-business development efforts, this demarcation would appear very much diminished in many organisations.

An example of such a change is risks related to branding and reputation in regard to e-business systems. Some evidence (Stevens and Fowell, 2003) suggests that these risks are now well within the domain of the IS developer. The customer facing nature of e-business systems means that any risk associated with the use of the system must be addressed to some extent in the development and operation of the system rather than by the business unit for whom the e-business system has been built.

LEGACY INFORMATION SYSTEMS AND RISK

The Keil et al., (1998) study, like other studies within this area, did not appear to have captured the software project risk factors in regard to any particular type or class of application, so these factors may be considered generic and applicable to systems development in general. A simple review of the Keil et al. (1998) 'Top 11' risk factors with respect to legacy information system modernisations reveals that many of these factors would be germane to almost all modernisation projects. A closer inspection reveals some of the risk factors appear more salient to particular types of modernisations than others. For example, Factor 9 (Introduction of new technology) and Factor 7 (Lack of required knowledge / skills in the project personnel) would appear to be of more importance to modernisations that are more towards the replacement end of the continuum. Conversely Factor 2 (Failure to gain user commitment to the project) and Factor 5 (Failure to manage end user expectations in regard to the projects outcomes) would seem more pertinent issues to modernisations that involve less change, although this would depend on the context.

A number of factors that have been found to be very problematic in regard to modernisations are not addressed (in a general way) in the list provided by Keil et al., (1998). These legacy-specific risks include the poor state of documentation (both of the legacy system and the systems to which it may have be integrated) and the state and complexity of the systems with which it needs to interact (if that is required in the modernisation). While Keil et al (1998) does represent a 'top' issues list and other, more extensive, lists are likely to include items that resemble the risks discussed, it does serve to highlight the sense that legacy information system modernisations do carry a number of risks which are specific to the domain. Like the more general risks identified above, these legacy-specific risks are also likely to vary between modernisation approaches, suggesting that further investigation may be required.

Similarly, a review of the five software risk factors previously discussed shows that the different approaches may create different risk levels across the different factors. The risks involved with strategic direction would seem most germane in regard to modernisation approaches as each approach could be seen to offer a different level of risk. A minimal modernisation effort may result in considerable risk in this regard for the organisation, as the system, being relatively unchanged from the original, may not meet the newer competitive demands of a changing industry. Alternatively, a complete replacement may offer minimal risk in regard to strategic direction as the current and future strategic plans of the business are incorporated into that system. This circularity in regard to the software risk, in that the different approaches themselves embody different business risk (both short term and long term), suggests a complicated decisional environment. This is turn suggests the need to form an adequate understanding of the inputs into the process – hence the need to investigate the software risks associated with the different modernisation approaches.

In considering how to go about investigating these two areas of risk it is apparent that the two classes of risk are intimately intertwined, a situation that is not necessarily encountered in other systems development projects. For example, a minimal modernisation, such as a simple maintenance, will represent only a modest software project risk, as there is very little to go wrong during minor changes. However, a minor change may represent considerable software risk in terms of the lack of fit to the organisation (as the legacy system prior to modernisation no longer did what the organisation needed and the minimal enhancements do little to address that) and strategic direction (the lack of change in the system means the business may become less able to compete). With these sorts of interactions in mind, any investigation of the software project risks and the software risks associated with legacy information systems modernisations should occur together.

The IS risk management literature contains two competing schools of thought regarding the way in which organisations manage risk (or 'uncertainty') in order to ensure organisational3 success. Structural contingency theories (e.g. Barki et al., 2001) state that the better the fit between a project's risk profile and the risk management approach applied to the project, the greater the chance of the project being successful. Risk-based theories, on the other hand, state that the effect on project success of a project's risk profile and the risk management approach adopted is mediated by another variable, performance uncertainty. Performance uncertainty is defined as the difficulty of estimating performance-related outcomes of a project; outcomes such as actual project cost, completion time and technical performance (Nidumolu, 1996).

Certain operationalisations of 'fit' within the structural contingency approach have been empirically shown to have questionable applicability in the real world (Nidumolu, 1996). However, structural contingency's basic premise of needing to tailor a risk management approach to each unique situation combines neatly with the postulate that different legacy system modernisation approaches exhibit different risk profiles. Together, these two ideas suggest that modernisation projects at different points along the continuum of strategies require different approaches to risk management if they are to have a maximal chance of succeeding.

Lyytinen et al. (1996) propose a hierarchical framework for risk management (figure 2), involving the management of risk in three environments: the management environment, the project environment and the system environment. These three environments incorporate both software project risk and software risk. The management environment lies at the top of the hierarchy, and is the environment that shapes software management activities. As the effects of risk management at this level propagate down the hierarchy, so too do risks themselves that arise as a result of insufficient managerial resources, skills or information. For example, management bias towards a particular architecture or supplier and a subsequent failure to give due consideration to all pertinent options may have a detrimental impact at lower levels of the hierarchy.

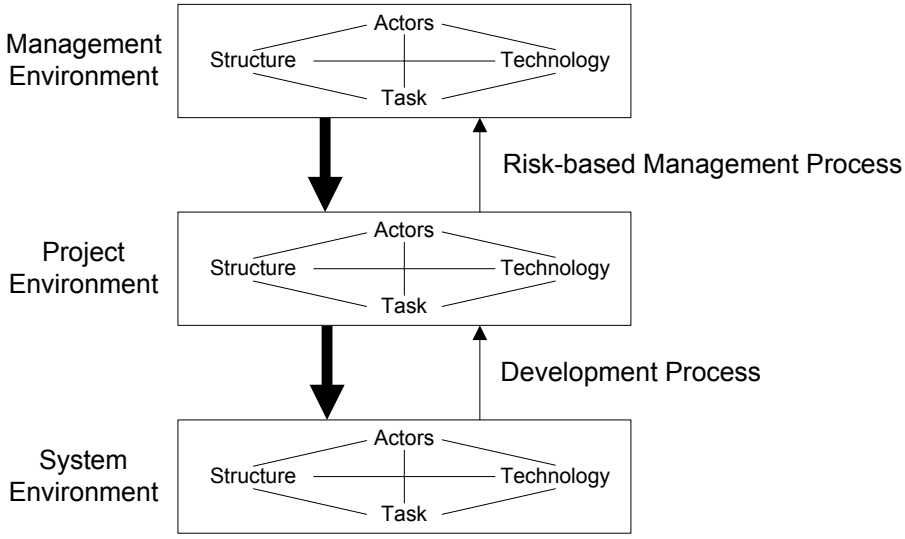

Figure 2: A framework for software risk management (Lyytinen et al., 1996)

It is in the project environment that the consequences of management environment risks begin to arise. The project environment is where the software development takes place, involving inquiry into the system environment in order to determine efficient ways in which to use software, and to subsequently implement a software system. As the project environment is concerned with describing a system environment and changing components of the existing system environment in order to achieve conformance with the newly articulated description, it follows that the system environment forms the context in which the components of the project environment are understood. Risks in the project environment may include lack of developer experience with management's chosen products or architecture

Finally, the system environment is the domain in which the software system operates. Risks that arise in the management and project environments will likely have impacts that are felt at this level. Poor management and/or poor project execution will more than likely lead to a system that is less successful than anticipated. A key risk in the system environment is that the delivered system will be of no benefit to the users, resulting in poor adoption of the system.

[3] Note that these approaches are applicable both to permanent organisations, such as the enterprise, and to temporary organisations, such as a systems development project.

Lyytinen et al. (1996) applied this framework in two case study situations, and found it a useful tool for describing, understanding and managing software risk. This study does not seek to empirically validate the framework. Instead, the framework will be used to structure and inform the data-gathering process in two ways. Firstly, the three levels of the framework indicate several stakeholder groups whose different perspectives on the project should be understood:

- Management Environment: Top-management decision makers
- Project Environment: Project manager(s), development and implementation staff
- System Environment: Operations manager(s), operations staff and business users

Secondly, the flows between the levels of the framework point to the need to consider the impact decisions taken and risks arising in each level of the framework will have in each of the other levels.

The above discussion suggests that firstly the risks associated with the legacy information systems modernisation process may be, at a minimum, sufficiently different from other development projects to warrant investigation. Secondly it is also apparent that the various approaches to modernisation that an organisation may consider have differing risk profiles, in terms of both the software project risk and the software or business risks they represent. In conjunction with the contingency model of risk management, this suggests that organisations should be customising a risk management approach to fit the chosen approach's risk profile. Once again these differences seem sufficient to warrant further investigation.

By understanding the role that risk plays in the legacy system modernisation process, legacy system stakeholders' awareness of risk can be enhanced. As a result, top management can take risk into account when deciding which legacy information system modernisation approach is the most appropriate, and project managers can adapt their risk management approach to the risk profile of the selected approach. This should ultimately lead to more successful legacy system modernisation projects.

PROPOSED RESEARCH METHOD

This research study seeks to answer the following questions:

- What are the risks associated with legacy information system modernisations?
- How important are these risks in the decision amongst alternative modernisation approaches?
- What are the current management and mitigation strategies adopted by organisations?
- Are these management and mitigation strategies unique to legacy systems modernisation projects?

Few studies have investigated software project risk with respect to a specific domain. Consequently, there appears to be no work in the fragmented legacy systems literature that investigates the risks peculiar to these systems. Given the increasing organisational emphasis on information systems risk, and the prevalence of legacy information systems within organisations, we consider this an important gap in the existing literature.

This study will seek to close this gap by attempting to identify the risks involved in the legacy system modernisation process. Our research questions include examining whether each modernisation process has a particular set of associated risks, and what impact these risks have on the choice of modernisation strategies within organisations.

Given the newness of this area of investigation and the highly individualised nature of the legacy system problem (both across organisations and across legacy systems within an organisation), the most appropriate research instrument was considered to be a single case study. Yin (1989) states that the single-case model is appropriate in new areas of investigation (the "revelatory case"). This study is not seeking to develop a set of 'one size fits all' guidelines for managing risk in legacy system modernisations. Rather, it aims to gain an understanding of the role risk plays in the modernisation process, and to examine current practices for managing and mitigating risk in legacy system modernisations.

Multiple sources of evidence will be used, including interviews with key stakeholders, documentation from the modernisation project, and information from the organisation's centralised risks and issues register. Interview questions will be developed with guidance from existing risk management literature (e.g. Barki et al.'s (2001) instrument). Questions will be composed to gain a greater understanding of the legacy system modernisation process, seeking to identify the role risk plays in the initial choice of a modernisation strategy, during development and implementation of the modernisation initiative, and during ongoing post-implementation operation. Interviews will be sought with one or more representatives from each of the stakeholder groups identified within the Lyytinen et al. (1996) framework. Content analysis will be performed on the collected data in order to discover the dominant themes regarding risk, management and mitigation in the case.

CONCLUSION

The modernisation of legacy information systems is a problematic process for many organisations. The preliminary analysis undertaken in this study suggests that:

- The risks associated with legacy information systems are not well understood, especially in regard to the risks presented by the various approaches to modernisation.
- The risks associated with legacy information systems may differ from the risks of other projects and that an understanding of these specific risks is important not only for their management during the development process, but as an input into the process for deciding which modernisation approach to pursue.
- Understanding these risks would seem worthwhile and would be of use to:
  - Decision makers choosing amongst alternative modernisation approaches
  - Project managers in the management of the modernisation projects

At this early stage of the research it is difficult to anticipate the outcomes. However, at the very least the research will enhance legacy system stakeholders' awareness of the risks involved in these projects, leading to a greater consideration of risk in the initial choice amongst modernisation approaches. In addition, this study will provide much needed research into risk management within the information systems context.

REFERENCES

Bass, A. (2003), 'Cigna's Self-Inflicted Wounds', CIO, March 15 2003

Barki, H., Rivard, S. and Talbot, J. (1993), 'Toward an Assessment of Software Development Risk', Journal of Management Information Systems, 10 (2), 203-225

Barki, H., Rivard, S. and Talbot, J. (2001), 'An Integrative Contingency Model of Software Project Risk Management', Journal of Management Information Systems, 17 (4), 37-69

Battaglia, M., Savoia, G. and Favaro, J. (1998), 'RENAISSANCE: A Method to Migrate from Legacy to Immortal Software Systems', Proceedings of the Second Euromicro Conference on Software Maintenance and Reengineering (CSMR'98), 197-200

Bergey, J., Smith, D., Tilley, S., Weiderman, N. and Woods, S. (1999), 'Why Reengineering Projects Fail', Carnegie Mellon University Software Engineering Institute Technical Report CMU/SEI-1999-TR-010

Bisbal, J., Lawless, D., Wu, B. and Grimson, J. (1999), 'Legacy Information Systems: Issues and Directions', IEEE Software, 16 (5), 103-111

Boehm, B.W. (1991), 'Software Risk Management: Principles and Practices', IEEE Software, 8 (1), 32-41

Brodie, M. and Stonebraker, M. (1995), Migrating Legacy Systems: Gateways, Interfaces and the Incremental Approach, Morgan Kaufmann, San Francisco

Canfora, G., Cimitile, A., De Lucia, A. and Di Lucca, G.A. (2000), 'Decomposing legacy programs: a first step towards migrating to client-server platforms', The Journal of Systems and Software, 54, 99-110

Chorafas, D.N. (1999), 'How long into the 21st century will the aftermath of the millennium bug last?', Information and Software Technology, 41, 951-956

Comella-Dorda, S., Wallnau, K., Seacord, R.C. and Robert, J. (2000), 'A Survey of Legacy System Modernisation Approaches', Carnegie Mellon University Software Engineering Institute Technical Report CMU/SEI-2000-TN-003

Crawford, W. (2001), 'Y2K: Lessons from a Non-Event', Online, March/April 2001, 73-74

Edwards, J. (2002), 'Reaching Back in Time', CIO, 15 (17)

Gamble, R.H. (2000), 'A Credit Manager's Guide to Systems Integration', Business Credit, January 2000, 36-40

Gruhn, V. and Schöpe, L. (2002), 'Software processes for the development of electronic commerce systems', Information and Software Technology, 44, 891-901

Jaklevic, M.C. (2001), 'Revenue stopper', Modern Healthcare, 31 (27), 36-38

Jiang, J. and Klein, G. (2000), 'Software development risks to project effectiveness', Journal of Systems and Software, 52, 3-10

Jiang, J. and Klein, G. (2001), 'Software Project Risks and Development Focus', Project Management Journal, 32 (1), 4-9

Kaplan, S. (2002), 'Now is the Time to Pull the Plug on Your Legacy Apps', CIO, 15 (11)

Keil, M., Cule, P.E., Lyytinen, K. and Schmidt, R. (1998), 'A Framework for Identifying Software Project Risks', Communications of the ACM, 41 (11), 76-83

Kuipers, W. (1995), 'Surviving a system conversion', Catalog Age, 12 (3), 67-70

Li, F., Williams, H. and Bogle, M. (1999), 'The "Millennium Bug": its origin, potential impact and possible solutions', International Journal of Information Management, 19, 3-15

Lientz, B.P. and Swanson, E.B. (1980), Software Maintenance Management: A Study of the Maintenance of Computer Application Software in 487 Data Processing Organizations, Addison-Wesley Publishing Company, Reading

Liu, K. and Sharp, B. (1994), 'A Strategic Attempt to Management of Legacy Information Systems', IEE Colloquium on Legacy Information System Migration – Barriers to Process Reengineering, 4/1-4/6

Lyytinen, K., Mathiassen, L. and Ropponen, J. (1996) 'A framework for software risk management', Journal of Information Technology, 11, 275-285

McNurlin, B.C. and Sprague, R.H., Jr. (2002), Information Systems Management in Practice, Prentice Hall, New Jersey, 323-328

Moynihan, T. (1997), 'How Experienced Project Managers Assess Risk', IEEE Software, 14 (3), 35- 41

Nidumolu, S.R. (1996), 'A Comparison of the Structural Contingency and Risk-Based Perspectives of Coordination in Software-Development Projects', Journal of Management Information Systems, 13 (2), 77-113

Nosek, T. and Palvia, P. (1990), 'Software Maintenance Management: Changes in the Last Decade', Journal of Software Maintenance: Research and Practice, 2, 157-174

Ransom, J., Sommerville, I. And Warren, I. (1998), 'A Method for Assessing Legacy Systems for Evolution', Proceedings of the Second Euromicro Conference on Software Maintenance and Reengineering (CSMR'98), 128-134

Ropponen, J. and Lyytinen, K. (2001), 'Components of Software Development Risk: How to Address Them? A Project Manager Survey', IEEE Transactions on Software Engineering, 26 (2), 98-112

Schneider, A. and Feffer, S. (1999), 'Lessons Learned: Managing the Systems Conversion Process', Trust & Investments, May/June 1999, 30-35

Scott, J.E. and Vessey, I. (2002), 'Managing Risks in Enterprise Systems Implementations', Communications of the ACM, 45 (4), 74-81

Seacord, R.C., Plakosh, D. and Lewis, G.A. (2003), Modernizing Legacy Systems, Addison-Wesley, Boston

Song, W.W. (1996), 'Integration Issues in Information System Reengineering', Proceedings of the Twentieth International Computer Software and Applications Conference (COMPSAC'96), 328-335

Standards Australia (1999), Risk Management, Australia/New Zealand Standard AS/NZS 4360:1999

Stevens, K. and Fowell, S. (2003), 'Perspectives on E-Business Software Risk', Proceedings of the 7th Pacific Asia Conference on Information Systems (PACIS 2003)

Weiderman, N., Northrop, L., Smith, D., Tilley, S. and Wallnau, K. (1997), 'Implications of Distributed Object Technology for Reengineering', Carnegie Mellon University Software Engineering Institute Technical Report CMU/SEI-97-TR-005

Wu, B., Lawless, D., Bisbal, J., Grimson, J., Wade, V., O'Sullivan, D. and Richardson, R. (1997), 'Legacy System Migration – A Method and its Tool-kit Framework', Proceedings of the Asia Pacific Software Engineering Conference and International Computer Science Conference (APSEC'97 and ICSC'97), 312-320

Yin, R. (1989), Case Study Research: Design and Methods, Revised Edition, Sage Publications, Newbury Park

Zou, Y. and Kontogiannis, K. (2002), 'Migration to Object Oriented Platforms: A State Transformation Approach', Proceedings of the International Conference on Software Maintenance (ICSM'02), 530-539

COPYRIGHT