The Journal of Systems and Software 81 (2008) 2118–2133

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Risk and risk management in software projects: A reassessment

Paul L. Bannerman

NICTA, Australian Technology Park, Eveleigh, NSW, Australia

ARTICLE INFO

Article history:
Received 31 July 2007
Received in revised form 11 January 2008
Accepted 26 March 2008
Available online 23 April 2008

Keywords:
Software projects
Risk management
Project management
Threat management

ABSTRACT

Controlling risk in software projects is considered to be a major contributor to project success. This paper reconsiders the status of risk and risk management in the literature and practice. The analysis is supported by a study of risk practices in government agencies in an Australian State, contributing to a gap in research in the public sector. It is found that risk is narrowly conceived in research, and risk management is under-performed in practice. The findings challenge some conventional conceptions of risk management and project management. For example, it was found that software projects do not conform to a uniform structure, as assumed in much of the literature. This introduces variations in the risk and project management challenges they face. Findings also suggest that formal project management is neither necessary nor sufficient for project success. It is concluded that risk management research lags the needs of practice, and risk management as practiced lags the prescriptions of research. Implications and directions for future research and practice are discussed.

1. Introduction

Is risk management up to the task of improving outcomes in software projects? If you believe some of the industry surveys on project success rates you could be excused for being unsure.

Software projects are high risk activities, generating variable performance outcomes (Charette, 2005). Industry surveys suggest that only about a quarter of software projects succeed outright (that is, they complete as scheduled, budgeted and specified), and billions of dollars are lost annually through project failures or projects that do not deliver promised benefits (Charette, 2005; Johnson, 2006). Evidence suggests that this is a global issue (KPMG, 2005), impacting private and public sector organizations alike (Sauer and Cuthbertson, 2003).

The promise of risk management in commercial software projects is that it can improve project outcomes (consideration of specialist domains such as safety–critical systems is not included in this study). According to the literature (Simister, 2004; Ward and Chapman, 2004), risk management can lead to a range of project and organizational benefits including:

- identification of favorable alternative courses of action;
- increased confidence in achieving project objectives;
- improved chances of success;
- reduced surprises;
- more precise estimates (through reduced uncertainty);
- reduced duplication of effort (through team awareness of risk control actions).

Project-related risk management has attracted a steady stream of interest in the academic literature (Taylor, 2006), practice-based methods (e.g., CMMI and PRINCE2), and standards (e.g., PMBOK and AS/NZS 4360:2004). Industry survey data suggests that while there has been some improvement in project success rates, software projects are still more likely to fail on some key performance criterion than succeed outright (Johnson, 2006; Rubenstein, 2007). Furthermore, empirical studies have found that risk management practices often vary from prescriptions in the literature (March and Shapira, 1987; Ropponen and Lyytinen, 1997; Taylor, 2006).

In re-examining the question, this paper makes two main contributions. The first is a review and reassessment of the literature on software project risk and risk management. The second is an empirical study on risk management practices that is assessed against the prescriptions in the literature. From these analyses, implications are drawn for future research and practice.

Implicit in the empirical study is a further contribution. The study reports on software project risk management practices in agencies of an Australian State government, contributing to a paucity of research on public sector software projects in the literature. Ten major risk factors were found. The study also reports unexpected findings relating to the existence and risk implications of different project types.

The paper is structured as follows: In the next section, the literature is reviewed and reassessed. Following this, the empirical study is described and the main findings are presented before conclusions are reached on the comparison of practices with the prescriptions in the research literature. Finally, limitations of the study and implications for research and practice are discussed and conclusions are drawn on directions for future research and practice.

E-mail address: paul.bannerman@nicta.com.au

0164-1212/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2008.03.059

2. Literature review

In this section, first the importance of risk and risk management is briefly recounted. Then the concept of risk in software projects in the literature is reconsidered and four limitations are discussed in comparison to the needs of practice. Risk management and associated practice prescriptions are then reviewed before conclusions are drawn from the literature.

2.1. Why are risk and risk management important?

Conceptually, from the organizational perspective, risk arises when organizations pursue opportunities in the face of uncertainty, constrained by capability and cost. The challenge is to find a position on each of these dimensions that, in combination, represents a risk profile that is appropriate to the initiative and acceptable to internal and external stakeholders. Consequently, risk and risk management are strategic and governance issues that usually involve a compromise: a risk-averse strategy can limit distinctive achievement; however, a risk-embracing strategy can increase project losses. Explicitly managing this balance is often under-played or overlooked in the pursuit of desired goals (Charette, 2005).

At the project level, software projects have long been recognized as high-risk ventures prone to failure (Brooks, 1975; Abe et al., 1979). Boehm and Ross (1989) argue that there are two classes of software project risk: generic risks common to all projects, and project-specific risks. Some of these risks are easy to identify and manage. Others are less obvious or it is more difficult to predict their likelihood and/or impact. This is complicated by multiple project dimensions including size, structure, complexity, composition, context, novelty, long planning and execution horizons, and volatile change (Ward and Chapman, 2004; Willcocks and Griffiths, 1997). Therefore, risk management in software projects is important to: help avoid disasters; avoid rework; focus and balance effort; and stimulate win–win situations (Boehm, 1989). While not all risks have their source in software practices, they all have the potential to impact the outcome of the software process via the project mechanism through which the software artifact is usually delivered.

Risk and risk management are also important because IT projects (including software projects) can be vehicles of delivering IT-enabled organizational change, so achieving business objectives can be critically dependent upon their success.

How, then, is risk conceived and what management practices are prescribed in the literature to improve project outcomes?

2.2. What is risk?

The most common definition of risk in software projects is in terms of exposure to specific factors that present a threat to achieving the expected outcomes of a project. On this basis, risk in software projects is usually defined as the probability-weighted impact of an event on a project (Boehm, 1989; Charette, 1989, 1996). Simplistically, $\mathbf{R} = \mathbf{P} \times \mathbf{I}$ where **R** is the risk exposure attributable to a particular risk factor, **P** is the probability the undesirable event will be realized and **I** is the impact or magnitude of the loss if the event occurs. Risk exposure is usually measured in dollars or time in commercial projects.

This view of risk was adapted from management theory in the 1980s (March and Shapira, 1987). In classical decision theory, risk was viewed as reflecting variation in the probability distribution of possible outcomes, negative or positive, associated with a particular decision. However, March and Shapira (1987) found that 80% of managers consider only negative outcomes as 'risk'. They found that possibilities of positive outcomes were primarily considered only in assessing the attractiveness of alternative choices. Furthermore, opportunity management can require different processes to threat management. Consequently, risk became viewed as a danger, hazard or threat of a poor outcome, although some researchers still define risk as encompassing both threat and opportunity.

The general notion as used today in software projects is that to reduce the likelihood of an adverse project outcome, all potential risk factors should be identified at the start of the project. The risk exposure for each factor is then estimated (using the above formula) and the exposures are prioritized to identify the risks that represent the greatest threat to the project. Attention is then focused on the high risk factors to minimize the likelihood of their occurrence and/or the magnitude of impact if they are realized, through control measures such as mitigation strategies and/or contingency plans. Risk factors are monitored progressively to detect, as early as possible, when they materialize or if the threat changes (in likelihood or impact). A progressive status of identified risk factors is maintained and periodically updated. Realization of a risk is often recognized through the onset of a predefined risk trigger or the reaching of a predetermined risk threshold, at which time predefined contingency plans are activated to minimize the impact.

This common conception of risk has some limitations.

First, even before software engineering adopted the definition, management research found that this approach does not match actual managerial behavior. It was found that, in practice, the likelihood of outcomes and their impacts tend to enter into managers' calculations of risk independently, rather than as their products (March and Shapira, 1987). Managers see risk in less precise ways. First, they tend to be more concerned with the magnitude of the potential loss than the probability it will occur. They also tend to prefer verbal characterizations of risk than probabilistic representations because they are skeptical that the broad dimensionality of risk can be reduced to a single number. Finally, managers tend not to accept risk estimates given to them because they see risk as subject to control. They believe that risks can be reduced or dissolved by using their managerial skills to control the dangers. That is, "managers look for alternatives that can be managed to meet targets, rather than assess or accept risks" (March and Shapira, 1987, p. 1414).

A second limitation of this definition is that it is very difficult in practice to estimate the probability of impact of many risk factors, especially in software projects. Probabilities can only be meaningfully determined for activities that are repeated many times, under controlled circumstances. The one-off nature of many software project activities mitigates against accurately estimating probabilities. In classical decision theory, this problem was handled by conceiving risk as variation in a distribution of probable outcomes, not one probable outcome.

These issues reflect an unresolved question about whether the management of risk is a science, an art, or some combination of both (Bernstein, 1996). Are the best decisions based on quantification and numbers, determined by the patterns of the past, or are they better based on more subjective assessments of the uncertain future? We cannot quantify the future with any certainty, but through probability mathematics, we have learned how to extrapolate from the past. However, since software projects are often about enabling *change* through *new applications* using *new technologies* in *dynamic environments*, the degree to which past patterns are relevant to the future is fundamentally uncertain in these projects.

While it may be possible to generate metrics of low-level software engineering processes that enable probabilistic quantification of some important risk factors, there is likely to be many other critical software project risk factors that cannot be probabilistically assessed.

A common response to this problem in software projects is to view risk more generally in terms of *uncertainty* and to assess it qualitatively. Risk factors are assessed and ranked against a categorical scale of relative values such as low, medium and high (or, more typically, a five-point Likert scale) on the two dimensions of risk: likelihood and impact. Under this approach, 'high-highs' attract the most attention in applying risk control strategies, subject to cost; moderate risks ('medium–mediums') might only be monitored to see if they change status; while 'low–lows' might be ignored. However, a generally accepted definition of risk that is not based on the notion of probability has not yet emerged in the literature.

A third limitation of this definition is that it tightly couples the risk event with the risk consequence, ignoring the mediating influence of organization-specific vulnerabilities and capabilities to mitigate and respond (Zhang, 2007). Vulnerability is the capability of an organization to respond to a threat. Vulnerability may increase or decrease an organization's exposure to a risk event, depending upon the characteristics and response capability of the organization. These variables are not explicitly accounted for in the traditional probabilistic definition of risk. They are usually left to be implicitly considered during risk identification and evaluation processes. Few risk management methodologies incorporate a vulnerability assessment process.

A fourth limitation is that the definition encompasses only known or foreseeable threats. It provides limited options for managing realized threats and it does not recognize unforeseeable threats. This is a consequence of defining risk in terms of probability of impact. To assess the probability of an impact you need to be able to foresee an eventuality.

Knowledge of events can be distinguished by four categories of awareness, which chart the spectrum of certainty–uncertainty confronting projects:

Known–knowns are 'things we know we know'. These are usually called *issues* when they present as problems for projects. In the literature, threats that have occurred are not *risks* but *issues* because **P** = 1. Therefore, they fall outside of the domain of risk management, even though their impact (**I**) still needs to be managed downwards. When they are proactively managed it is usually through processes of issue management. However, issue management is often not integrated with risk management. Also, issues are often managed less formally at lower organizational and priority levels than risks, and often have less visibility within governance arrangements. Therefore, issues still have great potential to impact a project.

Known–unknowns are 'things we know we don't know'. These are the traditional domain of risk and risk management, which aim to identify and mitigate foreseeable threats.

Of greater concern, however, are *unknown–knowns* ('things we don't know we know', or knowledge that others have that we do not) and *unknown–unknowns* (things we don't know we don't know', or completely unforeseeable threats). Unforeseen threats are not directly accounted for by current project management processes (Pender, 2001). This limitation is usually resolved in practice through iterative risk identification. However, either threat can materialize rapidly and unexpectedly, leaving no time to respond and plan mitigation actions, thereby threatening the integrity and survival of a project. When they do materialize, at best, they are handled by an issue management or, more rarely, a crisis management process, if either exists. Often, however, they are responded to reactively by uncoordinated 'firefighting'.

This suggests that, theoretically, traditional risk management is closely related to other important control processes such as issue and crisis management. They share a common *impact* construct. However, this inter-relationship is usually not recognized or explicitly accounted for in risk-related research or practice-based methodologies.

In sum, based on this brief review of risk, it is concluded that the conceptualization of 'risk' in the research literature may be narrower than the nature of the problem in practice requires.

2.3. What is risk management?

As foreshadowed above, software project risk management is usually defined as a set of *principles and practices* aimed at *identifying*, *analyzing* and *handling* risk factors to improve the chances of achieving a successful project outcome and/or avoid project failure (Boehm, 1989, 1991; Charette, 1989; Kerzner, 2003). Any variation in approach is usually in the 'principles and practices' employed within this conceptual understanding of risk management.

Most commonly, one or more of four inter-related approaches to risk management are found in the literature and practice. These are checklists, analytical frameworks, process models, and risk response strategies. Each approach is briefly reviewed.

2.3.1. Checklists

Lists of the top risk or success factors in software projects are common in the literature and practice. Examples of well-known 'top-ten' lists are provided by Boehm (1991) and Johnson et al. (2001). More extensive lists can be found in Addison and Vallabh (2002), Barki et al. (1993), and Schmidt et al. (2001). These lists are usually compiled from surveys of the experiences of stakeholders such as project managers who have been involved in software projects. The risk management value of these lists is that the factors may also be important in other projects (that is, they may be generic risks). Therefore, a rudimentary form of risk management is to use the list as a checklist against which other projects can be reviewed and assessed, to ensure that each factor in the list is appropriately accounted for in the project.

The main benefit of the checklist approach to risk management is that it provides a quick, low cost way of identifying and assessing the risk exposure of a project against the major factors found by others to be important in determining the outcome of software projects. There are, however, several problems with this simple approach.

First, how do we choose which list to use? There are many different lists available. Note, for example, that Boehm's (1991) list focuses on low level development risks while Johnson et al.'s (2001) focuses on higher level project risks. Also, other lists exist that are generic to all projects, not just software projects (see, for example, Schultz et al., 1987). The chosen list may not adequately cover the factors relevant to a particular project.

Second, research shows that the perception of risk in software projects varies between stakeholder groups, over time, across project and life cycle stages, and between cultures (Boehm, 1988; de Camprieu et al., 2007; Keil et al., 2002; Mursu et al., 2003; Schmidt et al., 2001). This raises the prospect that risk assessment based on published checklists may be biased and/or limited in scope.

For example, Ropponen (1999) re-ranked Boehm's (1991) list of risk factors in a different cultural context at a different time and found significantly different rankings. Only 3 of the 10 were ranked the same and Boehm's highest ranked item was placed seventh in Ropponen's list. In another study, Keil et al. (2002) found marked differences between project managers and users in the risk factors they identified and their relative importance in software projects. Also, Schmidt et al. (2001) conducted simultaneous surveys in Hong Kong, Finland and the United States to develop an authoritative list of common risk factors. They found significant differences in the risk factors identified and their perceived relative importance across the three cultural environments. In a composite list of 29 identified and ranked software project risk factors, only 11 items were common to all three countries.

Third, research also shows that stakeholder groups tend to identify and rank highly risks that are perceived to be outside their own

control. That is, they tend to identify risks in the responsibility domains of other stakeholders, rather than point to factors as risks within their own areas of responsibility (March and Shapira, 1987; Schmidt et al., 2001). For example, Keil et al. (2002) found that users ranked six factors that project managers did not consider to be important, five of which related to project management. Conversely, seven of the 10 factors ranked by project managers but not users related to users. The effect of this tendency can be to limit the likelihood that the full range of relevant risk exposures is identified, especially if the checklist is based on the perceptions of a single stakeholder group, which is usually the case with published lists.

A final problem with the checklist approach to risk management, as noted above, is that managers' risk perceptions tend to be based more on the magnitude of the potential loss than the probability a loss will occur. However, risk surveys and checklists typically focus mainly on factors that contribute to the likelihood of project failure rather than on the magnitude of loss should failure occur (Keil et al., 2000).

Based on these research findings it can be concluded that software project risk factor checklists are unlikely to be universally applicable, and great care should be taken in using published lists as tools for risk management in practice. The best use of risk/success factor checklists is as a starter list in evolving a customized in-house set of risk factors from the software projects conducted in the organization over time. Factors on the generic list that are not found to be relevant to the organization's projects can be replaced by factors that are identified as risks. However, it is critical that the views of all key stakeholder groups are taken into account in the ongoing iterative risk identification and review processes, not just during project planning.

2.3.2. Analytical frameworks

The second major approach to risk management found in the literature and practice is closely related to checklists. Non-process based analytical frameworks provide an alternative way to think about and manage software project risks.

There are often too many potential risk factors to effectively identify and manage in a checklist, even if the focus is only on 'high-highs' (Cule et al., 2000). Furthermore, due to causal ambiguity, controlling individual risk factors may be unproductive. Risk factors often cluster into categories according to related themes (Barki et al., 1993), so individual control measures can often be applied effectively to one or more whole categories of risk, rather than treating each individual factor (Addison and Vallabh, 2002). This approach significantly leverages risk management effort.

High level sources of risk, such as *technology*, *requirements* or *expertise* can each account for multiple related risk factors. On this basis, categories of risk (also called risk dimensions, risk drivers, or risk components) can provide a broader framing for thinking about what risks might threaten a particular project, rather than to simply work through a pre-defined checklist of specific factors. Categories can also represent target areas for applying risk control strategies.

A variety of risk categories and frameworks has been proposed, mainly in the academic literature. Examples include the following:

- Frameworks categorizing risks according to their perceived *source* are the most common (e.g., Barki et al., 1993; Boehm and Ross, 1989; Cule et al., 2000; Davis, 1982; DeMarco and Lister, 2003; Jiang et al., 2002; Keil et al., 1998; Lucas, 1981; McFarlan, 1981; McKeen and Smith, 2003; Ropponen and Lyytinen, 2000; Tiwana and Keil, 2004; Wallace et al., 2004; Zmud, 1979). Taking one example, Cule et al. (2000) classify risks into four major types according to their source (client, self, task, environment), each with up to 20 indicative risk factors. A dominant risk management strategy is identified for each type rather than each factor (relate, assess, control, and monitor, respectively).
- In contrast, another researcher suggests a lifecycle-based risk management approach for large enterprise integration projects in which risk is assessed in each major phase of the project (Lam, 2004).
- A third example applies a generic socio-technical model to the software development context at three different levels of analysis: system, project, and management (Lyytinen et al., 1996, 1998). The idea is that the model elements (task, actors, structure, and technology) and their inter-relationships provide a structure within which to think through the risk exposures in each major environment impacting the project (the system environment, project environment and managerial/organizational environment).

Categorical and other non-process analytical risk management frameworks can be very helpful tools in framing thinking about risks and risk management actions in software projects, especially in support of risk identification and analysis, at a higher level of abstraction than checklists. However, many of the same limitations of checklists apply. Which framework should be used? Is it sufficiently representative of another project's context? If not, should multiple frameworks be used, a composite formed, or a framework tailored to the target environment? And how much risk framing is enough to identify all relevant risks?

Given an appropriate framework, the main limitation of this approach is closely related to its major benefit. On its own, the framework will do nothing to improve risk management. As with any tool, its value (or otherwise) is totally dependent on how well it is used. For example, the quality of risk identification and analysis is dependent on the representation, participation, perception, and insight of the stakeholders in the risk brainstorming workshops who think through the various pointers offered by the analytical tool. If the analysis is cursory or superficial then the risk management benefits are likely to be low.

2.3.3. Process models

The third and most common risk management approach found in the literature and practice is process models. Process models specify stepwise tasks for managing risks. Typically, they specify the individual activities believed to be necessary to manage risk in software projects (for example, risk identification, analysis, response and control). Usually they also specify how these activities should be sequenced to effectively manage risk and, less frequently, they may also suggest tools and techniques to use in individual steps to aid in the risk management process. Conceptually, most models include a similar set of process steps which include, for example: risk strategy, risk identification, risk analysis, risk responses, and risk control (Simister, 2004). The ordered steps are usually intended to be executed iteratively throughout the project, to manage known and new risk factors as the project proceeds and as environmental circumstances change.

Many prominent examples of risk management process models can be found in practice and in the literature. The two most dominant models in software engineering are associated with Boehm (Boehm, 1989, 1991; Boehm and Ross, 1989) and PMI's *PMBOK Guide* (ANSI/PMI 99-001-2004). Other influential models are found in the Software Engineering Institute's CMMI (CMU/SEI-2006-TR-008), various industry and national standards (e.g., PRINCE2; ISO/IEC 16085:2004, IEEE 1540-2001; AS/NZS 4360:2004), and the academic literature (e.g., Charette, 1989, 1996; Simister, 2004).

Checklists, analytical frameworks and process models are interrelated and often used together. For example, checklists and

analytical frameworks may be used in the *risk identification* and *risk analysis* steps of a process model.

The major contribution of process models is that they guide and direct risk management *action* rather than just analytical *thinking*. However, process models provide no 'cookie cutter' solutions to software project risk management. They require skill, judgment and persistence to effectively apply them and their associated tools and techniques in practice. For example, having 'identified' and 'analyzed' the risks, it is then necessary to determine what, if anything, can and should be done about them. This requires reasoned context-specific actions.

In the *risk response* and *risk control* steps of a process model, the literature provides some support in deciding a course of action through the prescription of several generic response strategies, the final risk management approach considered here.

2.3.4. Risk response strategies

The literature describes generic options for responding to project risks (e.g., DeMarco and Lister, 2003; Frame, 2003; Kerzner, 2003; Schwalbe, 2007). Within these high-level options, specific responses can be formulated according to the circumstances of the project, the threat, the cost of the response and the resources required for the response. Typically, risk response strategies aim to either reduce or eliminate the likelihood of the threat occurring (that is, to reduce **P**); limit the impact of the risk if it is realized (reduce **I**); or a combination of both. These strategies are formulated and implemented in response to new risks whenever they are identified and assessed as a threat that must be controlled. Four common risk response strategies are found in the literature:

- *Avoidance*. Avoidance strategies aim to prevent a negative effect occurring or impacting a project. This may involve, for example, changing the project design so that the circumstance under which a particular risk event might occur cannot arise, or so that the event will have little impact on the project if it does. For example, planned functionality might be 'de-scoped' to remove a highly uncertain feature to a separate phase or project in which more agile development methods might be applied to determine the requirement (Boehm and Turner, 2003).
- *Transference*. This strategy involves shifting the responsibility for a risk to a third party. This action does not eliminate the threat to the project; it just passes the responsibility for its management to someone else. Theoretically, this implies a principal–agent relationship wherein the agent is better able to manage the risk, resulting in a better overall outcome for the project. This can be a high risk strategy because the threat to the project remains, which the principal must ultimately bear, but direct control is surrendered to the agent. Common risk transfer strategies include insurance, contracts, warranties, and outsourcing. In most cases, a risk premium of some kind is paid to the agent for taking ownership of the risk. The agent must then develop its own response strategy for the risk.
- *Mitigation*. Risk mitigation is one or more reinforcing actions designed to reduce a threat to a project by reducing its likelihood and/or potential impact before the risk is realized. Ultimately, the aim is to manage the project in such a manner that the risk event does not occur or, if it does, the impact can be contained to a low level (that is, to 'manage the threat to zero'). For example, using independent testers and test scripts to verify and validate software progressively throughout the development and integration stages of a project may reduce the likelihood of defects being found post-delivery and minimize project delays due to software quality problems.
- *Acceptance*. Risk acceptance can include a range of passive and active response strategies. One is to passively accept that the risk exists but choose to do nothing about it other than, perhaps, to monitor its status. This may be an appropriate response when the threat is low and the source of the risk is external to the project's control (Schmidt et al., 2001). Alternatively, the threat may be real but there is little that can be done about it until it materializes. In this case, contingencies can be established to handle the condition when and if it occurs. The contingency may be in the form of provision of extra funds or other reserves, or it may be a detailed action plan (contingency plan) that can be quickly enacted when the problem arises. Validation and maintenance of contingency plans is a critical part of this strategy to ensure that contingency plans work as expected when required.

Overall, risk response strategies are effective in providing general options for consideration in formulating responses to foreseen project threats. Each requires a specific response to be formulated, executed and re-assessed throughout the project as the nature of the risk unfolds or significantly changes. However, consistent with a narrow definition of risk, they provide no generic response options for unforeseen threats.

2.4. Improving software projects

The need for mechanisms to improve project outcomes is graphically illustrated by Charette's (2005) "Software Hall of Shame", which lists over 30 major software project 'failures' in a little over a decade and their associated costs. However, the literature reports that risk management can significantly improve software project outcomes (Charette, 2005; Elkington and Smallman, 2002; Jiang et al., 2002; Remenyi, 1999; Ropponen and Lyytinen, 1997).

Unfortunately, research also finds that risk management is not always well-applied in practice (Ibbs and Kwak, 2000; Morris, 1996; Pfleeger, 2000; Ropponen, 1999; Ropponen and Lyytinen, 1997). For example, in a multi-industry study of project management maturity based on PMI's knowledge areas, Ibbs and Kwak (2000) found that the risk knowledge area had the lowest maturity of all knowledge areas in the IS industry, and the risk maturity level of the IS industry was the lowest of the four industries in the study. Also, Ropponen found that 75% of project managers did not follow any detailed risk management approach, and only vaguely understood the software risk concept and its managerial implications. However, most reported using some type of risk management method (Ropponen, 1999; Ropponen and Lyytinen, 1997).

Based on this review, therefore, we conclude that: first, the notion of risk is relevant to software projects, and there is a need and potential for risk management to contribute to project outcomes; second; the development of risk and risk management in the research and practice literature lags the requirements of the threat phenomenon in practice, and; third, the adoption of risk and risk management concepts and methods in practice lags the understanding and prescriptions found in the literature. In sum, there is a need for better risk management in research and practice.

No causality is implied in these findings. That is, it is not argued that the practice of risk management is low because risk management narrowly fits the needs of practice. Rather, the conclusion highlights a need for research to extend the relevance of the risk concept and risk management to improve software project outcomes.

The following empirical study supports, reinforces and extends these conclusions.

3. The study

The study investigated software project and risk management practices in government agencies in an Australian State. The primary purpose of the study was not a private–public sector comparison.

Rather, it was to investigate the practices of a State government that had experienced notable successes and failures in software projects in recent years and contribute to a gap in the literature in public sector studies.

Access to agency projects was facilitated by the government's CIO office. Agencies operate semi-autonomously, responsible for provisioning their own system needs. The central CIO office provides the agencies with a policy framework for IT acquisition, process standards, and some funding for special projects. Agencies were invited to participate in the study by nominating one or more recently completed software projects.

After issues of access and availability of informants was resolved, the study sample comprised 23 informant perspectives on 17 projects from 17 agencies. The correspondence was not one project per agency. Ten agencies contributed one project each and three agencies contributed two projects each. One perspective was obtained on each of these projects except for three, on which two perspectives were obtained. A further project was investigated from the perspectives of four different, but related, agencies. Each project was different.

Structured interviews were conducted with informants and a case study was prepared based on the informant's perspective of the project. The interview comprised 150 questions spanning nine topic areas: informant, organization, project, governance, project and risk management, development, implementation, third parties, and other. The questions were developed from a review of risk and success factors in the research literature on project management and risk management. Opportunity was also provided for informants to disclose context-specific information.

Answers to questions reflected informants' perceptions of the project. In most cases, participants were asked to answer by choosing a value from 1 to 10, where one signified 'no' or the lowest value in response to the question, and 10 indicated 'yes' or the highest value. They were then given the opportunity to explain their answer. For example, "Was the project completed within budget?" might have been answered by the informant with a value of eight and the explanation that a slight cost overrun occurred, but it was within limits considered acceptable by the agency.

Interviews lasted from one to two hours and were recorded. Interview recordings were later used by the researcher to generate descriptive statistics and prepare a descriptive case study of the informant's account of the project. Case study descriptions were validated by the respective informants.

Each case study was qualitatively analyzed to identify thematic patterns and artifacts that appeared to be relevant or important in enabling or inhibiting the performance and/or outcome of the project. Over 300 artifacts were identified from the cases studies independently by two researchers as evidentiary data. Interest focused on novel characteristics and events that directly related to the projects studied and their contexts (that is, on project-specific rather than generic factors). The researchers then ordered the artifacts by theme, resulting in 10 categories of related software project risk factors. Further analysis of the categories provided support for the literature as well as gave rise to novel insights into software project risk and risk management practices in the cases studied.

While not random, the study sample comprised considerable variety in agencies and projects. Supported by Table 1, the following summarizes profiles of the State government agencies and projects included in the study.

3.1. Agency profile

Agency size. Agencies varied significantly in size, ranging from very small (31 employees) to very large (130,000 employees). Half of the agencies had less than 1000 employees. The median size was 650. The majority of agencies comprised less than 5000 employees, as shown in Table 1.

CIO reporting line. The reporting level of the CIO (or CIO equivalent) is an indicator of the perceived importance and role of the IT function in the agency. As shown in Table 1, IT typically reported to a divisional or functional director, such as Corporate Services. Only 38% reported to the CEO (or equivalent).

Informants. Study informants were mostly project managers, as shown in Table 1.

Project management experience. For the informants with project management experience, the length of experience was typically more than five years. The average duration of experience was 5–10 years, but there was some variation, as shown in Table 1.

Table 1
Study profile

	Percentage (%)
Agency size (in employees)	
Less than 999	50
1000–4999	38
5000–9999	6
10,000-99,999	0
More than 100,000	6
CIO reporting level	
CEO (or equivalent)	38
Divisional Director	56
Business Unit Manager	6
Study informants	
Project Manager	70
IT Manager	22
Business Manager	8
Project management experience	
Less than 2 years	12
2–5 years	18
5–10 years	35
More than 10 years	35
Project scope	
Development and implementation	53
Implementation only	32
Development only	10
Other	5
Applications	
Web-based	35
Transaction-based	29
Package implementation	24
Data publishing	6
Statewide package rollout	6
Project size (duration)	
<6 months	22
6–12 months	34
1–2 years	22
>2 years	22
Executive involvement in project	
CEO	33
Divisional Director	50
Business Unit Manager	17

3.2. Project profile

For the study, a software project was defined as a project involving software development (internally or by a commercial software developer) and/or implementation of custom-built or a COTS application system (with or without software customization). Here, 'implementation' is used in the sense of deploying a completed application, not coding a solution.

Project scope. More than half of the projects in the study involved development and implementation of software (see Table 1). Another third involved implementation only.

Applications. Web applications development, transaction-based systems development, and package implementation dominated the projects studied, as summarized in Table 1.

Project size. More than half of the projects in the study were less than 12 months long in duration on completion (see Table 1). Three were large ongoing projects, for which interviews focused on completed phases.

Third parties. Thirteen projects (76%) involved and were significantly dependent upon one or more external third parties (such as a vendor, developer or consultant). The number of participating third parties varied, with an average of two per project.

Strategic importance. Most projects (89%) were perceived to have some importance in achieving the strategic objectives of the agency, with two-thirds (67%) seen as being highly relevant, strategically. Only 11% of projects were considered not to be strategically important to the agency.

Executive involvement. In contrast to the perception of strategic importance, only one third of the projects had direct CEO (or equivalent) level involvement in the project. In another 50% of projects, the most senior executive involved was a divisional or functional director; and in the remaining projects, the most senior person involved was a business unit manager.

4. Findings

This section reports three main findings of the study. First, findings relating to key practice areas are outlined. Second, the major risk factors found in the study are described. Third, an unexpected finding of multiple project types, each with different implications for risk and project management, is described and discussed. Additional details on some findings are available in Bannerman (2007).

4.1. Key practice areas

Selected responses relating to key practice areas investigated in the study are summarized in Table 2 and described following. In the table, 'Score' is the average score out of 10 and 'Range' is the spread of scores from 1 to 10 for all respondents.

Table 2
Responses on key practices

Questions	Score	Range
Project outcome		
How *successful* was the project?	8	5–9
Were the *business objectives* achieved?	8.9	7–10
Was the project completed *within budget*?	9	6–10
Was the project completed *within schedule*?	8.4	7–10
Was the project completed *within scope*?	8.8	6–10
Project governance		
How *effective* was the steering committee?	7.9	3–10
How *effective* was the project sponsor?	8	1–10
How *committed* was top management?	9.2	4–10
How *involved* was top management?	7.7	3–10
Did this commitment/involvement *benefit* the project?	7.9	1–10
Project management		
How well was the *project managed*?	7.8	6–9
Did the project manager have a *clear vision* of the project?	7	4–10
How appropriate was the *methodology* used?	7.4	1–10
How well was *progress controlled* against the plan?	7.1	2–10
Did the project have effective *change control*?	8.2	2–10
Was a formal *post-project review* held?	6.5	1–10
Risk management		
Did you use a specific risk management *methodology*?	4.6	1–10
How well were *risks identified* as the project's start?	6.9	1–10
How well were *risks managed* throughout the project?	6.6	1–10
How well were *risks prioritized* when identified?	5.3	1–10
Were *mitigation/contingency plans* pre-determined?	7.6	1–10
Was *responsibility assigned* for *monitoring* risks?	6.4	1–10
Did *unanticipated problems* arise during the project?	7.7	1–10
Implementation		
How *easy* was implementation? (0 = trivial)	6.5	2–10
How *successful* was the implementation?	8.4	4–10
How great was the *impact* on the organization?	7.1	2–10
How *important* was the project to strategic objectives?	7.9	2–10
How well were the organizational *changes managed*?	6.9	4–9
Third parties		
How *effective* was third party input to the project?	7.5	2–10
How well were third party *relationships* managed?	8.1	5–10

Project outcome. All of the projects in the study except one were perceived to have been 'successful', with an average score of 8 out of 10 for the question "How successful was the project?" The exception project was suspended indefinitely and ultimately cancelled due to conflicts with contingent projects. Key practice measures were not provided for this project, so they are not included in these responses. Informants' claims of project success were accepted at face value.

As expected from the project management literature, informants varied in how 'project success' was defined (Atkinson, 1999; Wateridge, 1998). Overall, business managers tended to view project success as achievement of business objectives, while IT and project managers emphasized various technical criteria, achievement of project objectives, or achievement of desired organizational outcomes.

Project governance. Three quarters of all projects reported to a steering committee and nearly all had a business sponsor. However, as shown in Table 1, only a third of projects had direct CEO involvement. There was wide variation in responses about how effective the project sponsor was and whether top management commitment or involvement benefited the project.

Project management. Ratings on questions relating to project management tended to be lower than those for project outcome and displayed greater variance (refer to Table 2). Variance was greatest for questions on the appropriateness of the methodology used, how well progress was controlled against plan, how effective change control was, and whether a formal post-project review was held. The last question attracted the lowest score for questions discussed so far (average of 6.5), with responses varying across the full range of scores (from 1 to 10).

Risk management. Formal risk management was practiced in five projects (29%), no risk management was practiced at all in another five (29%), while the remaining seven projects (41%) adopted a range of semi-formal or informal practices. These typically included formal identification of risks in the business case, request for tender document, or at the start of the project, followed by informal monitoring that dissipated as the project progressed or no further action other than to respond to issues as they arose during the project. These figures are consistent with the findings of Ropponen and Lyytinen (1997) and Ropponen (1999), reported in the literature review. Consistent with this finding, average scores on questions relating to risk management practices dipped in value further to the above practice areas, with all questions attracting responses across the full range of scores (Table 2). The responses indicate that agencies in the study tended not to use a specific or formal risk management methodology, but did tend to identify risks at the start of the project, although informants were quite equivocal about how well the risks were prioritized. However, mitigation actions or contingencies were determined in advance and responsibility was assigned for monitoring risks, but this was mostly to the project manager. Overall, informants believed that risks were reasonably managed (average of 6.6). As indicated, interview evidence suggested that project management practices tended to wane as the project progressed.

No agency reported using quantitative risk assessment. Agencies that assessed risk used qualitative scales.

These responses are curious considering that a majority of projects (15 of the 17) encountered unanticipated threats (average 7.7) and 89% of projects were considered to be strategically important, suggesting the importance of having strong risk management. As one experienced project manager explained:

"We have limited time and resources to get the system in. We keep a [risk] register because that's what we're required to do. Otherwise, we know what we have to do so we get on with it. We have a feel for areas where problems might arise, and, if they do, we handle them at the time. We don't spend time thinking about disasters because they rarely happen."

Implementation. Informants scored the success of system implementation highly (average of 8.4), which is consistent with the perceptions on project outcome (above), although the implementation was not seen as being a difficult part of the project (average of 6.5). There was also strong consistency between perceptions of the scope of impact of the system being implemented on the organization (average 7.1) and the project's relevance to achieving strategic objectives of the agency (average 7.9). Informants scored the projects moderately high on how well organizational changes associated with the system implementation were managed, but interview data showed that few projects explicitly included organizational change management within their scope. In most cases (69%), organizational change management was handled independently by the business areas impacted by the new system, either proactively or reactively as the system was introduced. Variance was high on all responses relating to implementation.

Third parties. Finally, as noted earlier, most projects had third party ('vendor') involvement. Overall, their input was considered to be effective, with some variation, and the relationships were perceived to have been well-managed.

4.2. Major risk factors

Analysis of the projects in the study uncovered 10 categories of risk factors listed in Table 3. The table also shows the number of projects (out of 17) in which the category was found to be a risk factor. Overall, the factor clusters identified are consistent with those found in the literature and best practice prescriptions.

No assessment was made of the relative importance of the risk categories so they are not numbered. They were all found to be important in reducing software project risk in the agencies studied. Rather, the categories are loosely sequenced according to their positioning in the software development life cycle. Each category is briefly described following.

4.2.1. Governance

A range of governance-related issues was found in the projects studied. Projects tended to struggle if governance bodies were passive, had insufficient or inadequate representation, mixed or declining participation rates, did not adequately resolve escalation issues, or did not actively engage and drive the project as a business or organizational initiative. Effective project governance was found to facilitate project alignment with the business, executive involvement and business ownership; clarity and relevance of objectives, scope and requirements; provide guidance, direction and a common sense of purpose; and did not curb the project team's responsibilities or stifle initiative. Absence of effective governance resulted in risk exposures in these areas.

Two forms of effective project governance were found. The most common was the project steering committee (PSC), chaired by a committed and involved senior business executive (found in twelve projects). In the remaining five projects, there was no formal governance. Less commonly, in projects over which the PSC adopted a very passive role, 'governance' was provided through the proprietorship of a business owner or project sponsor at the project level. In three cases where this was effective, the business person and project manager formed a close operating relationship based on mutual dependence, and worked through project issues together as they arose. Executives were informed only on major issues with organizational impacts, issues that might affect achievement of project objectives, and progress against key milestones.

Table 3
Major risk factors

Risk factor categories	Project occurrences
Project governance	12
Project setup	9
Partner engagement	8
Business proprietorship	11
Project management	9
Change management	10
Management of projects	11
Recognition of red flags	15
Management of risk	14
Benefits realization	8

4.2.2. Project setup

Many projects encountered problems due to poor project setup. Critical activities to get right at the start of a software project were found to be: determining the most appropriate project design and development methodology; setting the right budget; securing the necessary funds; choosing the right vendor partner(s); and objectively assessing risk. For example, risks and problems arose from using rigid, plan-based methodologies when requirements were highly uncertain and/or contexts were volatile. Also, some projects were constrained by funding arrangements that allocated funds before project costs were fully known, setting the project up to under-deliver from the outset. Other risks and issues arose from project setups that did not align to value-adding business objectives or where the project setup was left to a dominant vendor whose priorities and actions were driven mainly by self-interest.

4.2.3. Partner engagement

Several projects found that external third parties can be either an asset to the project or a significant risk, depending on how they are managed. The key challenges were in *engagement* and *control*. For *engagement*, an 'arms length' outsourcing approach was found to work well in formally structured, plan- and specification-driven contexts, but lacked the flexibility and interaction necessary under conditions of greater uncertainty. Some, especially smaller, agencies learned over time that better value and control could be achieved by having the third party developer's project staff located in-house (insourcing). This physical integration greatly improved project communication, interaction, issue resolution and progress tracking. It also enabled a degree of incremental, cyclical development, which resulted in delivery of a system that more closely fitted the agency's needs.

With respect to *control*, three agencies did not recognize the risks in not retaining project control. For example, rather than prepare their own plan and use their own methodologies, these agencies defaulted to using whatever the dominant third party proposed or used, placing themselves fully in the hands of their project partner(s) and thereby surrendering control. In one project, the vendor struggled to deliver, so the agency was forced to take back control.

In another case, great effort was taken to outsource the technical risk associated with a project to the vendor via the contract

(discussed above as a risk transference response strategy). This appears to have been done effectively as technical risk did not impact the project (although it is not known if the foreshadowed risk materialized or not). However, the agency's view was that it had given the risk away and no longer had to worry about it. This is dangerous thinking. Ultimately, organizations cannot offload software project risk. They can outsource risk management responsibility for certain risk factors and, perhaps, even offset the cost of the impact through penalty clauses if the risk materializes and negatively impacts the project. However, if the required project outcome is not fully delivered then it is the client that takes the direct impact of any sunk costs not covered by the penalties, as well as any impact from the loss of opportunity and failure to fully achieve business plans.

Other agencies found that rather than being a source of threat to the project, the prime contractor contributed positively to the project and its outcomes through the experience and competence of their project manager and specialist staff, compensating for a lack of internal expertise.

Finally, managing input from *internal* third parties was also a problem for six projects. Some projects relied on the IT department, for example, to provide infrastructure or implementation services. Conflicting priorities often made these unreliable partners. To overcome this problem, one agency planned to implement a project-based service level agreement during project initiation, as a formal engagement of internal service providers.

Overall, projects that benefited from partner involvement adopted a value-adding mindset to the engagements and sought to complement internal capabilities.

4.2.4. Business proprietorship

Business ownership, sponsorship and participation were found to be critical project risk factors. In fact, several project managers claimed that committed and capable business sponsorship was the most important factor in the success of their projects.

Two effective approaches were found (identified, above, under *Governance*): formal proprietorship by the business sponsor through project governance structures; and informal relationship through the business owner teaming with the project manager.

Strong business proprietorship was found to remove obstacles and drive projects toward their objectives. It also fostered a significantly different mindset toward the project wherein project commitment and participation was driven by a desire to achieve the outcomes promised by the project, in contrast to the mindset of resistance found in users who felt that an IT solution was being imposed upon the business.

Projects with weak or no substantive business proprietorship encountered major problems throughout the project, peaking when the 'IT solution' was eventually delivered.

4.2.5. Project management

Project management experience and capability were found to be critical in the performance and outcome of the software projects studied. This is not a novel finding. Curiously, it was also found that while good project management is *necessary*, it is *not sufficient* for success.

In one project, managed by a certified project management professional, project and risk management appeared to be both rigorous and 'by the book', but the project stalled and was eventually abandoned. Other factors came into play to which the steering committee could not effectively respond (the project was displaced by another from a higher level within the agency's hierarchy that offered an inferior solution to the agency's needs).

The case illustrates that the project manager is no super-hero or lone crusader. Organizational support is also needed. Good project management requires a complementary framework of individual, team and organizational capabilities and effort to optimize their contribution (Jugdev et al., 2007; Sauer et al., 2001). Organizational capabilities may include effective planning and governance frameworks; learning and development programs; mentoring arrangements; knowledge databases; career structures; and incentive schemes to deliberately promote, build, reward and support in-house skills in project activities.

Agencies that recognized the need to grow organizational capabilities in project management were found to build mechanisms into their project governance frameworks – typically through a project office – to capture project experience and knowledge and pass it on to others for the benefit of future projects. Agencies that relied on project managers alone, tended to be exposed to the limitations and vagaries of project 'heroes in action'.

4.2.6. Change management

Many projects in the study encountered implementation and user-related issues due to inadequate management of organizational impacts (as noted earlier, nearly 70% of projects did not explicitly include change management within their scope). Typically, these projects viewed their role narrowly as delivering a software system rather than a new business solution. In these cases, managing the organizational impacts of the change was considered to be a separate responsibility or left by default to the business user to resolve after delivery.

Four familiar issues were found: (1) in one case, a software solution was imposed on the business with no business input or involvement in specification, selection, and implementation; (2) in others, business processes were not always realigned to the new applications during or after implementation; (3) user buy-in to the new business solution and resistance to change were not always managed; (4) in several cases, transitioning to post-project operational and technical support arrangements was ignored.

Projects tended to have fewer implementation problems when organizational change was managed concurrently from the beginning of the project. In these cases, the project was viewed as an IT-enabled organizational change event. That is, as a technical means to a business end.

4.2.7. Management of projects

Paradoxical observations in the study raised questions about the traditional view of *project management* as a formal discipline of defined methods and practices that are critical for project success. For example, one project used no formal project management methodology or practices whatsoever but succeeded, and another (mentioned above) rigorously followed project management 'by the book' but failed. In contrast to the earlier finding that project management is *necessary but not sufficient* for success, these observations challenge whether project management, as formally conceived, is even *necessary* for project success. This is consistent with a recent review which notes that there is little data supporting the claim that formal project management produces better project outcomes (Shenhar and Dvir, 2007).

The critical distinction evident from effectively managed projects is that *project management* is fundamentally about *the management of projects* as a *management* activity and capability. There is no inherent determinism in the engineering disciplines of 'project management' *per se* (Morris, 1996). That is, these projects indicate that project success is an outcome of good management *per se*, not necessarily 'project management' as a body of knowledge and practice. What is critical to software project success is good management of a discrete, temporary, joint business-technical activity.

This finding is not a call to abandon formal disciplines of project management in favor of an "anything goes" approach. History shows that formal project management can be a very effective

way of framing how to manage software projects (Morris, 1994). But it is no panacea. Rather, this finding is a reminder to manage software projects within the organization's managerial structures and practices, whether through formal methodologies or otherwise, rather than to naively submit to a canned approach and assume that it will deterministically deliver the right outcome. This finding supports the above view that it is necessary to build individual and organizational capabilities in managing projects, rather than simply applying formularized methods.

4.2.8. Recognition of red flags

The study found that certain characteristics of individual projects tend to increase risk. This is consistent with Boehm and Ross' (1989) categorization of generic and project-specific risks.

For example, if the project is to implement an enterprise-wide suite of integrated applications (an ERP system) then a large range of risks are introduced relating to the implementation approach adopted (e.g., phased or 'big bang'), gaining participation from multiple independent business units, customizing and configuring the system and its modules, data set-up and migration, process realignment, redefinition of roles, and user training, among others.

The ability of the organization to recognize these risk indicators ('red flags') and take them into consideration in building and deciding the business case, and designing, planning and executing the project, was found to be critical to good risk management and, ultimately, the outcome of the project. Projects varied widely in this ability.

The existence of project-specific risks is a major reason why generic top-ten checklists may not focus attention on all relevant threats to a project. The study reinforces the view in the literature that a critical competency for organizations to build is the ability to recognize project-specific red flags and assess and treat the associated risk exposures to the project.

4.2.9. Management of risk

The study suggests that risk management, like project management, is more than a process or methodology; it is also a real-time threat management capability that is developed within an organization, through learning, practice, and other mechanisms, over a long period of time. Risk management is not just about identifying and assessing risks, and putting in place mitigation and contingency strategies. It is also about being able to respond quickly and effectively to realized threats as they arise. These threats may or may not have been foreseen but they have the potential to significantly impact the project and its outcomes.

This was graphically illustrated by one agency that was exposed to an extreme risk of security invasion through a web site. The agency thoroughly planned and prepared for the threat, taking every reasonable mitigation action, including predefining contingency plans and responses in the event that a breach occurred. A security breach did occur, but in an unexpected way. To the uninformed observer (and the management hierarchy of the agency), this looked like a failure of risk management. However, this is an unreasonable conclusion because it is highly unlikely that every possible threat could have been foreseen and catered for. In this case, the breach was sealed off within minutes of it being identified, with no damage to the system.

This incident demonstrates an effective and legitimate form of risk management to contain a risk that was not identifiable until after it materialized. The response to the threat was an effective utilization of an organizational risk management capability that had been built up over many years within the IT department, being rapidly applied to contain the impact and eliminate the exposure. The case illustrates that this capability dimension of risk management is essential to handle unforeseen threats that suddenly arise and cannot be responded to through planning-based responses. It also supports the finding in the literature review that risk management, as conventionally conceived, falls short of the needs of practice to handle some project threats.

4.2.10. Benefits realization

Finally, the agency cases also indicated that benefits are unlikely to flow automatically from software projects. At the business level, IT projects carry the cost, but business efficiencies generate the benefits. Therefore, benefits have to be both sought and captured within the business–IT collaboration to realize business value from the investment. Delivering a project 'on time, within budget and to specification' is of limited value if the original driving goals of the project are not also actively pursued and achieved (Bourne, 2007).

For example, one agency realized at the end of a major project that little more had been achieved than to install a new computer system. No real organizational benefits had been achieved. Therefore, the agency planned another project cycle to focus on realigning processes, deploying new functionality available in the system, and consolidating its integration with other critical enterprise systems. The driver for this approach was a change of mindset that occurred within the project steering committee from viewing the new application as an administrative system to seeing it as an enabler of new business value.

At the project level, the study also highlighted that seeking and capturing benefits includes securing returns to future software projects, not just organization-level stakeholders. For example, in a multi-phase/release project, one agency held post-implementation reviews at the end of each software release to ensure that incremental benefits were delivered, and lessons learned were carried forward to the next release.

In the projects studied, this mindset ultimately delivered real benefits to the agency from the IT investment. In other cases, however, the same mindset for pursuit of benefits realization was not so evident. There were cases, for example, where a new system generated limited value because insufficient data had been loaded into a newly installed system. The original funding proposal had not included the cost of resources needed to load the database. In another case, it was noted at the end of a project that benefits realization was not yet evident. However, neither was it evident that there were any mechanisms in place or momentum to seek benefits. Finally, in one other case, there was no known business case or statement of expected benefits for the project or organization to target.

In sum, agencies that practiced effective risk management did the following:

- had an effective project governance framework (formal or informal);
- practiced careful, realistic and context-specific project setup;
- adopted a value-adding approach to partner engagements;
- exercised strong business proprietorship of projects;
- developed project management capabilities within the organization;
- managed organizational change concurrently with technology change;
- recognized that projects are an engineering and management activity;
- had a strong ability to recognize project-specific red flags;
- recognized that risk management is more than a methodological process;
- recognized that organizational benefits must be explicitly sought and captured.

4.3. Project types

We tend to think of software projects in uniform terms as discrete activities subject to the joint disciplines of project management, risk

management and software engineering as espoused in best practice prescriptions. However, conceptually, four different project types were found in the study, suggesting that the reality of practice in the public sector may be more complex than this conventional view. I call the four types the *'pure' project form*, *hybrid form*, *operational activity* and *breakthrough event*.

These types vary along the dimension of formality from complying (or aiming to comply) with formal project management disciplines at one end, to adopting no conventional formality at all at the other, in the order of 'pure' project form, hybrid form, operational activity, and breakthrough event. The subject of each of the four types is a 'project' as conventionally defined (that is, a temporary endeavor undertaken to create a specific software-based result). However, the nature of the management practices employed varies from formal project management disciplines and practices at one end of the spectrum to practices that are operationally and contextually expedient at the other. This variation is not simply a reflection of different maturity levels in project management but rather of different organizational arrangements to meet the task challenge within each context.

The characteristics of the types suggest that different issues and challenges arise with respect to project and risk management in each type, as summarized in Table 4.

4.3.1. 'Pure' project form

This is the traditional project that is structured, operated and managed under conventional project management disciplines and practices (to varying degrees of compliance to formal prescriptions). Five projects of this type were found in the study. These projects practiced either formal risk management or informal risk management as described in Section 4.1.

For example, one project was tasked with choosing and implementing a replacement COTS administration system for 256 of an agency's operating facilities, spread across nine geographic areas. The project was one of several inter-related projects in a program that had higher level strategic objectives. The project had a set budget ($160 M), schedule (spanning 5 years), scope (specified in various planning documents), dedicated project team (15 people), and a full-time project manager. The project reported to a project steering committee, which reported to the program steering committee, which, in turn, reported to the agency's IT steering committee. Each geographic area also had a small implementation team (usually two or three people) as did each facility (usually up to six part-time resources). The central project followed a formal in-house-developed project methodology and practiced formal risk management. The project manager maintained a risk register and reported the status of 'high-highs' and the identification of any new risks monthly to the project steering committee. Local implementation teams were less formally managed, typically not following any specific project methodology, not practicing formal risk management, and not following the schedule defined at the start of their sub-projects. In sum, for this large-scale implementation, this project substantially followed the classic ('pure') model of a structured software project according to generally accepted 'best practice' project management principles and practices, with the exception of the local implementation teams.

In the public sector, the key advantage of this type is that it enables attention and effort to be focused on the target objective(s) with minimal distraction from the normal day-to-day work responsibilities that project team members might otherwise be engaged in. The main weakness of this type in the projects studied was its limited inherent authority in getting buy-in from key stakeholders. Securing representation, for example, on the steering committee or project team, did not necessarily equate to secondment of the most suitable resources or commitment to the project's goals and activities by business unit stakeholders. Similarly, the major challenges of 'pure' projects in the study related to the adequacy of the project and risk management skills available to the project, and to achieving and maintaining alignment of the project with organizational objectives and strategies. For example, in the case of the COTS system implementation, at the start of the project the agency's facilities were spread across seventeen geographic areas. However, a restructure mid-project amalgamated many areas reducing the number to nine. This substantially reduced the scale of work for the central project team. However, no adjustments were made to the central project's plans to assist the local areas with their enlarged scope of responsibility.

Table 4
Types of projects

	'Pure' project form	Hybrid form	Operational activity	Breakthrough event
Characteristics	Traditional project. A discrete, temporary activity, structured, operated and managed under conventional project management disciplines (to varying degrees of formality and completeness)	A combination of *project form* and *operational activity*. A core project structure exists, but key elements of the project are delivered by one or more functional units that exist and operate independently of the project	A recurring activity that is conceptualized as a project but is executed by functional units within their normal operational space and structure, using elements of project management control (like schedule, budget, reporting, etc.)	A focused effort by a small designated team to achieve a specific, high priority objective in a short timeframe without the constraints of conventional disciplines, practices or methods
Number of projects	5	6	5	1
Time horizon	Variable	Short–medium	Short–medium	Short
Locus of control	Project manager; governance framework	Project and functional management	Functional and line management	Executive
Project management	Formal	Semi-formal	Informal	None
Risk management	Formal	Control limited	Control limited	Intuitive
Advantages	Enables dedication of effort to target objective	Optimizes use of specialist resources	Focuses operational effort and avoids artificial overheads	Focuses effort on quick results
Weaknesses	Getting buy-in and participation from stakeholders outside of the project team	Difficult to enforce compliance and accountability	Project/risk management disciplines subordinated to business unit management/ practices	Highly dependent on 'heroes' (the skills and dedication of individual team members)
Key challenges	Adequacy of project and risk management skills; achieving/ maintaining project and organizational alignment	Balancing competing objectives, practices and resource demands	Balancing operational practices and expedience with accountability for delivery and quality control	Tends to leave 'loose ends', ill-fitting solutions and unresolved issues that hamper subsequent activities/operations

4.3.2. Operational activity

This is an activity that is conceptualized as a project but is executed by functional units within their normal operational space and organizational structure, using elements of project management control (such as scheduling, monitoring, and reporting). Five projects of this type were found in the study. Of these, three practiced no risk management while two practiced informal risk management as described in Section 4.1.

For example, the operations of one agency were dominated by a politically sensitive monolithic core business system that was continuously developed (enhanced or changed) and maintained. The agency's IT Division had dedicated departments responsible for specific parts of this operation, including multiple development teams (each responsible for a particular business function within the system), testing and quality control, and implementation and front-line support. These departments were permanently staffed (although some were contractors). The system was upgraded on a release basis, typically with two major and two minor releases per year. Each development team worked on changes targeted for a particular release. Each release was considered to be a project. All work within departments was attributed to one or more projects but all staff held positions within a functional organization structure and reported substantively to a functional unit manager. A project manager was assigned for the overall release, and each contributing department also had its own project manager for its input to that release. The formality of the project plan varied, depending on the release project manager, although it usually was limited to a set of key hand-over dates between functional units (from development to testing to implementation to production). There was no steering committee, and interaction with the business units requiring the changes was informal and limited to the manager of the development team responsible for the respective business function in the system. A formal in-house-developed project management methodology existed but was not rigidly or consistently followed. Formal risk management tended to occur only for critical major releases, and then the emphasis was on initial risk identification. Ongoing monitoring and maintenance of the risk register was left to the release project manager, and tended to dissipate as the project progressed. In sum, the dominant form was functional. Projects were more a means of segmenting work within the operational flow of the functional units than a framework for applying conventional project management disciplines and practices.

The key advantages of this type of project in the study were that it avoided establishing and maintaining project overheads that were non-critical to the operating function itself, and it focused the activities of the functional units involved through segmentation of their work. The main weakness of this type was that project and risk management disciplines tended to be subordinated to functional unit priorities and practices. The project's objectives were the objectives of the operating unit, not a contributing subset. Furthermore, the project's independent existence, from a work management perspective, was essentially a mirage. For example, speaking of the project from one agency, the IS Director held the view that "the plan is not the gospel for us. Delivery of services and value is more important than keeping to a plan. I had the vision of the grand master plan but that was not burdened on the people working on the project".

The main challenge for 'operational activities' in the study was balancing operational practices and pressures for expedience, with accountability for project delivery and quality control. Most projects still had targets of some kind (in terms of budgets, target due dates, and/or scope), but the formal practices and structures of project management that are needed to enable these objectives to be met were often missing because they were not directly relevant to the needs of the functional unit. Furthermore, project failure could always be attributed to some problem in the operating chain.

4.3.3. Hybrid form

This is a combination of the *'pure' project form* and *operational activity*. A core project structure exists, but key elements of the project are delivered by one or more functional units that exist and operate independently of the project. Six projects of this type were found in the study. Five practiced formal or informal risk management and one practiced no risk management.

For example, one project was tasked with developing a small specialist registry application for a business unit within an agency. The project was expected to take 6–9 months to complete. A project team was established comprising a project manager, project sponsor, and a user representative. Requirements and development work were contracted to a vendor while, internally, the team was dependent on the agency's infrastructure group to provide and set up the required operating environment, and a separate testing and implementation group to put the system into production. The main roles of the project team were to monitor, control and facilitate resolution of any issues raised by the contributing parties. The project reported to a project steering committee (which, in turn, reported to a technology steering committee) on an exception and milestone achievement basis. A tailored version of the PRINCE2 methodology was used, commensurate with the size of the project, however, no formal plan existed for the whole project. Rather, a best 'guesstimate' of the full duration and scope was formed, but detailed planning was only done for immediate tasks through to the next milestone. Issues and changes were formally managed but risks were not. Critical issues arose in getting commitment and delivery from the internal infrastructure and testing/implementation groups due to conflicting operational priorities. However, the project team and steering committee had limited ability to influence the contribution of these functional units because these departments had to juggle ongoing operational commitments to other business activities. The project did not have full control over the delivery of all of its work packages.

This type had the advantage of optimizing the use of specialist resources, but the disadvantage of difficulty in enforcing compliance and accountability because of limited control over the deployment of those resources. The main challenge for 'hybrid form' projects was balancing competing objectives, practices and resource demands. Take testing, for example. Some organizations funneled testing of integrated systems through a central quality control and testing function to preserve the integrity of existing systems and assure the quality of the new functionality or associated integrated system. This required an ongoing established group with specialist systems knowledge and skills, and dedicated test environments and facilities. However, such groups were often working on multiple projects as well as providing frontline support for resolving problems in production. At any one time, they might not have been able to perform the planned work for a particular project because resources had been redeployed or test environments had been held up by other activities. The longer the project, the greater this challenge becomes for hybrid form projects as unforeseen demands on the specialist resources can arise. This presents a specific challenge for risk management.

4.3.4. Breakthrough event

The last project type is characterized as a focused effort by a small dedicated team to achieve a specific, high priority objective in a short timeframe without the constraints of conventional project management disciplines, practices or methods. One project in the study was of this type.

The head of one agency directed a manager to create a new 36 seat telephone call center with a supporting Customer Relationship

Management (CRM) system to quickly meet a sensitive inter-agency need. He was set a target of three weeks for completion. No infrastructure existed at the start of the project. The manager put together a small team of three consultants, a vendor specialist and himself and, together, they energetically set about making the center a reality. There was no project manager; no formal project management; no project plan; no requirements specification; no risk assessment or risk management; no user participation; and no formal testing. Work proceeded based on broad initial directions, the manager's best guess about what was needed, intuition, and emergent discovery. The CRM package chosen was highly configurable, which was important because the intended operations of the call center were atypical. The team failed to meet the three week target but the center did commence operations six weeks after the start of the project, at the start of the business's busy season. Call-takers initially reacted negatively to the new system because they were unfamiliar with it (there had been no training); they had not been consulted on how it was set up (they were all familiar with the operations of other centers that had been disbanded); and it did not fully match what they needed. Over the next 6 months, however, business operations stabilized sufficiently for the team to determine exactly what changes were needed. A revised system was then implemented, paid for with funds left over from the initial implementation. Despite their initial disaffection, the call center team viewed this as a better outcome than if a functionally rich system had been initially delivered that did not match their ultimate requirements, because there would have been no money left in the budget to do anything about it.

The main advantage of 'breakthrough event' projects is that they focus effort on quick results. They can be a means of achieving significant progress in a short period of time. However, their success is highly dependent on 'heroes' (the skills and dedication of individual team members). The key organizational challenge with this type is that it tends to leave 'loose ends', ill-fitting solutions and unresolved issues that hamper subsequent operations.

Considering the project types overall, some variation in the relative importance of risk factor categories was found between types, especially the two least formal types (operational activity and breakthrough event), as might be expected.

Taking *operational activity* first, the least important risk factors were *project setup* and *partner engagement*, reflecting the fact that formal project management was not a major consideration for this type and that most of the work tended to be performed with internal resources. Other factors of low importance also reflect the finding that the operational activity type tended not to adopt formal project management practices. These were *project governance*, *business proprietorship*, *project management*, *change management*, and *benefits realization*. The most important factors in this project type were *management of projects*, *recognition of red flags* and *management of risk*.

In the case of the *breakthrough event* project type, only four factor categories were identified as relevant: *project setup* (it was important to get the right support and team in place); *change management* (largely ignored by the project examined); *management of projects* (good management was needed to achieve a hard target in a short period of time with minimal resources); and *recognition of red flags* (project-specific factors were very important in achieving success). However, this finding should be interpreted with caution because there was only one project of this type in the study.

There were no discernible differences in the pattern of factors identified between the *'pure' project form* and *hybrid form* types. This is not surprising considering that, in contrast to the others, projects in these types tended to practice some level of formal project management and all but one practiced either formal or informal risk management. The differences tended to be in the emphases within risk factor categories. For example, the challenges facing *hybrid form* projects in securing the commitment and delivery of work packages from contributing functional units influenced the nature of several risk factors and red flags in these projects.

In sum, these project types illustrate that there is a range of structural arrangements under which software projects operate. The view may be taken that some types inadequately apply the intentions and principles of project management. The view taken here, as reflected in the agency cases studied, is that they represent a need for more flexible project arrangements to meet the contextual circumstances of the task and agency at the time. The direct implication of this finding is that a uniform view of, and approach to project and risk management is unlikely to fully address the specific challenges associated with all project types found in practice. That is, 'one size' project and risk management does not 'fit all' (Shenhar, 2001).

In sum, the study of risk management in the public sector found a mix of effective and ineffective practices, similar to those encountered in private sector projects. Agencies that did manage risk well appeared to have benefited from the investment. Others succeeded due to fortuitous arrangements such as strong business sponsor–project manager relationships or good fortune. Overall, risk management practices in the public agencies studied appeared to lag the benchmark of 'best practice' prescribed in the literature, supporting the finding in the literature review. Risk was not well understood or well managed in many agencies, and risk management tended to be unsystematic and informal, even when projects were otherwise formally managed. In nearly a third of projects, risk was not intentionally managed at all. These findings are surprising considering the high level of scrutiny and accountability that usually operates in this sector.

5. Discussion

This paper has examined risk and risk management in the literature and in a study of software projects in government agencies in an Australian State. Three overall conclusions are reached. First, a risk management capability can play an important role in managing software projects. Second, the conceptualization and development of risk and risk management theory in the literature lags the requirements of practice to handle the threats associated with the full spectrum of uncertainties faced by software projects. Third, the practice of risk management lags the understandings and prescriptions of risk and risk management found in the research literature.

5.1. Study limitations

The findings of the empirical study need to be interpreted cautiously because it has some limitations. First, it is based on the public sector. Practices in government agencies may not generalize to private sector projects. However, Ferlie (2002) argues that, despite differences, public sector agencies are not radically different as an organizational form. Also, contrary to the usual perception, a US study found no significant difference between sectors in cost and schedule overruns but client satisfaction tended to be higher for public sector projects (Baker et al., 1988); and a UK study found that for IT projects, the public sector performed marginally better than the private sector (Sauer and Cuthbertson, 2003). Indeed, Baker et al. (1988) concluded that many preconceptions of difference between the two sectors are not supported. Therefore, this may not be an important limitation.

Second, the data sample is small, not random and dominated by cases that were perceived by the informants to be successful. This may have limited or biased the findings of the study. However,

adding more failure cases may not have improved the perception of risk management in practice.

Third, the risk factors described above flow from the findings of the study and therefore have high 'face validity', but they have not otherwise been empirically validated. Despite their intuitive appeal, corroboration is needed from other public sector studies.

Fourth, study findings were based solely on the perceptions and reports of the informants. No attempt was made to validate informants' perceptions from the agencies' perspective or against any objective benchmarks of project performance. From the researcher's perspective, the 'success' of some of the projects was equivocal.

Finally, the study focused only on software projects. Other kinds of projects such as infrastructure, process redesign and outsourcing projects are also common in government agencies and may contribute practice-based insights that are also relevant to software projects.

5.2. Research implications

The analyses in the paper provide insights relevant to future research in the field.

First, research may be limited by the current dominant conceptualization and definition of risk as probability of an impact. We have seen that managers tend to be more concerned about the magnitude of potential impacts and that impact is a consideration in both foreseen and unforeseen threats. We have seen that risk practices in software projects tend not to apply the current view literally, instead adopting more qualitative assessments of risk. A broader view of risk in terms of uncertainty or threat may enable research developments that better equip practitioners to manage project threats and reduce the high variance reported in project performance outcomes (Johnson, 2006).

Second, it has been argued from the literature review and experiences of particular agencies that the scope of risk management is narrow compared to the potential threats that can and do impact software projects. It was also speculated that a broader, integrated view of threat management may be appropriate. Restricting treatments within projects to foreseeable impacts falls short of the range of uncertainties that can disrupt attention from fulfilling project plans and objectives. Organizations that are more sophisticated in project and risk management may circumvent the problem by including separate issue and crisis management processes, for example, in their portfolio of project management practices. Integrating risk management with other threat-related management processes is one way forward. Other approaches may also be relevant. Issue and crisis management are underdeveloped areas of inquiry (Pearson and Mitroff, 1993; Jaques, 2007), particularly in software projects. There is both an opportunity and need for further research to extend the scope of threat management, especially in support of software projects. Bannerman (2008a) investigates this issue further.

Third, the study suggests there is also need for further research on the integration of risk and project management and the interaction between the two in practice. For example, one of the findings of the public sector study was that many of the risks that threaten software projects are built into the project design at the start of the project life cycle. For example, rigid plan-based waterfall development and structured project management methodologies dominated the projects studied. However, rigid design in uncertain and changing environments can be a major source of system and project risk in itself (and was in some of the projects studied). It was also found in both the literature and the projects that there are many sources of risk and each project has context-specific 'red flags' that need to be recognized.

One approach to this problem may be to match the project and risk management approaches used and tailor them to the characteristics and contexts of each software project. That is, to adopt a contingency or method engineering approach to project and risk management design (Barki et al., 2001; Ropponen and Lyytinen, 1997, 2000; Shenhar, 2001). Another may be to apply adaptive learning (Sommer and Lock, 2004). Few guidelines are available in the literature to support organizations in adopting these approaches. A notable exception is Boehm and Turner (2004), who argue that "an organization should have a repository of 'plug-compatible' process assets that can be quickly adopted, arranged, and put in place to support specific projects" (p. 23). Further research is needed to investigate and develop this option as an integrated contingency view of project and risk management.

Finally, the study reaffirms management as a fundamental anchor in risk and project research. The study findings suggest that agencies tended to expect too much from engineering solutions to management problems. Engineering-based methods and techniques are valid tools for managers but are of limited value in unskilled hands, when used naively, or used as ends in themselves. There was a tendency to undervalue and under-employ critical organizational management capabilities to enable and support software projects in favor of formularized and often generic planning and process methodologies.

Finding the right amount of rigor or process in project control is a management issue (Boehm and Turner, 2004). Unexpected and unforeseen threats require rapid response management borne of deeply embedded capabilities in adaptation, improvisation, and value creation (Pavlak, 2004; Bannerman, 2008a). These management capabilities are developed in-house over long periods of time by deliberate learning from experience and by institutionalizing accumulated project and risk management competencies into the structures, processes and practices of the organization (Jugdev et al., 2007; Bannerman, 2008b). Practitioners would benefit from further research that focuses on the execution end of risk management as well as on front-end risk evaluation and monitoring processes.

5.3. Practice implications

Taking the state of risk management research in the literature as the current benchmark, the agency study found that organizations tend to lag in full application of this knowledge in practice. For example, after the initial round of risk identification, risk management tended to be relegated to the project manager, who often did little more than informally update the risk register before each steering committee meeting. Furthermore, checklists tended to be used naively, and risk management practice was often not sustained throughout the whole project or assessed at the end of the project. If this finding can be generalized to other organizations, several practice implications arise for project managers and stakeholders.

One strategy that may increase adoption and awareness is to explicitly include assessment of the role and contribution of risk management in post-implementation reviews.

One of the challenges facing the practice of risk management in organizations is that business executives and managers are driven by demonstrable outcomes, usually performance related. If a major project is successful, it can be difficult to unequivocally attribute any part of that outcome to risk management. It is also very unusual for project success to be attributed to risk management. Rather, success is usually attributed to good fortune (even luck) or, more likely, claimed by various individuals involved as resulting from their skills and unique contributions to the project.

On this basis, it can become easy for an organization that has had a project success to play down the importance of risk management in the next project. This can happen implicitly, by not being quite so formal in executing risk management processes next time

around, or even explicitly, as some form of resource- or cost-cutting measure.

This possibility points to the importance of conducting a post-implementation review (PIR) and explicitly including a formal assessment of the role, performance and contribution of risk management in the project in that review. The PIR is an opportunity for evaluating and improving any risk checklists, frameworks, risk management processes, risk response strategies, and tools and techniques used by the organization, and ensuring that the lessons learned are carried forward for the benefit of subsequent projects. It may also raise the profile of risk management in facilitating business outcomes. It has been argued that if we 'fail to learn' from our experiences in software projects, we will 'learn to fail' (Lyytinen and Robey, 1999).

Another strategy for managers is to be more proactive in developing quick response capabilities for handling realized threats – especially unforeseen ones. The paper suggests that these capabilities could be integrated with existing risk management processes rather than adopted as separate methods. A good way to begin is to develop a generic contingency response plan for a major disruptive event. This would include, for example, high level processes for establishing a response team, analyzing the problem, determining actions, containing impacts with workarounds, resolving the problem, implementing longer term remediation, and extracting lessons learned. Within this generic process framework, action plans could then be developed for specific organization- and project-specific vulnerabilities that can be readily identified. This is fundamentally what the agency referred to above did in responding to an unforeseen security breach of its web site. For other threats for which it is not possible or feasible to pre-plan, the generic response process might include mechanisms for drawing on individuals with high diagnostic skills in impact domains. A last step, which the above-mentioned agency had not achieved, is to institutionalize and legitimize this approach to threat management in the project governance framework of the organization and gain support for developing risk management as an ongoing real-time threat management capability, not just as a method for periodic risk planning and review.

Finally, in practice-oriented disciplines, it is not unusual for research to lag the needs of practice. Researchers learn from observation of effective practices and generalize these into emergent disciplinary theory. Risk managers and project managers should not wait for research to catch up to their needs. Rather, learn from experience – from what works and what does not work in particular situations – and experiment. Try new ideas if normative approaches do not improve the projects' performance. In this way, practice will lead research.

6. Conclusion

This paper has reviewed and reassessed the status of risk management research in the literature and practice in a sample of Australian public sector agencies. Software projects are complex multi-dimensional endeavors in any context, private or public, that are particularly susceptible to failure. It was found that the notion of risk as a threat of negative impact is relevant to software projects, and that there is a need to manage such threats to achieve beneficial project outcomes. It was also found that the development of risk and risk management in the literature lags the needs of the phenomenon in practice, and that adoption of risk concepts and risk management methods in practice lags the understandings and prescriptions found in the literature.

Given the potential cost and losses from failed software projects, researchers and practitioners must continue to learn from each other to reduce project failures and develop practices that consistently generate better project outcomes. Better risk management, as a project and organizational capability, is critical to achieving these objectives.

Acknowledgements

The paper benefits from comments provided by Mark Staples. NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

References

Abe, J., Sakamura, K., Aiso, H., 1979. An analysis of software project failure. In: Proceedings of the Fourth Software Engineering Conference, pp. 378–385.
Addison, T., Vallabh, S., 2002. Controlling software project risks – an empirical study of methods used by experienced project managers. In: Proceedings of SAICSIT, pp. 128–140.
Atkinson, R., 1999. Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria. International Journal of Project Management 17 (6), 337–342.
Baker, B.N., Fisher, D., Murphy, D.C., 1988. Project management in the public sector: success and failure patterns compared to private sector projects. In: Cleland, D.I., King, W.R. (Eds.), Project Management Handbook, second ed. Van Nostrand Reinhold, New York, pp. 920–934.
Bannerman, P.L., 2007. Software project risk in the public sector. In: Proceedings of the 2007 Australian Software Engineering Conference, 10–13 April, Melbourne, pp. 389–398.
Bannerman, P.L., 2008a. Toward an integrated framework of software project threats. In: Proceedings of the 19th Australian Software Engineering Conference, 26–28 March, Perth, pp. 139–148.
Bannerman, P.L., 2008b. Macro-processes informing micro-processes: the case of software project performance. In: Proceedings of the International Conference on Software Process, LNCS 5007, 10–11 May, Leipzig, pp. 12–23.
Barki, H., Rivard, S., Talbot, J., 1993. Toward an assessment of software development risk. Journal of Management Information Systems 10 (2), 203–225.
Barki, H., Rivard, S., Talbot, J., 2001. An integrative contingency model of software project risk management. Journal of Management Information Systems 17 (4), 37–69.
Bernstein, P.L., 1996. Against the Gods: The Remarkable Story of Risk. John Wiley & Sons, New York.
Boehm, B.W., 1988. A spiral model of software development and enhancement. IEEE Computer 21 (5), 61–72.
Boehm, B.W., 1989. Software Risk Management. Tutorial. IEEE Computer Society, Washington.
Boehm, B.W., 1991. Software risk management: principles and practices. IEEE Software 8 (1), 32–41.
Boehm, B.W., Ross, R., 1989. Theory-W software project management: principles and examples. IEEE Transactions on Software Engineering 15 (7), 902–916.
Boehm, B., Turner, R., 2003. Using risk to balance agile and plan-driven methods. Computer 36 (6), 57–66.
Boehm, B., Turner, R., 2004. Balancing Agility and Discipline: A Guide for the Perplexed. Addison-Wesley, Boston.
Bourne, L., 2007. Avoiding the successful failure. In: PMI Global Congress, Asia Pacific, Hong Kong, 29–31 January.
Brooks Jr., F.P., 1975. The Mythical Man Month: Essays on Software Engineering. Addison-Wesley, Reading.
Charette, R.N., 1989. Software Engineering Risk Analysis and Management. McGraw-Hill, New York.
Charette, R.N., 1996. The mechanics of managing IT risk. Journal of Information Technology 11 (4), 373–378.
Charette, R.N., 2005. Why software fails? IEEE Spectrum 42 (9), 42–49.
Cule, P., Schmidt, R., Lyytinen, K., Keil, M., 2000. Strategies for heading off project failure. Information Systems Management 17 (2), 65–73.
Davis, G.B., 1982. Strategies for information requirements determination. IBM Systems Journal 21, 4–30.
de Camprieu, R., Desbiens, J., Feixue, Y., 2007. 'Cultural' differences in project risk perception: an empirical comparison of China and Canada. International Journal of Project Management 25 (7), 683–693.
DeMarco, T., Lister, T., 2003. Waltzing with Bears: Managing Risk on Software Projects. Dorset House Publishing, New York.
Elkington, P., Smallman, C., 2002. Managing project risks: a case study from the utilities sector. International Journal of Project Management 20 (1), 49–57.
Ferlie, E., 2002. Quasi strategy: strategic management in the contemporary public sector. In: Pettigrew, A., Thomas, H., Whittington, R. (Eds.), Handbook of Strategy and Management. Sage Publications, London, pp. 279–298.
Frame, J.D., 2003. Managing Risk in Organizations: A Guide for Managers. Jossey-Bass, San Francisco.
Ibbs, C.W., Kwak, Y.H., 2000. Assessing project management maturity. Project Management Journal 31 (1), 32–43.

Jaques, T., 2007. Issue management and crisis management: an integrated, non-linear, relational construct. Public Relations Review 33 (2), 147–157.
Jiang, J.J., Klein, G., Ellis, T.S., 2002. A measure of software development risk. Project Management Journal 33 (3), 30–41.
Johnson, J., 2006. My Life is Failure: 100 Things You Should Know to be a Successful Project Leader. Standish Group International, West Yarmouth, MA.
Johnson, J., Boucher, K.D., Connors, Y., Robinson, J., 2001. Project management: the criteria for success. Software Magazine 21 (1), S3–S11.
Jugdev, K., Mathur, G., Fung, T.S., 2007. Project management assets and their relationship with the project management capability of the firm. International Journal of Project Management 25 (7), 560–568.
Keil, M., Cule, P.E., Lyytinen, K., Schmidt, R.C., 1998. A framework for identifying software project risks. Communications of the ACM 41 (11), 76–83.
Keil, M., Wallace, L., Turk, D., Dixon-Randall, G., Nulden, U., 2000. An investigation of risk perception and risk propensity on the decision to continue a software development project. Journal of Systems and Software 53 (2), 145–157.
Keil, M., Tiwana, A., Bush, A., 2002. Reconciling user and project manager perceptions of IT project risk. Information Systems Journal 12 (6), 103–119.
Kerzner, H., 2003. Project Management: A Systems Approach to Planning, Scheduling, and Controlling, eighth ed. John Wiley & Sons, Hoboken.
KPMG., 2005. Global IT Project Management Survey. KPMG, Australia.
Lam, W., 2004. Technical risk management on enterprise integration projects. Communications of the AIS 13, 290–315.
Lucas, H.C., 1981. Implementation: The Key to Successful Information Systems. Columbia University Press, New York.
Lyytinen, K., Robey, D., 1999. Learning failure in information systems development. Information Systems Journal 9 (2), 85–101.
Lyytinen, K., Mathiassen, L., Ropponen, J., 1996. A framework for software risk management. Journal of Information Technology 11 (4), 275–285.
Lyytinen, K., Mathiassen, L., Ropponen, J., 1998. Attention shaping and software risk: a categorical analysis of four classical risk management approaches. Information Systems Research 9 (3), 233–255.
March, J.G., Shapira, Z., 1987. Managerial perspectives on risk and risk taking. Management Science 33 (11), 1404–1418.
McFarlan, F.W., 1981. Portfolio approach to information systems. Harvard Business Review 59 (5), 142–150.
McKeen, J.D., Smith, H.A., 2003. Making IT Happen: Critical Issues in IT Management. John Wiley & Sons, Chichester.
Morris, P.W.G., 1994. The Management of Projects. Thomas Telford, London.
Morris, P.W.G., 1996. Project management: lessons from IT and non-IT projects. In: Earl, M.J. (Ed.), Information Management: The Organizational Dimension. Oxford University Press, Oxford, pp. 321–336.
Mursu, A., Lyytinen, K., Soriyan, H.A., Korpela, M., 2003. Identifying software project risks in Nigeria: an international comparative study. European Journal of Information Systems 12 (3), 182–194.
Pavlak, A., 2004. Project troubleshooting: tiger teams for reactive risk management. Project Management Journal 35 (4), 5–14.
Pearson, C.M., Mitroff, I.I., 1993. From crisis prone to crisis prepared: a framework for crisis management. Academy of Management Executive 7 (1), 48–59.
Pender, S., 2001. Managing incomplete knowledge: why risk management is not sufficient. International Journal of Project Management 19 (2), 79–87.
Pfleeger, S.L., 2000. Risky business: What we have yet to learn about risk management? Journal of Systems and Software 53 (3), 265–273.
Remenyi, D., 1999. Stop IT Project Failures through Risk Management. Butterworth Heinemann, Oxford.
Ropponen, J., 1999. Risk assessment and management practices in software development. In: Willcocks, L.P., Lester, S. (Eds.), Beyond the IT Productivity Paradox. John Wiley & Sons, Chichester, pp. 247–266.
Ropponen, J., Lyytinen, K., 1997. Can software risk management improve system development: an exploratory study. European Journal of Information Systems 6 (1), 41–50.
Ropponen, J., Lyytinen, K., 2000. Components of software development risk: How to address them? A project manager survey. IEEE Transactions on Software Engineering 26 (2), 98–112.
Rubenstein, D., 2007. Standish Group Report: There's Less Development Chaos Today. Software Development Times, 1 March, <http://www.sdtimes.com/article/story-20070301-01.html>.
Sauer, C., Cuthbertson, C., 2003. The State of IT Project Management in the UK. ComputerWeekly.com, <http://www.cw360ms.com/pmsurveyresults/index.asp>.
Sauer, C., Liu, L., Johnston, K., 2001. Where project managers are kings? Project Management Journal 32 (4), 39–49.
Schmidt, R., Lyytinen, K., Keil, M., Cule, P., 2001. Identifying software project risks: an international Delphi study. Journal of Management Information Systems 17 (4), 5–36.
Schultz, R.L., Slevin, D.P., Pinto, J.K., 1987. Strategy and tactics in a process model of project implementation. Interfaces 17 (3), 34–46.
Schwalbe, K., 2007. Project Risk Management. Information Technology Project Management, fifth ed. Thomson Course Technology, Boston. pp. 446–488.
Shenhar, A.J., 2001. One size does not fit all projects: exploring classical contingency domains. Management Science 47 (3), 394–414.
Shenhar, A.J., Dvir, D., 2007. Project management research: the challenge and opportunity. Project Management Journal 38 (2), 93–99.
Simister, S.J., 2004. Qualitative and quantitative risk management. In: Morris, P.W.G., Pinto, J.K. (Eds.), The Wiley Guide to Managing Projects. John Wiley & Sons, Hokoben, pp. 30–47.
Sommer, S.C., Lock, C.H., 2004. Selectionism and learning in projects with complexity and unforeseeable uncertainty. Management Science 50 (10), 1334–1347.
Taylor, H., 2006. Risk management and problem resolution strategies for IT projects: prescription and practice. Project Management Journal 37 (5), 49–63.
Tiwana, A., Keil, M., 2004. The one-minute risk assessment tool. Communications of the ACM 47 (11), 73–77.
Wallace, L., Keil, M., Rai, A., 2004. How software project risk affects project performance: an investigation of the dimensions of risk and an exploratory model. Decision Sciences 35 (2), 289–321.
Ward, S., Chapman, C., 2004. Making risk management more effective. In: G Morris, P.W., Pinto, J.K. (Eds.), The Wiley Guide to Managing Projects. John Wiley & Sons, Hoboken, pp. 852–875.
Wateridge, J., 1998. How can IS/IT projects be measured for success? International Journal of Project Management 16 (1), 59–63.
Willcocks, L.P., Griffiths, C., 1997. Management and risk in major information technology projects. In: Willcocks, L.P., Feeny, D.F., Islei, G. (Eds.), Managing IT as a Strategic Resource. McGraw-Hill, Berkshire, pp. 203–237.
Zhang, H., 2007. A redefinition of the project risk process: using vulnerability to open up the event-consequence link. International Journal of Project Management 25 (7), 694–701.
Zmud, R.W., 1979. Individual differences and MIS success: a review of the empirical literature. Management Science 25 (10), 966–979.

Paul Bannerman is a Research Scientist at NICTA, Australian Technology Park, Sydney; a Visiting Research Fellow at the University of New South Wales; and Adjunct Faculty member of the Australian School of Business (formerly Australian Graduate School of Management, AGSM). His research interests are in IT-enabled organizational change and performance, strategic alignment of business and IT, and project and risk management. He has a BA in economics and finance and an MBA from Macquarie University; an MSc in computing from University of Technology, Sydney; and a PhD in management from the AGSM (University of Sydney and University of New South Wales). He can be contacted at paul.bannerman@nicta.com.au.