Managing Structure-Related Software Project Risk:  
A New Role for Project Governance

Paul L. Bannerman  
NICTA and School of Computer Science and Engineering  
University of NSW  
Sydney, Australia  
paul.bannerman@nicta.com.au

Abstract—This paper extends recent research on the risk implications of software project organization structures by considering how structure-related risk might be managed. Projects, and other organizations involved in projects, are usually structured according to common forms. These organizational entities interact with each other, creating an environment in which risks relating to their structural forms can impact the project and its performance. This source of risk has previously been overlooked in software project research. The nature of the phenomenon is examined and an approach to managing structure-related risk is proposed, responsibility for which is assigned as a new role for project governance. This assignment is necessary because, due to the structural and relational nature of these risks, the project is poorly placed to manage such threats. The paper argues that risk management practices need to be augmented with additional analyses to identify, analyze and assess structural risks to improve project outcomes and the delivery of quality software. The argument is illustrated and initially validated with two project case studies. Implications for research and practice are drawn and directions for future research are suggested, including extending the theory to apply to other organization structures.

Keywords - software project; risk management; organization structure; governance

I. INTRODUCTION

In the ongoing effort to improve software processes and practices to improve product quality and project outcomes, recent research has pointed to an area of project risk that has previously been overlooked, namely, the structural makeup of projects. That is, the potential impact that a project’s design, as an organizational structure, can have on project risk and performance [5]. This includes how a project entity interacts with other structural entities in the software development process, such as other business units, clients, vendors and consultants that may be involved in the project. Common organization structures include functional bureaucracies, project-based organizations, matrix-based organizations, and more flexible team-based structures. We know that the more complex a project becomes, the lower the likelihood of success [9, 34], but research has paid little attention to the role of organization structures in contributing to project complexity and resultant outcomes.

This paper extends prior work [4, 5] by considering further the nature of structure-related risk in software projects and how it might be managed. It is argued that because the perspective of the project manager and project team is focused internally on project activities after it is formed, rather than on the structural context of the project, there is need for an external authority to facilitate and coordinate risk management of structure-related risk. The project governance framework is viewed as being ideally positioned to perform this role.

Project governance is still an emerging field in research and practice. Governance is a multi-dimensional construct whose functions are variously conceived as including alignment, stewardship, accountability, compliance, decision rights allocation, oversight, control, propriety assurance, risk management, resource allocation and issue resolution. However, it is an important field as early research has found that companies with better than average IT governance earn at least 20% higher return on assets than organizations with weaker governance [30]. The aim of this paper is not to explicate or rationalize the functions of governance, but to propose an additional role in managing structure-related risk.

Why is structure-related risk important to software engineering? Primarily because it forms part of the macro process domain of software engineering and this research is motivated to improve the delivery of quality software. Most software development takes place through the vehicle of a software project. While not all software project risk results from software engineering practices, the performance of software projects reflects directly on our ability to engineer software, so it is critical that the delivery context of software engineering receives concurrent research attention.

The paper addresses three questions: What are common sources of structure-related risk? How might structure-related risks be managed? And what techniques might help manage structure-related risks?

The paper proceeds as follows. Section II reviews prior research on organization structures, structure-related risk, and project governance. Section III then outlines the central propositions of the paper in response to the research questions. This is supported by two case studies in Section IV. Finally, the contribution is discussed in Section V before the paper is concluded.

II. PRIOR RESEARCH

This section reviews prior work on organization structure-related risk and governance.
A. Organization structures and risk

Fundamentally, organization structure is about how organizational work is designed (an architecture of work), which is reflected in a corporation’s organization chart. Common approaches to organization design are to group work together under an authority structure based on common business functions or processes, products or projects, customer segments or geographic areas, according to the purpose of the organization. Common structures include the functional form (including the machine bureaucracy, simple form and multidivisional form), project-based organization, matrix organization, professional bureaucracy and adhocracy (also called the innovative organization form) [15, 18, 22].

Some research has been done on projects of different structural types (e.g., [18, 28]) and the relationship between structure and project risk and performance (e.g., [19, 21]). Recent research has extended this work by studying the risk implications of software project organization structures [5]. Contrary to a common assumption of standards and bodies of knowledge that projects are uniform in structure, evidence is reported that organizations adopt a range of structural arrangements in practice. Furthermore, these arrangements have project and risk management implications. By analyzing characteristics of the project structures, specific sets of risk factors (risk profiles) are identified for common organizational forms. To illustrate and support the case studies reported later in the paper, Tables I to III summarize risks profiles for three of the four main structural forms identified in [5]: the functional; project, and; matrix forms.

The functional form is the most common way of organizing, namely, according to functional expertise. This form features strong hierarchical authority and control; highly structured and routinized work; formal coordination and communication, and; a drive for achieving efficiencies based on economies of scale. Mintzberg aptly names this form the ‘machine bureaucracy’ [22]. Risks associated with the functional form are summarized in Table I.

<table>
<thead>
<tr>
<th>Risk</th>
<th>Structure-related drivers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Projects fall behind</td>
<td>Functional specialization dominates; inefficient in one-off, discrete activities</td>
</tr>
<tr>
<td>Engagement distraction</td>
<td>Functional conflicts; conflicting objectives and tensions distract focus</td>
</tr>
<tr>
<td>Operational inflexibility</td>
<td>Hierarchical control; tight vertical authority and reporting lines; rigid roles</td>
</tr>
<tr>
<td>Innovation suppressed</td>
<td>Drive for functional efficiency and scale economies favor the status quo &amp; stability</td>
</tr>
<tr>
<td>Size-related inefficiencies</td>
<td>Drive for scale efficiencies leads to large organizations</td>
</tr>
</tbody>
</table>

The project form emerged during and after World War II as a reaction to the rigidities of bureaucracies (the functional form). The aim and design of the project form is the opposite of the functional form. Here, projects are the main unit of organization rather than functions and the aim is organizing around achieving specific outcomes rather than routinizing repetitive work. Also, activities and reporting (usually to a single project manager) are designed to optimally suit the nature of each project. In a project-based organization, multiple projects are usually clustered under central management and shared support units. In this form, practicing project management according to ‘best practice’ standards is paramount. Risks associated with the project form are summarized in Table II.

<table>
<thead>
<tr>
<th>Risk</th>
<th>Structure-related drivers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost pressures</td>
<td>Non-routine; does not scale well; weak in coordinating processes</td>
</tr>
<tr>
<td>Strategic control</td>
<td>Detached; can lose connection with the strategic centre</td>
</tr>
<tr>
<td>Operational control</td>
<td>Operationally decoupled from functional controls; can be responsive and reactive</td>
</tr>
<tr>
<td>Stakeholder participation</td>
<td>Stakeholders can become detached and operationally detached</td>
</tr>
<tr>
<td>Contextual blindness</td>
<td>Myopic; insularity can result in loss of grounding in the originating purpose</td>
</tr>
<tr>
<td>Adequacy of personnel</td>
<td>Team resources are acquired sparingly; the acquired skills may not match needs</td>
</tr>
<tr>
<td>Capability development</td>
<td>Weak in learning; limited mechanisms or incentives for learning from experience</td>
</tr>
</tbody>
</table>

Finally, the matrix form aims to achieve the best of the functional and project forms. Typically, matrices are structured functionally vertically, overlaid by projects horizontally, with dual authority and reporting lines. Contrary to the design intentions, matrix-based organizations often become unworkable in practice, with decision making paralyzed and/or politicized. Risks associated with the matrix form are summarized in Table III.

<table>
<thead>
<tr>
<th>Risk</th>
<th>Structure-related drivers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unclear accountabilities</td>
<td>Dual reporting; has competing authority structures and split loyalties</td>
</tr>
<tr>
<td>Poor decision-making</td>
<td>Stronger authority lines can dominate decisions, creating an authority bias</td>
</tr>
<tr>
<td>Slow response</td>
<td>Power, authority and accountability are often separated</td>
</tr>
<tr>
<td>Control problems</td>
<td>Tendency toward anarchy; control can break down; self interest can take over</td>
</tr>
<tr>
<td>Staff stress and turnover</td>
<td>Dual reporting, conflict and ambiguity can result in personnel issues and stress</td>
</tr>
</tbody>
</table>

The tradition of risk and risk management in software engineering was established in [8] and [11] and is discussed in [4]. Having established the risk implications of common software project organization structures in [5], this paper extends this work by considering how structure-related risks might be managed. It is argued that (a) due to the nature of the threat, a different locus of control may be required to manage these risks (specifically, the project governance framework), and (b) that existing risk management practices require additional techniques to identify, analyze and assess structure-related risks (a process is proposed that integrates structure analysis, stakeholder analysis, and traditional risk
technology (IT) governance and project governance. Subordinate to corporate governance lie other domain-specific governance arrangements, including information technology (IT) governance and project governance.

At the Board and executive level, corporate governance is concerned with enacting arrangements to direct, administer and control a corporation to the satisfaction of major stakeholders (usually shareholders) and regulators. Subordinate to corporate governance lie other domain-specific governance arrangements, including information technology (IT) governance and project governance.

First, at the IT level, it is generally considered that IT governance aims to achieve a better alignment between the business and IT. Its scope includes [33]:

- Strategic alignment
- Risk management
- Delivery of business value through IT
- Control and accountability
- Performance management
- Capability management

IT governance is viewed as comprising structures, processes and relational mechanisms [29, 30]. Apart from some early interest in the structuring of IT in organizations (for example, whether it should be centralized, decentralized or positioned in a hybrid/federated form) [10], the focus of IT governance research has been mainly on strategic and operational alignment rather than structural alignment. More recently, IT governance standards have started to emerge; in particular, ISO/IEC 38500:2008, based on AS 8015-2005, and a derivative draft Australian Standard on the corporate governance of projects involving IT investments (08145). Industry frameworks such as ITIL and COBIT are also popular in practice, providing guidance on IT governance, among other things. A recent integrative view is provided by [29]. Also, a comprehensive list of IT governance frameworks is provided in [25].

Interest in project governance is more recent than IT governance, but is not yet well-established as a stream of research or a formally prescribed discipline in practice. Müller defines project governance as “the value system, responsibilities, processes and policies that allow projects to achieve organizational objectives and foster implementation that is in the best interests of all the stakeholders” [23, page 4]. Its overall aim is “a consistent and predictable delivery of projects and programs in accordance with their planned contribution to corporate strategy and stakeholder expectations” [23, page 16].

Methodologies such as PRINCE2 and bodies of knowledge such as PMI’s PMBOK Guide imply governance through the structured frameworks they propose but tend not to describe a discrete project governance function. For example, PRINCE2 contains a process group called Directing a Project (DP), targeting a Project Board that is “accountable for the project’s success by making key decisions and exercising control while delegating day-to-day management of the project to the Project Manager” [24, page 136]. Its processes cover initiation, authorization, stage boundaries, ad hoc direction, and project closure. The PMBOK Guide is less specific. Its only statement on the topic is that “Project governance provides a comprehensive, consistent method of controlling the project and ensuring its success. The project governance approach should be described in the project management plan. A project’s governance must fit within the larger context of the program or organization sponsoring it.” [26, page 20]. Otherwise, the PMBOK Guide appears to assume that the project, program and portfolio management structures and practices it describes provide de facto governance (presumably supported by OPM3, PMI’s Organizational Project Management Maturity Model). Similarly, SEI’s CMMI for Development [27] includes organizational and project process areas that are related to the notion of governance, but governance is not explicitly addressed.

By contrast, APM does provide a guide to governance but its focus is governance of project management rather than projects per se [2]. The guide aims to inform boards on project portfolio direction, project sponsorship, project management effectiveness and efficiency, and disclosure and reporting. Consistent with [23], however, effective project governance is viewed as including governance of an organization’s project management capabilities and practices as well as the direction, execution and control of its projects (individually and collectively in programs).

It would appear, therefore, that project governance is still an emerging practice. This is further supported by work on standards. The draft Australian Standard on the corporate governance of projects involving IT investments, mentioned above (DR 08145), applies the framework and principles of AS 8015-2005 to projects. It defines three main processes (direct, evaluate and monitor) and six principles:

1. Roles for the governance and management of projects and responsibility for project outcomes are clearly defined and understood.
2. Projects are aligned with the organization’s business and IT strategies.
3. Projects are undertaken validly and transparently.
4. Projects are governed to achieve agreed objectives with risks to the organization managed.
5. Projects conform to policies and regulations.
6. Projects demonstrate respect for human behavior.

This reflects a strong orientation towards strategic and operational alignment. However, there is no mention of mediation or alignment of the disparate organization structures involved in projects, or management of their associated risks.
Recent research on software development and software architecture governance (as distinct from project governance) argues that the principles underlying corporate and IT governance apply to other subordinate IT-related domains as well as projects [6, 7]. However, proliferating governance arrangements horizontally within the IT framework in addition to vertical mechanisms is likely to create a coordination and integration challenge. So far, however, research in governance in these other software-related domains is even less mature than in project governance.

Finally, other recent research has focused on the project management office (PMO) as a locus of project, and project management, governance (e.g., [16]). Much of this research is empirically based and is struggling with the diversity of PMOs found in practice (e.g., [3, 13, 17]). However, the research does point to the potential importance of the PMO as a governance mechanism – if it is developed beyond being a pure project administration body (that is, beyond a centre only for collecting and collating project status reports).

In the next section we extend this research by proposing an additional role for project governance beyond strategic and operational alignment, namely, to mediate structural entities in software projects and mitigate structure-related risks. We then illustrate this role with two case studies.

III. MANAGING STRUCTURE-RELATED RISK

This section addresses the three research questions.

A. Sources of structure-related risk

The first question concerns the potential sources of structure-related risk in software projects. Since projects are embodied within temporary organization structures (the project forms described in the previous section), one might intuitively assume that the project itself is the main source of structure-related risk. However, other organizational entities, with concomitant organization structures, are also involved in software projects. These can also be sources of significant structure-related risks for a project, as can interactions between the different organizational entities involved. Four main sources of structure-related risk factors can be identified from projects in the public sector study [4]:

Project organization. The project is usually organized according to one of the project types described above. This structuring relates to the project team itself, independently of any associated project governance structures, contributing business units and participating third party organizations that might also be involved in the project. Each project is subject to potential risk factors from the risk profile developed for each project form (project, matrix, functional or adhocracy). For example, most projects were structured as temporary project teams under the direction and control of the project manager, either within a business unit or corporate function (such as IT), or independently of all other functional and divisional line management structures.

Parent organization. The ‘parent’ organization is the organizational entity with the most direct responsibility for establishing and executing the project, and ensuring that the objectives of the project are achieved. The parent may or may not be funding the project investment (depending on whether it is for internal use or external delivery to a client). This source relates to the organization design of the parent organization independent of the project structure that operates under its control. Again, the structural form of the parent can create form-specific risk factors (from the risk profile of the associated organization structure type) that may impact its interactions with the project. In the public sector study, all of the agencies conformed to the classical functional (bureaucratic) structural form.

Note that for project-based organizations (a common form in software developing companies), the project and parent organizations may be closely related. The project may be one of the projects in the company’s portfolio of client-based projects. The key difference is that the project organization would relate to the project ‘silo’ (conceptually, the equivalent of a function in the functional form or division in the divisional form) while the parent organization would relate to the overall entity, encompassing the corporate office components and all projects that are active at a particular point in time.

Other influencing organizations. It is common for other parties to be involved in software projects in addition to the project and parent entities. These third parties can include, for example, clients, consultants, vendors and developers. These organizations are often in a position to significantly influence the project. Therefore, the potential risk factors arising from their structural forms, and/or the interaction of these entities with the other entities involved in the project activity, may also critically impact the project’s performance and outcomes. For example, the second case study in the next section features a multi-vendor project involving eight vendor companies of either functional/divisional or matrix organization forms.

Other structural entities within and external to a parent organization may also influence a project. Examples include governance-related structures, standards bodies, regulatory authorities and compliance entities. However, analysis of the structure-related risks associated with such entities and their influence on software projects are subjects for future research.

Interactions. Finally, interactions between these organizational entities during their involvement in a project can compound the threat to software projects. From the study, common interactions were found to occur between: (a) the project organization and the parent organization (and/or its subunits, such as functional divisions or business units); (b) the project organization and external organizations, and; (c) the parent organization and external organizations.

Different combinations of organizations of different structural forms can result in many project configurations and, consequently, many combinations of structure-related risks. Analyzing these configurations is outside of the scope of this study. However, this diversity implies that while it may become possible to identify common or generic patterns of structure-related risk associated with common project configurations (known as generic risks), analysis of context-specific structure-related risks is likely to remain essential.
B. Governance as a risk management mechanism

The second research question concerns how structure-related risk can be managed in software projects. The default assumption is that structure-related risks are still risks so the full theory and practice of risk management applies (for example, as recently reviewed in [4]). However, traditional software project risk management has some significant limitations in dealing with structure-related risk:

- First, these risks tend to have low visibility within projects. After a project is initiated, the structural context of the project is already determined and is often fixed and assumed to be invariable. The focus of risk management tends then to focus on operational rather than structural or contextual concerns. Consequently, structure-related risk may not receive the attention it deserves.
- Second, these risks are essentially external to the project once it is initiated so, as well as lying outside of the project manager’s immediate purview, they may also lie outside of his/her direct control. The project manager is constrained in the risk response strategies the project can apply to risks that arise in relation to other organizational entities (including the parent) involved in the project, and to the interactions between the multiple parties involved.
- Finally, additional considerations often come into play in dealing with other organizational entities that may be independent of the project engagement. These might be strategic, political, financial, legal, reputational or purely relational. The project may not be in a position to understand or act on these broader considerations.

Consequently, while the project has a role to play in identifying and managing structure-related risk, it is unlikely that it can manage all such risks alone. An authority external to the project and with the span of influence to consider any broader implications is required to own structure-related risk management and work with projects to mediate and mitigate the effects of the different organizational structures involved. The fundamental proposition of this paper is that governance (and project governance in particular) is ideally positioned to take on this additional role.

As seen from the literature, IT governance and project governance have traditionally focused on business value delivery from IT investments through business-IT alignment, and IT risk management through accountability and control. Managing structure-related risk is highly consistent and fits the domain of project management. Typically, a peak program or project steering committee mediates the activities of one or more projects or programs with higher level governance mechanisms responsible for the overall portfolio. These might be an IT Steering Committee, an executive management team or committee, or even the Board of Directors. Beneath the program or project steering committee might be a program or project management office (PMO), which mediates and/or supports the activities of individual programs and projects. Each governance structure strategically and operationally aligns the activities below it with the interests of the stakeholders above. Without these mechanisms, the project is detached from its stakeholders.

The strategic and operational alignment role of this framework is illustrated by considering some common ‘top 10’ project success/risk factors:

- **Executive support.** Governance can make projects visible to executives and enable them to directly participate and contribute to project outcomes.
- **User involvement.** Governance can ensure that suitable business unit representatives and users are involved and business interests drive the project.
- **Clear business objectives.** Business representation in project governance can facilitate formulation, expression and understanding of objectives.
- **Realistic plan.** Governance can provide a forum for overseeing the project plan and monitoring progress to ensure the project’s scope remains relevant and realistic.
- **Firm requirements.** Participation in governance can enable stakeholders to influence project design to ensure requirements are satisfactorily determined.
• **Experienced project manager.** Governance can ensure that competent managers are assigned to projects.

• **Skilled resources.** Governance can ensure that the required capabilities are deployed for the project, sourced internally or externally.

• **Appropriate methodology.** Governance can play a role in supporting projects to choose and apply suitable processes and practices (this is usually via the PMO).

• **Tools and infrastructure.** Similarly, PMOs can support projects by facilitating required project technology infrastructure and tools.

• **Organizational change management.** Depending on the project’s scope, transitioning software into operation in a client’s site can be facilitated by governance.

In a similar manner, governance capabilities can also be applied to managing structure-related risk, including structuring projects and managing interactions between the parties involved. Governance is an ideal structural, relational and processual mechanism to mediate relationships between different structural entities and mitigate the effects of any risks resulting from these structures on the projects under its control. (Note that there is no assumption of determinism here. Governance can mitigate project risk and success through these factors but only if it is effectively applied.)

On these bases, it is concluded that governance is ideally placed to own and manage structure-related risk in software projects.

C. **Specific structure-related risk management techniques**

The third research question extends consideration of how structure-related risks might be managed to practical techniques. Since structure-related risk, as a source category of risks, extends beyond the traditional domain of software project risk analysis, existing risk management techniques (designed for strategic and operational project threats) may not be sufficient. In this section, first, a new analysis technique is proposed (structure analysis), and an existing technique (stakeholder analysis) is adapted to support the other techniques, before an integrated process of structure-related risk management is outlined.

**Structure analysis.** This is proposed as a new technique. The aim of structure analysis is to determine the structural form of each organization involved in the project. The analysis assumes the structural forms described in [5] (viz., project, matrix, project and adhocracy forms). Table IV lists indicative structural characteristics of the forms as a reference guide to assist the analyst in identifying the form of each entity. The analysis process involves examining each

<table>
<thead>
<tr>
<th>TABLE IV. STRUCTURAL CHARACTERISTICS OF MAJOR ORGANIZATION TYPES</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Organizational Forms</strong></td>
</tr>
<tr>
<td><strong>Work organization:</strong></td>
</tr>
<tr>
<td>Form</td>
</tr>
<tr>
<td>Nature of work</td>
</tr>
<tr>
<td>Individual contribution</td>
</tr>
<tr>
<td>Learning orientation</td>
</tr>
<tr>
<td><strong>Coordination:</strong></td>
</tr>
<tr>
<td>Coordination</td>
</tr>
<tr>
<td>Centralisation</td>
</tr>
<tr>
<td>Standardisation</td>
</tr>
<tr>
<td>Formalisation</td>
</tr>
<tr>
<td>Reporting line</td>
</tr>
<tr>
<td><strong>Other structure elements</strong></td>
</tr>
<tr>
<td>Duration</td>
</tr>
<tr>
<td>Dominant force</td>
</tr>
<tr>
<td>Timeframe focus</td>
</tr>
<tr>
<td>Typical size</td>
</tr>
<tr>
<td>Scope</td>
</tr>
<tr>
<td>Importance of economies of scale</td>
</tr>
<tr>
<td>Flexibility</td>
</tr>
</tbody>
</table>
organization against its organization chart, organizational characteristics defined in the literature (such as summarized in [5]) and in Table IV, and classifying them according to the most likely matching form. (Note that in organization theory, these forms are ideal generic types; actual organizations may not exactly match the form of the ideal type.)

A key question can be based on the attribute categories in Table IV: How is work organized in the entity? How is work coordinated? What other structural characteristics point to a specific form?

Structural form is usually stable so the analysis will most likely only need to be done once, as early as possible in the project life cycle. However, monitoring for structural change should be included in the risk management cycles.

Understanding of the characteristics of the organization structures involved is also carried forward to assist in other risk management processes, such as stakeholder analysis.

**Stakeholder analysis.** Stakeholder analysis is an existing technique that can be included in risk management practices for structure-related risks. In general, stakeholders are individuals or groups who have a vested interest in the project, what it produces and/or its outcomes. Stakeholders may be involved in or affected by the project. Here, the interest is in stakeholder groups (organizations or parts thereof). Stakeholders with convergent interests may be enlisted to support the project and help it progress while stakeholders with divergent interests may impede the project, either overtly or covertly, necessitating their impact being neutralized.

For the present purposes, stakeholder analysis has two main steps: first, to identify the key organizations that have a stake in the project, and; second, to identify potential stakeholder interests relating to the characteristics of their organizational structure or interactions with other stakeholders involved in the project, that might positively or negatively impact the project.

An example of a potential structure-related stakeholder threat might be that business user resource, provided to a vendor project structure.

**Stakeholder theory,** analysis and mapping techniques can be readily found in the literature (e.g., [14]) and on the Web (e.g., [1]). Typically, stakeholders are assessed against three dimensions:

- Power (high, medium, low)
- Support (positive, neutral, negative)
- Interest (high, medium, low)

These two techniques (structure analysis and stakeholder analysis) can be combined with traditional software project risk management processes to form an integrated approach to managing structure-related risk, as follows.

**Structure-related risk management.** By adapting the traditional risk management process, structure-related risk can be managed via the following high level steps (the relevant management techniques are shown in parentheses):

1. Identify the key organizations involved in the project (stakeholder analysis).
2. Determine the structural form of each key organization (structure analysis).
3. Identify potential structure-related risks (structure analysis, stakeholder analysis and risk management).
4. Assess the identified risks (that is, prioritize them and decide which require treatment) (risk management).
5. Formulate and apply risk response strategies (as described, for example, in [4]) (risk management).
6. Monitor and review (that is, iterate the process) (risk management).

Within the governance framework, accountability for this process is most likely assigned to the program or project steering committee. The role of the governance framework in this risk management process is to:

- assign a representative ‘owner’ of the structure-related risk process (this may or may not be the project manager, depending on the expected scope of the role and the potential to distract or overload the project manager);
- ensure that the above process is in place, appropriately integrated with the project’s risk management process, resourced and followed;
- if a process owner is assigned other than the project manager, ensure that the project manager is involved in the process (so that the project does not become isolated from managing structure-related risks)
- facilitate resolution of stakeholder-related issues that cannot be managed within the project and/or directly by the representative process owner;
- monitor the risk register and ongoing risk management process, adapting the process as appropriate; and
- ensure that ‘lessons learned’ are captured by the PMO for the benefit of future projects.

Not that assigning a representative process owner does not release the governance body from responsibility and accountability for managing structure-related risk. Rather, it assigns the operational activities of risk management to a suitable person to act under its direction and control. (It is not essential that governance owns the process – some other body may be able to perform the role – however, in most projects, governance is ideally positioned for the role.)

This role of project governance is illustrated and initially validated by two case studies of software projects, following.

### IV. CASE STUDIES

The following two case studies illustrate and motivate the problem of structure-related risk in software projects and demonstrate the role that governance can play in managing this risk. They are not intended to causally validate the propositions but provide initial *prime face* support. The first case (drawn from the study reported in [4]) describes a project structured in the matrix form, as described in Section II, while the second case describes a more complex multi-vendor project structure.
A. Case 1: Matrix form-based project

In the first case, a web-based system was developed by a government agency to manage the process of tendering for government contracts. Planned as a six month project, the project was sponsored and managed by the Procurement business unit, responsible for tendering, and the IT department, whose primary role was to provide system infrastructure. The system was to be offered on a software-as-a-service basis to the procurement units of other government agencies. The system was developed by an external developer firm to specifications prepared by the agency. There was no project governance. The main structural relationships are shown in Figure 2.

![Figure 2. Case 1 structural relationships.](image)

Three matrix form risks (from Table III) and three project form risks (from Table II) were encountered by this project (the existence of risk factors from the project form structure in a matrix form organization is most likely due to the matrix form being a composite of the functional and project forms):

- **Unclear accountabilities** – the project experienced difficulties with functional silos, especially from IT, which provided inadequate input to the project because the required system was non-standard.

- **Poor decision-making** – the vendor was chosen in isolation of other decision-making authorities based on its prior experience with the application; however, it went into receivership before the project finished.

- **Control problems** – the vendor focused more on features it had delivered to prior clients than understanding the agency’s requirements; change control was a problem, and; people-related change management was ad hoc.

- **Strategic control** – Sponsors were ‘dreamers’; there was no strategic grounding (as evidenced by the absence of project governance).

- **Stakeholder participation** – the main business sponsor had no involvement after initiation; IT buy-in was also low.

- **Adequacy of personnel** – IT was unskilled in the targeted Unix-based platform.

The project was delivered late and software development had to be brought in-house because the contracted developer organization failed.

The project team was able to superficially recognize the issues as they arose but not their underlying causes. It is likely that an effective project governance body could have had the necessary oversight of the project and vision of the structural drivers, enabling them to ensure that actions were taken to avoid or minimize the effects of these issues (realized risks), especially relating to selection of the developer and the contribution of the IT department.

B. Case 2: Multi-vendor project

The second case provides a different, more complex example of a project involving multiple parties. It also provides a rudimentary illustration of how governance is ideally placed to resolve structure-related risk. This case also involved a government agency (at the national level) but, here, the agency played a comparatively passive role. Instead, the focus is on a major hardware vendor organization, which was appointed prime contractor to complete a large project on a ‘turn-key’ basis. The project’s aim was to replace the agency’s national IT infrastructure in 40 sites with a variety of new hardware, systems software, networks, and application software from multiple suppliers. Our interest is in the structural arrangement that was established to execute the project.

The agency wanted the security and simplicity of dealing with only one vendor so it chose a high profile hardware vendor to act as prime contractor, with the other suppliers subcontracting to it. The main structural relationships are shown in Figure 3. Each corporation conformed to either the functional or matrix structural form. However, in this case, the structure of each participant was less an issue than the structure of the project.

![Figure 3. Case 2 structural relationships.](image)

The arrangement proved to be less than satisfactory for all parties involved. Apart from needing to comply with an overall project schedule, each supplier was largely able to act independently. Furthermore, each had to coordinate with the agency so the agency adopted a practice of dealing directly with each supplier, bypassing the prime contractor. While this was not a major issue for the prime contractor (after all, it was still being paid the prime contractor fee), when the principal application developer encountered financial difficulties (a small owner-managed software house), and looked like it might fail, the agency enforced the contract terms and insisted that the major vendor take responsibility.
for the performance of the developer. The vendor had to take over management and funding of the final stages of software development, at a cost that far exceeded its prime contractor compensation. Rollout of the integrated procurement and asset management system under development was to be the main deliverable in the second half of the project.

A mid-term milestone at which maintenance agreements were due for renewal (for equipment already operational) provided the opportunity to review the arrangement. The prime contractor set up a working party to consider how to resolve the problem. This governance mechanism comprised a senior manager, the project director and legal counsel from the prime contractor, a senior manager from the government agency and the owner of the developer organization. The working party considered various options and financial models to find a way forward. The net result was that to retain the present arrangement, the agency would have to pay a huge risk premium to cover the vendor’s costs if the software house failed. However, neither party liked this option (the vendor did not want to take ownership of the application software and the agency could not pay the risk premium without retendering for the software).

Finally, it was agreed to restructure the project, through legal/contractual means, to release the vendor from the prime contracting role and for the agency to deal directly with each supplier, including the developer, thereby accepting all risk. To avoid the agency having to retender, this was done by retaining the head contract between the agency and vendor, and novating each supplier subcontract from the vendor to the agency. With the commissioning of additional work by the agency, the developer was able to trade through its financial crisis, successfully completing the project.

In sum, through the mediation of the governance working party, the vendor identified, assessed and mitigated the risk by renegotiating the contract to restructure the project, thereby avoiding the risk. In turn, the agency mitigated its risk of direct responsibility for the developer by giving the developer another cash stream to help it through its financial difficulties. The prime contractor’s project team did not have the visibility of the problem or executive ‘clout’ to have achieved this outcome.

This case graphically illustrates how structure-related risk (a complex contractual arrangement and the near failure of the principal software developer) can significantly threaten the performance and success of a software project, and how these threats could not be adequately handled from within the project itself. Rather, they required a project governance body with the visibility of the issues and scope of influence to remove the structural barriers to save the project. As such, this case lends direct support to the principles proposed in this paper for managing structure-related risk.

V. DISCUSSION

This paper has extended prior work on structure-related risk by proposing that: the organization of software projects, the parent and other involved corporate stakeholders, and interactions between these entities can be sources of structure-related risk; that management of structure-related risk requires a perspective and span of influence external to the project, and the project governance framework is ideally placed to provide this, and; finally, that traditional risk management processes require augmentation with specific techniques to facilitate identification, analysis and assessment of structure-related risk (structure analysis and stakeholder analysis are proposed). A management approach and process is proposed to manage this risk. Through extending and reinforcing the emergent role of project governance, management of this previously overlooked source of project risk is likely to contribute to improved performance of software projects and the timely delivery of quality software.

The study has some limitations. Work so far has focused only on common organization structures and traditional configurations of organizational interactions in projects. This does not reflect the full diversity of arrangements in practice. Further work is needed to identify the scope of structural forms and interaction configurations between participating entities to establish the boundaries of the problem. Also, empirical validation of the propositions is only preliminary, based on two cases from the studies from which they were abstracted. Further empirical investigation is required to validate the nature and scale of importance of structure-related risk in software projects and the role of project governance (as opposed to the possibility of some other mechanism), as the primary authority for managing such risks. Finally, the proposed risk management processes are generic and preliminary. They will likely need to be adapted to specific project contexts. Further work is needed on the development of management processes, guidelines to support the control of these risks in practice, and the capabilities of practitioners to conduct the key analyses.

The paper has implications for research and practice. For research, the paper further highlights that in addition to risks that are endogenous and exogenous to software projects, risks relating to the structural makeup of the project are also critically important. The paper also reinforces the importance of the macro process context of software engineering and the value that can be drawn from related disciplines such as organization theory in delivering quality software. These associated management processes are critical in ensuring that well-engineered software reaches its intended usage domains as required and expected. For software engineering to continue to improve its record of performance, research needs to proceed in parallel in both the micro and macro contexts.

For practice, the paper suggests that the project is not well-positioned to manage all project-related risks and that the role of project governance needs to the extended beyond the passive receipt of risk register updates and general risk management oversight to actively own and manage structure-related risk. The initial risk management process proposed in Section III was abstracted from the actions of managers struggling to work through partially understood problems encountered in projects for which no packaged wisdom was available. It is important that this effort continues in refining the proposed approach.

Some topics for future research have already been mentioned. One other important future direction for this
work could be to extend the principles of structure-related risk to other structural forms such as virtual organizations, common in SOA and cloud computing environments. Furthermore, the focus so far has been on development environments, but this could be usefully extended to include operational contexts in a virtual world. Virtual organization tends to encompass both structural forms of interaction as well as strategic considerations, which may further broaden the scope of this research.

VI. CONCLUSIONS

Project structure is a little like infrastructure. It is assumed to be there but, otherwise, it is usually ignored in everyday conscious activity. However, where we may look at infrastructure if something goes wrong (such as when a pothole is encountered on a highway), in projects, we have tended to overlook the possibility of structures as a source of problems. This paper has reported preliminary work in highlighting the potential role and importance of the structural context of projects in successfully delivering software, and has suggested some initial approaches to managing the associated risks. In partnership with the practitioner community, further work is needed to consolidate and extend these contributions.

ACKNOWLEDGMENTS

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