Risk Implications of Software Project Organization Structures

Paul L Bannerman

NICTA and School of Computer Science and Engineering, University of NSW, Sydney, Australia

paul.bannerman@nicta.com.au

Abstract

Researching organization structures is no longer in fashion. However, a recent study uncovered gaps in our knowledge about the structure of software projects and their associated risks. This paper develops risk profiles from the literature of four common structures (functional, project, matrix and adhocracy forms) and validates them against data from a public sector study. Understanding structure as a source of project risk could unlock deeply embedded performance problems in software projects that have been overlooked or attributed to other causes.

Keywords: Risk management, organization structure, software project, software process.

1. Introduction

Risk management is an important process in software development that can improve product and project outcomes [7, 9]. However, progress in risk management research and practice has been slow [5], leaving opportunities for improvement.

Recent research in software projects in the public sector encountered unexpected findings relating to the organization of projects which may have implications for risk management [3]. It was found that, in contrast to the uniform organization of projects assumed by project and risk management standards and bodies of knowledge, agencies in the study used a range of structural arrangements for their projects. Four types of arrangement were identified: pure project form; hybrid form; operational activity, and; breakthrough event. Each appeared to have type-specific project and risk management characteristics.

Some research has been done on projects of different types [e.g., 16, 34], including identifying that project structure impacts project performance and project risk [17, 20, 21]. However, the project and risk management implications of project structures have received little attention in the literature.

This paper explores these types further, with the aim of developing an understanding of the risk implications of software project organization structures. To understand is the first step toward improving the ability to manage and succeed.

It is recognized that projects differ on a range of dimensions that require different management [21, 33, 34]. For example, Shenhar & Dvir [34] identify 13 project types on the dimensions of project complexity and pace, and product novelty and scope. They draw on classical contingency theory to argue that project success depends on adapting project management practices to match the specific project type.

Researchers have also distinguished project types based on the dimension of organization structure. Structure is about how organizational work is designed [13]. Early views in the first half of last century espoused one best form – bureaucracy. This changed in the second half of the century to the view that there is no one best way to design organizations. Rather the design must fit the organization’s strategy and environment, and be internally consistent (known as structural contingency theory) [13].

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 and/or geographic areas, according to the purpose of the organization. Common structures today include the functional form (including the divisional form), the project-based organization, the matrix organization and free-form structures such as the adhocracy [15, 16, 22]. Each structure has strengths and weaknesses which imply certain structure-related risks. Despite the long history of organization design, these risks have not been investigated in the literature to-date.

How is this relevant to software projects? First, as already mentioned, research indicates that project structure does impact project performance and project risk [17, 20, 21]. It is important, therefore, that we understand the risk implications of project structure to improve software project management. Second, there is evidence that the role of structure in project and risk
management is recognized in extant standards-based bodies of knowledge, but this is not explicated or applied in project and risk processes. Rather, they assume a common form that can be managed by a generic set of practices, implying a gap in our knowledge. For example, PMI’s PMBOK Guide recognizes that organizational structure can have differential effects on projects (it distinguishes functional, matrix and projectized forms), but these effects are not developed or applied in the knowledge process areas it prescribes (ANSI/PMI 99-001-2004). Similarly, the Australian risk management standard has a process for “establishing the [risk] context” that includes consideration of “structure”, but this is seen in terms of “governance, roles and accountabilities” rather than organization structure (AS/NZS 4360:2004). Finally, project and risk management are critical macro processes of software engineering [4, 26]. Not all risks in software projects are caused by software engineering. However, all have the potential to impact project outcomes, so their handling reflects on the perceived quality of software processes. It is important, therefore, to understand and manage these effects.

Structure is relevant to software projects in two ways. First, there is the organization of the project itself and how its structure can affect its performance. Second, there is the structure of a project’s “parent” and/or any influencing organization. Many project arrangements are possible within a single organization and between multiple organizations. For example, a software developer may be structured as a project-based organization and, therefore, not have a parent, but have various structural arrangements with client organizations. Alternatively, a large functionally-structured organization, such as a bank or government department, may run one or more internal team-based projects (project structures) that draw on resources from its various functions and from outside. Here, we limit consideration to the project organization and the possible influence of a parent organization.

In many views, researching organization structure is no longer fashionable. The implicit assumption is that questions of structure have already been answered, so interest has moved on. This paper takes the contrary view and demonstrates that such thinking is premature. Indeed, we may have passed over or forgotten the fundamental role that work organization can have on its performance and outcome. Take, for example, two project risk factors that appear on most ‘top ten’ lists: executive support; and business participation. It makes little sense that these factors should be risks in an organization that is investing in software-based systems to improve business effectiveness. Yet they are among the most common issues software projects face. Why is this? Understanding the risk implications of project and related structures can provide substantial insight into these issues and potentially point to deeper causes. How a project is structured in relationship to its parent organization may create significant barriers that elevate these and other factors as major risks and/or actual issues in a software project [6].

The paper proceeds as follows. In the next section, four organization structures are considered from the literature. The four structures chosen (functional, matrix, project, and adhocracy forms) represent the closest fit to the four project types found in the public sector study (operational activity, hybrid form, pure project form, and breakthrough event, respectively) [3]. A profile is developed from the literature for each structural type by identifying structure-related characteristic strengths and weaknesses that may have risk implications for projects of that structure or projects operating within a context dominated by an organization of that structure (such as a parent). Section 3 then presents a reanalysis of the public sector data against the identified set of structure-related risks to validate that risks cluster to the identified project structure types. Support is provided if agency projects in the study that were identified as being of a particular structural type exhibit risk characteristics of that type as derived from the literature. Two cases are then used to illustrate and explicate the structure-related risks. Section 4 discusses the findings and implications before conclusions are drawn.

2. Risk implications of project structures

In a study of the project and risk management practices of 17 projects in Australian public sector agencies, an unexpected finding was that the projects did not subscribe to any uniform organization structure as implicitly assumed by bodies of knowledge such as PMI’s PMBOK Guide and the Australian risk management standard [3]. The archetypical view of a software project is as a temporary endeavor undertaken to accomplish a unique purpose within a set of constraints (usually time, cost and scope), under the direction and funding of a sponsor or customer, and organized as a project team under the control of a project manager, comprising people with skills and/or knowledge relevant to the project and following best practice project and risk management prescriptions. Rather, many agencies viewed projects more broadly and adopted a range of organizational arrangements (project types). These are summarized in Table 1.
The study speculated that these organizational arrangements may have type-specific risks associated with them that were not recognized in the study.

With the aim of developing a risk profile for each type, the literature is briefly examined, following, to identify structure-related characteristics that may have risk management implications for projects of that type or for projects operating within parent organizations of that type. The analysis follows the chronological order each structure is encountered in the literature.

### 2.1. Functional form

The functional form is the traditional and most common way of structuring organizations. People and work are organized according to their functional expertise such as engineering, production, marketing, sales, distribution and finance. Authority and control are strongly hierarchical, with reporting lines that follow the chain of command upwards to the top rather than horizontally across departments and functions. Work is highly structured and routinized with rigid rules, regulations, processes and strictly defined boundaries to optimize the use of functional resources and achieve economies of scale. As a result, functionally structured organizations can grow to very large sizes but also be somewhat dull and unexciting for workers. This form works best in stable and predictable environments so that it can labor away at each function, without having to think too much about what is being done or the need for change. Another name for the functional form found in the literature is the “machine bureaucracy” [22]. This is the structure commonly found, for example, in banks, insurance companies, government departments, and retail department stores.

For Mintzberg, the distinguishing characteristic of the machine bureaucracy is standardization to achieve operational efficiency in functional activities [23, 24]. Its mains strength is its ability to perform standardized work in a highly efficient manner under stable operating conditions. Its main weakness is that specialization can create conflict between subunits and functional goals can override organizational goals. From a customer perspective, a machine bureaucracy can work well for you if your case fits within its established rules and processes. If not, it can be very inflexible and frustrating to deal with.

Clearly, this structure does not sound like one that is well-suited to venturing into change and uncertainty through technology projects. Indeed, the inflexibility of the functional form prompted the move toward project-based organization and management [10, 25]. The functional form is not structured for innovation. Nevertheless, at least two types of software project are commonly found in functional form organizations.

The first type is the conventional project team, created as a temporary organization for a specific purpose, that is resourced with staff from various

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Pure Project Form</th>
<th>Hybrid Form</th>
<th>Operational Activity</th>
<th>Breakthrough Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>Characteristics</td>
<td>Traditional project. A discrete, temporary activity, structured, operated and managed under best-practice project management disciplines (to varying degrees of formality and completeness)</td>
<td>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 existing functional units that operate independently.</td>
<td>A routine viewed as a project, executed by functional units within their normal operational space and structure, using some elements of project management control (like schedule, budget, scope, reporting)</td>
<td>A focused effort by a small specialist team to achieve a specific, high priority objective in a short timeframe without the constraints of conventional disciplines, practices or methods</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Structural form</th>
<th>Project</th>
<th>Matrix</th>
<th>Functional</th>
<th>Adhocracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dominant force</td>
<td>Completion</td>
<td>Optimization</td>
<td>Efficiency</td>
<td>Innovation</td>
</tr>
<tr>
<td>Locus of control</td>
<td>Project manager; governance framework</td>
<td>Project and functional management</td>
<td>Functional and line management</td>
<td>Executive</td>
</tr>
<tr>
<td>Project mgmt</td>
<td>Formal</td>
<td>Semi-formal</td>
<td>Informal</td>
<td>None</td>
</tr>
<tr>
<td>Risk mgmt</td>
<td>Formal</td>
<td>Variable</td>
<td>Variable</td>
<td>Intuitive</td>
</tr>
</tbody>
</table>

Adapted from [3]

Table 1: Public sector project types
business and IT functional units (as well as from outside) and with a primary association with one main functional department (often IT for software projects). The project usually seeks to apply best practice project management processes.

A key challenge of this type of project is alignment. Due to the structural differences and separation from the command and control authority lines, and potential for functional conflict, artificial integration mechanisms are needed to connect the project to the parent’s control centre at the top of the organization. The most common mechanism is the project steering committee. Such structures are not necessary to link projects to the top of project-based organizations. Over the years we have tended to lose sight of why steering committees are a success/risk factor. Alignment of structural differentials is a driving cause [6].

The second type of software project commonly found in functionally structured organizations, especially government departments with a dominant business system, is what the public sector study called an “operational activity” (refer to Table 1). Here, the IT department is structured functionally in line with the business (with separate development teams for each functional specialization), with work executed in job units as separate ‘projects’. Often the output of multiple projects is combined into one upgrade release of the major business system. Line authority still dominates, but work is partitioned into project units so that elements of project control can be applied (like schedule, scope, reporting and, maybe, budget). In this type, steering committees are not as important because functional reporting lines are used for inter-departmental communication. The main problem with this project type is that functional line management tends to override project management.

Table 2 summarizes key characteristics of this structure from the literature that may have risk implications. For each characteristic, the literature view is summarized with supporting references. An implied risk is proposed for each characteristic. Five risks are identified:

First, the preoccupation with routinized functional specialization has the potential to make it difficult for organizations of this form to complete one-off activities within schedule, budget and to specification, resulting in projects tending to fall behind.

Second, the tendency for functional objectives to interfere in inter-departmental interactions (leading to tensions and conflicts between functional units), is likely to divert attention and distract and/or delay their engagement in project activities.

Table 2: Functional form

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Description</th>
<th>References</th>
<th>Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional specialization dominates</td>
<td>Resource specialization makes achieving the iron triangle very difficult; focus is on functional delivery within an integrated value chain; inefficient in one-off activities; efficient in routinized tasks; customer service orientation may be lost</td>
<td>[11], [15], [16], [27]</td>
<td>Projects fall behind</td>
</tr>
<tr>
<td>Functional conflicts</td>
<td>Conflicting objectives can cause tension between functions, distracting focus and attention from projects</td>
<td>[27]</td>
<td>Engagement distraction</td>
</tr>
<tr>
<td>Hierarchical control</td>
<td>Operations tightly controlled and governed through lines of authority; permits little operational flexibility; communication is hierarchical, rather than across functions; roles are rigidly and distinctly defined; can become compulsive</td>
<td>[11], [23], [27], [36]</td>
<td>Operational inflexibility</td>
</tr>
<tr>
<td>Economies of scale</td>
<td>Mass production and functional specialization orientation confers scale and efficiency advantages; tends towards large size; standards and rules of operation promote continuity and stability; favors stability over novelty; change interrupts smooth functioning; the quest for stability and functional efficiency can become mundane, demotivating staff; innovation capacity can be suppressed</td>
<td>[11], [16], [24]</td>
<td>Innovation suppressed</td>
</tr>
<tr>
<td>Large size</td>
<td>Economies of scale pressures leads to large sized organizations</td>
<td>[23]</td>
<td>Size-related inefficiencies</td>
</tr>
</tbody>
</table>
Third, the hierarchical nature of command and control in functional structures is likely to lead to operational inflexibility at lower levels in the face of unfamiliar, uncertain and non-routine project activities. Fourth, the standardization-based driver toward efficiency and economies of scale in the functional form is unlikely to be conducive to innovation and change, which characterizes most technology projects. Finally, size brings many inefficiencies which can spell the “death march” for a software project [38]. Some of these inefficiencies (other than those described above) may also impact projects of any size operating within large functionally structured contexts.

2.2. Project form

As indicated, the project form, or project-based organization, emerged as a reaction to the bureaucracy of the functional form and its inability to handle change and innovation responsively. In the project form, projects are the primary unit of organization rather than functions. In a project-based organization, multiple projects are usually clustered under central management (which provides governance directly, rather than through artificial mechanisms) and support units. This structure provides the freedom and flexibility to organize activities to optimally suit the nature of each project. In this form, project managers are kings and project management is paramount [29]. The project form is fundamentally the inverse of the functional form: it is strong where the functional form is weak; and weak where the functional form is strong. Organizations in the business of software engineering and providing software-related services to clients often adopt this form of structure (it is also common in engineering, construction, aerospace and defense industries).

Table 3 describes seven characteristics of the project form, derived from the literature, which may have risk implications. These are as follows:

- First, it is inefficient compared to the functional form, and may experience cost pressures as a result.
- Second, its independence can lead to strategic detachment and loss of direction in its activities.
- Third, its flexible and adaptable nature can, if left unchecked, lead to problems of operational control.
- Fourth, getting stakeholders to participate in the project can be difficult due to structural separation.
- Fifth, the insular nature of projects, focused on a specific goal and practices, can lead to blindness with respect to important considerations in its environment.
- Six, projects are dependent on a range of skills and, 

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Description</th>
<th>References</th>
<th>Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inefficient</td>
<td>Does not scale well; weak in coordinating processes, resources and capabilities across-project; not well-suited to routine mass production</td>
<td>[15], [16]</td>
<td>Cost pressures</td>
</tr>
<tr>
<td>Goals and governance</td>
<td>Can become detached from and misaligned with its strategic centre; can miss major changes</td>
<td>[30], [36]</td>
<td>Strategic control</td>
</tr>
<tr>
<td>Flexible and adaptive</td>
<td>Not constrained by functional controls; operations decoupled from governance; can respond to uncertainty, changes and new requirements; strong in non-routine tasks and one-off projects; can be innovative</td>
<td>[16], [36]</td>
<td>Operational control</td>
</tr>
<tr>
<td>Stakeholder buy-in</td>
<td>Strategic/operational separation and focus on functional priorities can limit buy-in and participation from stakeholder outside the project</td>
<td>[1], [18]</td>
<td>Stakeholder participation</td>
</tr>
<tr>
<td>Myopic</td>
<td>The focused, temporary nature of projects, separated from functional business operations, can lead to insularity from its context and purpose; can become methodology-bound; can lose its grounding</td>
<td>[2], [11], [15], [36]</td>
<td>Contextual blindness</td>
</tr>
<tr>
<td>Adequacy of skill set</td>
<td>People are acquired sparingly; available skills may not fully match needs; skills may have to be shared across projects; weak in developing and retaining skills; not structured for personnel or career development; technical leadership is dispersed across projects; can suffer from a lack of technical leadership and direction; experts are highly mobile</td>
<td>[8], [14], [16], [31], [35]</td>
<td>Adequacy of personnel</td>
</tr>
<tr>
<td>Weak in learning</td>
<td>Limited structures/incentive for within-project and cross-project learning</td>
<td>[16]</td>
<td>Capability development</td>
</tr>
</tbody>
</table>
due to their tight constraints, may not always be able to obtain the full set or the best skills needed for the job.

Finally, unlike the functional form, projects are not resourced or scoped for developing capabilities for future projects. They offer strong learning opportunities but weak potential for accumulating and passing on learning. The project management office (PMO) is an artificial mediating device that can help to overcome this limitation.

2.3. Matrix form

The matrix form lies on a continuum between the functional and project form extremes. Matrices are typically structured functionally vertically, and have projects overlaid horizontally across them (or some other specialization, in a non project-based business). Conceptually, they offer the best of both the functional and project forms. However, they potentially also offer the worst of both worlds. The matrix features dual authority and reporting lines (to the function and the project), which can paralyze and/or politicize decision-making. Indeed, the matrix organization structure has been described as the ultimate in unworkable structures - easily becoming bureaucratic and non-creative, and prone to degeneration into anarchy [28].

Software projects in matrix organizations are often for the development and/or implementation of enterprise-wide business systems. Multiple concurrent projects in such structures can create complex and high risk conditions for each project.

In Table 4, five potential risks of the matrix form are described:

First, the dual reporting nature of the matrix often results in unclear accountabilities.

Second, localized claims to authority can result in decisions and actions being taken in isolation of other relevant interests, resulting in poor decision-making.

Third, political infighting and conflict resulting from the dual authorities and multiple interests within the organization can distract attention from project objectives and result in slow response times.

Fourth, internal preoccupation with sectional interests and infighting can result in a degradation of operations and project-related control problems.

Finally, such environments can be highly stressful for staff, resulting in project staffing problems and turnover.

### Table 4: Matrix form

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Description</th>
<th>References</th>
<th>Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dual reporting</td>
<td>Has competing authority structures and loyalties; makes monitoring, controlling and evaluating managers difficult; can foster matrix layering down and across the organization</td>
<td>[12], [15], [19]</td>
<td>Unclear accountabilities</td>
</tr>
<tr>
<td>Authority bias</td>
<td>Prone to “groupitis” (decision-making in isolation of other authorities); loyalty bias to functional manager (who does performance appraisals and influences career development); control tends to dominate at lower levels rather than higher corporate levels; stronger authority line can dominate decisions</td>
<td>[12], [28], [36]</td>
<td>Poor decision-making</td>
</tr>
<tr>
<td>Power and conflict</td>
<td>Boundaries of authority and responsibility overlap, prompting people to maximize their own advantage; power, authority and accountability often rest with different people; conflict through competition for resources; need for consultation and shared decision making can delay, stifle and/or paralyze decisions; interminable meetings</td>
<td>[12], [19], [27], [28]</td>
<td>Slow response time</td>
</tr>
<tr>
<td>Tendency toward anarchy</td>
<td>Tendency for controls to break down and individual/sectional interests to take over; “navel gazing” (internal preoccupation instead of focusing on objectives); when business declines, the matrix can become the scapegoat and be discarded</td>
<td>[12]</td>
<td>Control problems</td>
</tr>
<tr>
<td>Personnel issues</td>
<td>Dual reporting can be stressful for staff, contributing to role ambiguity and conflict; competing objectives and priorities can lead to resource shortages and discontinuities</td>
<td>[19]</td>
<td>Staff stress and turnover</td>
</tr>
</tbody>
</table>
2.4. Adhocracy form

In comparison to the other forms, the adhocracy, as encountered in software projects, is a free-form organization, with minimal structure and overhead to enable it to achieve breakthrough change. As described by Mintzberg, “it is a tremendously fluid structure, in which power is constantly shifting and coordination and control are by mutual adjustment through the informal communication and interaction of competent experts” [23]. Its design is optimized for innovation. For software projects, it is an ideal form for achieving quick leaps of change. It can be used as a precursor to more formal projects and/or for standalone initiatives. Often, breakthrough projects comprise a small number of skilled people, limited resources and a very specific change target to achieve in a short period of time. No constraints are imposed on how the team is organized or how the target is to be achieved, so formal methodologies and project processes are often not used. Instead, the team proceeds with the resources provided on the basis of sheer expertise.

Not all such projects are small and informal, however. Lockheed Martin, for example, adopted the adhocracy during World War II for special projects. Called skunk works, specially formed development groups are given a high degree of autonomy, unhampered by bureaucracy, to work on advanced or secret aircraft projects.

Table 5 identifies four potential risks associated with adhocracies in the form of breakthrough projects:

First, achieving the change objective overrides all other considerations so there is a risk of stakeholders being alienated and resisting project endeavors.

Second, when breakthrough projects are tightly constrained by time, funds and/or resources, there is a high risk that an 80% solution is delivered, leaving unresolved issues (jagged edges) to resolve later.

Third, the success of breakthrough projects is often more dependent on the capabilities of team members than on the use of best practices, creating a risk of exposure to the vagaries of ‘heroes’.

Finally, because there is often little opportunity to involve or prepare stakeholders for the system outputs of breakthrough projects, acceptance of implemented solutions by stakeholders can be a major problem.

3. Empirical study

With the aim of seeking initial validation of the proposed risk profiles of software project organization structures (Tables 2 to 5), the interview data and case descriptions of the public sector study [3] were reanalyzed for evidence of the risk factors in projects of each type. The results are summarized in Table 6. In the table, a check mark (‘tick’) in a cell indicates that supporting data was found for that risk factor in an agency project. Only evidence relating to organization structure, as described in the profile tables, was considered in checking a cell. For example, a number of projects fell behind (the first factor in the functional form risks) but only the checked one showed evidence of this being linked to structural issues described in the literature and summarized in Table 2.

The results support the proposed structure-related risks. Evidence of each risk factor identified from the literature was found in the projects. Furthermore, for each project structure (project form, matrix form, functional form, and adhocracy form) structure-related risks were found that clustered in the region of the factors proposed for that structure.

Outliers were also found, including some clustered against another risk set. These require explanation.

Consider, first, agency projects with a project form (Agencies 1 to 4) and functional form (Agencies 10 to 13) in Table 6. On first consideration, one might expect that agency projects adopting a project form would

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Description</th>
<th>References</th>
<th>Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bias for change</td>
<td>Oriented toward innovation and change through creative autonomy; intentionally transformative; cuts across conventional boundaries; flexible and responsive; unconstrained by functional operations</td>
<td>[11], [37]</td>
<td>Stakeholder resistance</td>
</tr>
<tr>
<td>Oriented toward expedience</td>
<td>Operates informally; focused on quick results; can be inefficient; can produce ill-fitting solutions with unresolved issues</td>
<td>[6], [11], [24]</td>
<td>Jagged edges</td>
</tr>
<tr>
<td>Resource dependent</td>
<td>Success is highly dependent on the expertise and effort of team members ('heroes'); can be waylaid by ego(s); can become anarchic</td>
<td>[11], [24], [28], [32]</td>
<td>Exposure to heroes</td>
</tr>
<tr>
<td>Change management</td>
<td>Institutionalizing outputs can be problematic due to limited stakeholder involvement in the project</td>
<td>[5]</td>
<td>Low solution acceptance</td>
</tr>
</tbody>
</table>
experience risks only from the project form risk factors set. However, two projects (from Agencies 1 and 2) also encountered a functional form risk ("Projects fall behind"). The most likely reason is that these projects were, in fact, operating within the context of parent agencies that were functionally structured bureaucracies, so it is reasonable (according to the proposed theory) that the structure of the parent may also have impacted the project.

Functional form projects have risks clustered in the functional form risk set, as predicted, but also three risk occurrences from the project form risk set. The most likely explanation is that functional form projects are still projects and may experience some risks in common with projects of a purer project form.

Similarly, the matrix form projects (Agencies 5 to 9) clustered strongly against matrix form risk factors, as expected, but also encountered three project form risks and one functional form risk. This is most likely because matrix form projects have to balance issues of both project and functional environments in their parent organizational context.

3.1. Case examples

Two cases illustrate structure-related risk.

Agency 3 (Project 1) developed and implemented a web-based online search engine system for public access to department records. The six month project was set up formally as a project team operating within the agency under the CEO as sponsor and the CIO as project manager. A vendor was contracted to do the system development, whose project manager was also on the agency’s project team. Due mainly to the involvement of an external organization, operational control of the project became an issue for the agency, leading to the adoption of the vendor’s project management methodology because it was designed for multiparty projects. Business unit stakeholders (who were responsible for providing content for the system) were not members of the team, but were approached only when their input was required. As a result (and due to pressing work within their business units) stakeholder participation was low. Furthermore, since the team lacked users, the vendor was concerned about the system being tested properly (adequacy of personnel). Also, the team was so focused on the system solution, that the implemented system was too short on content to satisfy public needs and justify using the system (contextual blindness). Similar issues also arose in the project that followed (capability development).
In Agency 14, a manager was tasked with setting up a new Customer Relationship Management-based call centre in three weeks. Together with a software vendor representative and three consultants, the team set about building the entire call centre infrastructure from nothing (exposure to heroes). A CRM package was purchased and configured by the team according to their best guess of what might be required. No users were consulted. On opening day, user resistance to the system was strong because they had not been trained and the setup did not fully match their needs (stakeholder resistance and low solution acceptance). It took a year to fix the loose ends (jagged edges).

4. Discussion

Improving the quality of software and the success of software projects remain major goals and challenges for researchers and practitioners. With the passage of time, the way software projects are structured has slipped from visibility as a contingency variable in project performance and potential source of project risks. Indeed, it is argued that structure-based considerations could underpin and explain some of the greatest issues faced by software projects today. Motivated by unexpected findings in a study of project and risk management practices in a sample of Australian public sector agency projects, this paper develops structure-based risk profiles for the four project types identified in the study and validates them against the original data from the study. Twenty one risk factors are identified across the four project types.

The research has some limitations. The literature review was not ‘systematic’ and the subsequent risk profile development was done by one researcher (the author). A more rigorous literature search and analysis by other researchers may identify additional factors. The aim of the paper, however, was to highlight structure as a source category of project risk rather than to identify all possible risk factors. Reanalysis of the earlier data set provided positive initial validation for the risk factors identified, but further testing is required to test the relevance of the identified risks in other projects in different settings. In particular, they need to be validated for projects in the private sector and in different cultural contexts.

The study has implications for research and practice. For research, the paper suggests that the classical theory of organizational structure and design still has relevance and value for the micro structure context of software projects and their alignment with organizations of other designs (parents and other influencing organizations). There is an opportunity and need to extend current project and risk management bodies of knowledge by integrating consideration of the contingency effects of structure on project design, risk and outcomes.

For practice, organizational managers involved in software projects, and project managers, would benefit from being aware of the operational issues and risk implications associated with different project structures and the structural context of any parent in which they operate. A critical implication of the findings is that structural characteristics can build risks into software projects from inception. The structural implications of project designs can be considered during the feasibility and planning stages as risk analysis and risk mitigation processes. Having chosen a particular design and initiated the project, the project manager can then manage the outstanding risks accepted for that design throughout the project.

For future research, as well as refining and further validating the risk factors, attention could be given to suitable mechanisms for integrating consideration of project structures into existing bodies of knowledge. At least three options may be relevant. First, acknowledge organization structure as an analysis category during risk identification, as a potential source of risk. Second, consider process changes to standard risk process cycles to explicitly include consideration of the risk implications of the structural context (such as during the ‘establishing the context’ process in the Australian standard). Finally, consider risk techniques that might support risk identification and mitigation of structure-related risk. Structure analysis and stakeholder analysis might be candidates, for example.

5. Conclusions

This paper has examined an area of research, organizational structure, which is often overlooked as a potential source of risk in software projects. By reexamining the literature, structure-specific risks have been identified for four structures found in Australian public sector projects. Work remains to extend and consolidate organization structure as a source of project risk. However, a stake has been firmly placed in the ground. Projects are not immune from the antecedent effects of their design. To understand these effects is the first step toward improving our ability to manage them and further improve project performance.

6. References


7. Acknowledgements

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and Digital Economy and the Australian Research Council.