Software Development Governance: A Meta-management Perspective

Paul L Bannerman

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

Abstract

Software development governance is a nascent field of research. Establishing how it is framed early, can significantly affect the progress of contributions. This position paper considers the nature and role of governance in organizations and in the software development domain in particular. In contrast to the dominant functional and structural perspectives, an integrated view of governance is proposed as managing the management of a particular domain (that is, a meta-management perspective). Principles are developed and applied to software development governance and illustrated by a case study.

1. Introduction

Research has found that governance of information technology (IT) domains can add business value [11]. Software development is a vital IT sub-activity that has aggressively pursued improvement opportunities over the years, in research and practice, to keep up with the pace of hardware technology developments and deliver software to meet demands for increasingly broader and sophisticated applications. Software developing businesses are well-positioned to benefit from the application of governance principles to the software development domain. However, little research has been done specifically on software development governance (SDG). Indeed, most governance research has focused higher up the 'organizational stack' than product engineering activities. While established governance principles are likely to apply in common to SDG, the distinctive characteristics and challenges of software development must also be considered.

This paper contributes a synthesized view of governance, abstracted from the central tenants of mainstream approaches, and applies it to the software development domain. In Section 2, prior research on governance is overviewed before issues in defining governance are discussed in Section 3. Section 4 then outlines the proposed view of governance as meta-management before these concepts are applied to SDG in Section 5. Section 6 illustrates one approach to SDG with a case study before conclusions are reached in the final section.

2. Prior research

Software development governance is a new field of research which extends governance principles down from the corporate and IT domains to the software development domain in software-developing organizations. It has received little attention in the literature outside of the inaugural SDG 2008 workshop at ICSE last year in Leipzig.

Research in organizational governance has its origins in political science and institutional economics [2]. According to [13], the spirit of governance is good order and workable arrangements. Its focus is on the means aspect of the means-end relation. Governance is about considering the efficacy of alternative modes (means) of organizing a business and selecting the best mechanisms to suit the circumstances at hand.

At the corporate level, governance is concerned with enacting arrangements to direct, administer and control a corporation to the satisfaction of major stakeholders and regulators. At the IT level, governance research has tended to focus on forms of governance (that is, on the structuring of IT activities and locus of IT decision-making within organizations) or contingencies that may influence governance (such as structure, strategy, industry and size). Structure-related research has focused on whether IT should be centralized, decentralized or located in a hybrid form such as a federal IT structure. Other researchers have viewed the form of governance more broadly as a composite of structures, processes and relational mechanisms (for example, [11]), a position that is adopted in this paper. A more detailed overview of the IT governance research literature is available in [2]. It is likely that similar considerations to the above also apply to software development governance, depending on the nature of the particular software-developing firm.
SDG 2008 considered a range of opportunities and challenges for software development governance, including governance as empowering and enabling software developers through managing value, developing flexibility, and controlling risk and change [5]; as a process framework [4] or iterative process [6]; as a tool-enabled responsibility assignment mechanism [14]; as a facilitation and control mechanism for distributed software development [8]; as non-deterministic criteria to guide adaptive and emergent software development processes within organizations [12] [9], and; as linkage mechanisms between business and agile software development [10]. The workshop identified a need for further research to understand and define the nature and role of software development governance in software-developing organizations.

Consensus has not yet emerged on a definition of IT governance. Software development governance is even less mature. Rather than attempt a definition, this paper aims to understand the underlying nature of governance and how it might apply to software development. The view is proposed of SDG as framing the management of software development to ensure good order and workable arrangements. That is, as a meta-management capability of the organization.

In the next section, we consider some generic distinctions that are critical to this understanding and the potential role of organizational governance.

3. Issues in defining governance

As described in the literature and experienced in practice, governance is a multi-dimensional concept, encompassing elements of organizational stewardship, accountability, risk management, compliance, control, propriety, functional oversight, resource allocation and capability. It tends to be defined from one of two perspectives: functionally, in terms of what governance does (e.g., assigning and administering decision rights, responsibilities and accountabilities) or; structurally, in terms of what it looks like (a framework of inter-related boards, councils, and committees). This paper argues that both perspectives are necessary for a balanced representation of governance. Furthermore, the two approaches are brought together in a meta-management perspective of governance, outlined in the next section. In preparation, this section considers eight issues that can influence how governance is viewed.

Governance versus management. It is easy to define governance in terms of management concepts and functions because management is the dominant lexicon at hand, but governance is not management (and vice versa). Both share common drivers of organizational stewardship, transparency and accountability but their nature and purpose are different. Defining governance in terms of management functions such as planning, organizing, leading, coordinating, controlling, staffing and motivating the activities of an organization can blur and confuse the distinctive contributions. Indeed, it has been argued that, at the corporate level, the amount of governance and management that is applied is an organizational choice [7]. Pound distinguishes between the managed corporation, in which senior managers are responsible for leadership and decision making while the board and shareholders have limited power, and the governed corporation, in which the board exercises appropriate power on behalf of shareholders to ensure that decisions are made effectively [7].

Notwithstanding this, governance can be viewed as managing the management function of the organization (as distinct from directly doing the work of managing the fulfillment of the business’s mission).

Governance as a noun and a verb. Prior research has tended to focus on governance as a verb. That is, on what governance does (the functional perspective). This view typically sees governance as one or more processes. Less commonly, governance is viewed as a noun, in terms of what it looks like (the structural or framework perspective). This view sees governance as an inter-related structural arrangement comprising one or more linked entities such as boards, committees and councils. This paper views governance as both noun and verb. A focus on one at the expense of the other provides a limited and potentially biased view of the phenomenon.

Governance as strategy. For some researchers, the central purpose of governance is strategic. It is a mechanism for ensuring that the goals and objectives of the board of directors are implemented. Two qualifiers are important here. First, while the work of governance can be strategic (especially in companies with active boards that have growth aspirations), management and organization theory assigns strategy as a function of management more so than of governance. The role of governance can be strategic in the sense of providing vision, direction and capabilities for the organization [11], but strategic management is more directly a management role. Second, focusing on the strategic potential of governance ignores its tactical side. Governance is also concerned with establishing and maintaining efficient and effective operations for the business as usual activities of the organization. It is not necessarily or exclusively about strategic direction.

Governance as process. Governance is also viewed predominantly as a process, for example, of assigning decision rights, monitoring the exercise of those rights,
and reviewing and adjusting these assignments and processes. Again, to balance viewing governance as both noun and verb, the position taken here is that governance includes processes but is not limited to processes. Governance is seen as comprising structures, processes and relational mechanisms, as in [11].

Governance as formal and explicit. Governance is assumed to be explicit and formal arrangements within organizations. Recent research, however, suggests that, particularly at lower levels of governance, implicit informal arrangements may be equally or even more effective. A study of project and risk management in software projects in the public sector found that an informal partnership between the project’s business owner and the project manager was seen by the latter as providing more effective project proprietorship and governance than the project steering committee [1]. This suggests that the application of governance may be more important than its form. This paper, however, focuses only on formal and explicit governance.

Governance as active or passive. As implied by Pound’s distinction between governed and managed corporations [7], governance may be actively interventionist in the affairs of the organization or maintain an arm’s length, passive position. Interventionist governance extends its activities into what would otherwise be the domain of management, thus supporting a blurred distinction between the two. Active, interventionist governance can be appropriate during times requiring strong directional guidance such as when initiating repositioning or major growth strategies. As a standard operating mode, however, strong centralist direction is at risk of being insensitive to customer/market requirements and changes. As implied by [7] (and some SDG 2008 papers), having the right balance between the two can be an ongoing challenge for organizations in volatile environments.

Governance as absolute versus situational. Several of the above issues (and SDG 2008 papers) imply that there is no ‘one size fits all’ solution to governance. Effective governance is contingent upon the characteristics of the organizational circumstance. Therefore, the aim of research should be to identify governance principles and the situations in which they are most likely to apply. These may be presented in a generic governance framework but not with the intent of representing one normative model of governance. Governance is likely to add greater value if it fits the organization’s business model, environmental context, stage of growth, and size (among other contingencies).

Software development governance versus software project governance. Interest in project governance pre-dates research in software development governance.

There is substantial overlap between software project governance and software development governance as much software development in software organizations is done via software projects. The question therefore arises as to whether there is a need for both forms and, if so, how they might be distinguished to enable their co-existence? This question is not answered here. It is likely that it will only be determined with the passage of time (and the processes of normal science). The position taken here is that there is a distinctive role for both, which is illustrated in the case study in Section 6.

4. Governance as meta-management

Conceptually, governance is the arrangements and practices that an organization puts in place to ensure its activities are adequately and appropriately managed. ‘Adequately and appropriately’ reflect the nature and context of the organization and stakeholder interests in its operations. Governance is not the direct operation and control of the organization’s business activities but the infrastructure needed to ensure their management to the satisfaction of direct and indirect stakeholders. As such, governance is viewed as meta-management. That is, as managing the management of the organization.

A governance cell is conceptualized as comprising the following elements, as illustrated in Figure 1:

Purpose. The anticipated outcome that is intended and motivates the actions of the cell.

Structures. One or more organizational entities through which the work of the cell is enacted, such as a board, steering committee, council or working party.

Processes. A collection of interrelated tasks that are enacted to accomplish the purpose of the cell.

Relational mechanisms. Arrangements that support and enable the purpose of the cell by facilitating interaction and communication with stakeholders and interested parties within and outside of the domain.

![Figure 1. A conceptual governance cell](image-url)
A governance cell applies to the activities of a particular stewardship domain (such as the whole corporation, IT, data management, architecture, projects or software development) and is associated with a set of stakeholders (such as governance proponents, governance actors, domain members and influential third parties). Each cell usually interlinks with one or more other related governance cell, vertically and/or horizontally to form an integrated governance infrastructure for the organization. This enables a pattern or framework of governance to be established that suits the needs of the organization (a simple example is illustrated in Figure 2).

The remainder of the paper focuses on the software development governance cell within a conceptualized integrated governance framework.

5. Software development governance

Extending these concepts, this section considers how governance might apply to the software development domain.

Software-developing organizations vary greatly in size and product(s) produced from small single product startups to large diversified product/service providers. Focusing only on the production of software, Figure 3a provides a classification of the main categories of software that might be produced. While it has been argued that governance is most effective when tailored to the needs of each organization, general principles about the components of governance can be stated that apply to SDG in most (if not all) software developing organizations. These follow.

The major purpose of SDG is to establish how the organization’s software development capability is sustained (in terms of structures, processes and relational mechanisms) to meet its engineering and business needs. Its role is not to directly manage the capability development. That is the role of software development management. Small firms are likely to govern via informal structures, processes and relational mechanisms. Larger organizations will typically adopt more explicit, formal governance arrangements.

For organizations with formal SDG structures, the peak body is best located in the organizational hierarchy at the highest position below which systems and software development-related capabilities may be shared or reused. Most organizations will require only one peak SDG body (organizations with multiple fully autonomous divisions may be an exception). The peak body is responsible for coordinating the governance activities of any subordinate capability-specific entities or cells. This structure will interlink with any other IT-related governance structures and with the operational and strategic governance of the business. For example, SDG structures might comprise a Systems & Software Capability Board that oversees the activities of several Councils (e.g., for architecture, platforms, tools, processes, change control, and skills), and interface with application domain-specific advisory groups, and a project management office.

Governance processes are likely to include the assignment and maintenance of software development capability decision rights, governance responsibilities and accountabilities; software development capability planning, monitoring and review processes; issue escalation procedures, and; stakeholder consultation and communication processes. Note that these SDG processes are distinct from the systems and software engineering processes that a process council, for example, might govern, and software process managers might manage.
Finally, SDG relational mechanisms might include client project steering committees, user groups, design working parties, and technical working groups, as well as web-based information support systems, knowledge management systems, customer relationship management systems, product marketing initiatives, and other coordination and liaison mechanisms.

Figure 3b suggests how some governance dimensions might apply to develop the four software types shown in Figure 3a. Governance of Type 1 software (development tools and operating systems) is likely to be highly centralized to ensure coordination and integration of products with target platforms, component specifications and standards, many if not all of which are likely to be determined externally by other software developers. Compliance with software standards and specifications is likely to be important to maximize the utility of the products produced for use by developers (who, in turn, may use them to create other software-based products). Strong, standardized methods and process-oriented controls for testing, configuration management and distribution are likely to be important to ensure high levels of product quality, consistency and reliability across successive releases of the product.

By contrast, governance of Type 4 software (customized applications) is likely to require decentralized arrangements to facilitate close contact and interaction with customers, even if detailed requirements are specified in a contract. Methods and processes are likely to be adapted to suit the needs of the customer and contract. Maintenance of a positive relationship with the customer is likely to influence outcome success as much as delivering the contracted software on time, within budget and to specification.

Software Types 2 (customized components) and 3 (application suites and shrink-wrapped software) share characteristics of Types 1 and 4 software. Therefore, these are likely to benefit from hybrid forms (that is, a mix) of the governance arrangements of the other two ‘extremes’.

6. Case illustration
NCW Systems Inc. (a fictitious name for a real company) is a subsidiary of a major aerospace/defense contractor [3]. The expertise of the defense division is as lead systems integrator in projects that build and/or support large software-enabled defense systems under government contracts. NCW runs a small number of high-budget, long duration projects. NCW views its expertise as an organizational capability and governs and manages it accordingly. It has comprehensive governance structures that extend down from the corporate level to the divisional level.

At the corporate level, capability governance is established for Engineering, Quality Innovation and Improvement, Supplier Management, and Human Resources. These are replicated in the defense division with the addition of Project Management and Integrated Logistics Support capability domains. Each governance domain is interlinked vertically and horizontally with the other domains at each level.

The Engineering capability governance structure (our domain of interest) is divided into four capability cells: software engineering, systems engineering, integration and test, and configuration management and data management. A Capability Manager is assigned to each area, responsible for ensuring that everything is in place for the organization to apply the capability when and where it is needed, including processes, people, tools and environments. Support with processes is provided by subordinate Process Councils within each capability area. Process councils ensure that adequate processes are maintained, project adherence to processes is assessed, and needs for process improvement are identified.

At the software engineering (SE) level, the defense division’s SE Capability Manager has established a tailored reference framework and capability planning and development program. The framework details capabilities under four areas: process; team behavior; technology; and enabling environment. The aims of the framework are to align SE capabilities to the division’s strategy; sustain existing SE capability; create an agile SE capability that can adapt to new business needs; drive change and innovation to improve business outcomes, and; develop the division’s SE capability as a competitive differentiator.
The framework is intended to be used by functional managers to assess the maturity of their current capabilities. This is then analyzed against the results of a sophisticated strategic planning process to identify gaps in capabilities that may be required, as well as existing capabilities that may be below required levels of maturity. Based on this analysis, capability and functional managers prepare development plans to fill the identified capability gaps according to the timeframes outlined in the strategic plans.

These governance structures and responsibilities are independent of the functional management structure of the division and of its projects.

7. Discussion and conclusions

This paper has considered the nature and role of governance with the aim of framing how governance might apply to the software development domain of software developing organizations. A challenge for this approach is the diversity of software developing enterprises, making generalization of models and practices difficult. Consequently, the analysis has focused on developing an understanding of governance issues and general principles that might apply to SDG.

Viewing governance as meta-management avoids the limitations of functional and structural approaches and forces a clear distinction between governance and management. However, many software practices such as requirements engineering, product and process quality assurance, configuration management and process improvement have evolved with elements of both governance and management embedded within them. It may be difficult to unwind this trend and segment explicit, uniform governance for all software development practices. This is an opportunity and challenge for future SDG research and practice.

Governance in other domains has been shown to add significant value to the business. The considerations in this paper suggest that value can also be derived from SDG in software developing organizations. However, how SDG is framed and applied in practice is likely to determine how effectively managing the management of software development emerges as a distinct value-adding activity in these organizations.

8. References


9. Acknowledgements

The paper benefits from comments by Mark Staples. NICTA is funded by the Australian Government via the Department of Broadband, Communications and Digital Economy and the Australian Research Council.