

Extending Software Development Governance to meet IT Governance

Carlos Juiz

Department of Mathematics and
Computer Science
University of the Balearic Islands,
Palma, Spain
cjuiz@uib.es

Ricardo Colomo-Palacios
Department of Computer Sciences
Østfold University College
Halden, Norway
ricardo.colomo-
palacios@hiof.no

ABSTRACT

Given the importance of IT for organizations worldwide, IT Governance is an increasing concern for C-suite officers. Inside IT, software is a key aspect in the governance scenario. The increasing pressures by regulatory, and compliance efforts are changing the software governance arena, thus there is a need to focus on the current state of the topic. Despite this acknowledged need, the studies on Software Governance are still scarce. In this paper, authors expand the software governance model introduced by Chulani et al. [1] with new concerns derived from the alignment of the model with the IT Governance standard, ISO/ IEC 38500 and the experience of authors. Moreover, the new model proposes the categorization of these concerns to govern software development activities aligned with IT Governance.

CCS CONCEPTS

Applied computing → Enterprise computing → IT governance

KEYWORDS

IT Governance, Software Governance, Mapping, ISO 38500.

1 Introduction

Information Technology (IT) is one of the main assets and investments for organizations. Given the strategic role of IT in today's world, all organizations must develop their strategic IT capabilities including IT Governance (ITG) [16]. Deprived ITG can lead to value destruction through lost opportunities or innovation lag and increased exposure to IT risks [11]. ITG is key to ensure effective IT decision-making is aligned with organization's strategy, but also to control the alignment of IT as a whole to the needs of the business [3]. ITG is connected to the allocation of decision rights and accountability to align IT decisions to strategic objectives [14]. Maybe the most important ITG initiatives are the Control Objectives for Information and Related Technologies (COBIT) by the American Information System Audit and Control Association (ISACA), based mainly in [12, 13] and the Massachusetts Institute of Technology (MIT) approach for ITG [15]. Despite these initiatives, there is also a specific standard for ITG, launched back in 2008, and the current version is ISO

38500:2015 [2]. This initiative presents a framework including three different tasks for ITG, namely:

- 1) Assess the use of IT;
- 2) Preparation and implementation of plans and policies
- 3) Monitor conformance to policies and performance against the plans.

According to Paré et al. [9], the core activities of the IT function are the acquisition, parametrization, development and the deployment of software as well as the management of projects. This leads to the asseveration that Software Governance is part of ITG, as was also underlined by [1, 5]. Most of the previous works devoted to software governance is dedicated to software development governance (these ones mostly published in the ICSE workshop devoted to the topic that ended by 2011) and, connected to that, aspects like global software development governance [10] and Open Source Software Governance [4]. The goal of Software development governance is to ensure that business processes results/outcomes of the software company meet the strategic requirements of the organization [1].

Given that the topic of the workshop is "Governance in Software Engineering" and software engineering is a discipline that goes beyond software development, it is necessary to check the popularity of the subject in the literature. Although the aim of this paper is not to perform a systematic literature review on the topic, it would be interesting to check the popularity of the subject in the literature. Authors performed a query in Google Scholar with the keywords: "Software Engineering Governance" OR "Governance in Software Engineering". The query returned 38 results (by mid December 2019), Google Scholar, due to duplicates just presented 31. Then, authors refined the list to find 5 more duplicates, leaving just 26. A first filter was applied to discard the papers that were inaccessible by means of the current subscriptions at authors' institutions or unavailable in open access. Consequently, six more papers were discarded. The second criteria for inclusion was the presence of a discussion of governance aspects regarding software engineering and not just citations or mentions. To these matters, 17 papers were discarded. Just 3 papers remained as primary studies. All the 3 papers are authored by Nguyen et al. [6–8] and they deal with the alignment of requirements engineering to ITG by means

of introducing governance in requirements-driven software development process. The mentions of IT governance in software engineering point out to the definition of software development governance by Chulani et al. [1]. To conclude, this search leads us to the conclusion that this paper is not only the seminal one, but also one of the most influential in the topic, receiving 35 citations by the end of 2019 (according to Google Scholar).

Table 1: Software development governance concerns (adopted from [1])

Phase Goal	Inception & Elaboration	Construction	Transition
Manage Value	Allocate development organization's investments to maximize ROI	Manage projects based on alignment between goals and Software Engineering concerns	
Control Risk & Change	<p>Determine resource availability and staffing to mitigate risks identified</p> <p>Perform risk measurement and analysis</p> <p>Perform risk prioritization by analyzing risk exposure and determine go/no-go</p> <p>Assess, evaluate & minimize development risks Plan for compliancy</p>	<p>Do prototyping & simulation to minimize risk exposure</p> <p>Measure and reassess risk with different stakeholders based on initial prototyping results</p> <p>Measure ongoing cost and effort</p>	<p>Do risk prioritization to minimize risk exposure & leverage risks</p> <p>Identify team communication patterns to meet development goals</p> <p>Minimize risk exposure from 3rd party packages & components</p> <p>Measure costs & resources</p> <p>Prioritize operational risks</p> <p>Track actual values of effort, quality and other metrics to enable control & future planning</p> <p>Meet compliance needs</p>

Develop Flexibility	Define an information architecture	Continuously manage business goals & requirements	Monitor and manage effort and other metrics to enable control and future planning
	Create framework for technology planning	Design & develop resource	Manage applications & information to maximize usage and flexibility
	Define organization and processes	Validate & measure quality	
	Define development investment framework	Measure development ongoing costs	
	Manage human resources	Measure / estimate value	
	Develop quality management system		
	Develop project management framework		

Maybe the most important contribution of the paper is the identification of Software Development Governance (SDG) concerns. These concerns are represented in a matrix in which a set of actions are classified according to two dimensions: Software development phases (inception and elaboration, construction and transition) and SDG aims (Manage value, Develop flexibly and Control risk and change). Concerns are extracted by authors from initiatives such as COBIT, ITIL or the Sarbanes-Oxley (SOX) Act of 2002. The set of concerns presented by these authors is presented in Table 1.

Being to date the most important contribution in the area, in this paper authors aim to study the set of concerns identified by Chulani et al. [1]. This work is performed in two steps. Firstly, and taking into account the traditional confusion between management and governance, as stated by [3], authors classify each concern into Governance, Management and mixed practices. Secondly, authors propose a new set of concerns, in addition to the ones identified by Chulani et al. [1]. In the next section, authors will present the conducted study. Finally, in Section III authors present main conclusions and propose future work.

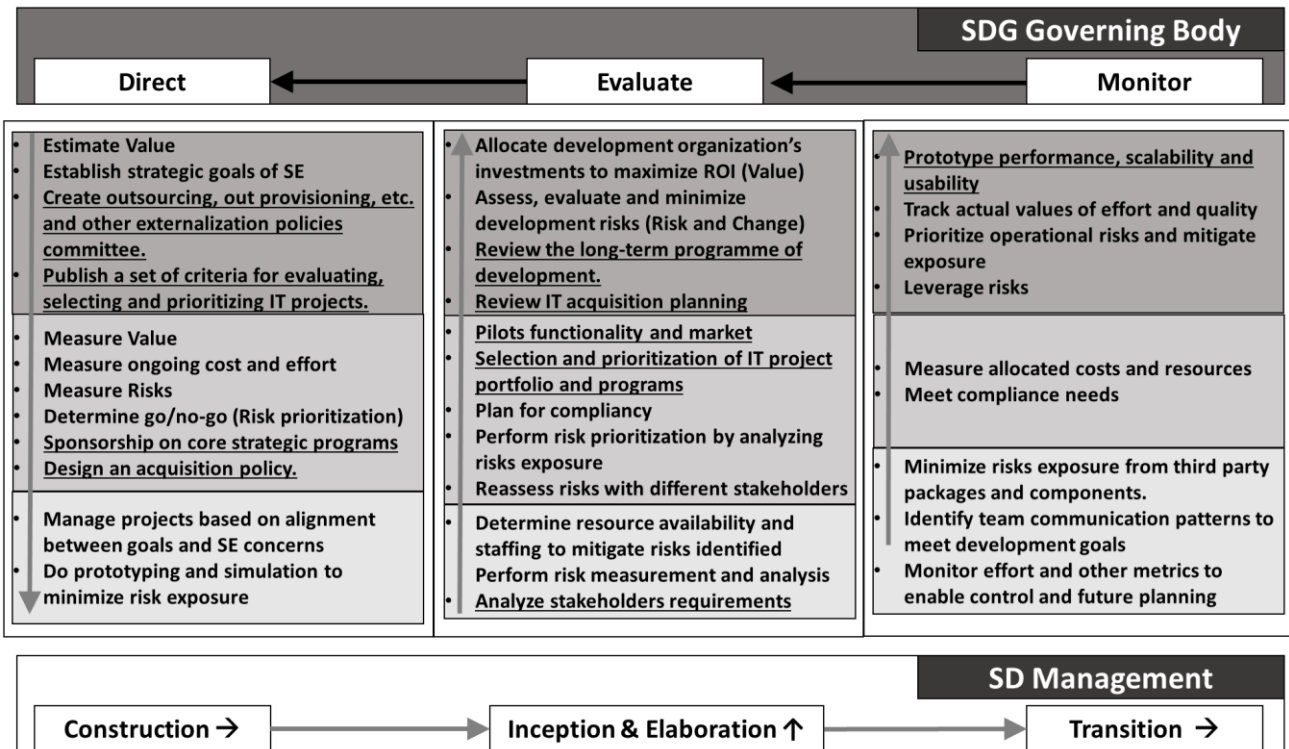


Figure 1: Enhanced Software Development Governance Model

2 Software development governance concerns

In this paper, authors aim to study software development governance concerns and expand them with new concerns. Firstly, authors tried to fit Table 1 into the meta-model of ISO/ IEC 38500 adapted to SDG, in which those activities that are typical of governance, that is, evaluate, direct and monitor (EDM), must be distinguished from management. Authors used three columns to classify the concerns as also adopted in [1], namely Inception & Elaboration, Construction and Transition. These three aspects are connected to governance activities and mapped as follows; Construction as Direct, Inception and Elaboration as Evaluation and Transition as Monitor based on their correspondence. In each of the three columns there are three different levels corresponding to the closeness to governance. The darker the gray, the closer to governance and vice versa, the lighter the gray, the closer to management. Thus, on one hand, the upper level is devoted to including concerns that are almost purely governance under the EDM model. On the other hand, the lower level consists of almost management concerns but not directly included into develop flexibly. Whereas, the intermediate one includes concerns that incorporate management and governance aspects. Taking into account Table 1, all aspects identified in [1] in the last row (Develop Flexibility) are considered purely software management and not governance. The rest of the aspects are placed under the

other two levels assessing their identification with governance practices. These concerns are closer to governance than to management. This is because, on the one hand, producing business value with IT is the main leitmotif of ITG and, on the other hand, because governing is basically directing and controlling. Thus, the contents (concerns) of these two rows and columns have been rearranged according to EDM activities, SD management activities and the proximity between them. That is, the closer they are to ITG, the higher they appear in the communication arrows of the meta-model that connects SD governance with SD management. On the contrary, the closer they are to the SD management, the closer they are to the phases of flexible development. We have reordered these concerns, according to our experience in ITG, since not all the concerns that are placed in Table 1 have the same proximity to the ITG. In fact, the closer these concerns are to ITG, the more possibilities that they will be EDM's activities and therefore belonging to the governance body, top management committees (CxO) or even ITG steering committees. However, as these concerns get closer to Plan-Do-Check-Act (PDCA) activities, software management and even project management actions, the lower position in the meta-model.

Apart from this classification, authors have included (underlined) other concerns that are not in Table 1, especially with regard to the portfolio of software projects, the construction of prototypes and pilots, outsourcing policies, long-term development policies, software acquisition policies and important issues such as

performance, scalability and usability. A total of nine new concerns have been added by authors to the original list.

Finally, the three columns include arrows indication of the direction of the action. For instance, Construction is top down given the fact that in Direct activities decisions are top-down, while in Evaluate (Inception & Elaboration) and in Monitor (Transition) information and decisions are coming bottom-up.

We believe that the model presented in figure 1 clarifies and classifies the governance and management concerns for the different phases Chulani et al. [1] pointed out and matches better with the ISO/IEC 38500 meta model for ITG. In fact, figure 1 shows two interesting direct/control cycles for software governance/management intercommunication throughout the development of any software artifact:

Inception/Construction cycle: The software artifacts are incepted and elaborated from management (central column) and being evaluated for governance. Once all concerns are considered, then the construction is directed from governance (left column) and all these concerns should be treated from governance to management. This virtuous cycle runs until the software artifact is considered in the next phase.

Transition/Construction cycle: Once the software artifact is performed, governance is monitoring (right column) the concerns communicated from management. This monitoring should be evaluated (considering the other management concerns) from governance in order to direct a new construction (if necessary).

3 Conclusions and outlook

In this paper, authors present an enhanced model for software development governance aligned to the IT Governance standard, ISO/IEC 38500. The model classifies current and new suggested concerns in the three phases Inception & Elaboration, Construction and Transition, which are mapped to governance activities, respectively, evaluate, direct and monitor. Acknowledging the traditional confusion between governance and management, these issues are also classified into three levels, namely, governance concerns, management concerns and mixed ones. The new model is of importance for CxOs and especially for CTOs and CIOs in their endeavors.

As future work, authors propose several aspects. Firstly, it is aimed to enrich the model with mappings to other IT Governance and IT Management initiatives. Secondly, it is aimed to tailor a deployment method for the model in order to ensure its applicability in real-world scenarios. To do so, a qualitative study with practitioners is planned.

REFERENCES

- [1] Chulani, S., Williams, C. and Yaeli, A. 2008. Software Development Governance and Its Concerns. *Proceedings of the 1st International Workshop on Software Development Governance* (New York, NY, USA, 2008), 3–6.
- [2] ISO/IEC JTC 1/SC 40 2015. *Information technology — Governance of IT for the organization*. Technical Report #ISO/IEC 38500:2015.
- [3] Juiz, C. and Toomey, M. 2015. To Govern IT, or Not to Govern IT? *Commun. ACM*, 58, 2 (Jan. 2015), 58–64. DOI:<https://doi.org/10.1145/2656385>.
- [4] Kendall, K.E., Kendall, J.E., Germonprez, M. and Mathiassen, L. The Third Design Space: A postcolonial perspective on corporate engagement with open source software communities. *Information Systems Journal*, n/a, n/a. DOI:<https://doi.org/10.1111/isj.12270>.
- [5] Manteli, C., van den Hooff, B. and van Vliet, H. 2014. The effect of governance on global software development: An empirical research in transactive memory systems. *Information and Software Technology*, 56, 10 (Oct. 2014), 1309–1321. DOI:<https://doi.org/10.1016/j.infsof.2014.04.012>.
- [6] Nguyen Huynh Anh, V., Kolp, M., Wautelet, Y. and Heng, S. 2017. Aligning Requirements-driven Software Processes with IT Governance. *Proceedings of the 12th International Conference on Software Technologies, ICSOFT 2017* (2017).
- [7] Nguyen, V.H., Kolp, M. and Wautelet, Y. 2018. Knowledge Management Governance in Software Development Process with GI-Tropos. *The 30th International Conference on Software Engineering and Knowledge Engineering* (2018), 468–467.
- [8] Nguyen, V.H.A., Kolp, M., Wautelet, Y. and Heng, S. 2019. Mapping IT Governance to Software Development Process: From COBIT 5 to GI-Tropos. (Dec. 2019), 665–672.
- [9] Paré, G., Guillemette, M.G. and Raymond, L. 2019. IT centrality, IT management model, and contribution of the IT function to organizational performance: A study in Canadian hospitals. *Information & Management*. (Aug. 2019), 103198. DOI:<https://doi.org/10.1016/j.im.2019.103198>.
- [10] Sievi-Korte, O., Richardson, I. and Beecham, S. 2019. Software architecture design in global software development: An empirical study. *Journal of Systems and Software*, 158, (Dec. 2019), 110400. DOI:<https://doi.org/10.1016/j.jss.2019.110400>.
- [11] Turel, O., Liu, P. and Bart, C. 2019. Board-Level IT Governance. *IT Professional*, 21, 2 (Mar. 2019), 58–65. DOI:<https://doi.org/10.1109/MITP.2019.2892937>.
- [12] Van Grembergen, W., De Haes, S. and Amelincckx, S. 2003. Using COBIT and the balanced scorecard as instruments for service level management. *Information systems control journal*, 4, (2003), 1–7.
- [13] Van Grembergen, W., Saull, R. and De Haes, S. 2003. Linking the IT Balanced Scorecard to the Business Objectives at a Major Canadian Financial group. *Journal of Information Technology Case and Application Research*, 5, 1 (Jan. 2003), 23–50. DOI:<https://doi.org/10.1080/15228053.2003.10856015>.
- [14] Weill, P. and Ross, J. 2005. A matrixed approach to designing IT governance. *MIT Sloan management review*, 46, 2 (2005), 26.
- [15] Weill, P. and Ross, J.W. 2004. *IT Governance: How Top Performers Manage IT Decision Rights for Superior Results*. Harvard Business Press.
- [16] Wilkin, C.L. and Chenhall, R.H. 2010. A Review of IT Governance: A Taxonomy to Inform Accounting Information Systems. *Journal of Information Systems*, 24, 2 (Sep. 2010), 107–146. DOI:<https://doi.org/10.2308/jis.2010.24.2.107>.