

## Chapter 1.23

# Governance of Software Development: The Transition to Agile Scenario

**Yael Dubinsky**

*IBM Haifa Research Lab, Israel  
Technion–Israel Institute of Technology, Israel*

**Avi Yaeli**

*IBM Haifa Research Lab, Israel*

**Yishai Feldman**

*IBM Haifa Research Lab, Israel*

**Emmanuel Zarpas**

*IBM Haifa Research Lab, Israel*

**Gil Nechushtai**

*IBM Haifa Research Lab, Israel*

### ABSTRACT

Governance is the exercise of control and direction over a subject such as a society, an organization, processes, or artifacts, by using laws and policies that are defined, deployed, and executed. In this chapter we develop this definition into a formal conceptual model that can be applied to a variety of governance domains. At the heart of this model lies the concept of the governance solution and its lifecycle. The governance solution

embodies the set of mechanisms—decision rights, policies, controls, and measurements—applied to a governance scope in order to achieve some governance goals. As part of the lifecycle, the effectiveness of the governance solution is measured, and corrections and alignments are made as necessary. We demonstrate how this model can be applied to multiple governance domains by providing examples from IT governance as well as software-development governance. We conclude by providing a detailed scenario in the

software-development governance space, which looks at large software organizations undergoing transition to agile development methodology. We further demonstrate how the governance model is instantiated and evolved in the context of this scenario.

## INTRODUCTION

The field of information technology (IT) governance has garnered an increased amount of attention in recent years. However, it is still struggling to provide a universally agreed-upon definition and a complete model for IT governance, along with the required tools and techniques.

The definitions of IT governance that can be found in the literature from Broadbent (1998), Chulani, Clay, Yaeli, Wegman, and Cantor (2006), Van Grembergen and De Haes (2004), Weill and Ross (2004), and Williams (2005) and they all share common ideas, such as the need to increase the value of IT to the organization while reducing risk. For example, *Weill and Ross (2004) focus on decision rights and define IT governance as “specifying the decision rights and accountability framework to encourage desirable behavior in the use of IT”* (p. 8). Van Grembergen and De Haes (2004) address the alignment of the IT organization with the business needs, and define IT governance as “the leadership and organizational structures, processes, and relational mechanisms that ensure that the organization’s IT sustains and extends the organization’s strategy and objectives” (p. 1).

Chulani et al. (2006) include both decision rights and the alignment with business needs: “Within IBM, a widely accepted definition for IT governance is:

- Governance that pertains to an organization’s information technology activities and the way those activities support the goals of the business

- Decision making rights associated with IT as well as the mechanisms and policies used to measure and control the way IT decisions are made and carried out within the organization” (p. 10).

In recent years, several IT governance and control frameworks, such as CobiT<sup>1</sup>, ITIL<sup>2</sup>, ISO-17799<sup>3</sup> have been developed. These frameworks help business management, IT management, quality practitioners, and auditors understand what needs to be done; yet they are far from being complete. Dahlberg and Kivijärvi (2006) outline the limitations of CobiT as a process-centric framework and suggest a new framework that takes an integrated process and structural approach, and links into corporate governance.

Another limitation stems from the fact that CobiT is a high-level framework targeted at IT organizations that support a business unit or a business organization. CobiT considers software development activities only within the context of providing a supporting service in a value chain for another business unit, rather than as a central business activity in itself. Software development activities are briefly described in CobiT as part of the high-level control objective AI2, “Acquire and Maintain Application Software.” CobiT thus lacks a description of governance mechanisms that are appropriate for organizations with a large focus on software development. To that end, organizations need to refer to other standards and frameworks that focus more on software development and control of software development activities.

This chapter is aimed at bridging the gap between high-level IT governance and software development governance. We first present a model for governance in general, and then use the model to describe IT and software development domain-specific governance. The model is built based on a review of the literature and a set of scenarios, as explained in the next section. We use the process of transition to agile software development (Beck & Andres, 2004; Dubinsky, Hazzan, Talby, &

17 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

[www.igi-global.com/chapter/governance-software-development/29395](http://www.igi-global.com/chapter/governance-software-development/29395)

## Related Content

---

### Capturing Consumer Preference in System Requirements Through Business Strategy

Constantinos Giannoulis, Eric-Oluf Svee and Jelena Zdravkovic (2013). *International Journal of Information System Modeling and Design* (pp. 1-26).

[www.irma-international.org/article/capturing-consumer-preference-in-system-requirements-through-business-strategy/103315](http://www.irma-international.org/article/capturing-consumer-preference-in-system-requirements-through-business-strategy/103315)

### Improving Novice Programmers' Skills through Playability and Pattern Discovery: A Descriptive Study of a Game Building Workshop

Thiago Schumacher Barcelos, Roberto Muñoz Soto and Ismar Frango Silveira (2015). *Human Factors in Software Development and Design* (pp. 141-172).

[www.irma-international.org/chapter/improving-novice-programmers-skills-through-playability-and-pattern-discovery/117300](http://www.irma-international.org/chapter/improving-novice-programmers-skills-through-playability-and-pattern-discovery/117300)

### Identifying Biased Reviews: An Analysis on Amazon Electronic Products

Md. Niaz Imtiaz, Md. Toukir Ahmed and Md. Rakib Hasan (2022). *International Journal of Software Innovation* (pp. 1-10).

[www.irma-international.org/article/identifying-biased-reviews/297991](http://www.irma-international.org/article/identifying-biased-reviews/297991)

### User Interface Design in Isolation from Underlying Code and Environment

Izzat Alsmadi (2018). *Application Development and Design: Concepts, Methodologies, Tools, and Applications* (pp. 530-538).

[www.irma-international.org/chapter/user-interface-design-in-isolation-from-underlying-code-and-environment/188222](http://www.irma-international.org/chapter/user-interface-design-in-isolation-from-underlying-code-and-environment/188222)

### A New Method for Writing Assurance Cases

Yutaka Matsuno and Shuichiro Yamamoto (2013). *International Journal of Secure Software Engineering* (pp. 31-49).

[www.irma-international.org/article/new-method-writing-assurance-cases/76354](http://www.irma-international.org/article/new-method-writing-assurance-cases/76354)