A FRAMEWORK FOR INTEGRATING SARBANES-OXLEY COMPLIANCE INTO THE SOFTWARE DEVELOPMENT PROCESS

Sushma Mishra
Virginia Commonwealth University
mishras@vcu.edu

Heinz Roland Weistroffer
Virginia Commonwealth University
hrweistr@vcu.edu

Abstract

The Sarbanes-Oxley act introduces a new set of requirements into software development. Corporations need to assess their internal control effectiveness for business processes to show compliance with the act. This paper proposes a conceptual framework for integrating Sarbanes-Oxley compliance needs into the software development process by mapping the various stages of the software development process with an established framework for internal controls.

Keywords: Sarbanes-Oxley, Internal Control, Software Development Life Cycle

Introduction

In the aftermath of the accounting scandals at Enron, HealthSouth, Tyco and Worldcom, many companies saw huge depreciations in the value of their stocks. Many more companies, other than the ones publicly exposed, were suspected of being involved in similar accounting malpractices, and potential investors demanded more accountability from top management. Government intervention in the form of regulation is a viable option under such conditions (La Porta et. al., 2000). Thus, the Sarbanes-Oxley (SOX) act was passed by the American congress in 2002.

SOX is one of the most sweeping act since Securities and Exchange act of 1934 (Coates, 2003), with big implications for ethics in corporate governance practices and thus with direct implications on information technology (IT) governance practices. SOX establishes new standards for corporate accountability by requiring companies to assess and report the effectiveness of internal controls and procedures for financial reporting. CEOs and CFOs must certify and provide quarterly and annual reports to the Security Exchange Commission. Management must accept responsibility for the effectiveness of its internal controls, evaluate the effectiveness using suitable control criteria, and support this evaluation with sufficient evidence.

SOX places an extra burden on IT in corporations, as a strong IT infrastructure becomes a necessity to show compliance with this law. IT provides an effective foundation of efficient internal control in an organization (Fox, 2004). IT divisions must provide not only various kinds of control documentation (as seen in the form of manuals, flowcharts, memoranda, etc.), but also documentation about the effectiveness of those controls. This new business requirement of SOX compliance significantly affects the software development process in two ways: IT applications need to provide the means for effective internal controls and monitoring of other business operations to show compliance with SOX, and the development process itself also needs to be controlled and monitored. Thus, SOX compliance issues become an additional set of requirements that need to be considered in the software development process.

Impact of SOX on IT Governance

SOX is divided into eleven titles or sections, some of which concern executives and others involved in IT. The sections that most directly impact IT are section 404 and section 409. To meet compliance with respect to section 404, executives must attest not only to their companies' financial statements, but also on control processes surrounding the collection of the data behind them—down to the transaction level (Gallagher, 2003). Section 409 requires real time disclosure of financial and operating events. Compliance with these two sections require that each step in a transaction—from order, to payment, to storage of data, to aggregation into financial reports—will need to be audited, verified, and monitored (Volonino, Kermis, & Gessner, 2004).

To meet SOX compliance, corporations must continuously check if sufficient internal controls are in place, and the effectiveness of these controls must be verified by outside auditors and reported quarterly. Companies need to assess various risks involved in IT projects. It is management's responsibility to balance risk and control investment in an unpredictable business environment. This applies even more so for IT development companies, where software development is a major revenue generating business process. Particularly for companies that develop software as their core business, internal controls must be integrated into development methodologies. IT has always been considered an enabler for effective deployment of an organization's strategy (Orlikowski, 1992), but IT should be considered an integral part of a company's strategy itself. IT governance, defined as "structure of relationships and processes to direct and control the enterprise in order to achieve the enterprise's goals by adding value while balancing risk versus return over IT and its processes" (ISACA, 2004, p. 53), provides the structure that links IT processes, IT resources and information to meet business objectives. It enables the enterprise to create value from information and maximize benefits from it. SOX will force companies to effectively manage their internal controls.

COBIT Framework

Control Objectives for Information and Related Technology (COBIT) is a rich and robust IT governance framework. This framework, which accommodates managerial as well technical issues, is considered standard all across industry for developing and maintaining IT controls. It has become a leading international model to establish and maintain IT controls (Damianides, 2004; Fox, 2004). The main objective of the COBIT framework is to support clear policies and good practices for security and control in IT, with worldwide endorsement by commercial, governmental and professional organizations. COBIT is designed for three distinct audiences:

- *Management*: COBIT helps management balance and mitigate risk in an unpredictable IT environment.
- *Users*: COBIT helps assure users of the security and controls of IT services provided by internal or third parties.
- *Auditors*: COBIT helps auditors in fairly assessing company claims regarding its control at work.

In COBIT, control objectives are defined in a process-oriented manner following the principle of business reengineering. Control is defined here as "the policies, procedures, practices and organizational structures designed to provide reasonable assurance that business objectives will be achieved and that undesired events will be prevented or detected and corrected" (ITGI, 2000). The underlying idea of the COBIT framework is that IT control should be approached by looking at information that can support business objectives or requirements. COBIT also looks at information as being the result of the combined application of IT related resources that need to be managed by IT processes.

COBIT comprises four domains, thirty-four IT processes or high-level control objectives, and 318 detailed control objectives. The basis of this classification is three levels of IT efforts, which are required for effective management of IT resources. At the bottom-most level are activities and tasks. These need to achieve a measurable result. Processes are then defined, which is one level above activities and tasks. Processes are a series of joined activities and tasks with natural controls. At the highest level are domains, which are groups of processes. This conceptual framework can be approached from three different points of view: information criteria, IT resources, and IT processes. The four domains of COBIT are:

- **Planning and organization** – covers the strategic importance of IT and assesses how IT is able to meet business objectives in a better way.

- **Acquisition and implementation –** defines the ways or means to achieve output from IT, once the importance and use of IT in meeting the strategic goals is decided.

- **Delivery and support –** considers the delivery of required services for the tools adopted in the acquisition and implementation domain.

- **Monitoring –** focuses on supporting all functions to ensure smooth operations. Identifying solutions to business needs and acquiring these solutions does not automatically realize the goals of an organization.

Each of these domains has a series of sub domains that extensively cover all the required control points for internal control assessment purposes.

Unified Process

A SOX compliant company has to have its internal controls in place for all its processes including software development. The Unified Process (UP) is a software development life cycle approach, which is characterized by iteration and incrementation, and which has emerged as a *de facto* standard software development process (Krucheten, 2000; Booch, Rumbaugh, and Jacobson, 1999).

In UP, relatively short iterations produce artifacts, which are modified or incremented in later iterations. Iterations may be like a mini-waterfall life-cycle, going through some or all of the workflows of requirements, analysis, design, and implementation. UP is typically divided into four consecutive phases:

- **Inception:** The business needs addressed by the new application are defined. At the end of this phase, stakeholders should have a reasonable view about the scope, requirements, and feasibility of the project, with a tentative project plan.

- **Elaboration:** The solution to the problem defined in the inception stage is addressed. Some of the major decision points in this phase include: acceptability of the project to stakeholders, reliability of projected estimates, and dependability of the estimated time frame for deliverables. These factors then contribute to the commitment to funding the entire project, with a detailed project plan and good design and/or prototype for the solution.

- **Construction:** The focus is on the actual creation of the solution. All components and features are created, tested, and integrated. The deliverables of this phase are a functional version of the application, together with adequate documentation, and a transition plan.

- **Transition:** The application is deployed. Additional user testing and refinements take place here, together with training and data transitioning. The primary evaluation criteria for this phase are user satisfaction and cost over-run estimation.

A Conceptual Framework Mapping COBIT with Unified Process

The framework, summarized in Figure 1, maps the control objectives of COBIT to various stages of UP. This mapping provides a means to identify issues during the software development process which impact SOX compliance. Though there are other software development approaches (e.g. the traditional waterfall life cycle model, agile processes, prototyping, etc.), we use UP as perhaps the most encompassing methodology. As pointed out in the previous section, individual iterations in UP may go through all the same workflows that constitute the stages of the traditional waterfall approach, agile processes may be implemented within a UP structure, and prototyping is just one tool that is often employed within UP.

Given below is a brief discussion on the reasoning for mapping certain control objectives with one phase and not the other. The mappings are based on logic reasoning by the authors, using the authors' definitions of the phases. It is a subjective assessment of the fit of objectives and phases.

Inception

In this phase, inputs like requirements assessment, feasibility assessment, information needs and data input requirement are defined (IBM, 2005). All the above processes primarily fall under the ***planning and acquisition*** domain of the COBIT framework. Thus most of the objectives in this domain are mapped to the inception phase of UP (except ***manage human resources***, ***manage projects*** and ***manage quality***, which we feel are not the focus at this stage in UP). The inception phase could be critical for identifying governance issues. Some objectives from other domains also need to be considered at this stage, as shown in Figure 1.

Elaboration

This phase iteratively builds the core architecture and resolves the technical risks of the project. Decisions regarding funding of the entire project take place here and requirements analysis is completed in this phase through feedback and adapting to the changes. All control objectives of the ***acquire and implement*** domain of COBIT are to be considered in this stage of UP. Some objectives from other domains also need to be considered, as shown in Figure 1.

Construction

This phase iteratively integrates the components developed, prepares for deployment and provides quality assurance features. This phase is important for considering most of the objectives in the ***deliver and support*** domain (other than the ones regarding service, security and costs). Managing operations, facilities and configuration along with user training is emphasized in this phase. Some objectives from other domains also need to be considered, as shown in Figure 1.

Transition

This phase completes beta testing of the system, resolves implementation issues and deploys the system. Documentation is completed; assessment of effectiveness as per the requirements, and an external audit of the development process may be done. All the objectives of the ***monitor and evaluate*** domain of COBIT are important for this phase. Some objectives from other domains also need to be considered, as shown in Figure 1.

COBIT Control Objectives	UP Components			
	Inception	Elaboration	Construction	Transition
Plan and Organize (PO)				
Define a strategic IT plan.	●			
Define the information architecture.	●	●		
Determine technological direction.	●	●	●	
Define the IT organization and relationships.	●	●		
Manage the IT investment.	●			●
Communicate management aims and direction.	●	●		
Manage human resources.			●	●
Ensure compliance with external requirements.	●	●	●	
Assess risks.		●		
Manage projects.		●		●
Manage quality.		●	●	●
Acquire and Implement (AI)				
Identify automated solutions.	●	●		
Acquire and maintain application software.		●	●	●
Acquire and maintain technology infrastructure.		●	●	
Develop and maintain procedures.		●	●	●
Install and accredit systems.			●	●
Manage changes.		●	●	●
Deliver and Support (DS)				
Define and manage service levels.	●	●	●	●
Manage third-party services.		●	●	●
Manage performance and capacity.			●	●
Ensure continuous service.	●			●
Ensure systems security.		●		
Identify and allocate costs.	●			
Educate and train users.			●	●
Assist and advise customers.			●	●
Manage the configuration.			●	
Manage problems and incidents.			●	●
Manage data.		●	●	
Manage facilities.			●	●
Manage operations.			●	●
Monitor and Evaluate (M)				
Monitor the processes.			●	●
Assess internal control adequacy.	●	●		●
Obtain independent assurance.	●			●
Provide for independent audit.	●	●	●	●

Figure 1. A Conceptual Mapping of UP Phases with COBIT Control Objectives

Conclusion

The full repercussions of SOX are still being assessed by businesses. It seems definitive though, that to show compliance with the act, companies need to have a strong IT infrastructure. All the business processes have to be mapped to internal control objectives to meet internal control assessment compliance. This paper presents a conceptual mapping of COBIT control objectives to various phases of a commonly used software development approach. This framework may facilitate internal control assessment during the course of software development. This paper recognizes SOX compliance as another requirement, which has to be considered by all software development projects, irrespective of the approach adopted for development.

Currently the framework presented here is purely conceptual, based on logic and the authors' experiences. Future work would include empirically validating the framework, possibly using an action research approach.

References

Booch G., Rumbaugh, J. and Jacobson, I. (1999). *The Unified Modeling Language User Guide*. Addison-Wesley Longman.

Coates, B. E. (2003). Rogue corporations, corporate rogues & ethics compliance: The Sarbanes-Oxley Act, 2002. *Public Administration and Management, 8*(3), 164-185.

Damianides, M. (2004). How does SOX change IT? *The Journal of Corporate Accounting & Finance*, September/October.

Fox, C. (2004). Sarbanes-Oxley-considerations for a framework for IT financial reporting controls. *Information Systems Control Journal* 1.

Gallagher, S. (2003) Gotcha! Complying with financial regulations. *Baseline Magazine* August 1, 2004. Retrieved December 21, 2005 from, http://www.baselinemag.com/article2/0,1397,1211224,00.asp.

IBM (2005). Rational Unified Processes: Best practices for software development teams. Retrieved December 21, 2005 from, http://www.augustana.ab.ca/~mohrj/courses/2000.winter/csc220/papers/rup_best_practices/rup_bestpractices.

ISACA (2004). *CISA ReviewManual, 2004 Edition*. Information Systems Audit and Control Association, Rolling Meadows, IL.

ITGI (2000). *COBIT Framework.* Released by COBIT Steering Committee and IT Governance Institute.

Kruchten, P. (2000). *The Rational Unified Process: An Introduction.* 2nd ed. Addison Wesley Longman, Reading, MA.

La Porta, R., Lopez-de-Silanes, F., Shleifer, A. and Vishny, R. (2000). Investor protection and corporate governance. *Journal of Financial Economics*. 58, 3-27.

Orlikowski, W.J. (1992). The duality of technology: Rethinking the concept of technology in organizations. *Organization Science*, 3(3), 398-427.

Volonino, L., Kermis, G., Gessner, G. (2004) Sarbanes-Oxley links IT to Corporate Compliance. *Proceedings of the Tenth Americas Conference on Information Systems*, New York, New York, August 2004.