Communications of the Association for Information Systems

Volume 20 Article 44

November 2007

A Framework for Integrating Sarbanes-Oxley Compliance into the Systems Development Process

Sushma Mishra
Virginia Commonwealth University, mishras@vcu.edu

Heinz Roland Weistroffer
Virginia Commonwealth University

Recommended Citation
Mishra, Sushma and Weistroffer, Heinz Roland (2007) "A Framework for Integrating Sarbanes-Oxley Compliance into the Systems Development Process," *Communications of the Association for Information Systems*: Vol. 20 , Article 44.
DOI: 10.17705/1CAIS.02044
Available at: https://aisel.aisnet.org/cais/vol20/iss1/44

Communications of the Association for Information Systems

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

Sushma Mishra
Heinz Roland Weistroffer
Department of information Systems
School of Business
Virginia Commonwealth University
Richmond, VA 23284-4000
mishras@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 software development by mapping the activities of an established framework for internal controls to the various workflows of the systems development process. Theoretical and practical contributions are discussed and future research directions are explored.

Keywords: SOX, internal controls, software development, COBIT, unified process, workflows, requirements, analysis, design, implementation, testing

I. INTRODUCTION

In the aftermath of the accounting scandals at Enron, HealthSouth, Tyco, and Worldcom, many companies saw significant depreciations in the value of their stocks. Many more companies, in addition to 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; Moore 2004]. Thus, the Sarbanes-Oxley (SOX) Act was passed by the American Congress in 2002.

SOX is one of the most sweeping acts since the Securities and Exchange Act of 1934 [Coates 2003; Burrows et al. 2004], with major consequences for ethics in corporate governance, 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. Chief executive officers (CEOs) and chief financial officers (CFOs) must certify and provide quarterly and annual reports to the Security Exchange Commission [Dietrich 2004]. Management must accept responsibility for the effectiveness of its internal controls, evaluate the effectiveness using suitable control criteria, and support this evaluation with adequate evidence. IT provides an effective foundation for efficient internal controls in an organization [Fox 2004; Damianides 2005].

Lack of sufficiently documented requirements and lack of a change control process have been identified as major reasons for project failures [Kappelman et al. 2006]. These concerns are clearly recognized in the course of action for SOX compliance as SOX places an extra responsibility on IT in corporations and a strong IT infrastructure becomes a necessity to show compliance with this law. IT divisions must provide not only various kinds of control documentation (in the form of manuals, flowcharts, memoranda, etc.) but also documentation about the effectiveness of those controls. The IT governance structure must be designed so that IT adds value to the business and IT risks are mitigated [Braganza and Desouza 2006]. This also includes an IT organization structure that supports adequate segregation of duties and promotes the achievement of the organization's objectives [Information Systems Audit and Control Association 2004]. The new "business requirement" of SOX compliance significantly affects the systems development process in two ways: First, IT applications provide the means for effective internal controls and monitoring of other business operations to show compliance with SOX; second, the software development process itself needs to be controlled and monitored. Thus, SOX compliance issues present an additional set of requirements that needs to be considered in the systems development process. Good IT governance with respect to planning and life cycle control objectives should result in more accurate and timely financial reporting.

SOX has brought many information systems related business decisions into the purview of the investor's scrutiny. Software development projects have particular control requirements and needs to be fulfilled using appropriate tools such as workflow and event management or integrated development environment [Basham 2006]. Decisions on, for example, whether to develop in-house or buy applications, the adequacy of return on investments in software development projects, and the effectiveness of internal controls in the maintenance of systems are of utmost importance, and management, by law, is accountable to shareholders on these decisions. Thus in order to maintain the trust of their investors, business organizations must be careful in the choice of methodologies adopted for selecting or developing new IT applications. The process of software development includes design, building, and deployment of systems that help in achieving business objectives of an enterprise. Deficiencies in this development process can impact financial reporting and disclosure. For example, without sufficient controls for application interfaces, financial information may not be complete or accurate [IT Governance Institute 2006a]. Thus it is important to have a compliance plan from the start of the development process.

It is argued here that all organizations that develop business software, whether as in-house applications, or for commercial distribution, need to consider SOX compliance as a systems requirement. While defining the functional and non-functional requirements for particular applications, compliance issues must be considered as well and may impact the choice of development methodology adopted for a project. Investments in software development are high and so are the associated risks with such projects. Project failures eat into earnings of companies and may lead to extra auditor scrutiny. Software development and implementation requires the evaluation of the enhancements to the existing system, extensive testing, user retraining, and the rewriting of procedures. These processes require IT controls to be in place, such as authorization of change requests, review of the changes, approvals, documentation, testing, and assessment of change impact on other IT components, and implementation protocols [IT Governance Institute 2006]. The change management process also needs to be integrated with other IT processes, including incident management, problem management, availability management, and infrastructure change control. Thus software development and implementation functionality require a deeper understanding of the applicability of IT controls during the life cycle of the development process and the identification of appropriate control points to ensure compliance with regulations.

This paper presents a framework for incorporating and thus integrating control objectives of IT governance into the various workflows of software development. A mapping of an established IT governance framework, *COBIT* (*Control Objectives for Information and Related Technology*) [IT Governance Institute 2000; 2006a], to the core workflows which are common to most established systems development approaches, is presented.

The rest of this paper is organized as follows: Section II presents the concepts on which the framework is based, which includes an overview of SOX and its implications on IT governance practices; a brief discussion of the most commonly used IT governance framework, COBIT, and an overview of the UP approach to systems development. Section III presents our framework, a conceptual mapping of the COBIT control objectives to the core workflows of systems development, with reference to the traditional waterfall approach as well as to the *unified process* (*UP*) [Jacobson et al. 1999; Kruchten 2004], with a discussion of the framework's constituent parts. In the concluding section, contributions of this paper are highlighted along with avenues for possible future research.

II. FRAMEWORK FUNDAMENTALS

SARBANES-OXLEY

SOX is divided into 11 titles or sections, some of which concern executives and others involved in IT [US Securities and Exchange Commisision2003]. 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 to 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 et al. 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 to 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. SOX forces companies to effectively manage their internal controls.

COBIT

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 the company controls that are in place.

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" [IT Governance Institute 2000]. The underlying idea of the COBIT framework is to approach IT controls by assessing the 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, 34 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. Activities and tasks need to achieve a measurable result. Processes are defined 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. The COBIT framework can be approached from three different points of view: information criteria, IT resources, and IT processes. The four domains of COBIT are:

1. *Plan and Organize.* This domain covers the strategic importance of IT and assesses how IT is able to meet business objectives in a better way. This domain reaffirms the importance of planning and organizing effectively to realize the strategic vision of a company. To achieve the desired vision, proper technological infrastructure is required which can support the long-term and short-term goals of the organization.

2. *Acquire and Implement.* This domain defines the ways or means to achieve output from IT, once the importance and use of IT in meeting the strategic goals is decided. In this domain, importance of methods and tools to implement strategic goals at the operational level is highlighted. IT solutions have to be developed or acquired and integrated with the business processes to meet the business objectives. The focus in this domain is to provide the means for realizing the objectives of the planning and organization domain. Continuous improvement in IT systems and their maintenance is the goal in this domain.

3. *Deliver and Support.* This domain deals with the delivery of required services by means of the tools adopted in the acquisition and implementation domain. Ensuring the proper functioning of the systems implemented and providing support to the systems in use are the core purposes of this domain. To deliver services, necessary support processes must be set up to help in processing data or support application controls. Acquiring such services requires decisions such as whether a new system should be outsourced or be developed in-house. The critical support services must be provided for optimal functioning and uninterrupted operations.

4. *Monitor and Evaluate.* This domain focuses on checking and 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. A critical factor for smooth functioning of IT systems is to provide continued support to such IT solutions so that business objectives are realized. This domain monitors all IT processes for its quality and compliance with control requirements. Continuous monitoring ensures that all the controls are in place and working effectively. It also addresses any oversight on behalf of management in the control process.

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

COBIT is a high-level framework that suggests what needs to be done by an organization in order to be compliant and effective. But how to address the requirements of COBIT in detail and

achieve the desired outcomes must be resolved by each organization individually. People in various roles and assigned various responsibilities in the organization may be empowered to make decisions on how to best reach the COBIT objectives in accordance with their business needs. COBIT by itself does not guarantee SOX compliance, but rather the appropriate customization and implementation of the COBIT guidelines will lead to success in this area. Also, there may be additional financial control requirements for SOX compliance that are not sufficiently emphasized in COBIT. Thus, simply following COBIT will not automatically result in SOX compliance but can definitely strengthen general and application level controls.

SOFTWARE DEVELOPMENT PROCESS

Software development methodologies can generally be classified as either waterfall or iterative development. In the waterfall approach, all the workflows for systems development are sequential in nature, i.e. constitute different, chronologically sequential phases of the development process, where one starts after the completion of the other. In iterative development, the various workflows are repeated several times, producing new system artifacts or modifying previously produced artifacts [Jacobson, Booch, and Rumbaugh 1999].

UP, perhaps the most popular and widely used iterative approach, is an object-oriented iterative and incremental systems development methodology using the unified modeling language (UML) as its primary modeling notation [Kruchten 2004; Booch et al. 1999]. In UP, relatively short iterations produce artifacts such as business, analysis and design models, prototypes, and system components, which may be modified or added to in later iterations. Individual iterations may resemble mini-waterfall system development life cycles, going through some or all of the workflows of requirements determination, analysis, design, and implementation. UP is typically divided into four consecutive phases: *Inception, elaboration, construction, and transition*.

Inception: This phase of the UP generally focuses on planning and requirements determination [IBM 2005]. In this phase, the business problem that is to be addressed by the new application is defined. At the end of this phase, stakeholders should have a reasonable view about the scope, requirements, and feasibility of the project. Some of the artifacts typically created in this phase include a tentative project plan, an initial risk assessment, a business model, and a vision document for the project.

The inception phase is probably the most critical for identifying governance issues. Some of the milestones of this phase include: scope definition of the project, requirements understanding, cost schedules, architectural prototype, and expenditure estimates [Kruchten 2004]. The IT compliance program should be properly scoped and planned at the beginning of the development process.

Elaboration: The business problem defined in the inception phase is further developed here. The primary focus is on analysis and design. Some of the major decision points in this phase relate to the acceptability of the project to stakeholders, the reliability of projected estimates, and the dependability of the estimated time frame for deliverables. This phase iteratively determines the core architecture of the IT application and resolves the technical risks of the project. All these factors contribute to the commitment to funding the rest of the project. Typical artifacts created in the elaboration phase include a revised, more detailed project plan, and a conceptual design and/or prototype for the IT application.

Construction: In this phase of the UP the focus is on the actual creation of the solution. All components and features of the IT application are created, tested, and integrated. The major deliverables of this phase are a functional version of the application, together with adequate documentation, and a transition plan. The essential activities of this phase are resource management and the evaluation of product releases against acceptance criteria.

Transition: The new application is deployed and documentation is completed in this phase. User acceptance testing, user training, and data transitioning are major activities in this phase. This

phase may also include parallel operations with the legacy systems that the project is replacing, conversion of operational databases, and rollout of the product to the marketing, distribution, and sales teams [Kruchten 2004]. The primary evaluation criteria for this phase are user satisfaction and cost over-run estimation.

While all of the workflows may take place within each of the four phases, the amount of time spent for requirements determination, analysis, design, implementation and testing typically varies in the different phases, the early phases focusing more on requirements and analysis, and the later phases more on design, implementation, and testing.

III. A CONCEPTUAL FRAMEWORK MAPPING COBIT WITH DEVELOPMENT WORKFLOWS

SOX impacts various aspects of the information systems discipline, such as project management, software development, IT governance, and data monitoring [Dhillon and Mishra 2006]. The software development process is becoming more formalized with SOX [Leih 2005] because compliance needs make development more process centric and encourage organizations to follow all the controls and documentation requirements. But the authors are not aware of any published research which has directly explored the impact of various internal controls within the software development life cycle.

The proposed framework, summarized in Table 1, maps the control objectives of COBIT to various core workflows in the development process. This mapping provides a means to identify tasks and issues during systems development which impact SOX compliance the most. In the following we discuss the reasoning for mapping specific control objectives with each of the five core workflows: requirements, analysis, design, implementation, and testing, which form part of all structured systems development methodologies, and thus our framework can be applied to most development approaches The authors want to add a note of caution though that all the control objectives provided by COBIT are important and have to be considered during the whole systems life cycle.

In this next section, we identify the major control objectives which are crucial for a particular workflow during systems development. The mapping has been done on the basis of general experience with and knowledge about UP, and guidelines provided by the IT Governance Institute for IT controls in SOX compliance.

REQUIREMENTS

The main purpose of the requirements workflow (Figure 1) is to ensure that the right system is developed. This is a difficult discipline, as determining and specifying what is required in a system is challenging on the part of end users as well as analysts. End users are rarely able to easily communicate their exact requirements to the analysts, and in fact, they may not always know themselves what their requirements are. In the UP, requirements analysis is done primarily in the inception and elaboration phases. Critical requirements, around 10 percent, typically are identified during the inception phase and about 80 percent are captured during the elaboration phase [Jacobson, Booch, and Rumbaugh 1999]. The remaining requirements are determined during the construction phase.

In this workflow, business models are created to set the context of the system and use case models are defined to captures the functional requirement [Jacobson, Booch, and Rumbaugh 1999]. User interface sketches or throw-away prototypes may also serve to correctly identify some of the requirements.

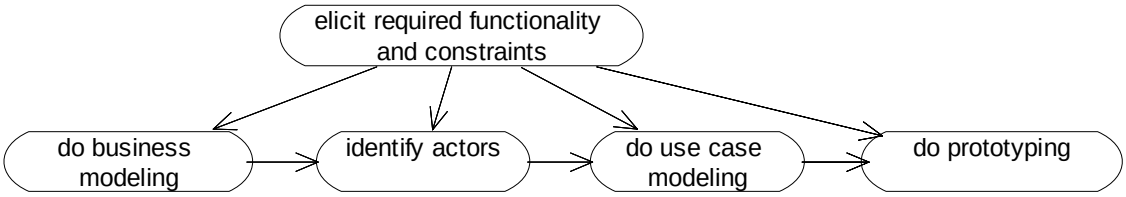

Figure 1. Requirements Workflow

Depending on how the requirements workflow is conducted, control objectives in all the four domains of COBIT may map to this workflow. The control objectives relevant to the requirements workflow are shown in Table 1. Thus, for example, in order to identify the requirements for a system, it is important to define a strategic IT plan and determine the technological direction for the organization. COBIT emphasizes the importance of communicating management aims and objectives and "setting the tone at the top" for a successful project outcome and an effective control environment [Damianides 2004]. If compliance is considered another requirement in systems development, it is important to plan for creating an audit trail for the entire project such that independent audit assurance can be provided.

SOX compliance requires a justification of a development project in terms of the viability for shareholders. Thus having a clear strategic plan and identifying the impact of the project on the business are some of the important control points that need to be considered, from a compliance perspective. Also, risk assessment is a crucial process during planning, and all the steps taken in this direction need to be documented for compliance purposes. Not all IT systems or projects pose a high risk to financial statements; hence not all projects need to be included in a compliance program. Assessing the inherent risks of a project is essential to determine the nature of the controls required to manage such risks. Documenting controls is required to manage the risks that threaten reliable financial reporting. Identification of controls in the project and providing for independent audits should be done while determining system scope and requirements, as it helps in preparing for a better compliance plan.

Table 1. Mapping of COBIT Control Objectives with Development Workflows

	Development Workflows				
COBIT Control Objectives	Requirements	Analysis	Design	Implementation	Testing
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.	•		•		•

ANALYSIS

This workflow (Figure 2) deals with analyzing the existing system and the identified requirements for the new system to be developed. Analysis models are used to capture and communicate the

proposed system requirements in more detail. An analysis model structures a requirement in a way that facilitates understanding and allows further modification when required [Jacobson, Booch, and Rumbaugh 1999]. This model acts as an essential input to the design and implementation workflows. In UP, analysis is the focus of the early iterations during the elaboration phase. The analysis workflow helps in determining the system architecture and facilitates better understanding of the requirements. Analysis models serve as an intermediate tool, and are later discarded or evolved into design models.

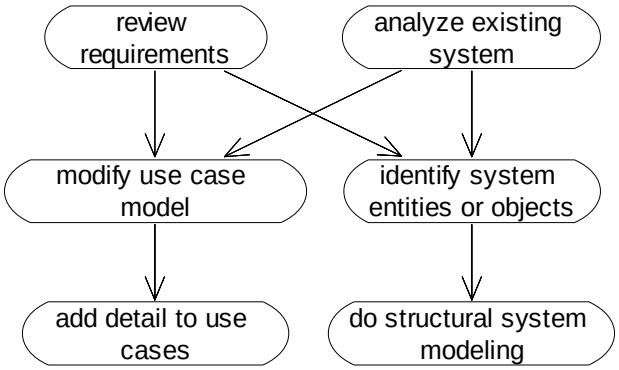

Figure 2. Analysis Workflow

The control objectives of the *acquire and implement* domain of COBIT must be considered for this workflow, but controls from other domains of COBIT are important as well. The control objectives that need consideration from a compliance perspective are shown in Table 1. SOX emphasizes the importance of identifying risks related to IT systems, designing and implementing controls intended to mitigate such risks, and monitor the risks continually [IT Governance Institute 2006]. Activities in the analysis workflow require risk assessment for ensuring systems security and identifying technical solutions accordingly.

The analysis workflow plays a decisive role in defining the security aspects of the IT application, as well as outsourcing needs and change management constraints. SOX emphasizes security of information systems and has important implications for change management functionality. As part of analysis, decisions on data models, architectural plans, data security at the enterprise level or interorganizational level, access control mechanisms, segregation of roles among employees, process models, and disaster recovery techniques, need to be made. Making SOX compliance a constraint while deciding on these issues may avoid the need for expensive changes to the system later on. Approaching compliance as an opportunity to establish better systems development processes provides manifold benefits, such as better alignment between project initiatives and business requirements, more efficient system operations, and potentially reduced system breaches [Damianides 2004]. SOX requires that policies and procedures regarding program development be periodically reviewed, updated and approved by management [IT Governance Institute 2006b].

DESIGN

The design workflow (Figure 3) applies in-depth understanding of the issues dealing with nonfunctional requirements. It helps in conceptualizing the constraints related to programming languages, component reuse, operating systems, database technologies, transaction management technologies, etc. [Jacobson, Booch, and Rumbaugh 1999]. It creates an appropriate input to and point of departure for the implementation workflow, like collecting requirements on individual subsystems. The design workflow also captures major interfaces between sub-systems and creates a seamless abstraction of the proposed systems

implementation, such that the implementation workflow becomes a refinement of the design workflow by filling in essentials without changing the structure.

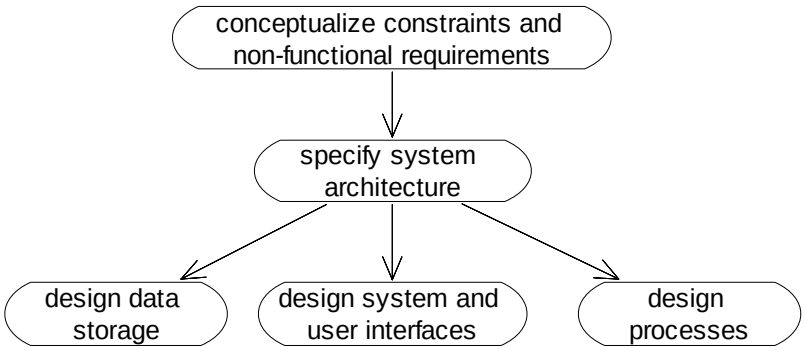

Figure 3. Design Workflow

In UP, design is the focus during the later iterations of the elaboration phase and the early iterations of the construction phase [Jacobson, Booch and Rumbaugh 1999]. The design workflow creates a stable architecture and a blueprint for the implementation workflow. Design models are very close to the actual system and are maintained through the end of the development life cycle.

The main SOX compliance issues that need to be considered for this workflow are fulfillment of the initial objectives of the project, adequate return on investment for shareholders, extensive documentation maintenance for external auditing, and security standards for business transactions. The business logic needs to include compliance constraints and the system should be prepared to meet compliance standards and detect errors. Thus the design workflow is important for considering most of the COBIT control objectives from the *deliver and support* domain. Some control objectives from other domains need to be considered as well. The control objectives important for the design workflow are shown in Table 1. Various aspects of project management become important in this workflow to ensure service levels, allocate costs, and manage facilities and operations. Leih [2006] claims that SOX compliance leads the greater part of project management to ensure internal controls are being followed.

Documenting controls is required to manage the risks that threaten reliable financial reporting. Identification of controls in the project helps in preparing for a better compliance plan. SOX requires companies to document controls for financial reporting and perform an assessment of their design and operating effectiveness [IT Governance Institute 2006b]. No specific form of documentation is mandated by SOX, thus documentation may take any form, such as policy manuals, IT policies and procedures, narratives, and flowcharts. Identifying and documenting disaster recovery plans is mandatory for SOX compliance. With a plethora of security breaches across the nation, it is important that a contingency plan exists in case of such breaches, and SOX mandates notifying all shareholders about breaches in a stipulated time period. Thus identifying such control points during the design workflow makes it easier to meet the requirements of compliance.

IMPLEMENTATION

In this workflow (Figure 4), system components are integrated into the user environment. The system is distributed by mapping executable components onto nodes in the deployment model. The design classes and subsystems found in the design workflow are implemented as file components that contain source code. The implementation workflow interacts strongly with the testing workflow, as components are unit tested, integrated and compiled together into executables before integration and system testing [Jacobson, Booch, and Rumbaugh 1999]. This workflow results in functioning components of the system, and eventually in the complete system.

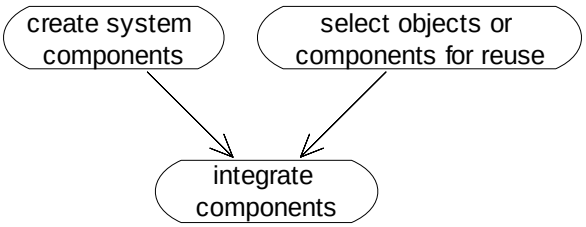

Figure 4. Implementation Workflow

In UP, the implementation workflow is a major discipline of the construction phase. The implementation workflow is also a significant part of the elaboration phase to create the executable architectural baseline, and it is a major focus of the transition phase to handle late defects during release [Jacobson, Booch, and Rumbaugh 1999].

The control objectives of COBIT that are important during this workflow are shown in Table 1. Change management is a big issue during implementation, and an effective transition can be achieved through appropriate training of users, troubleshooting, effectively managing problems, and providing assistance to users to adapt to the new system. Documentation plays an important role in these activities and also helps in preparing for auditing requirements. The IT Governance Institute emphasizes the importance of managing the human element of change for compliance purposes. It is important to use insights into cultural and people management issues when implementing new compliance requirements into a newly developed system component. Alter and Browne [2005] claim that the tools, processes and practices in systems analysis and design activities are important for organizational change management and reengineering efforts. Such efforts improve those work processes that are unsatisfactory.

The implementation workflow has significant implications for SOX compliance. External auditing is mandatory for SOX compliance and needs to be addressed during this workflow. Adequate audit trail facilities need to be provided to help the auditors in tracing the cause of any incident. Independent or third party auditor attesting of compliance is a requirement for SOX. For example, there should be controls such that transactions cannot be recorded outside of financial cut-off dates, and system controls must be in place for appropriate approval of write-offs.

Any issue with the new system has to be logged, documented and subjected to formal change management practices, for compliance purposes. SOX requires informing shareholders about incidents that could impact the financial health of the company. Such activities require upfront preparations for compliance on the part of organizations. Since SOX mandates quarterly reporting of the internal control effectiveness, it is essential for organizations to closely monitor the transition process of a system.

TESTING

The testing workflow (Figure 5) requires planning tests in all iterations, including integration and system tests. Integration tests are required for every build, and system tests are required at the end of each iteration. Test cases and test procedures are created that specify what to test and how to perform the tests. The results of each test are systematically handled. Builds with defects are retested and sent back to other core workflows for fixing the issues. The results of the testing workflow are test models which include: test cases to specify what to test; test procedures to specify how to perform test cases, and test components to specify how to automate the test procedures.

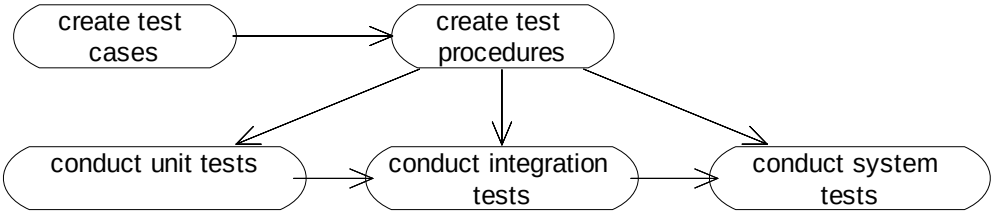

Figure 5. Testing Workflow

In the UP, testing may begin in the inception phase when the system is scoped. However, it is primarily employed when each build is implemented. Thus testing is a focus during the elaboration phase, where the architectural baseline is created, and during the construction phase, when the bulk of the system is implemented [Jacobson, Booch, and Rumbaugh 1999]. The iterative nature of development requires that some of the test cases be used as regression test cases that specify how to regression test subsequent builds.

The control objectives that need extra consideration during this workflow are shown in Table 1. Safeguarding of organizational records, prevention of misuse of information infrastructure, establishing systems audit controls are required during this workflow. For compliance purposes, it is important to ensure that incorrect code and data are not introduced into financial reporting systems. It is necessary to identify where input data validation is performed in the computer programs, and documentation focused on data integrity checks or other security issues needs to be maintained [Haworth and Pietron 2006]. Identifying where input data validation is performed in programs is not an easy task. Most computer programs are documented to show function and process flow; rarely is the documentation focused on data integrity checks or other security issues [Haworth and Pietron 2006]. In the testing workflow, integrity checks should be performed to ensure the security and compliance requirements of the system.

Note that providing for independent assurance and audit has been emphasized in most of the workflows. Assurance and audit are the two main components of obtaining SOX compliance and should be planned for right from the beginning of the project. Considering compliance requirements while a system is being developed helps in establishing policies and procedures and gets it documented during the development itself. This ensures that a written standard is created for reference, and independent interpretation from people is avoided. Clear roles and responsibility structures enhance accountability. A well-documented control structure shows a mature management and also increases the reliability of the financial statement.

IV. FUTURE RESEARCH AND 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 business processes have to be mapped to internal control objectives to meet internal control assessment compliance, and setting standards is essential to achieve consistency in the implementation of the control objectives [Braganza and Desouza 2006]. A framework that helps identify which controls to implement when and where may make it easier to effectively integrate such controls into the IT infrastructure.

This paper presents a conceptual mapping of COBIT control objectives to the core workflows of systems development. This framework, which hopefully will help facilitate internal control assessment during the course of software development, recognizes SOX compliance as another requirement that has to be considered by all systems development projects. Even though there has been significant research on SOX implications in the area of accounting auditing, the information systems discipline lags considerably behind in investigating the impact of SOX on IT and organizations [Leih 2006]. Compliance-related publications in information systems are

primarily focused on the areas of: data reliability, SOX mandates related to enterprise systems, SOX related IT project management issues, and the impact of SOX on IT governance [Leih 2006]. This paper addresses a part of this gap by looking at the impact of SOX on a pertinent area of information systems, viz. systems development, which is a distinctive contribution to the IS field. The conceptual framework suggested in this paper provides a theoretical basis for further research on compliance issues in systems development.

This paper contributes in three ways: First, by introducing the concept of incorporating compliance issues into systems development methodology and providing a conceptual basis for further researching compliance needs during systems development. Even though SOX compliance plays a significant role in business planning today, there is a dearth of research connecting systems development to compliance needs [Duffy 2004; Ivancevich et al. 2003]. Second, the paper contributes by providing a framework that can be applied as a practical guide or tool by systems development managers in the business world. Planning for compliance needs, right from the beginning of a development project, may potentially save extra dollars and time in the completion of the project [Basham 2006]. Third, this paper conceptually establishes the importance of regulatory forces shaping the IT industry in current times [Chin and Mishra 2006] and identifies compliance as a business requirement that has technical as well as organizational preparedness implications.

The framework presented here is conceptual, based on reason and the authors' general knowledge and experiences. As yet there is no empirical validation to support the framework. Future work may include empirically validating the framework, possibly using an action research approach. Another limitation of this work in its current form is that the proposed framework is rooted in the context of projects using structured development approaches, such as the UP or the traditional waterfall approach. A follow-up study may look at the impact of SOX compliance on agile practices. Also, the impact of other regulations such as the Health Insurance Portability and Accountability Act (HIPAA) and the Gramm-Leach-Bliley Act (GLB) on systems development methodologies needs to be explored.

In summary, this paper takes the position that SOX compliance needs to be included as a requirement in the software development process. A conceptual mapping of COBIT control objectives with the core workflows that make up most structured development methodologies is proposed.

REFERENCES

EDITOR'S NOTE: The following reference list contains the address of World Wide Web pages. Readers, who have the ability to access the Web directly from their computer or are reading the paper on the Web, can gain direct access to these references. Readers are warned, however, that:

1. These links existed as of the date of publication but are not guaranteed to be working thereafter.

2. The contents of Web pages may change over time. Where version information is provided in the References, different versions may not contain the information or the conclusions referenced.

3. The authors of the Web pages, not CAIS, are responsible for the accuracy of their content.

4. The author of this article, not CAIS, is responsible for the accuracy of the URL and version information.

Alter, S and G. J. Browne. (2005). "A Broad View of Systems Analysis and Design," *Communications of the Association for Information Systems,* (15), pp. 981-999

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

Basham, R. (2006). "Procedure Guidelines and Controls Documentation: SDLC Controls in COBIT 4.0," *Information Systems Control Journal*, (6).

Braganza, A. and K. Desouza. (2006). "Implementing Section 404 of the Sarbanes Oxley Act: Recommendations for the Information Systems Organizations," *Communications of the Association for Information Systems*, (18), pp. 464-487

Burrowes, A. W., J. Kastantin, and M. M. Novicevic. (2004). "The Sarbanes-Oxley Act as a Hologram of Post-Enron Disclosures: A Critical Realist Commentary," *Critical Perspectives on Accounting,* 15, 797-881.

Chin, A. G. and S. Mishra. (2006). "Increasing Governmental Regulations and Their Impact on IT: SOX and HIPAA," *Proceedings of the International IRMA Conference*, May 2006, Washington DC, May 21-24

Coates, B. E. (2003). "Rogue Corporations, Corporate Rogues and Ethics Compliance: The Sarbanes-Oxley Act," *Public Administration and Management,* (*8*)3, pp. 164-185.

Damianides, M. (2004). "How Does SOX Change IT?" *The Journal of Corporate Accounting & Finance*, (5) 6, September/October, pp. 35-41.

Damianides, M. (2005). "Sarbanes-Oxley and IT Governance: New Guidance on IT Control and Compliance," *Information Systems Management,* Winter 2005, pp. 77-85.

Dhillon, G. and S. Mishra. (2006). "The Impact of Sarbanes-Oxley (SOX) Act on Information Security Governance," In *Enterprise Information Security Assurance and System Security, Managerial and Technical Issues,* Warkentin, M. and R. Vaughan, (Eds.), Hershey, PA: Idea Group Publishing, pp. 62-79.

Dietrich, R. (2004). "Sarbanes-Oxley and the Need to Audit Your IT Processes: An MKS White Paper," *MKS*, www.mks.com, (current 2/27/05).

Duffy, M. N. (2004). "Section 404 Opens a Door," *Journal of Accountancy*, (197), p.8.

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, 2003 http://www.baselinemag.com/article2/0,1397,1211224,00.asp (current Dec. 21, 2005)

Haworth, D. and l. Pietron. (2006). "Sarbanes-Oxley: Achieving Compliance by Starting with ISO 17799," *Information Systems Management*, Winter, pp. 73-87.

IBM. (2005). "Rational Unified Processes: Best Practices for Software Development Teams," http://www.augustana.ab.ca/~mohrj/courses/2000.winter/csc220/papers/rup_best_practices/rup_bestpractices (current Dec. 21, 2005).

Information Systems Audit and Control Association (ISACA). (2004). *CISA Review Manual,* Rolling Meadows, IL: Information Systems Audit and Control Association.

IT Governance Institute. (2000). *COBIT Framework*, COBIT Steering Committee and IT Governance Institute.

IT Governance Institute. (2006a). *COBIT 4.0*, Rolling Meadows, IL: IT Governance Institute.

IT Governance Institute. (2006b). *IT Control Objectives for Sarbanes Oxley: The Role of IT in the Design and Implementation of Internal Control over Financial Reporting*, 2nd Edition, Rolling Meadows, IL: IT Governance Institute.

Ivancevich, J. M., T. N. Duening, J. A. Gilbert, and R. Konopaske. (2003). "Deterring White-Collar Crime," *The Academy of Management Executive*, (17) 2, pp. 114-127

Jacobson, I., G. Booch, and J. Rumbaugh. (1999). *The Unified Software Development Process*, Addison-Wesley.

Kappelman, L., R. Mckeeman, and L. Zhang. (2006). "Early Warning Signs of IT Project Failure: The Dominant Dozen," *Information Systems Management*, Fall 2006, pp. 31-36.

Kruchten, P. (2004). *The Rational Unified Process: An Introduction,* 3rd ed., Reading, MA: Addison-Wesley.

La Porta, R., F. Lopez-de-Silanes, A. Shleifer, and R. Vishny. (2000). "Investor Protection And Corporate Governance," *Journal of Financial Economics*, (58), pp. 3-27.

Leih, M. (2005). "The Impact of the Sarbanes-Oxley Act on IT Project Management: A Case Study," Proceedings of the Twelfth Americas Conference on Information Systems, Omaha, NE, USA, August 11th-14th 2005.

Leih, M. (2006). "IT Governance and the Sarbanes-Oxley Act," Proceedings of the Twelfth Americas Conference on Information Systems, Acapulco, Mexico, August 04th-06th 2006.

Moore, C. (2004)."The Growing Trend of Government Involvement in IT Security," Proceedings from InfoSecCD Conference '04, October, 2004.

Orlikowski, W. J. (1992). "The Duality of Technology: Rethinking the Concept of Technology in Organizations," *Organization Science*, (3)3, pp. 398-427.

U.S. Securities and Exchange Commission. (SEC). (2003). " Management's Reports on Internal Control over Financial Reporting and Certification of Disclosure in Exchange Act Periodic Reports," 2003, Retrieved on 06/25/06 http://www.sec.gov/rules/final/33-8238.htm.

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

ABOUT THE AUTHORS

Sushma Mishra is a Ph.D. candidate in the Information Systems Department at Virginia Commonwealth University. Her research interests lie in the areas of information security governance, internal controls, regulatory compliance, systems audit, health informatics and systems analysis and design.

Heinz Roland Weistroffer is an associate professor of information systems in the School of Business at Virginia Commonwealth University. Roland received his doctorate degree from the Free University of Berlin. His research interests include systems development, computer assisted decision support, security models, and economics of information technology. He has published in *IEEE Transactions on Software Engineering*, the *Journal of Multi-Criteria Decision Analysis*, *Socio-Economic Planning Sciences*, *Computational and Mathematical Organization Theory*, *Journal of Information Systems Security*, *Journal of Computer Information Systems*, and *Electronic Journal of Information Systems in Developing Countries* among other journals.

Communications of the Association for Information Systems

ISSN: 1529-3181