IT Audit Basics

Auditing Applications, Part 1

Tommie W. Singleton, Ph.D., CISA, CGEIT, CITP, CPA, is an associate professor of information systems (IS) at Columbus State University (Columbus, Georgia, USA). Prior to obtaining his doctorate in accountancy from the University of Mississippi (USA) in 1995, Singleton was president of a small, value-added dealer of accounting using microcomputers. Singleton is also a scholar-in-residence for IT audit and forensic accounting at Carr Riggs & Ingram, a large regional public accounting firm in the southeastern US. In 1999, the Alabama Society of CPAs awarded Singleton the 1998–1999 Innovative User of Technology Award. His articles on fraud, IT/IS, IT auditing and IT governance have appeared in numerous publications.

Auditing applications is a common type of audit for medium and large companies, especially when some of the applications are developed in-house. There are some basic principles of auditing applications that IT auditors need to know and understand. This two-part article describes one framework for performing effective audits of applications.

A FRAMEWORK

A process-oriented framework includes steps similar to the following:

- Plan the audit.
- Determine audit objectives.
- Map systems and data flows.
- Identify key controls.
- Understand application's functionality.
- Perform applicable tests.
- Avoid/consider complications.
- Include financial assertions.
- Consider beneficial tools.
- Complete the report.

Some of the steps, such as mapping systems and data flows, are comprehensive. While mapping should occur near the beginning of the audit, it has a role in most of the other steps. Others, such as financial assertions, may or may not apply. However, the noted framework represents a fair body of steps that should allow for the effective audit of applications.

> "The noted framework represents a fair body of steps that should allow for the effective audit of applications."

The remainder of this article details the first three steps: planning, determining objectives and mapping. The remaining steps will be detailed in this space in volume 4, 2012.

PLAN THE AUDIT

Planning the audit includes the consideration of all the relevant factors that frame the purpose of the audit. This consideration is necessary to properly plan the audit.

Consideration of Purpose

One of the key drivers of an application audit throughout the process is the conditions or circumstances by which the audit arose. That is, what is driving the need for the audit? Is it a regular audit plan? Is it an *ad hoc* audit? The need is usually directly associated with the primary objective of the audit. For example, if management wants to gain assurance that a new application is performing as designed, that fact will drive the audit objectives and plan.

Consideration of Risk

A second key factor and driver is consideration of risk associated with a particular audit, given the purpose of the audit that was determined previously. The IT auditor, or the audit team, needs to identify risk associated with the application and its associated data, sources, infrastructure and systems. To follow the previous example, possible risk scenarios include a lack of functionality (i.e., does not actually meet the information requirements), errors and/or bugs, an inability to properly integrate/interface with other applications or systems, data errors, and other similar risk.

Naturally, once the risk scenarios are properly identified, the IT auditor needs to assess the impact on the audit objectives, audit plan, audit scope and audit procedures. For instance, if lack of functionality is a risk, the IT auditor should examine the original information requirements, review tests, review a user acceptance document (if one exists), test the application and perform other similar procedures.

Consideration of the Control Environment

Usually, the audit plan should take into account the control environment surrounding the application, within the context of the audit purpose. If the primary purpose of the audit is auditing proper

functionality, the controls might be application development controls or systems development life cycle (SDLC) controls. In particular, controls for testing the application are important.

Consideration of Pre/Postimplementation
Sometimes the application audit involves a preimplementation application, but most likely, it will be a postimplementation situation. A preaudit tends to involve proprietary objectives, scope and procedures that are peculiar to that application and purpose. Postaudits often follow a general set of objectives (see the Determine Audit Objectives section).

Consideration of Scope
A very important consideration in planning is to establish the boundaries of scope. That means determining the relevant technologies and controls associated with auditing the applications, such as:
- Interfaces to other applications
- Source systems
- Target/destination systems
- Infrastructure or components thereof
- Databases
- Staging area/testing facility

Consideration of Competencies
As in all audits, one of the leaders or managers of the audit team will need to assess the competencies of the staff against the needs of the audit. For example, if the interface involves Oracle, it is possible that an expert in Oracle will be needed to properly audit the application.

DETERMINE AUDIT OBJECTIVES

The objectives are somewhat tied to the consideration of pre/postimplementation. As stated previously, the objectives tend to be proprietary for preimplementation applications. The same could be true for certain purposes. For others, the objective tends to be one of those that are typical for audits:

> "Mapping is one of the most effectual tools that the IT auditor has for any IT audit."

- Efficiency (related to development cost, operational performance, etc.)
- Effectiveness (related to meeting information requirements/functionality, the original authorization purpose, integration with other IT, operational performance, etc.)
- Compliance (laws and regulations, contractual, etc.)
- Alerts (if alerts are involved with the application)
- Financial reporting implications

MAP SYSTEMS AND DATA FLOWS

Mapping is one of the most effectual tools that the IT auditor has for any IT audit. In auditing applications, it is important to properly scope other IT that either affects or is affected by the application. Experts believe that mapping can assist the IT auditor in gaining a thorough understanding of the relevant technologies, the process, the controls and how they all fit together. It also empowers the IT auditor to best perform the steps in this framework from planning to reporting—that is, it has a comprehensive impact on the quality of the IT audit.

Items that should be considered in properly mapping the application include, among others:

- Relevant IT components (description)
- The business owners or business lines
- Change management policies and procedures
- The role and impact of vendors
- Business processes
- Controls
- Access and security administration

These factors can guide the IT auditor in creating the map, determining what should be on the map or determining what columns should be used in a spreadsheet that depicts the mapping. **Figure 1** shows one way to map the auditing of an application.

Documenting and mapping risk may involve items such as the risk, risk area, objective, reference, procedures, audit days, percent done, days to complete, scope of systems and notes. **Figure 2** shows a spreadsheet document that may be helpful in mapping risk, and demonstrates how such a map may be useful throughout the audit and may assist in managing the audit.

IT auditors need to map the process and data flow using conventional data flow diagrams (DFD), use cases, systems flowcharts or Unified Modeling Language (UML). A nonconventional diagram may serve as a better model for depicting processes and data flows. For example, the matrix in **figure 3** may serve as a better model because it incorporates the time/delivery as well as systems, processes and data flows.

The particular schematic shown in **figure 3** depicts controls in such a way as to make them clear and understandable, e.g., the automatic reconciliation, error-checking system (IT-dependent) and manual review of CRM data before the target data are uploaded as a control in the flow of data and processes.

This process/data flow framework might be more effective if it is presented using the system model vs. the timeline and process dimensions. Inputs include the source data, such as the source data for the middleware application. They

Figure 3—Mapping Processes and Data Flows

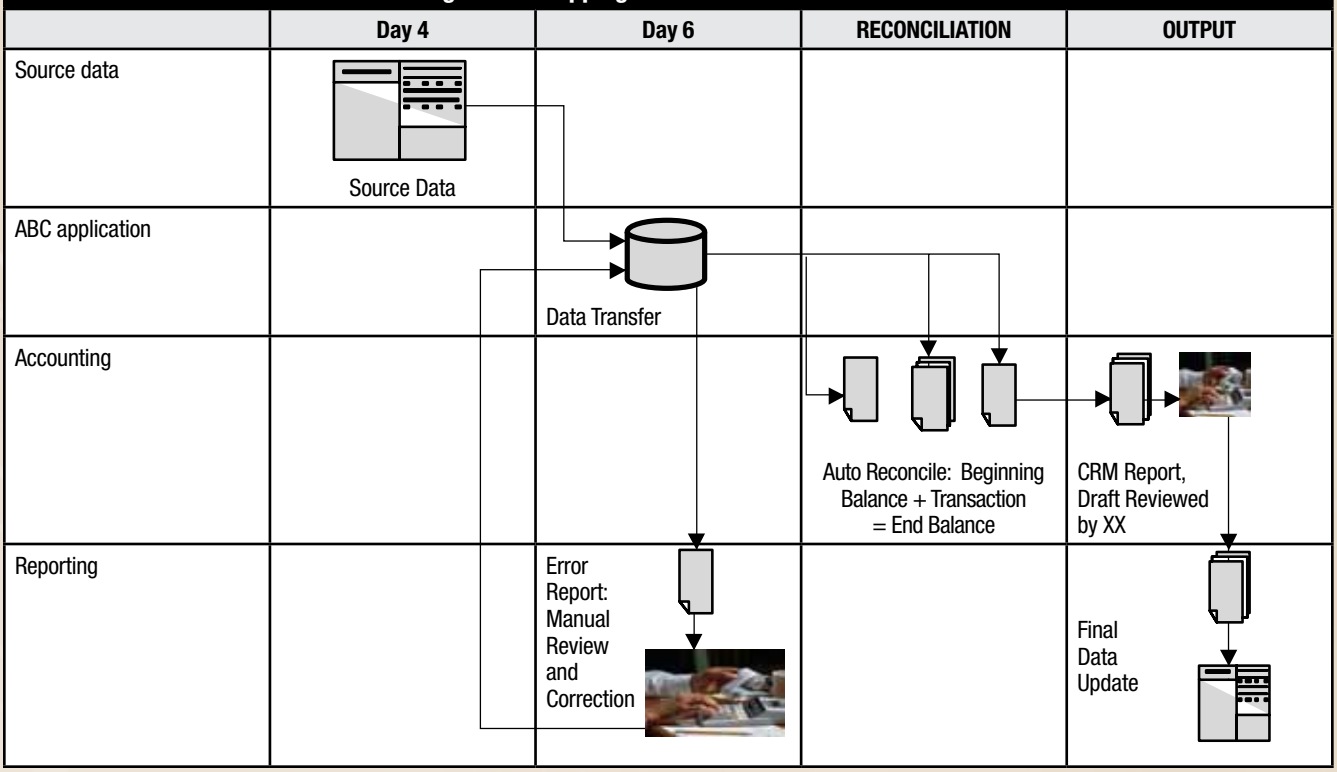

Figure 1—Mapping Example Using Spreadsheet, Part I					
IT	**Description**	**O/S**	**DBMS**	**DB Server**	**Data Location**
ABC App	Middleware designed to ...	N.A.	N.A.	XYZ	Birmingham
DEF App	CRM, target ...	Z/OS	DB2	Z mainframe	Nashville

Figure 1— Mapping Example Using Spreadsheet, Part II					
Developed	**Maintained**	**Owner**	**Access Admin**	**Change Control**	**Notes**
In-house	In-house	Sue	Active directory ...	Controls include ...	
Vendor	Vendor, SOC1/2 available	John	Security admin ...	Vendor ...	

Figure 2—Documenting and Mapping Risks, Part I					
Ref.	**Risk**	**Risk Area**	**Objective**	**W/P Ref.**	**Procedures**
1	Invalid, inaccurate or incomplete data may cause errors in reports or accounting.	Data integrity	Evaluate data integrity checks and controls between inputs and outputs.	CO.1.1	
2	Unauthorized or unintended changes to middleware may cause errors in reports/accounting.	Change management	Evaluate changes to the application for appropriate approvals, tests and segregation of duties (SoD).	CO.1.2	
3	Unauthorized access may cause unauthorized changes to middleware or target data, causing errors in reports/accounting.	Security	Evaluate logical access controls to the application and its folder.	CO.1.3	
4	Invalid, inaccurate or incomplete processing may cause errors in reports/accounting.	Operations	Evaluate processing and documentation for appropriate controls on development and support, and error identification and resolution.	CO.1.4	

Figure 2—Documenting and Mapping Risks, Part II					
Ref.	**Audit Days**	**Percent Done**	**Days to Complete**	**Scope of Systems**	**Notes**
1	0.5	100%	0	Middleware, stored procedures, views, CRM, DB2	
2	1.5	33%	1	Middleware	
3	1.0	0%	1	Active directory, middleware	
4	2.0	0%	2	INPUT: Source file PROCESS: Middleware OUTPUT: Target file/DB2, error report	

FIGURE 2—Documenting and Mapping Risks, Part III				
Ref.	**Inherent Risk**	**Control Risk**	**Assessed Risk**	**Notes**
1	High	Medium	Medium–High	To date, facts are ...
2	Medium	Low	Low	
3	High	Medium	Medium–High	
4	Medium	Low	Low–Medium	

also include intermediate data. Sources include the internal databases (DBs) and external providers of data—something not uncommon in data warehouses (DWs), for example.

The processing segment includes the processing function of the application (see **figure 3**, including automatic reconciliations and the error detection/correction routine). It also includes any process documents being created for the process functions. Certain processes are similar to those associated with DWs, such as ETL (extract, transform and load), which basically describe the process data go through to get into the DW from various sources. The ABC application example in **figure 3** is fairly consistent with ETL. Processing logic is of particular interest in auditing applications, as they are usually a chief component of data integrity and reliability.

Outputs include reports, screen information and other printed documents. Outputs also include the need to evaluate tools and templates being used to create those reports and screens.

CONCLUSION

This article explains the first portion of the framework. One of the key beneficial steps in this part of the application audit is to generate thorough and accurate maps or diagrams.

In the next issue (volume 4, 2012), the remaining steps of the framework will be explained. It is in these final steps that the bulk of the actual procedures and tests occur.

ADDITIONAL RESOURCES

Bitterli, Peter R., *et al*; "Guide to Audit of IT Applications," ISACA Switzerland Chapter, 2010

ERP Seminars, "Auditing Application Controls," 2008, *www.auditnet.org/docs/Auditing_Application_Controls.pdf*

SANS Institute, "The Application Audit Process," InfoSec Reading Room, *www.sans.org/reading_room/whitepapers/auditing/application-audit-process-guide-information-security-professionals_1534*