JOURNAL OF INFORMATION SYSTEMS
Vol. 22, No. 1
Spring 2008
pp. 47–62

Auditing System Development: Constructing the Meaning of "Systematic and Rational" in the Context of Legacy Code Migration for Vendor Incentives

A. Faye Borthick
Georgia State University

Paul L. Bowen
Florida State University

ABSTRACT: This simulation affords an opportunity for learning to audit system development for an accounting application. The simulation responds to the growing emphasis on controlling system development for complying with the internal control assurance requirements of Section 404 of the Sarbanes-Oxley Act of 2002 (U.S. House of Representatives). Because of the lack of detailed accounting standards for vendor incentives, learners have to construct a working definition of "systematic and rational" allocation of incentives in order to develop audit objectives and procedures. In the simulation, learners (1) develop objectives for auditing the specific project of migration of legacy code for vendor incentives and the system development for a group of projects, (2) design audit procedures to achieve the audit objectives, (3) execute the audit procedures by querying the databases, and (4) communicate objectives, procedures, and results in a report. The simulation is staged with conversations among audit staff members and the company's system development manager, databases containing application test data and program library transactions, and readiness questions. Although the databases are supplied in the form of Microsoft Access® files, the simulation can be worked with any database query tool. The simulation helps learners develop their capabilities for designing audit objectives and procedures for testing system development and for querying databases.

Keywords: audit simulation; audit software; database querying; integration of financial and IS auditing; query strategy; queries; system development audit; vendor incentives.

The authors are indebted to Chris Anderson, David Bateman, Elaine Mauldin, two anonymous reviewers, and participants at the 2007 AAA IS Section Midyear Conference and the 2007 UWCISA Symposium on Information Systems Assurance for helpful comments on earlier versions of the simulation.

Editor's note: This paper was awarded the Best Education Paper at the 2007 Information Systems Section Midyear Meeting.

I. LEARNING OBJECTIVES AND SIMULATION DESIGN

Learning Objectives

The learning objectives for this audit simulation are for students to learn to:

1. Develop objectives for auditing
   a. The migration of legacy code for vendor incentives
   b. System development for a group of projects
2. Design audit procedures to implement the audit objectives
3. Execute the audit procedures by querying the databases
4. Communicate objectives, procedures, and results in a report.

The simulation gives learners practice in auditing system development in two respects: auditing migration of code for vendor incentives and auditing system development for a group of projects. In the absence of specific accounting reporting standards for vendor incentives, learners have to construct for themselves an operational definition of "systematic and rational" with respect to booking entries for vendor incentives and apply it to develop audit objectives and procedures. The system development audit aspects include implementing queries for testing segregation of duties as explained in Hendrawirawan et al. (2007). Materials for staging the simulation include conversations among audit staff members and the company's system development manager, databases containing application test data and program library transactions, data attribute definitions, a report template, and readiness questions.

Working the simulation develops analytical and querying skills that accountants need to be productive with software tools for analyzing data (Elliott 2002; McCollum 2002; Fennel 2003; Jackson 2004). Like the due diligence audit of a fast-fashion retailer's inventory (Borthick and Curtis 2008), this simulation integrates financial and IS auditing. Unlike the practice in prior decades of IS auditors and general auditors working independently and having difficulty integrating their efforts (Stazyk 1992; Vendrzyk and Bagranoff 2003; Carmichael 2004), this simulation integrates the two skill sets by design in the following ways. First, developing audit objectives and designing audit procedures requires understanding the financial accounting application of vendor incentives, the application of "systematic and rational" as the accounting principle germane to allocation of expense and revenue amounts for vendor incentives, and the specific analyses that the available data afford. Second, executing the audit procedures requires expertise in querying exercised in the context of allocating vendor incentives. Third, communicating findings requires interpreting query results within a system development framework. These three integrations concern only the audit of migration of the vendor incentives code. An additional integration is needed to interpret the query results from examining program library transactions in the context of their implications for the integrity of application code generally. Thus, this simulation helps prepare students for auditing in the post-Sarbanes-Oxley (U.S. House of Representatives 2002) environment in which all auditors are expected to be skilled in using query tools and audit software to examine client systems.

Learning Theory: Constructing Situation Models

Like Borthick and Curtis' (2008) due diligence audit, this simulation invokes the theory of situation models, "in which learners construct their own mental models of a situation by making inferences and elaborations as they encounter new information" (11). In the theory of situation models, comprehending a problem entails constructing a situation model representing it (Johnson-Laird 1983; van Dijk and Kintsch 1983; Gernsbacher 1997;

Graesser et al. 1997; Zwaan and Radvansky 1998). In the simulation, successive utterances in the conversations reveal new information about vendor incentives, the company's approach to allocating them, and the company's system development practices. The data files reveal other information about the application and system development. Thus, the insights one needs to work the simulation emerge as one considers the conversations and the data files and makes inferences and elaborations that build one's situation model (Gernsbacher 1997; Zwaan and Radvansky 1998). In this theory, building mental situation models is how one prepares for acting in analogous future situations (Zwaan and Radvansky 1998; Barsalou 1999). In this simulation, learners practice building mental models like those they will need in subsequent engagements requiring analysis and querying skills.

Because easy-to-understand situations do not require inferences and elaborations (Myers et al. 1987) that lead to robust situation models, the simulation is staged with an unstructured, ambiguous situation featuring cognitive conflict tasks. Because there is no one right way to resolve ambiguities and conflicts, cognitive conflict tasks have no single correct answers (Laughlin 1980; McGrath 1984). In the simulation, there are overall conclusions to be reached, but there are multiple analysis paths that will reach them. Regardless of the analysis path, the learners' problem is to justify their conclusions through the most complete, plausible, or compelling understanding of the situation on the basis of the available evidence (King and Kitchener 1994).

Learners will likely notice that the simulation makes more demands on them than the intellective tasks they are probably accustomed to completing. Intellective tasks are those with demonstrably correct answers (Laughlin 1980; McGrath 1984). We believe the cognitive conflicts inherent in the situation are required to prompt learners to make the inferences required to construct situation models that will be useful to them later (Zwaan et al. 1995; Zwaan and Radvansky 1998). Although the learning process may be more effortful in the short run, the better situation models will be more useful in the long run (Barsalou 1999).

Simulation Design

Part 1: Readiness Questions

Five multiple-choice questions with feedback for each response (in the teaching notes) are provided for assessing learners' understanding of the Organofood situation and its representation in the data files. The questions can be administered any time after students have become familiar with the simulation materials. If staged in a learning management system, the response-level feedback can be provided automatically. As in Borthick and Curtis (2008), the readiness questions afford a potential *Saving Sergeant Pabletti* epiphany for learners (Prensky 2001), i.e., a wakeup call to learners indicating whether they really understand the business situation or have only deceived themselves that they understand it.

Part 2: The Simulation

In Part 2, learners (1) develop audit objectives, (2) design audit procedures for execution on data files, (3) implement audit procedures through data querying, and (4) communicate objectives, procedures, and results. The simulation suggests a reporting format with a report template. For a more formal presentation, instructors could require students to prepare a memo to the engagement partner summarizing their results. Part 2 materials can be staged on a website, which permits the opening conversation to offer links to supporting materials.

To make the simulation tractable for a learning experience, the databases were designed with just enough tables and attributes to represent the business context of vendor incentives

for a grocer. The data volumes are large enough to require manipulation with software such as a query manager.

We believe this audit simulation enhances the existing literature. Like Borthick and Curtis' (2008) due diligence audit, this simulation requires students to integrate analysis and query skills to design and execute audit procedures with software. It differs from Borthick and Curtis (2008) in offering a different accounting application (vendor incentives), in incorporating a system development audit, and in not requiring business process modeling (BPMI 2004; White 2004). Although the Norwood case (Gelinas et al. 2001) has integrative aspects, it was designed as an introduction to computer-assisted auditing techniques. Although Hunton et al.'s (2003) audit case requires use of audit software to audit data files, those case materials specify the audit objectives and procedures. In contrast, our audit simulation requires learners to make sense of an unstructured, ambiguous business situation and to develop their own audit objectives and procedures.

Student Reaction to the Audit Simulation

Students liked the readiness questions because they allowed them to calibrate their understanding of the audit situation early in their work on the simulation. After seeing the item-level response feedback, some students realized that they needed to understand the business situation more thoroughly in order to reason strategically about data relationships (Wilks and Zimbelman 2004).

In the first days of working on the simulation, some students have been frustrated, primarily because a clear understanding of "systematic and rational" allocation of vendor incentives seemed to elude them. It was a case of the more they read and studied about "systematic and rational," the more confused they became. As they thought longer about "systematic and rational" applied to vendor incentives, however, their frustration vanished.

Some students have said that, although the applications differed, the situation with migrating legacy code mirrored some that they had encountered on the job or vicariously through conversations with others. These students worked the simulation enthusiastically, looking for ways to sense when a development project might be flawed. These students appreciated the opportunity to test whether they were able to analyze a complicated situation and make sense of it (Weick 1995, 2001).

Although they may not have identified all the nuances, most students have been able to reach reasonable conclusions about the company's allocation of vendor incentives and the integrity of system development. Furthermore, students have been eager to participate in debriefings after seeing their marked audit reports because they were interested in how to identify all the salient issues.

Faculty Reaction to the Audit Simulation

Faculty members using the case were pleased to have one focusing on Sarbanes-Oxley issues, especially Section 404, that required analysis of transaction data. First, the case requires students to wrestle with ambiguity in laws and regulations. Dealing with these ambiguities challenges many students' perceptions of accounting as merely the application of well-defined rules that, at most, require accountants to select the most favorable rule from a set of clearly defined alternatives. Second, the case requires the innovative use of information retrieval tools to extract information about the extent to which the client is likely to meet its incentive contracts. This aspect of the case forces learners to use current data to forecast future economic activity and to think of reasons why future economic activity might differ from that already experienced. Third, the case requires integration and

interpretation of laws, regulations, business contracts, and incomplete economic data to evaluate the client's contractual performance and compliance with applicable regulations and laws.

One faculty member commented that although the case enables substantial learning for conducting IT audits, guiding students through the case poses some challenges to the instructor. First, students will struggle if their initial proficiency with the information retrieval tool is weak. Second, a significant portion of students experience difficulty coping with a less well-defined task than they are accustomed to encountering. Striking a balance between helping students overcome their current obstacle without giving them too much assistance requires judgment on the part of the instructor. Third, perhaps the most challenging issue, the instructor must help students break their mindset of accounting as a reasonably straightforward application and/or selection of clearly defined and easy-to-apply rules.

II. IMPLEMENTATION GUIDANCE

Matching to Courses and Assuring Prerequisite Skills

The simulation is suitable for information systems (IS) auditing courses and auditing courses for developing or assessing learners skills in developing audit objectives, designing audit procedures, executing audit procedures by querying databases, and communicating results. It could be used in accounting information systems (AIS) courses if the instructor provided audit objectives to students. The simulation is appropriate for an individual or a team assignment. According to student self-reports of time spent, the simulation takes between 8 and 16 hours per team member outside of class, where most of the hours are devoted to querying. In our experience, reported hours tend to increase with inadequately developed audit objectives and deferment of tasks too close to the due date.

Prerequisite skills for working the simulation include (1) querying proficiency, (2) familiarity with system development, and (3) some experience developing audit objectives and procedures. These skills are typically acquired in different courses. Some cases that develop querying proficiency include Borthick and Jones' (2005) warranty call center case and Borthick and Jones' (2007) case on analyzing wireless phone service. The system development familiarity should be at the level of a text such as Romney and Steinbart (2006). Information about the desirability and feasibility of using queries to test segregation of duties could be supplied with Hendrawirawan et al. (2007). If students have not had experience developing audit objectives and procedures, Borthick and Kiger's (2003) e-ticket travel revenue audit case would provide it. All the sources needed to make sense of "systematic and rational" allocation of vendor incentives are cited in the conversations, the most important ones of which are Bryan-Low (2003) and Zimmerman and Callahan (2003). No prior knowledge of accounting for vendor incentives is required.

Selecting the Software Tool

Students have worked the simulation using Query-by-Example (QBE) and Structured Query Language (SQL) interfaces to a database query manager and the audit software Interactive Data Extraction and Analysis (IDEA™). Any database query manager or audit software program, e.g., Audit Command Language (ACL™), could be used (McCombs and Sharifi 2004).

Realizing Learning Gains through Collaboration

The simulation can be assigned to individual learners or to teams of learners. Because team collaboration enables students to learn from each other (Rogoff 1998; Borthick et al.

2003) and replicates the practice of business professionals collaborating on problems requiring novel approaches (Schrage 1990; Raelin 1997), we believe it promotes greater learning than working individually. Individual assignment would be ideal, however, to assess individual learners' expertise.

Staging the Simulation

We stage the simulation on a password-protected course website where the first link is to a page with the conversation and the requirements. The conversation page has links to (1) pages containing the data attributes, authorization for the code migration, and report format; (2) database files; (3) pages at www.extremeprogramming.com; and (4) referenced articles such as those in *Computerworld*, *InformationWeek*, and *CFO Magazine*. With the assistance of library staff, instructors may be able to add links to their institution's electronic periodical holdings to the articles from the *Wall Street Journal*. We prefer web staging for the materials because it facilitates student access to them, especially the ones they need in digital form such as the report format and the database files. Web pages for staging the case on the web are available through the Teaching Notes.

Students answer the readiness questions in the quiz tool of a learning management system that grades the questions automatically and reveals the feedback for the students' responses. The quiz tool has a setting that allows students' responses to be marked without identifying the best response, which permits students to answer the questions again.

Conducting the Simulation

We believe a good way to launch the simulation is to give students a few days to think about the simulation and then respond to their questions by posing questions that help students learn to answer their questions for themselves. We believe it is helpful to hold these discussions after they read the conversations, before and after they complete the readiness questions, and during the period they are working on the audit. Students can use the report form as both a guide to the work to be done and as an organizing tool to capture interim outcomes while they are still fresh in their minds.

We have assigned the simulation so that students work on it over a period of about three weeks. On day one, students receive the assignment. A week later in class, students answer the readiness questions, get feedback on their responses, and are encouraged to ask any questions they wish about the audit. About half of two weeks' worth of class time is devoted to conversations about the audit. We let students direct the conversations because we believe they benefit from taking ownership of the audit. During the weeks students are working on the audit, we encourage them to post questions to an electronic discussion board topic for the audit, and they take advantage of the opportunity to receive answers (or post further questions) from other students and the instructor. In the first class session after marked audits have been returned to students, we conduct a short debriefing on the audit.

Posing a Single or a Two-Part Project

The audit report can be completed as a single project or as a two-part project. The single project approach has the advantage of students experiencing the audit as an integrative whole, as auditors on the job would experience it. The two-part approach has the advantage of minimizing student uncertainty about whether they are proceeding in productive directions. Independent learners seem to thrive on the single project approach while learners that are more hesitant appreciate the two-part approach. For a two-part approach, instructors would have students develop audit objectives in the first part and design audit

procedures, execute audit procedures, and communicate results in the second part. Before students begin the second part, instructors would ensure that students had complete audit objectives. Instructors could grade and return students' audit objectives or simply make a complete set of audit objectives available to students. For uses where the emphasis is on developing query proficiency, e.g., in an AIS course, student requirements would begin with designing audit procedures for a set of instructor-supplied audit objectives.

Closing the Simulation

A debriefing session is helpful because it allows instructors to ensure that all students understand the major issues and their resolutions. Students could drive the debriefing with presentations, or the instructor could lead a discussion. It is helpful to insist that students reflect on what they learned from the simulation and how that might be useful to them in the future. An especially useful skill is the ability to make sense of unstructured, ambiguous situations as the first step in completing work projects.

III. THE SIMULATION

The Engagement

Scene: Auditors planning a two-part system development audit. One part is an audit of the migration of legacy code for vendor incentives for Organofood, a grocer, to the company's ERP system, and the other part is an audit of system development generally at Organofood.

Betani (IS auditor at Jobert and Turin, LLP): "You lucky scoundrel! You get to do a development audit[1] that doesn't appear, at first glance, to be a candidate for the system development project failure-of-the-year award."

Alexie (IS auditor at Jobert and Turin, LLP): "Is this my reward for all the other tough ones?"

Betani: "Maybe. Organofood is moving its food purchases and store sales into the ERP it installed last year. Emil has already signed off on purchases and sales. That just leaves vendor incentives for you to deal with."

Alexie: "We're new at Organofood. What gives?"

Betani: "Fallout from SarbOx, courtesy of which Organo's external auditor can't do any system development auditing for it. Organo doesn't have any technical internal audit staff so we're it! The internal audit director is trying to hire a CISA,[2] but that may take a while."[3]

Alexie: "Taking in each other's laundry, eh? Show me the duds. Maybe they won't be too dirty."

Betani: "Here's the test data [OrganofoodTestdata.mdb] with data attributes [Figure 1], whose results match those from the legacy system. That's Emil's contribution to your part."

---

[1] A systems development audit "should ensure that systems under development meet the objectives of the organization, satisfy user requirements, and provide efficient, accurate, and cost-effective systems and applications. The audit should also ensure that these systems are written, tested, and installed in accordance with generally accepted standards for systems development" (Gallegos, F., S. Senft, D. P. Manson, and C. Gonzales. 2004. *Information Technology Control and Audit*, 577. Boca Raton, FL: Auerbach Publications.

[2] Certified Information Systems Auditor certification, sponsored by the Information Systems Audit and Control Association (ISACA). Available at: http://www.isaca.org.

[3] Hoffman, T. 2004. IT auditors coveted, hard to find: Companies compete for needed skills as Sarb-Ox deadlines near. *Computerworld* (May 3): Available at: http://www.computerworld.com/governmenttopics/government/policy/story/0,10801,92819,00.html.

FIGURE 1
Organofood Data Attributes: Testdata

Table/Attribute[a]	Explanation
account: Accounts in the general ledger	
accountID	Unique identifier for an account
title	Account title
generalLedger: General ledger transactions related to vendor incentives	
ID	Unique identifier for a transaction
accountID	Unique identifier for an account
vendorID	Unique identifier for a vendor
SKU	Stock keeping unit, unique identifier for an item
incentiveCode	Code indicating the type of vendor incentive, if any
date	Date transaction was entered into the general ledger
amount	Amount of the transaction
incentive: Terms of vendor incentives	
vendorID	Unique identifier for a vendor
incentiveCode	Code indicating the type of vendor incentive
SKU	Stock keeping unit, unique identifier for an item
incentiveUnit	Sales level in units, if applicable, required to attain the incentive
reductionPercent	Percent vendor-invoiced price reduced if incentive level attained
incentiveDays	Number of days over which incentive is to be attained
startDate	Beginning date for calculating incentive attainment
incentiveCode: Incentive codes by type	
incentiveCode	Code indicating the type of vendor incentive
title	Short title for an incentive type
explanation	Explanation of the incentive type
invoice: Vendor invoices	
purchaseID	Unique identifier for a purchase
invoiceDate	Date of an invoice from a vendor
vendorID	Unique identifier for a vendor
amount	Total amount for an invoice
purchase: Detail of a purchase	
purchaseID	Unique identifier for a purchase
SKU	Stock keeping unit, unique identifier for an item
caseOrderQty	Number of cases ordered
unitsPerCase	Number of units in a case
caseCost	Cost of a case
extension	Product of caseOrderQty and caseCost
SKU: Information about SKUs	
SKU	Stock keeping unit, unique identifier for an item
vendorID	Unique identifier for a vendor
markup	Markup proportion over cost to yield sales price
storeSales	
accountID	Unique identifier for an account
SKU	Stock keeping unit, unique identifier for an item
unitQty	Unit quantity sold
amount	unitQty times the sales price of the units
salesPeriodEnd	Ending date of sales period

[a] Table names and primary keys in bold.

Alexie: "Wait a minute. Why didn't Organo do purchases/sales and vendor incentives in the original ERP installation?"

Betani: "Think it was because they took the six-month mantra[4] seriously—divide projects into sub-projects that can be implemented in six months. Really reduces implementation risks in converting from legacy systems.[5] For the first ERP release, they implemented what they thought they could handle in the short run and deferred the rest."

Alexie: "And now they're picking up the pieces."

Betani: "The old system had a reputation of being a kluge when it came to handling vendor incentives.[6] Needed a lot of handholding. Wasn't a problem until the Wal-Marts of the world[7] made the grocery business more competitive, which thinned profit margins."[8]

Alexie: "What's that have to do with incentive handling in the accounting system?"

Betani: "As the price pressure increased, retailers and distributors demanded more incentives from food manufacturers, who complied."[9]

Alexie: "Because they didn't want to wake up one day selling only to Wal-Mart?"

Betani: "You got it!"

Alexie: "Something's missing still. What really prompted the audit?"

Betani: "The new CFO is uncharacteristically tech savvy—wants a blessing on Organo's systems. For vendor incentives, that means systematic and rational[10] allocation. Pervasively, that means controlled system development. The audit is Step 1 in Organo's SarbOx compliance."

Alexie: "Isn't Organo private?"

Betani: "Yep, but they're looking like a really good acquisition candidate.[11] It'd be a perfect fit for a chain grocer needing instant presence in the organic market, and easy compliance would up the price. The CFO is ambitious!"

Alexie: "Okay. I'll see whether systematic, rational, and controlled apply."

Betani: "Don't forget to note which system development controls should be documented.[12] If we get the contract to do the SarbOx documentation work, it would be helpful to have a starter list."

Later, at Organofood ...

Alexie: "At last, we get to talk."

---

[4] Keen, P. G. W. 2000. Six months—Or else. *Computerworld* (April 10): Available at: http://www.computerworld.com/news/2000/story/0,11280,44381,00.html.

[5] Whiting, R. 2003. Money Machines: Replacing trusted legacy apps has risks and up-front costs. *InformationWeek* (November 3): Available at: http://www.informationweek.com/story/showArticle.jhtml?articleID=15800298.

[6] Zimmerman, A., and P. Callahan. 2003. Bonuses paid distributors are focus of attention: Fleming inquiry upgraded. *Wall Street Journal* (February 26): A2.

[7] Taylor, D. A. 2003. Supply chain vs. supply chain. *Computerworld* (November 10): Available at: http://www.computerworld.com/industrytopics/manufacturing/story/0,10801,86908,00.html.

[8] Ellison, S., and S. Kilman. 2003. SEC Expands food-industry probe—Kraft, Dean and Frito-Lay could face civil lawsuit in revenue-inflation case. *Wall Street Journal* (November 6): A3.

[9] Zimmerman, A., and A. Raghavan. 2003. Special deals for distributors draw scrutiny. *Wall Street Journal* (November 7): B1.

[10] Bryan-Low, C., and Schroeder, M. 2003. Questioning the books: Deloitte is familiar with supplier-rebate issues. *Wall Street Journal* (February 27): A7.

[11] Katz, D. M. 2003. Rites of privacy: With the dust settling on Sarbox compliance in the public sector, eyes turn to private companies. *CFO* (November 1): Available at: http://www.cfo.com/article.cfm/3010715.

[12] Hoffman, T. 2004. IT auditors seek Sarb-Ox guidance. *Computerworld* (April 12): Available at: http://www.computerworld.com/databasetopics/data/story/0,10801,92100,00.html.

Derilo (system development manager at Organofood): "Yes, it's been hectic but then that's the lot of developers trying to avoid being off-shored."
Alexie: "How likely is that for you and your crew?"
Derilo: "Dunno, but the rhetoric is getting louder. Management said we were at risk, so we're trying to be as effective as we can."
Alexie: "That's incentive! Are you doing anything different to keep your work off the boat?"
Derilo: "Well, we're spending less time trying to pull requirements out of users. Now, we make a formal request for specs and don't start on a project until we get approved ones. That tactic seems to have cut down on front-end discussions, which lets us start programming sooner."
Alexie: "Are users just as satisfied as before?"
Derilo: "Too soon to tell. They seem to take their responsibilities more seriously, even though they're as overworked as we are. The projects since the change have mostly been small ones, which makes it easier for users to freeze specs."
Alexie: "Did you make changes in version or configuration management to be more efficient?"
Derilo: "Not really, but we were just getting good with XP. Management thought pair programming[13] was wasteful, so we quit. What a bum legacy—programming as solitary activity."
Alexie: "X what?"
Derilo: "Short for 'Extreme Programming,'[14] a programming methodology for producing high-quality code through practices like planning based on user stories, coding unit tests first (as the definition of requirements), programming in pairs, keeping all code segments integrated, and refactoring as needed."
Alexie: "How often did you synch[15] the code in XP?"
Derilo: "At least once a day. After 'offshore' burrowed into our vocabulary, programmers cut back on the frequency of code check in to when they think they're done with it—typically a week to a month."
Alexie: "When did you quit XP?"
Derilo: "Let's see—with the new batch of programs starting in early September."
Alexie: "Tell me about vendor incentives. Is there anything unusual about the app?"
Derilo: "No, it's been pretty stable over the years. The number of transactions has grown, but the code's hardly changed. What we're doing now is porting the legacy code into the ERP [Figure 2]."
Alexie: "Did the legacy code require a lot of hand holding?"
Derilo: "Not really, although I remember tales of new operations managers not understanding that the incentives didn't show up on their P&Ls until the accounts were closed to cost of goods sold at the end of the period. That brouhaha occurred before I came."
Alexie: "So it was just lack of user understanding?"
Derilo: "Yeah—standard stuff."
Alexie: "Where can I get to program library transactions?"

---

[13] Wells, D. 1999. *Pair Programming.* Available at: http://www.extremeprogramming.org/rules/pair.html.
[14] Wells, D. 2006. *Extreme Programming: A Gentle Introduction.* Available at: http://www.extremeprogramming.org/index.html.
[15] Cusumano, M. A., and R. W. Selby. 1997. How Microsoft builds software. *Communications of the ACM* 40 (6): 53–61.

FIGURE 2
Organofood: Program Authorization

Organofood Program Authorization

Project	ERP migration for purchases/sales of store inventory
Project number	258484
Component	Vendor incentives
Component number	524383
Purpose	Port vendor incentives code from legacy system into ERP
Justification	Replace legacy code for vendor incentives with ERP code that enables real time posting. Upon completion, vendor incentives will flow through to GL without intervention.
Specifications	1. Duplicate existing functions from legacy system in ERP for the following vendor incentives except as noted in 2 below: 1. Promotion/advertising fees 2. Slotting fees 3. Volume discounts 2. Flow transactions to GL in real time to replace batch interface at period ends.
Authorization	
User name	Signature
A. J. Wellborn	
Sys Admin name C. K. Pralahad	Signature 

Derilo: "I'll get an extract [OrganofoodProgLibrary.mdb] with data attributes [Figure 3] for the project, including explanation of our development milestones. If this isn't enough, let me know. Here's my card in case you have questions."
Alexie: "Thanks! I'll be done as soon as I can."

Required

Part 1: Readiness Questions

After you have become familiar with the Organofood situation as represented by the conversations, cited materials, and data, answer the following questions to assess your

FIGURE 3
Organofood Data Attributes: ProgLibrary

Table/Attribute [a]	Explanation
libraryTransaction: Program library transactions	
transactionID	Unique identifier for a program library transaction
projectID	Unique identifier for a program
componentID	Unique identifier for a component within a program
date	Date of transaction
stageCode	Development stage completed in the transaction
personID	Unique identifier for the person checking in the component
stageCode: Definition of stageCodes	
stageCode	Code for a development stage
definition	Definition of a stageCode

[a] Table names and primary keys in bold.

readiness to begin the audit. The questions illustrate the kinds of thinking that will enable you to understand the situation in order to develop productive audit objectives. Select the best choice for each question based on this audit. The questions are independent of each other.

1. In the context of this audit, the best characterization of "systematic and rational" allocation of incentives would most likely involve or require:
   a. a basis for believing that complying with vendors' terms for incentives was probable
   b. evidence of having sold sufficient volumes to have earned the incentive payments
   c. reference to accounting rules defining the meaning of "systematic and rational"
   d. deferral of incentives booking until target sales levels have been achieved for the period
   e. verification of the absence of fraudulent intent in booking incentives before they are earned
2. Implications of purchasers of foodstuffs booking vendor incentives immediately are that the purchasers believe that they will:
   a. experience increasing demand over time
   b. be able to return unsold goods to vendors
   c. sell enough to warrant the vendor incentives
   d. experience wide swings in demand over time
   e. experience mostly constant demand over time
3. If appropriate data were available, detecting whether Organofood had over-booked promotion/advertising allowances from a vendor would require, for the period, comparing:
   1. The sum of allowances booked for promotion/advertising to the sum of vendor payments and discounts for promotion/advertising
   2. The sum of promotion/advertising expenses by vendor to vendor payments for promotion/advertising

   3. The sum of allowances booked for promotion/advertising by vendor to the sum of vendor payments and discounts for promotion/advertising
   4. The sum of vendor payments and discounts for promotion/advertising to expenses incurred for promotion/advertising
   5. The sum of vendor payments and discounts for promotion/advertising by vendor to expenses incurred for promotion/advertising

   a. 1 and 4
   b. 1 and 5
   c. 2 and 3
   d. 2 and 4
   e. 3 and 5

4. The data in OrganofoodProgLibrary.mdb permit verifying:
   a. completeness of requirements
   b. effectiveness of user participation
   c. completeness of quality assurance
   d. the adequacy of the SDLC method
   e. separation of duties in development
5. The best evidence for incompatible system development duties having been separated would be finding that:
   a. staff for each development stage were located in different physical facilities
   b. no person participated in more than one stage
   c. no person participated in more than one stage for the same program
   d. no person participated in both the requirements and programming stages
   e. no person participated in more than one stage for the same program component

Part 2: Analysis

1. Develop audit objectives for auditing:
   a. The specific project of Organofood's migration of legacy code for vendor incentives into an ERP system
   b. System development, including system development process effectiveness, for all projects for which program library transactions are available
2. Design audit procedures to achieve the audit objectives.
3. Execute the audit procedures by querying the data.
4. Communicate audit objectives, audit procedures, results from executing procedures, findings, recommendations, data limitations, lessons learned, and time spent by completing the report in Figure 4.

FIGURE 4
Organofood System Development Audit Report

Audit Objective Statement of the audit objective		**Audit Procedure** Explanation of the audit procedure for implementing the audit objective in terms of the data attributes in the database	**Results from Executing Queries** For each procedure: 1. The name(s) of the query(ies) that executes the audit procedure 2. A statement of the query results 3. An explanation of the meaning of the query results in the context of the audit	1. Findings from querying the data 2. Recommendations, if any 3. Data limitations, if any
Financial Accounting Objectives				
1				
2				
3	[add/delete rows as needed]			
System Development Objectives Including Development Effectiveness				
1				
2				
3	[add/delete rows as needed]			

Lessons Learned (e.g., insights/strategies that could be applied in other audit/analysis situations)		
#	Lesson	Explanation
1		
2		
3	[add/delete rows as needed]	

Time Log (hours spent on this engagement, by person)		
#	Auditor Name	Total Hours
1		
2		
3	[add/delete rows as needed]	

IV. TEACHING NOTES

The Teaching Notes for this case include:

1. Grading guidance
   a. Part 1: Solutions and item feedback for readiness questions
   b. Part 2: Analysis of querying with results presented in a completed report template
   c. Explanations of the motivation for and development of audit objectives
   d. Point allocation
2. A link to a zip file containing the following files:
   a. Access *.mdb files containing the data for student querying: OrganofoodTestdata.mdb and OrganofoodProgLibrary.mdb
   b. Access *.mdb files containing the data and QBE queries for instructor use that support the part 2 analysis of querying: OrganofoodTestdataQueries.mdb and OrganofoodProgLibraryQueries.mdb

c. HTML files of the case text to enable instructors to stage the case on a website. Any websites used for this purpose should be password-protected, and the passwords should be given only to students enrolled in courses using the case.

V. SUMMARY

This case simulates a system development audit in which students query supplied databases to audit (1) the migration of legacy code into an ERP for vendor incentives and (2) system development process compliance for a group of projects. In order to develop audit objectives and design audit procedures, learners have to make sense of an unstructured, ambiguous situation staged with conversations among audit staff members and the company's system development manager. The simulation requires the integration of financial auditing and IS auditing, which has been increasingly required in the post-Sarbanes-Oxley (U.S. House of Representatives 2002) environment for assuring the integrity of internal control.

TEACHING NOTES

Teaching Notes are available only to full-member subscribers to the Journal of Information Systems through the American Accounting Association's electronic publications system at http://www.atypon-link.com/action/showPublisherJournals?code=AAA. Full member subscribers should use their personalized usernames and passwords for entry into the system where the Teaching Notes can be reviewed and printed.

If you are a full member of AAA with a subscription to the Journal of Information Systems and have any trouble accessing this material, please contact the AAA headquarters office at office@aaahq.org or (941) 921–7747.

REFERENCES

Barsalou, L. W. 1999. Language comprehension: Archival memory or preparation for situated action? *Discourse Processes* 28 (1): 61–80.

Borthick, A. F., D. R. Jones, and S. Wakai. 2003. Designing learning experiences within learners' zones of proximal development (ZPDs): Enabling collaborative learning on-site and on-line. *Journal of Information Systems* 17 (1): 107–134.

———, and J. E. Kiger. 2003. Designing audit procedures when evidence is electronic: The case of e-ticket travel revenue. *Issues in Accounting Education* 18 (3): 275–290.

———, and D. R. Jones. 2005. Analyzing a potential warranty call center budget overrun: Using database queries to solve business problems. *Journal of Information Systems* 19 (1): 97–111.

———. 2007. Creating a business process diagram and database queries to detect billing errors and analyze calling patterns for cell phone service. *Journal of Information Systems* 21 (1): 107–122.

———, and M. B. Curtis. 2008. Due diligence on fast-fashion inventory through data querying. *Journal of Information Systems* 22 (1): 77–93.

Business Process Management Initiative (BPMI). 2004. *Business Process Modeling Notation (BPMN) Version 1.0.* Aurora, CO: BPMI. Available at: http://www.bpmi.org/downloads/BPMN-V1.0.pdf.

Bryan-Low, C., and M. Schroeder. 2003. Questioning the books: Deloitte is familiar with supplier-rebate issues. *Wall Street Journal* (February 27): A7.

Carmichael, D. R. 2004. The PCAOB and the social responsibility of the independent auditor. *Accounting Horizons* 18 (2): 127–133.

Elliott, R. K. 2002. Twenty-first century assurance. *Auditing: A Journal of Practice and Theory* 21 (1): 139–146.

Fennel, N. I. 2003. Make your data pay. *Internal Auditor* (June): 55–59.

Gelinas, U. J., E. S. Levy, and J. C. Thibodeau. 2001. Norwood Office Supplies, Inc: A teaching case to integrate computer-assisted auditing techniques into the auditing course. *Issues in Accounting Education* 16 (4): 603–636.

Gernsbacher, M. A. 1997. Two decades of structure building. *Discourse Processes* 23: 265–304.

Graesser, A. C., K. K. Millis, and R. A. Zwaan. 1997. Discourse comprehension. *Annual Review of Psychology* 48: 163–189.

Hendrawirawan, D., H. Tanriverdi, C. Zetterlund, H. Hakam, H. H. Kim, H. Paik, and Y. Yoon. 2007. ERP security and segregation of duties audit: A framework for building an automated solution. *Information Systems Control Journal* 2: 46–49.

Hunton, J. E., S. M. Bryant, and N. A. Bagranoff. 2003. *Core Concepts of Information Technology Auditing*. Somerset, NJ: Wiley.

Jackson, R. A. 2004. Get the most out of audit tools. *Internal Auditor* (August): 36–47.

Johnson-Laird, P. N. 1983. *Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness, Cognitive Science Series; 6*. Cambridge, MA: Harvard University Press.

King, P. M., and K. S. Kitchener. 1994. *Developing Reflective Judgment: Understanding and Promoting Intellectual Growth and Critical Thinking in Adolescents and Adults*. San Francisco, CA: Jossey-Bass.

Laughlin, P. R. 1980. Social combination processes of cooperative, problem-solving groups as verbal intellective tasks. In *Progress in Social Psychology*, edited by M. Fishbein. Hillsdale, NJ: Erlbaum.

McCollum, T. 2002. Data analysis with SQL. *Internal Auditor* (August): 25–27.

McCombs, G. B., and M. Sharifi. 2004. Utilization of generalized audit software in an information systems auditing course. *Information Systems Control Journal* 6: 37–38.

McGrath, J. E. 1984. *Groups: Interaction and Performance*. Englewood Cliffs, NJ: Prentice Hall.

Myers, J. L., M. Shinjo, and S. A. Duffy. 1987. Degrees of causal relatedness and memory. *Journal of Memory and Language* 26: 453–465.

Prensky, M. 2001. *Digital Game-Based Learning*. New York, NY: McGraw-Hill.

Raelin, J. A. 1997. A model of work-based learning. *Organization Science* 8 (6): 563–578.

Rogoff, B. 1998. Cognition as a collaborative process. In *Handbook of Child Psychology, Volume 2: Cognition, Perception, and Language*, edited by D. Kuhn and R. S. Siegler. New York, NY: Wiley.

Romney, M. B., and P. J. Steinbart. 2006. *Accounting Information Systems*. Tenth edition. Upper Saddle River, NJ: Prentice Hall.

Schrage, M. 1990. *Shared Minds: The New Technologies of Collaboration*. New York, NY: Random House.

Stazyk, R. E. 1992. Information systems auditing in the 1990s: A business approach. *Internal Auditing* 8 (1): 3–8.

U.S. House of Representatives, Committee on Financial Services. 2002. Sarbanes-Oxley Act of 2002. Public Law No. 107-204. Washington, D.C.: Government Printing Office.

van Dijk, T. A., and W. Kintsch. 1983. *Strategies of Discourse Comprehension*. New York, NY: Academic Press.

Vendrzyk, V. P., and N. A. Bagranoff. 2003. The evolving role of IS audit: A field study comparing the perceptions of IS and financial auditors. *Advances in Accounting* 20: 141–163.

Weick, K. E. 1995. *Sensemaking in Organizations*. Thousand Oaks, CA: Sage Publications.

———. 2001. *Making Sense of the Organization*. Oxford, UK: Blackwell.

White, S. A. 2004. *Introduction to BPMN*. Aurora, CO: BPMI. Available at http://www.bpmi.org/downloads/Introduction_to_BPMN89.pdf.

Wilks, T. J., and M. F. Zimbelman. 2004. Using game theory and strategic reasoning concepts to prevent and detect fraud. *Accounting Horizons* 18 (3): 173–184.

Zimmerman, A., and P. Callahan. 2003. Bonus paid distributors are focus of attention. *Wall Street Journal* (February 26): A2.

Zwaan, R. A., M. C. Langston, and A. C. Graesser. 1995. The construction of situation models in narrative comprehension: An event-indexing model. *Psychological Science* 6 (5): 292–297.

———, and G. A. Radvansky. 1998. Situation models in language comprehension and memory. *Psychological Bulletin* 123 (2): 162–185.