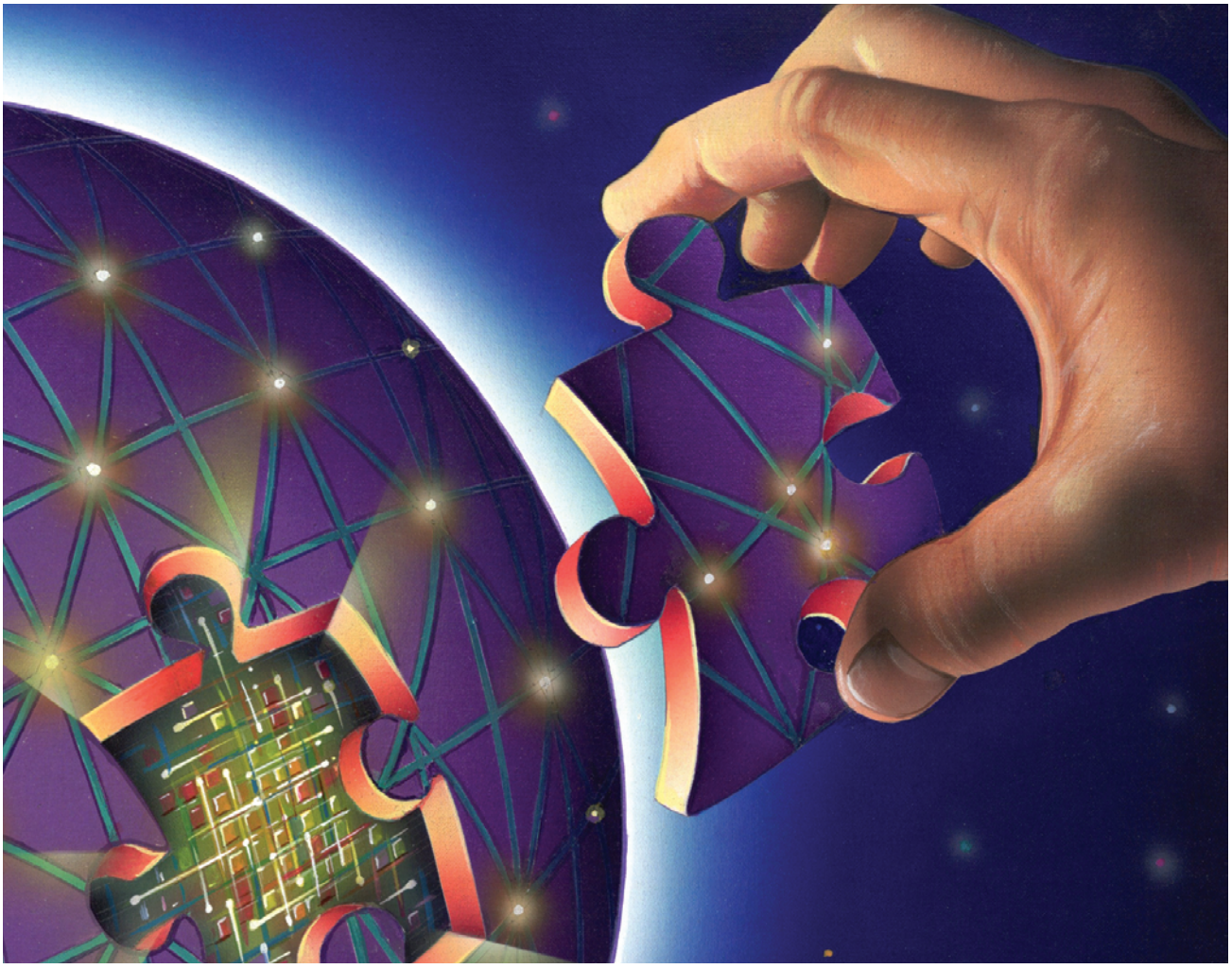
7-2011

End-User Computing Applications

Mary C. Hill
Kennesaw State University, mhill@kennesaw.edu

W. Alan Barnes
Assurant



End-User Computing Applications

Implications for Internal Auditors and Managers

By Mary Callahan Hill and W. Alan Barnes

Businesses today rely on the work being done by staff using personal computers. The proliferation of personal computers has led to widespread implementation of end-user computing applications. As their name implies, end-user applications are designed, implemented, and controlled by users rather than by IT professionals. End-user applications can be risky for organizations, both with respect to management decision making and to financial reporting. For public companies, the risk involved in these applications has been increased by the requirements of the Sarbanes-Oxley Act of 2002 (SOX), which call for management to document end-to-end financial operations and internal control structures. Following is a review of the reasons for the prevalence of end-user applications and their inherent problems, as well as strategies for the internal control of these applications for various-sized businesses.

Background

The following is a textbook definition of end-user computing: [A]n information system developed by the users themselves rather than IT professionals to meet company operational or management information needs. An end-user application often extracts or transfers data from a corporate database as a start-

ing point. (Marshall Romney and Paul Steinbart, *Accounting Information Systems*, 10th ed., 2006)

End-user computing can result in applications such as spreadsheets, databases, data extraction queries, specialized reports, and websites. End-user computing applications, particularly spreadsheets, are essential to business processes: A recent survey indicates that 70.1% of companies rely heavily on spreadsheets for critical portions of their business processes or to complete their financial reporting ("Spreadsheet Management: Not What You Figured," Deloitte, www.deloitte.com/assets/Dcom-UnitedStates/Local%20Assets/Documents/AERS/us_aers_Spreadsheet_eBrochure_070710.pdf).

A do-it-yourself mentality is prevalent in society today, and end-user computing fits well with this mindset. End-user computing gives the user control over a technology project such that explaining subject area or technical requirements to an IT specialist is not required. Further, the end user controls the time schedule of the development, which generally results in a quicker solution. The end user employs his own resources (time and knowledge) in developing the application and, thus, does not have to wait for the funding and scheduling of the project through an IT department's budgetary processes. End-user computing tends to become even more prevalent when budgets are tight and funds cannot be spent to acquire new software, and has become popular as users have grown more sophisticated and more confident in their abilities to develop technical solutions.

Risks

End-user computing, however, can result in four significant risks from an operational or financial reporting standpoint. First, there is the risk that the end-user application will have unintentional errors that result in poor decision making or inaccurate financial reporting. Second is the risk that scarce resources (money or employee time) will be wasted on developing these applications. The third risk is that end-user applications will be used to perpetuate fraud or hide losses. Finally, end-user applications increase the risk of data breaches.

A recent study of spreadsheets—one of the most popular end-user computing tools—has found that over 90% of spreadsheets have errors (Raymond R. Panko, "Spreadsheets and Sarbanes-Oxley: Regulations, Risks, and Control Frameworks," *Communications of the Association for Information Systems*, vol. 17, 2006). These unintentional errors can lead to poor decision making or additional costs. For example, in late 2008, Barclays Capital used a spreadsheet to determine which assets belonging to Lehman Brothers it wished to buy after Lehman's bankruptcy. In the rush to file before the bankruptcy court deadline, however, errors were made in spreadsheet use and formatting that caused Barclays to list assets in the final purchase offer that it did not want to purchase. As a result of this error, Barclays had to file a legal motion to exclude 179 Lehman contracts worth several million dollars that were mistakenly included in the asset purchase agreement (Frank Hayes, "Frankly Speaking: No. 1 Rule for Users: Keep It Simple," *Computerworld*, October 20, 2008). End-user applications have also been known to cause errors in financial reporting; recently, large accounting firms have issued client advisory documents that cite these applications as a significant threat. An example of a financial reporting error occurred in 2005, when Eastman Kodak was forced to restate financial results due to a spreadsheet that incorrectly calculated severance and pension-related termination benefits (Richard Morochove, "Tech at Work: Spreadsheet Slip-ups Cause Financial Errors," *PCWorld*, September 2006).

End-user applications are prone to unintentional errors from a variety of sources. One source is the lack of a systems development process. Because end-user applications are often developed in haste to meet an immediate need, the user/developer may decide to prioritize timeliness over the risk of errors (Jonathan P. Caulkins, Erica Layne Morrison, and Timothy Weidemann, "Spreadsheet Errors and Decision Making: Evidence from Field Interviews," *Journal of Organizational and End User Computing*, July–September 2007). These time pressures cause end users to omit standard systems development activities, such as a program walkthrough, testing, documentation, and independent review. Failure to utilize these steps can cause errors either in the formulation of the application (e.g., an incorrect understanding of the calculations required) or in the creation of the application (e.g., incorrect specifications of a formula in the software or incorrect report generation). Increasing the risk of errors is the problem that users tend to be overconfident in their system development abilities (Panko 2006). The selection of the end-user tool can also be a source of errors. End users frequently select the tool they know best, rather than the best tool for the job. The most common example is when end users develop applications using spreadsheets, when a database tool would be better. (A database tool is preferable when an application contains a large number of records or when the application requires searching and sorting capabilities.) Another source of errors arises from the need to input data into the application; data can be miskeyed or incorrectly or incompletely extracted from a database. Finally, lack of documentation of the application can be a source of errors. If the initial developer leaves the organization, other employees might not know how to use the application, which can result in errors (e.g., a spreadsheet user enters data into a calculated field, overwriting a formula).

Another risk is that companies will waste scarce resources developing end-user applications. Often, end users spend an inordinate amount of time developing an application, only to find that there is existing software that already performs the task (Stanley Earl Jenne, "Audits of End-User Computing," *Internal Auditor*, December 1996). Wasted resources also occur due to duplication of applications within the same company, as individual departments create similar end-user solutions for a common problem but do not share them across departmental boundaries. Another potential waste of resources is when users spend hours developing an application that an expert could have developed in a few minutes or using more efficient technology. Still another possible waste of resources is when a user chooses unusual development software that does not communicate with the company's standard platform. The lack of documentation on these applications means that if the initial developer leaves the organization and other employees do not know how to properly use it, the application can fall into disuse. When an end-user application is disused, the orga-

nization loses both the time spent developing it and the ability to perform an organizational task.

End-user applications can be risky from a perspective of intentional misstatements and frauds. The separation of duties that is built into systems developed via IT departments does not exist in end-user applications. In many cases, the developer is simultaneously sponsor, programmer, tester, and user (Deloitte 2009). The tools used for end-user applications are meant to be user friendly and flexible, but these qualities make applications developed with the tools easy to manipulate. An example of this risk is shown in the spreadsheet fraud that was perpetuated by Allfirst Bank trader John Rusnak. Rusnak lost close to $700 million in bad trading decisions. To cover his losses, he substituted a falsified spreadsheet as the input to a production trading system that his supervisors used for control (Gary Flood, "Spreading the Blame," *Financial Director*, May 25, 2006).

Finally, end-user applications can be risky with respect to data breaches. Many end-user applications extract data from a production database into the end-user application. The data then become much less secure, because the application can be stored on a user workstation, laptop, flash drive, or other portable device. A 2008 study found that more than 800,000 laptops are lost each year by users traveling through airports in the United States and Europe (Donna Fuscaldo, "Services Find Lost, Stolen Laptops," FoxBusiness, February 11, 2010). The press frequently reports of data breaches due to lost or stolen laptops containing confidential data, including Social Security or credit card numbers.

Control

End-user applications are an issue for all sizes of companies, but the approach to controlling these applications depends on the size of the company, its resources, and the number of end-user applications. Thus, we discuss control strategies for end-user computing based on size classifications as follows: Large companies are defined as those that have internal audit and IT departments, midsized companies are those that have some designated IT specialists, and small companies are those that have few departmental classifications. Large companies with internal audit staff are most likely to be aware of the risks related to end-user computing applications. Recently, the Institute of Internal Auditors issued Global Technology Audit Guide 14 on this topic (Christine A. Bellino, Douglas Ochab, and Jeffery S. Rowland, *Auditing User-developed Applications*, June 2010). Smaller companies have less guidance on this issue.

Two efforts are required to make sure that end-user applications are properly controlled and do not have a negative effect on either financial reporting or operational decision making. The first effort is to establish controls over the development of end-user computing applications. The second is examining the end-user applications themselves.

Control over development includes a policy on end-user applications, communication about end-user applications, and training on software to develop end-user applications. Below, we discuss each of these aspects of control over development of end-user applications in general and then, more specifically, how that aspect might be put in place for various sizes of companies.

Policy

For all companies, a policy should be put in place and communicated to employees about the use of end-user applications (Morochove 2009). The goals of having a policy are to promote consistency in development of end-user applications, to ensure that applications are developed with some form of control, and to make certain that the application is examined by at least one other employee. Meeting these goals should help to eliminate unintentional errors from the end-user applications.

A policy to address end-user computing is particularly important for large companies, because some large companies have been known to have hundreds of end-user applications (Panko 2006). Further, large companies are most likely to be subject to the requirements of SOX or other regulations such as the Payment Card Industry Data Security Standard. In large companies, the policy would most effectively be developed as a joint effort between the internal audit and IT departments. Policies may have to be approved or conform to standards set by the companies' external auditors. Once the policy is developed, it needs to be communicated to employees. IT staff (training specialists or help desk personnel) or internal auditors who interface with users should be aware of and promote the end-user computing policy.

One portion of the policy should include a description of a process that users should follow when developing applications. This type of process is generally referred to as a systems development life cycle (SDLC). In general, SDLC processes include how to define requirements for an application; how to confirm the requirements by walking through the proposed application with another knowledgeable user; how to select the appropriate software for the application; how to conduct appropriate testing; the type of documentation needed; and implementation standards, including backup and the need to cross-train at least one other user on the application (Romney and Steinbart 2006). Large companies are likely to already have an SDLC process in place in the IT department, and for simplicity and efficiency, the end-user computing policy should utilize the same SDLC process. The importance of testing should be particularly emphasized because end users have been found to be overconfident about their technical competency (Panko 2006).

End-user applications are an issue for all sizes of companies, but the approach to controlling these applications depends on the size of the company, its resources, and the number of end-user applications.

Another portion of the policy should include internal controls surrounding end-user applications once they are developed.

These controls include version standards, documentation standards, and access controls. Version standards include the development of a naming convention, the storage and backup of versions of the application, and a method to ensure that only the latest version is being used. Documentation standards should ensure that, to the greatest extent possible, the application is self-documenting, using tools such as clear field labels and built-in instructions. Access controls should require the application to have passwords and storing applications, with critical or sensitive data on secure servers that have frequent backups performed.

For large companies, the policy should provide a checklist to determine the level of risk in the application. Using the checklist, users should determine the risk in their application and report the risk level to the internal audit department for determination of potential inclusion of the application in the internal audit staff audit plans. The more "yes" answers to the questions below, the riskier the application. The following factors increase risk:

- Does the output of the end-user application significantly impact decisions about operations? If yes, what is the maximum dollar impact?
- Is the output used for accounting or financial reporting purposes? If yes, what accounts are affected?
- Would the loss of the end-user application or its output have a detrimental operational, financial, or legal impact? If yes, what losses would occur?
- Do multiple users rely on the end-user application or its output? If yes, list the downstream users.
- Are the data contained in the application confidential to the business or employees? If yes, specify the nature of the confidential data.
- Is the application particularly complex with respect to calculations? If yes, briefly describe the nature of the calculations.
- Does the input to the application rely on multiple applications, such as a database extraction query to input data into a spreadsheet? If yes, then the application's risk increases because either the end-user application itself or the process for loading data could contain an error.

While small and midsized companies have fewer resources to devote to controlling end-user applications, they must still make an effort to set up some policies to avoid the risks noted above. In midsized companies, the responsibility for developing a policy for end-user computing would normally rest with the controller or accounting manager. Department managers would communicate the policy to their employees. Midsized companies are less likely to have a formal SDLC process, but some aspects of those processes should be included as part of the end-user computing policy. Most importantly, the policy should emphasize how to conduct appropriate testing of an application and that the application needs documentation. Policies for midsized companies should also emphasize that end users should employ only standard and well-known application software packages (e.g., Microsoft Excel and Microsoft Access) so that other users within the company will be familiar with the development tool. Controls over the developed application should be similar to those for large companies and include version standards, documentation standards, and access control.

Small companies have even fewer resources available to develop and implement formal policies. Further, in small companies with informal cultures, implementing policies and controls can imply a lack of trust, thereby hurting morale and damping the initiative that is central to end-user application development (Caulkins, Morrison, and Weidemann 2007). A policy with respect to end-user applications should be part of a small company's control environment; in fact, simply raising awareness of the issue among staff members will help reduce risk. Small companies should also stipulate in their policies that end users utilize only well-known application software packages, that they use templates rather than starting from scratch on each new application, that they adhere to reporting standards (such as using brackets for negative amounts), and that they try to keep the application as simple as possible by avoiding complex formulas. Small businesses might also find it valuable to designate an employee as the "technology expert" and request that end-user applications be cleared with the company expert (Caulkins, Morrison, and Weidemann 2007). Like other companies, small companies should have policies to control the developed application: version standards, documentation standards, and access control.

Communication

Communication about end-user applications is important in order to eliminate waste. If end-user applications are developed without communication, there is more likelihood of duplicate or overlapping applications within a company. Thus, communication includes monitoring development across departments or branch offices that might develop overlapping or redundant end-user applications. Communication will help alleviate application losses that occur when developers leave the company. End users should communicate the purpose of the application, the tool that is being used to develop the application, their contact information, the location of the application, and the location of the application's documentation.

In large companies, communication should center on the IT support staff and user help desk. IT support staff should act as a clearing house for the coordination of end-user applications. Communication to the IT department and internal audit should occur at the beginning of the end-user development. Communication to the IT staff also facilitates any technical support that the user might need during the development and testing of an application.

In small and midsized companies, supervisors should encourage staff members to discuss possible end-user applications with them prior to initiating efforts to develop the application. Supervisors should share developments with peers in other departments and with relevant IT support staff.

Training

Training on end-user application software is critical to successful implementation of the end-user policy and the controls over the application itself. Training is important in order to avoid waste in the development of the applications. Training issues are the same for all sizes of companies—the only difference would be whether the training is offered within the company or by an outside vendor. End users must know how to use the development software, they must be able to evaluate the complexity of the application they are planning to develop, they must be able to estimate the time involved, and they must know where to get help if they are having trouble with the development software. Supervisors should encourage

employees to attend training classes. It is important also for supervisors to be knowledgeable about software in order to appropriately direct their employees' application development efforts.

As part of each application's training, users must be made aware of the appropriate—and inappropriate—tasks for the application. Training should include the control attributes that are built into the application, such as how to protect the application using passwords; how to prevent input errors using locked data fields, embedded edits, or drop-down lists; and how to use any embedded auditing tools (e.g., Microsoft Excel comes with an auditing tool that shows cell dependencies). Users also need to be trained in how to use the self-documentation features of the application, such as track changes and printing application structures. Finally, end users should be made aware of software that helps debug applications, such as Spreadsheet Detective or OpenGate Software for Microsoft Access.

Examining End-User Applications

For all companies, examining end-user applications consists of two activities. The first is gaining knowledge of the applications that exist and their purpose; the second is testing the application for accurate processing. Examining end-user applications is important in order to avoid all four potential risks associated with these applications: errors, waste, fraud, and data breaches.

In large companies, examining end-user applications would most naturally be performed by the internal audit staff. Internal audit would first conduct an inventory to identify end-user applications that are currently being used. An inventory can be conducted using either of two methods or a combination of both methods. One method is to survey users about their use of end-user applications to complete business processes or financial reporting. The survey would include questions on the purpose of an application, how frequently it is used, the number of copies or versions of the application, and whether there is adequate documentation of the application. The other option for conducting an inventory is to scan the company network for specific file extensions (e.g., ".xls" or ".mdb"). Each method has limitations: The survey relies on user responses, while the scan excludes laptops, other nonnetworked computers, flash drives, and other portable media.

Once the critical end-user applications are identified, internal auditors should perform tests on them. In large companies, testing might start with the following:

- A review of the application documentation or a review of the self-documenting features of the application
- A review of the version control processes in place surrounding the application and consistent use of naming conventions
- A review of the distribution list for the application or its output
- A review of the output of the application, such as making sure that the reports are transmitted as PDF files rather than as copies of the application itself (unless the application will require further downstream input)
- A comparison of input data to source material
- A review of the backup process for the most current version of the application
- A review of the termination control process, if an employee who "owns" a critical end-user application leaves the company.

These steps will also help an auditor gain an understanding of the application. The examination of the application would continue by having the auditor use the application itself. Some tests include the following:

- Testing the access control to the application by trying to log on using a password and user ID
- Recomputation of critical calculations
- Testing field edits or drop-down lists
- Trying to enter data into locked fields
- Generation of critical reports.

Test results should be documented, along with suggested remediation efforts.

In small and midsized companies, which generally do not have internal audit departments, examination of end-user applications would be conducted less formally. However, inspection is almost more important for small companies than large companies because large companies will create processes to build quality into end-user applications, while smaller companies tend to try and "inspect" quality into the applications (Caulkins, Morrison, and Weidemann 2007).

Midsized companies might want to keep a list of critical end-user applications, along with the developer name. In midsized companies, another employee should be assigned to both examine and be cross-trained on the application. The training would include how to access the application, how data is input to the application and how to verify that the input is correct, the critical calculation in the application, a review of the documentation for the application, and the backup and storage processes for the application. The testing would occur by assigning the cross-trained employee to utilize the application in the absence of the original developer.

For small companies, examination of end-user applications will primarily rest with the developer's supervisor. Companies may want to develop a checklist of things to look for in end-user applications, such as critical calculations or locked fields. Further, it is important that the supervisor perform a "sniff test" that examines both any critical assumptions used to develop the application and the bottom-line reasonableness of any output of the application (Caulkins, Morrison, and Weidemann 2007).

Controlling Risk

End-user applications are a critical business resource and a fact of life in today's organizations. While end-user computing has many benefits, there are risks involved in it that need to be recognized and controlled. Some audit staff and IT professionals argue that, given the spontaneous nature of the development of end-user applications and their immense number, these applications are impossible to control. However, uncontrolled development of end-user applications leaves an organization open to error, waste, and fraud. Therefore, even the smallest companies must make some effort to control them. Steps to control the development and use of end-user applications include implementing policies that govern their development, increasing communication about the applications, training users, and independent examination of critical applications. ❑

Mary Callahan Hill, PhD, CPA, *is a professor of accounting at Kennesaw State University, Kennesaw, Ga.* ***W. Alan Barnes, CPA, CIA,*** *is a director of risk and advisory services at Assurant, Atlanta, Ga.*