The Relationship of Application Risks to Application Controls: A Study of Microcomputer-based Spreadsheet Applications

Robert Schultheis and Mary Sumner
Southern Illinois University, Edwardsville

In most organizations today, end-user managers develop microcomputer-based applications using database, spreadsheet, project management, graphics, and other types of software. Many of these end-user systems are developed in response to the changing decision-making needs of managers. The developers themselves range in experience from relatively novice end-users to functional support specialists within various user departments.

A persistent concern expressed in the research literature and by practitioners is that end-user applications lack quality assurance (Pierson, Forcht, and Teer, 1990). Poor design, invalid data, and inadequate documentation introduce risks to the organization (Yeager, 1990). These risks may include incorrect outputs, unauthorized access to data, and invalid logic. Such risks may pose inconvenience, may impede transferability, and may cause needless re-invention of data and logic. The most serious risks occur, however, when the outputs of user-developed systems are invalid and when these data are used to make management decisions with financial impact on the business (Hayen and Peters, 1989).

The quality of user-developed systems is an issue which has relevance to user managers and to MIS professionals. User managers are increasingly aware of the hazards of applications which produce poor data. MIS professionals are also concerned about quality assurance issues and are introducing audit and control procedures for certain “mission critical” systems (Sumner and Klepper, 1987). However, the vast majority of user-developed systems lack adequate review and sufficient controls (Edge and Wilson, 1990).

The problem of user-developed systems represents a management control issue. Certain spreadsheet applications may pose greater risk to the organization...
than others. The risk of poor decision-making can occur when problems such as bad data, faulty logic, and unauthorized access introduce errors into a spreadsheet. A basic auditing premise is that the greater the risk, the greater the need for controls. Based on this premise, high-risk applications should have effective control mechanisms to assure data integrity, application reusability, and security. However, low-risk applications may not need to have extensive controls over their design and implementation.

This study addresses the issue of risks and controls in end-user development. It analyzes two questions related to the design and construction of spreadsheet applications by end-users. The first question focuses on the types of controls and the levels of risk associated with the applications themselves. The second question pertains to the extent to which control mechanisms are used to minimize the risk factors in development and use of spreadsheet data. In other words, a basic auditing premise was tested: If managers are aware of the risk factors in the spreadsheets they develop, do they apply controls in design and implementation to assure that these risks are effectively controlled?

Prior Research

A variety of research studies have focused upon quality assurance issues affecting user-developed systems. Many user-developed systems are constructed without adequate documentation, sufficient data integrity, and effective requirements definition (Sumner and Schultheis, 1990). Procedures for data validation and logic verification are often missing (Hayen and Peters, 1989). In general, user-developed applications lack design review and control procedures which are normally associated with application development by information systems professionals (Davis, 1984; Alavi and Weiss, 1985-1986; Sumner and Schultheis, 1990).

Spreadsheet applications pose potential risks to an organization. Perhaps the major risk of undetected spreadsheet errors is related to the consequences of a management decision based upon incorrect output. There are many anecdotes which report the consequences of spreadsheet errors. For example:

1. An undetected spreadsheet error in the application of the @SUM function at one construction company caused an erroneous bid on a contract, resulting in an underbid of $254,000 (Hayen and Peters, 1990).

2. An accounting firm consultant discovered over one hundred errors in several of his clients’ multibillion-dollar spreadsheets (Simkin, 1987).

3. A spreadsheet designed for market forecasting produced a forecast that was below the mark by $36 million. The error resulted because formulas caused amounts to be rounded to whole numbers, eliminating an inflation rate of 1.04 from the calculations (Krull, 1989).

Lack of reliability, lack of auditability, and lack of modifiability are all risks which are associated with poor spreadsheet design (Boaz, Paley, and Lucas Jr., 1989).

There are a number of studies that have identified the risk factors associated with end-user applications. Pierson, Forcht, and Teer (1990) found that an application may have a higher degree of risk and require a higher level of documentation if it has a financial impact, if its outputs are used by another application, and if it is broader in scope. If extensive maintenance requirements are expected, and if the application is highly complex, then a higher-level of risk is involved.

These same findings are true for spreadsheet applications. Some of the factors which contribute to the need to document a spreadsheet application are complexity, scope, frequency of use, and structure. If a spreadsheet has intricate logic, has an organization-wide impact, is extensively used, and is interrelated with other spreadsheets, then it may pose a more significant risk and may need more effective controls than spreadsheets that do not have these characteristics.

A number of studies recommend the use of controls in spreadsheet design and implementation. In general, testing, documentation, and audit procedures are effective control strategies (Ronen, Paley, and Lucas Jr., 1989). Effective controls include developing a paper model, conducting a design review, verifying logic, and establishing different areas in the spreadsheet for data and for formulas (Hayen and Peters, 1989). Documentation should include a description of the spreadsheet’s purpose, listings of formulas and cell contents, a description of macros, a diagram of flows between spreadsheets, and security and audit procedures. Informal audit procedures such as user reviews of the reasonableness of output data and use of historical data for which manual outputs are available to check spreadsheet results are also important (Edge and Wilson, 1990).

While many managers acknowledge the risks of spreadsheet errors, it is not clear if effective controls are being applied. This is particularly important in the case of high-risk spreadsheet applications.
Related Content

The Transformation Model
[www.igi-global.com/chapter/transformation-model/163868?camid=4v1a](www.igi-global.com/chapter/transformation-model/163868?camid=4v1a)

Microcomputer Software Evaluation and Selection Strategies
[www.igi-global.com/article/microcomputer-software-evaluation-selection-strategies/55647?camid=4v1a](www.igi-global.com/article/microcomputer-software-evaluation-selection-strategies/55647?camid=4v1a)

A Study of Computer Attitudes of Non-Computing Students of Technical Colleges in Brunei Darussalam
[www.igi-global.com/article/study-computer-attitudes-non-computing/3752?camid=4v1a](www.igi-global.com/article/study-computer-attitudes-non-computing/3752?camid=4v1a)

Studying the Documentation of an API for Enterprise Service-Oriented Architecture
[www.igi-global.com/chapter/studying-documentation-api-enterprise-service/62791?camid=4v1a](www.igi-global.com/chapter/studying-documentation-api-enterprise-service/62791?camid=4v1a)