Whitepaper

Spreadsheets and Sarbanes–Oxley: Regulations, Risks, and Control Frameworks

Raymond R. Panko
University of Hawai`i
panko@hawaii.edu

I. ABSTRACT

The Sarbanes–Oxley Act of 2002 (SOX) has forced corporations to examine their spreadsheet use in financial reporting. Corporations do not like what they are seeing. Surveys conducted in response to SOX have shown that spreadsheets are used widely in corporate financial reporting. Spreadsheet error research, in turn, has shown that nearly all large spreadsheets have multiple errors and that errors of material size are very common. The first round of Sarbanes-Oxley assessments confirmed concerns about spreadsheet accuracy. Another concern is spreadsheet fraud, which also exists in practice and is easy to perpetrate. Unfortunately, few organizations have effective controls to deal with either errors or fraud. This paper examines spreadsheet risks for Sarbanes-Oxley (and other regulations) and discusses how general and IT-specific control frameworks can be used to address the control risks created by spreadsheets.

II. KEYWORDS

KEYWORDS:
CobiT, controls, control deficiency, control framework, COSO, end-user computing (EUC), error, error rate floor, formula error rate (FER), fraud, 17799, ITIL, material error, spreadsheet.

III. INTRODUCTION

CONTROLS AND SARBANES-OXLEY

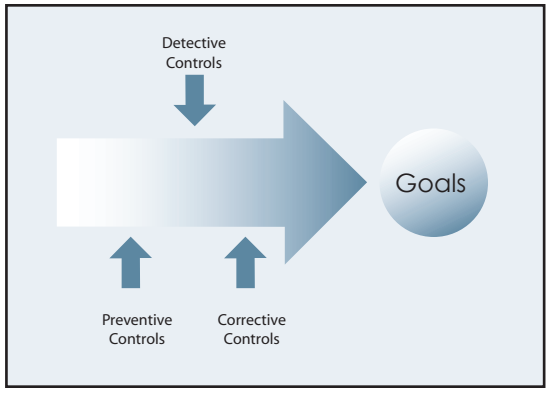
After financial reporting scandals at Enron and other major companies, the U.S. Congress passed the Sarbanes–Oxley Act (SOX) in 2002. Section 404 of the Act requires nearly every public company's chief corporate officers to assess whether the company's financial reporting system has been effectively controlled during the reporting period. Furthermore, it specifies that the company must hire an independent external auditor to assess the officers' assessment.

To oversee SOX, Congress created the Public Company Accounting Oversight Board (PCAOB) to create auditing standards. PCAOB's main guidance on 404 assessments of control attestations has been Auditing Standard No. 2, An Audit of Internal Control Over Financial Reporting Performed in Conjunction with an Audit of Financial Statements (PCAOB, 2004).

The focus of SOX and of Auditing Standard 2 is the creation of effective controls. Figure 1 illustrates that controls are ways to help a corporation achieve its objectives, such as producing accurate financial reports—despite the presence of threats.

Controls are ways to help a corporation achieve its objectives, such as producing accurate financial reports—despite the presence of threats.

Figure 1: Controls
Source: Panco (2005c)



Controls cannot guarantee that the goals will be met, but they reduce the risk that these objectives will not be met. In this context, effectively controlled financial reporting processes give reasonable assurance that the company will meet the goal of producing accurate financial reports.

Effectively controlled financial reporting processes give reasonable assurance that the company will meet the goal of producing accurate financial reports.

According to Auditing Standard 2, an internal control deficiency exists when the design or operation of a control does not allow for the timely prevention or detection of misstatements. The standard (PCAOB, 2004) defines two types of deficiencies:

- In a significant deficiency, there is more than a remote likelihood that the financial statements will be impacted in a manner that is consequential but not material.

- In a material deficiency, there is "a significant deficiency, or combination of significant deficiencies, that results in more than a remote likelihood that a material misstatement of the annual or interim financial statements will not be prevented or detected" (PCAOB, 2004). Vorhies (2005) indicates that a 5% error in revenues is the usual threshold for labeling a deficiency as material because a smaller difference is not likely to sway a reasonable investor.

This distinction between significant and material internal control deficiencies is important because if management finds even a single material deficiency, it may not assess its internal controls as having been effective during the reporting period.

According to the PCAOB's analysis, 12% of all audits in 2004 and the first part of 2005 assessed companies as not having effectively controlled their financial reporting function (Rankin, 2005). Actually, the situation may be much worse because only larger firms were required to assess their financial reporting systems during that period. In addition, auditors tended to focus on strikingly out-of-control aspects of financial reporting systems.

Failing an audit of the effectiveness of financial controls can be very costly to a company. The research firm Glass, Lewis & Company analyzed 899 cases in which firms reported material weaknesses (Durfee, 2005). They discovered that companies experienced an average stock price drop of 4% right after the announcement. In turn, the Dutch research firm ARC Morgan found in 2004 that in more than 60% of all cases, the chief financial officer (CFO) was replaced within three months after a companies reported material weaknesses (Durfee, 2005).

IV. WHAT ABOUT ALL THE SPREADSHEETS?

THE USE OF SPREADSHEETS IN FINANCIAL REPORTING

Auditing Standard No. 2 clarifies that controls must involve all forms of information technology (IT) used in financial reporting. One particular IT concern for corporations is the use of spreadsheets in financial reporting. There have long been indications that many spreadsheets are large (Cale, 1994; Cragg and King, 1993; Floyd, et al., 1995; Hall, 1996), complex (Hall, 1996), and very important to their firms (Chan and Storey, 1996; Gable, et al., 1991; Hall, 1996). When Comshare, Inc. surveyed 700 finance and budgeting professionals in the mid-1990s, it found that spreadsheets were already dominating budgeting (Modern Office Technology, 1994).

Although some people might doubt that companies use spreadsheets in critical financial reporting operations, the widespread use of spreadsheets is well documented, thanks to surveys motivated by concerns over SOX.

- In 2004, financial intelligence firm CODA reported that 95% of U.S. firms use spreadsheets for financial reporting (www.coda.com).

- RevenueRecognition.com (2004) (now Softtrax) had the International Data Corporation interview 118 U.S. business leaders. IDC found that 85% were using spreadsheets in financial reporting and forecasting.

- CFO.com (Durfee, 2004) interviewed 168 finance executives in 2004. The interviews asked about information technology use in the finance department. Out of 14 technologies discussed, only two were widely used—spreadsheets and basic budgeting and planning systems. Every subject said that his or her department used spreadsheets.

- In Europe, A.R.C. Morgan interviewed 376 individuals responsible for overseeing SOX compliance in multinationals that do business in the United States (TMCnet.com, 2004). These respondents came from 21 different countries. More than 80% of the respondents said that their firms used spreadsheets both for managing the financial reporting control environment and for financial reporting itself.

- In a webcast for Delloite on May 22, 2005, the author was able to ask a series of questions of the audience. The average response size was just over 800 financial professionals and officers in corporations. One question specifically asked, "Does your firm use spreadsheets of material importance in financial reporting?" Of the respondents, 87.7% answered in affirmative, while 7.1% said, "No." (Another 5.2% chose "Not Applicable.")

Furthermore, when companies use spreadsheets for financial reporting, they often use many. One firm used more than 200 spreadsheets in its financial planning process.

Today, companies are widely confused over what to do about spreadsheet controls. Obviously, if financial reporting spreadsheets contain a significant number of errors and a reasonable amount of testing has not been done, it is difficult to say that the reporting process is well controlled.

LACK OF CONTROLS, INCLUDING TESTING

One concern with spreadsheets is that they rarely are well-controlled (Cragg and King, 1993; Davies and Ikin, 1987; Fernandez, 2002; Floyd, et al., 1995; Gosling, 2003; Hall, 1996; Hendry and Green, 1994; Nardi, 1993; Nardi and Miller, 1991; Schultheis and Sumner, 1994). This is not surprising because few organizations have serious control policies—or indeed any policies at all—for spreadsheet development (Cale, 1994; Fernandez, 2002; Floyd, et al., 1995; Galletta and Hufnagel, 1992; Hall, 1996; Speier and Brown, 1996).

A specific concern is testing. Although there has long been evidence that spreadsheet error is widespread, organizations rarely mandate that spreadsheets and other end user applications be tested after development (Cale, 1994; Cragg and King, 1993; Floyd, et al., 1995; Galletta and Hufnagel, 1992; Gosling, 2003; Hall, 1996; Speier and Brown, 1996). Also, individual developers rarely engage in systematic testing on their own spreadsheets after development (Cragg and King, 1993; Davies and Ikin, 1987; Hall, 1996; Schultheis and Sumner, 1994).

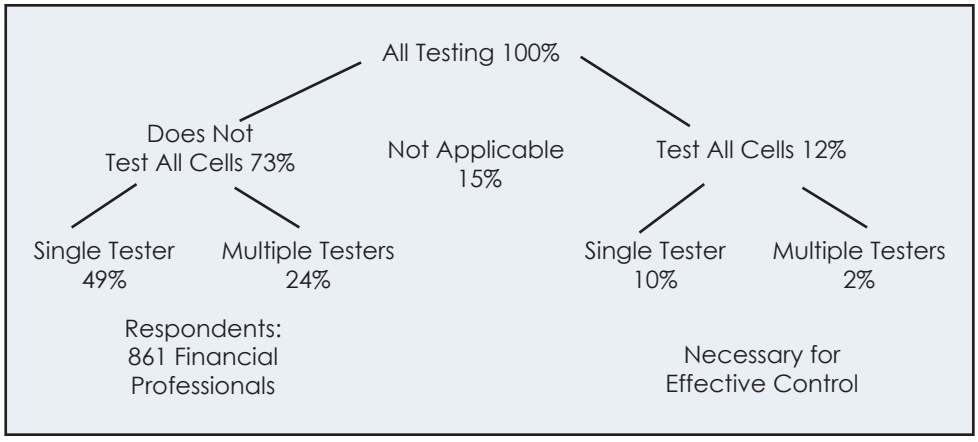
As noted earlier, the author was able to ask questions of corporate financial professionals and officers in a webcast. Figure 2 shows respondent answers to the question, "For spreadsheets of material importance used in financial reporting, what percentage does your company test?" Seventeen percent of the respondents said that their firm tests more than 25% of their material financial spreadsheets, and 16% said that their firm tests nearly all.

Figure 2: Testing for Material Financial Spreadsheets
Source: Panco (2005c)

Almost None	24%
Under 10%	20%
11% to 25%	12%
Over 25%	17%
Nearly All	16%
Not Applicable	11%
Total	100%
Respondents	862

These results make it appear that many companies do test their spreadsheets. However, what most respondents call testing appears to be "looking over the spreadsheet," rather than comprehensive cell-by-cell testing. Later in the webcast, participants were queried about their firms' testing of spreadsheets of material importance used in financial reporting. Figure 3 shows the results. Note that only 12% of the respondents said that their firms tested all cells. In addition, only 2% said that they both tested all cells and used multiperson testing. As we will see later, only testing all cells and using multiple testers is likely to be an effective control for spreadsheet errors.

Figure 3: Extent of Testing and Multiperson Testing
Source: Panco (2005c)



This lack of comprehensive testing may exist because developers tend to be overconfident of the accuracy of their untested spreadsheets. Certainly, widespread overconfidence, often in the face of widespread errors, has been seen repeatedly in spreadsheet research (Brown and Gould, 1987; Davies and Ikin, 1987; Floyd, et al., 1995; Panko and Halverson, 1997; Panko 2006c).

In a vicious cycle, organizations that do not test their spreadsheets get no feedback on real error rates and so do not realize the ubiquity of spreadsheet errors. Therefore, they see no need for testing. Rasmussen (1974) has noted that people use stopping rules to decide when to stop doing activities such as testing. If people are overconfident,

they are likely to stop too early. Consequently, if firms use spreadsheets to make decisions but do not test their spreadsheets, they may not realize how many errors there are in their spreadsheets.

One might argue that the real world would provide painful feedback if a spreadsheet were incorrect. For some situations, such as budgeting, errors would have to be small in order to pass undetected. Unfortunately, in this case, even small percentage errors can be very damaging. Hicks (1995) found that a relatively small percentage error in the capital budgeting spreadsheet he examined would have produced an error of several hundred million dollars. Yet this difference was too small, compared to the total, to be detected easily by "checking the result for reasonableness."

At the other extreme, when a new situation is modeled, such as the purchase of another company, even large errors in the spreadsheet might not be obvious. If a promising corporate purchase goes bad, furthermore, it is easy to dismiss the problem as being due to unforeseen factors, even if the real problem was a spreadsheet error. Without testing, real-world feedback may not be very effective.

THE PREVALENCE OF SPREADSHEET ERRORS

Are errors common in spreadsheets? For most people, the most convincing data on spreadsheet errors come from audits of real-world operational spreadsheets. Figure 4, which presents data from several audit studies, shows convincingly that spreadsheet errors are extremely common.

Figure 4: Audits of Real-World Spreadsheets
Source: Panco (2005c)

Authors	Year	Number of Spreadsheets Audited	Average Size (Cells)	Percent of Spreadsheets with Errors	Cell Error Rate	Comment
Hicks	1995	1	3,856	100%	1.2%	One omission error would have caused an error of more than a billion dollars.
Coopers & Lybrand	1997	23	More than 150 rows	91%		Off by at least 5%. At 5%, financial errors are considered to be material (Vorhies, 2005).
KPMG	1998	22		91%		Only significant errors that could affect decisions.
Lukasic value	1998	2	2,270 & 7,027	100%	2.2%, 2.5%	In Model 2, the investment's was overstated by 16%. Quite serious.
Butler	2000	7		86%	0.4%	Only errors large enough to require additional tax payments.*
Clermont, Hanin, & Mittermeier	2002	3		100%	1.3%, 6.7%, 0.1%	Computed on the basis of non-empty cells.
Interview I**	2003	~36 / yr		100%		Approximately 5% had extremely serious errors.
Interview II**	2003	~36 / yr		100%		Approximately 5% had extremely serious errors.
Lawrence and Lee	2004	30	2,182 unique formulas	100%	6.9%	30 most financially significant spreadsheets audited by Mercer Finance & Risk Consulting in previous year.
Total/Average			**88**		**94%**	**5.2%**

- First, these audits found errors in the vast majority (94%) of the spreadsheets they audited. This percentage would have been even higher, but several of the studies only reported serious errors. In other words, we should expect nearly all spreadsheets to contain errors. In fact, when the author discussed spreadsheet errors with the principals of two spreadsheet auditing firms in the UK, both said that they had never audited a major spreadsheet without finding errors.

- Second, these audits found many errors in the spreadsheets they audited. Specifically, studies that measured errors on a per-cell or per-formula basis (Butler, 2000; Clermont, et al., 2002; Hicks, 1995; Lawrence and Lee, 2004; Lukasic, 1998) found errors in an average of 5.2% of the cells or formulas in these spreadsheets. Most large spreadsheets have thousands of formula cells, so these large spreadsheets probably have dozens or even hundreds of errors.

If this cell/formula error rate (CER/FER) seems excessive, it should not. There has been a great deal of research on human error (cf. Panko, 2006a), and for tasks of comparable complexity, such as writing computer program statements, similar error rates are seen universally. Panko (2006a) has summarized results from a number of studies that measured fault rates in real-world software. Of particular value are four large studies (Ebenau and Strauss, 1994; Madachy, 1996; O'Neill, 1994; Weller, 1993). In these studies, the average error rate per line of code ranged from 1.5% to 2.6%. Note that this is close to the cell/formula error rates seen in Figure 4 for spreadsheet code inspections. Grady (1992) and Zage and Zage (1993) both found that software error rates depend on program difficulty. In both studies, fault rates were at least twice as high for difficult programs as for simple programs.

Humans appear to have an error rate floor (ERF) that exists even when they are working very carefully. Everyone has a similar error rate floor, and working more carefully can decrease one's error rate only modestly. Research has shown that the same human cognitive processes that allow us to respond to the world correctly most of the time have unavoidable trade-offs that create errors a few percent of the time (Reason, 1990). In most human cognitive activities, such small error rates are only minor nuisances, if anyone notices them at all. However, when dozens of formula cells are on a chain to calculate a bottom-line financial value, the probability of error in the bottom-line value becomes unacceptable.

BUT ARE THE ERRORS MATERIAL?

Errors are only bad if they are large enough to make a difference. Perhaps financial professionals in corporations catch all errors large enough to cause problems. Unfortunately, that is not the case.

An obvious issue for Sarbanes–Oxley is many spreadsheet errors are material. As noted earlier, a 5% error in an important bottom-line value in a key financial variable would probably be considered a material error (Vorhies, 2005). When Panko (2006b) interviewed the two spreadsheet auditing principals, both independently gave data suggesting that about 5% of all spreadsheets contain what one of the interviewees called "show stopper" errors. However, these show-stopper errors were far larger than simple materiality.

More to the point, the Coopers and Lybrand (1997) study shown in Figure 4 did not report an error unless there was at least a 5% error in a bottom line value, that is, a material error. The study found such errors in 91% of all spreadsheets. KPMG (1998) found a similar error rate and only reported spreadsheets to be incorrect if they contained errors that would make a difference to decision makers.

The Coopers and Lybrand (1997) study shown in Figure 4 did not report an error unless there was at least a 5% error in a bottom line value, that is, a material error.

More indirectly, we have data from software testing studies that classified errors found as major or minor. Although definitions about what constitutes a major error differ, all software audit studies that have used major/minor distinction found that major errors are very common.

- Bush (1994) and Jones (1998) both reported that a quarter of the errors in the inspections they examined were major errors.
- O'Neill (1994) found only 13% of errors to be major.
- Schulmeyer (1999) found that 42% of all errors were major.
- Ebenau and Strauss (1994) and Weller (1993) found major errors in 1.4% to 2.8% of the lines of code examined but did not report major errors as a percentage of total errors.

Given this data from software inspections, it certainly would be risky to assume that nearly all spreadsheet errors will be minor.

THE PROSPECT OF SPREADSHEET FRAUD

Although there has been a great deal of research on spreadsheet error, there has been no formal research on spreadsheet fraud. Legal definitions of fraud vary, but, generally speaking, a fraud exists when one person knowingly lies or conceals information whose nondisclosure would make other statements misleading, in order to get the victim to act in a way contrary to the victim's interests. Note that two elements are needed for there to be fraud: deception and harm.

There has long been concern that spreadsheet developers will manipulate their spreadsheet assumptions to make the results look better for their bargaining position (Levy, 1984). Few people would consider minor "puffery" to be fraud. However, when as the degree of deception and the damage due to deception increase in intensity, then spreadsheet misanalysis eventually rises to the level of fraud.

HM Revenue & Customs

Spreadsheet fraud is not just a theoretical concern. In England, HM Revenue & Customs collects certain types of taxes. When spreadsheets became prevalent in tax submissions, the agency began to audit submitted spreadsheets and found that many had substantial errors. In the late 1990s, the agency developed a program to automate many aspects of this auditing process (Butler, 2000). This was SpACE: Spreadsheet Auditing by Customs & Excise. (HM Revenue & Customs was previously called HM Customs & Excise.)

In addition to looking for innocent errors, SpACE also looks for certain types of fraud that the agency had found in earlier audits. For instance, the program highlights any numbers that are entered as text. Excel treats text cells as having the value zero. Consequently, entering a number in a column of numbers as text reduces the real total. This will create a fraudulent reduction in tax payments. Although HM Revenue & Customs has not published detailed information on its findings, staff members have said that nearly all adjustments require additional taxes rather than tax rebates.

The Allfirst Fraud Scandal

The most famous case of spreadsheet fraud occurred at Allfirst, which was a U.S. subsidiary of Allied Irish Banks (AIB) in Ireland. AIB commissioned Eugene Ludwig, former U.S. Comptroller of the Currency, to prepare a report on the incident. The story his report tells is an excellent cautionary tale.

Allfirst currency trader John Rusnak began losing money in his trades around 1997. He used a series of spreadsheet subterfuges to hide his losses, which continued to increase. When the fraud was finally discovered, his losses amounted to $691.2 million. Although neither Allfirst nor Allied Irish Banks went into receivership, the losses amounted to 60% of AIB's 2001 revenues and produced a major drop in AIB's stock price. After the scandal, AIB sold off its Allfirst subsidiary.

Rusnak began by entering two false option trades in the company's trading system—the receipt of a large premium and the payment of a large premium. The first option would expire the day of the trade, the other one later. Allfirst had no reports on options that expired the same day they were purchased and so did not detect what Rusnak was doing. The second option created a false asset on the company's books. This offset the real losing position that Rusnak wished to hide.

Initially, Rusnak used fake broker confirmations to validate his fictitious deals. This was risky because the back office staff reconciled trades with receipts. However, Rusnak convinced back-office personnel that they did not have to confirm the trades because they were offsetting deals with no transfer of cash.

In 2001, the head of treasury funds at Allfirst noted that Rusnak's trades were using up an unusually large portion of his balance sheet and that this was disproportionate to his earnings. He ordered Rusnak to reduce his exposure on the balance sheet. Rusnak did so, but he accomplished this using highly risky trades that saddled the company with massive potential liabilities.

One control at Allfirst was to compute a value-at-risk (VaR) ratio for each trader. The data for these calculations were supposed to have been computed independently by the back office staff, but Rusnak was able to persuade them to use data on his computer. He manipulated this data to make his VaR ratio look acceptable.

The fraud came apart when a back office supervisor noticed that Rusnak's trades were not being confirmed as required by procedures. The supervisor discovered that a number of trades were clearly bogus. He notified management of the problem. The fraud quickly unraveled.

Rusnak eventually entered into a plea agreement that got him seven years in jail. This relatively light sentence was a result of his agreeing to work with prosecutors to prosecute people in other companies whose actions prolonged the time it took for Rusnak's scheme to unravel (BBC, 2002).

SPREADSHEETS AND ACTUAL SOX DEFICIENCY REPORTS

Although it seems difficult to ignore the specter of spreadsheet error and fraud control deficiency, many firms have dismissed these issues as "theoretical." Based on the first round of Sarbanes–Oxley reports, this no longer seems to be an intelligent reaction.

Although most first-round Sarbanes–Oxley assessments only focused on the most glaring weaknesses, a number of firms did report material deficiencies because of spreadsheets. In 2005, RSM McGladrey studied the details of initial assessment reports and summarized this research by saying that "numerous" companies had already cited deficiencies because of spreadsheets (Kelly, 2005). Weaknesses included both operational control deficiencies and errors.

Jack Ciesielski, publisher of the Analyst's Accounting Observer, reported several specific cases of deficiency reports related to spreadsheets (MacDonald, 2005). Unfortunately, the wording in SEC filings typically is extremely vague. For instance, Eastman-Kodak merely reported that an major error in its financial reporting resulted from a "failure" in its "preventive and detective controls surrounding the preparation and review of spreadsheets that include now or changed formulas" (MacDonald, 2005). In many cases, reports are so vague that they do not even mention specific technologies, making it impossible to draw conclusions about the prevalence of spreadsheet errors and control weaknesses.

V. LEGISLATION: SOX AND BEYOND

SARBANES–OXLEY (SOX) AND FINANCIAL FRAUD

We have seen that the Sarbanes–Oxley Act of 2002, also known as SOX, requires the senior executives of U.S. firms and the many foreign firms listed on U.S. stock exchanges to have effective controls for their financial reporting sys-

tems and to report on the effectiveness of these controls. SOX also requires firms to hire an external auditor to assess their assessments. For most large firms, the deadline to implement effective controls either has passed or will pass soon.

SOX gave the Securities and Exchange Commission (SEC) overall responsibility for implementing the law. The SEC, in turn, created the Public Company Accounting Oversight Board (PCAOB) to develop specific rules and oversight functions to implement independent audits under Sarbanes-Oxley.

Although SOX was a response to specific high-profile cases of fraud, fraud in financial reporting has long been a major problem. In 2004, fraud through financial statements represented only 7% of all fraud cases studied in an ACFE survey (ACFE, 2004), but the median fraud loss for financial fraud was a million dollars, compared to a median loss of only $100,000 for frauds in general.

OTHER FINANCIAL REGULATIONS

Although SOX has received the most attention, a number of other recent pieces of legislation have also required corporations to reconsider their financial systems and other information systems.

SEC Accelerated Filing Deadlines

Since December 2002, the Securities and Exchange Commission has required firms to reduce the time they may take to produce their quarterly and annual reports. These tighter time limits will make designing controls more difficult because there will be less time to check for errors and violations.

SEC Accelerated Filing Deadlines

Since December 2002, the Securities and Exchange Commission has required firms to reduce the time they may take to produce their quarterly and annual reports. These tighter time limits will make designing controls more difficult because there will be less time to check for errors and violations.

IAS/IFRS

U.S. accounting standards are set by the U.S. Federal Accounting Standards Board (FASB), which creates the generally accepted accounting practices (GAAP). In turn, the International Accounting Standards, including the finance-specific International Financial Reporting Standards (IFRS), govern U.S. firms operating in Europe,.

SAS 99

In 2002, shortly after Sarbanes-Oxley was enacted, the Auditing Standards Board produced Statement on Auditing Standards 99 (SAS 99), Consideration of Fraud in a Financial Statement Audit. As its name suggests, this standard requires auditors to search aggressively for fraud.

Basel II

Banks who do business internationally will also have to comply with the Basel II accord. Basel II requires banks to maintain sufficient capital reserves to cover probable risks. Banks that do not have controlled financial reporting systems or risk controls in place must set aside more capital reserves to reflect the risk raised by inadequate control. This reduces the amount of loans they can support, which in turn limits profits. Basel II gives banks a direct incentive to invest in internal controls to reduce risks.

PRIVACY LAWS

Several laws now regulate privacy and the disclosure of private information, and the number of privacy regulations is increasing. Key regulations include the following (among others):

- The European Union (E. U.) Data Protection Directive of 2002, which is a broad set of rules ensuring privacy rights in Europe.
- Although the E. U. Data Protection Directive is the most important international privacy rule, many other nations with which U.S. firms do business are also developing strong commercial data privacy laws.
- The U.S. Gramm-Leach-Bliley Act (GLBA) of 1999 requires strong privacy protection in financial institutions.
- The U.S. Health Information Portability and Accountability Act (HIPAA) of 1996 requires strong protection for private data in health care organizations.

On the other hand, the U.S. Patriot Act of 2001 gives the U.S. government broad powers to see personal data. In some cases, there actually are conflicts between laws requiring the maintenance of privacy and laws mandating government access.

EARLIER COMPLIANCE ISSUES

Although Sarbanes-Oxley is a recent concern, the control of spreadsheets has been an issue for some time.

21 CFR Part 11 in the Medical Industry

In 1997, the U.S. Food and Drug Administration (FDA) published the Code of Federal Regulation, Title 21, Part 11, Electronic Records; Electronic Signatures. Better known as 21 CFR Part 11, this regulation mandates controls over electronic documents in pharmaceuticals and other medical industries, especially in research and development for new drugs. It came into effect in August, 1997. The 21 CFR Part 11 requires electronic signatures, limiting access to authorized individuals, operational system checks, device checks, authorization checks, written policies on accountability, education, appropriate experience, audit trails, records retention, and controls over system documentation.

The 21 CFR Part 11 regulations are relevant to spreadsheet controls because pharmaceuticals other medical companies have long used spreadsheets in regulated activities. Consequently, a number of vendors have developed software to protect repositories of spreadsheets from security violations (although not from errors). These products can serve a similar function in Sarbanes-Oxley compliance.

Y2K

As January 1, 2000 approached, corporations around the world had to remediate important computer systems for compliance with date problems. These systems included critical spreadsheet models. A number of software products were created to aid Y2K spreadsheet remediation. Some of these tools are applicable to compliance issues. For instance, SCANXLS is a program that can search a network for spreadsheet models and provide summaries of discovered spreadsheets. Knowing what spreadsheets a firm has obviously is a critical first step in developing an understanding of spreadsheet risks, and it is critical in creating actual controls.

DAYLIGHT SAVINGS TIME

At the time of this writing, Congress is considering extending the daylight savings time period. This could create an enormous number of date problems for spreadsheets and other software.

INDUSTRY-SPECIFIC ACCREDITATIONS

In addition to government regulations, many industries have specific accreditation bodies. To be accredited, a firm in the industry usually must comply with numerous requirements. Some of these requirements involve IT controls.

THE COMPLIANCE AGE

For working IT professionals, complying with regulations has already become a very important concern and is likely to continue to grow in importance. Quite simply, IT has entered the compliance age. Unfortunately, IS education has not kept pace with the growing importance of compliance in IT management.

VI. CONTROL FRAMEWORKS

To achieve compliance with SOX and other crucial regulations, companies typically adopt or are required to adopt a control framework. Control frameworks specify the actions that they need to take and how to take these actions.

TYPES OF CONTROLS

We saw in Figure 1 that the purpose of controls is to help organizations keep their organizational processes on track to achieve their firms' goals. Figure 1 specifically shows that controls generally fall into one of three categories.

- Preventive controls attempt to keep deviations from occurring in the first place. In movie theaters, for example, one person sells tickets but another collects them. This is the segregation of duties. Unless the two parties collude, the person accepting the money for tickets cannot collect money, pocket it, and then allow the moviegoer in without giving him or her a ticket.

- Detective controls attempt to detect deviations when they occur, so that action can be taken. Periodic reconciliations between independent processes will make it likely that deviations in one of the processes will be revealed. In the case of movie theaters, management reconciles the number of tickets sold with the number of tickets collected at the end of each day.

- Corrective controls actually fix deviations. The restoration of backup files on a computer compromised by an attack is a corrective control.

While the general taxonomy of preventative, detective, and corrective controls is useful in practice, it is not perfect. For instance, if people realize that detective controls are in place, this may deter them from misbehavior, so the measure would actually be a preventative control.

COSO

For Sarbanes-Oxley, the PCAOB explicitly requires corporations to use a well-developed comprehensive control framework. Although the PCAOB does not require corporations to use a specific framework, it has specifically listed only a single framework as acceptable, and most companies are using this framework to implement SOX. This is the COSO framework.

The COSO Framework

Although COSO is universally known by its acronym, the COSO framework actually is a document called Internal Control—Internal Framework (COSO, 1994). The acronym COSO comes from the organization that created the document, the Committee of Sponsoring Organizations of the Treadway Commission (www.coso.org).

Objectives

Control frameworks require objectives. In the COSO framework, there are three objectives.

- **Operations.** The firm wishes to operate effectively and efficiently. It is necessary for the firm to control its general internal operations to do this.
- **Financial Reporting.** The firm must create accurate financial reports. This, of course, is the focus of Sarbanes–Oxley.
- **Compliance.** The firm wishes to be in compliance with external regulations. In this paper, we are only directly concerned with SOX compliance.

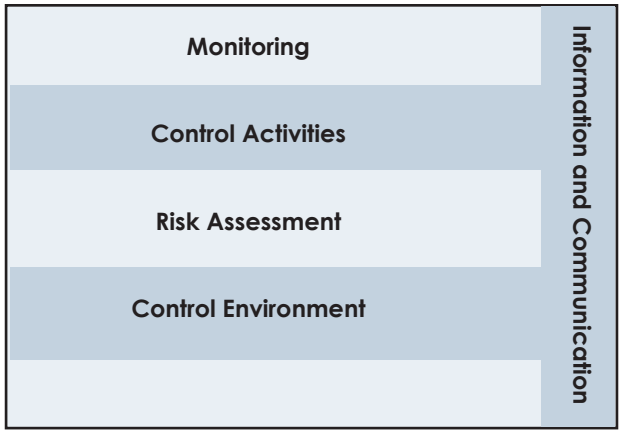
Reasonable Assurance

Good controls cannot completely guarantee that goals will be met. However, an effective control environment will give reasonable assurance that goals will be met.

COSO Framework Components

Figure 5 shows the COSO framework. It shows that the framework has five components. These are components rather than phases because there is no time ordering among them. All must occur simultaneously, and each feeds into others constantly.

Figure 5: The COSO Framework
Source: Panco (2005c)



- **Control Environment.** The component at the base of the COSO framework is the corporation's control environment. This is the company's overall control culture. It includes the "tone at the top" set by top management, the company's commitment to training employees in the importance of control, the punishment of employees (including senior managers) who violate control rules, attention by the board of directors, and other broad matters. If the broad control environment is weak, other control elements are not likely to be effective.
- **Risk Assessment.** More specifically, a company needs to assess the risks that it faces. Without systematic risk analysis, it is impossible to understand what level of controls to apply to individual assets. Risk assessment must be an ongoing preoccupation for the firm because the risk environment constantly changes.

- **Control Activities.** An organization will spend most of its control effort on the control activities that actually implement and maintain controls. This includes approvals and authorization, IT security, the separation of duties, and many other matters. Controls usually have two elements. One is a general policy, which tells what must be done. The other is a set of procedures, which tell how to do it.

- **Monitoring.** Having controls in place means nothing if organizations do not monitor and enforce them. Monitoring includes both human vigilance and audit trails in information technology. It is essential to have an independent monitoring function that is free to report on problems even if these problems deal with senior management.

- **Information and Communication.** For the control environment, risk assessment, control activities, and monitoring to work well, the company needs to ensure that it has the required information and communication across all levels of the corporations.

Types of Control Activities

Internal Control—Internal Framework (COSO, 1994) does not list a comprehensive set of control activities, probably because it is impossible to create a complete list of potential controls. However, the document does provide several lists of types of control activities. For instance, on Page 49, the framework notes the existence of manual controls, computer controls, and management controls. On the following page, it provides the following list to consider:

- **Top Level Review**—comparing budgets with actual performance, tightly monitoring major initiatives, and so forth.

- **Direct Functional or Activity Management**—managers who run individual operations must examine the appropriate reports for their level, for instance, loan performance in a bank's lending operations.

- **Information Processing**, including the enforcement of manual procedures, such as checking if a customer's accounts payable value is below a certain amount before accepting an order. Note that information processing must focus on business processes, not merely on IT processes.

- **Physical Controls**—inventories, locked cash drawers, write-only archival media, and so forth.

- **Performance Indicators**—relating different sets of data to each other to check for inconsistencies, noting deviations from normal performance (in either direction), unusual trends, and so forth.

- **Segregation of Duties**—requiring sensitive processes to be completed by two or more people, so that no single person can engage in improper activities without this becoming apparent. Earlier, we saw how movie theaters do this. To give another example, it is normal to require that one person may purchase and order, but another person will record it. It is also normal to ensure that no single person can both authorize and make a purchase.

Controls for Information Systems

On Pages 52-55, *Internal Control—Internal Framework* specifically lists some controls over information systems. At a most basic level, the framework discusses the differences between application controls and general controls.

Application Controls

Application controls, as the name suggests, involve individual applications (accounting applications, spreadsheets, and so forth), including manual operations in using them. This includes interfaces to other systems for input data,

checks on input, and internal checks during processing to flag errors, misbehavior, and other problems.

General Controls
General controls cover everything beneath the applications—computers, operating systems, the network, and so forth, together with manual operations in using them. This includes making purchases, application systems development, maintenance, access controls, evaluating packets software, and so forth. The controls needed in individual applications will depend on the quality of general controls.

Controls for "Evolving Issues"

The report spends approximately half of Page 55 on "evolving issues." Only two brief paragraphs are devoted to end-user computing (EUC). The first paragraph simply says that EUC exists. The second gives the following meager guidance:

To provide needed control for EUC systems, entity-wide policies for system development, maintenance, and operation should be implemented and enforced. Local processing environments should be governed by a level of control activities similar to the more traditional mainframe environment. (COSO, 1994, p. 55)

Internal Control–Integrated Framework does not give *any* specific guidance on spreadsheets. In fact, it does not even mention them. In general, the framework is an old (1994) document that was written before spreadsheets became important, or, probably more accurately, before IT control professionals realized that spreadsheets were important.

CobiT

COSO is a general control planning and assessment tool for corporations. For IT controls, there is a more specific framework, CobiT (*Control Objectives for Information and Related Technologies*) (IT Governance Institute, 2000). In addition to creating the broad control objectives framework, the IT Governance Institute also has developed detailed guidance for implementing the CobiT framework.

The CobiT Framework

Figure 6 illustrates the CobiT framework. This framework has four major domains, which follow the general systems development life cycle:

Figure 6: COSO/CobiT Framework
Source: IT Governance Institute (204), Page 50

			COSO Components				
Corporate Level	**Activity Level**	**CobiT Objectives**	**Control Environ.**	**Risk Assess.**	**Control Activities**	**Info & Comm.**	**Monitoring**
		Planning and Organization					
x		IT strategic plan	x	x		x	x
x		Information architecture			x	x	x
		Technological direction					
x		IT organization/relationships	x			x	
		Manage IT investment					
x		Communication aims & directions	x			x	x
x		Manage human resources	x			x	
x		Ensure compliance				x	x
x		Assess risks		x			
		Manage quality	x		x	x	x
		Acquisition and Implementation					
		Identify automated solutions					
	x	Acquire/Develop app. software			x		
	x	Acquire technological infrastructure			x		
	x	Develop & maintain procedures			x		x
	x	Install and test systems			x		
	x	Manage changes			x		x
		Delivery and Support					
	x	Define and manage service levels	x		x		x
	x	Manage third-party services	x	x	x		x
x		Manage performance and capacity			x		x
		Ensure continuous service					x
	x	Ensure systems security			x	x	x
		Identify and allocate costs					
x		Educate and train users	x			x	
		Assist and advise customers					
	x	Manage the configuration		x	x		
	x	Manage problems and incidents			x	x	x
	x	Manage data			x	x	
x		Manage facilities		x			
	x	Manage operations			x	x	
		Monitoring					
x		Monitor the process				x	x
x		Assess internal control adequacy					x
x		Obtain Independent assurance	x				x
x		Provide for independent autiditing					

- **Planning and Organization**. The planning and organization domain has 11 high-level control objectives that cover everything from strategic IT planning and the creation of a corporate information architecture to the management of specific projects.

- **Acquisition and Implementation.** After planning, companies need to acquire and implement information systems. This domain has six high-level control objectives.

- **Delivery and Support.** Most of an IT project's life takes place after implementation. Consequently, the CobiT framework has 13 high-level control objectives for delivery and support. This is more than any other domain.

- **Monitoring.** Finally, firms must monitor their processes, assess the adequacy of internal controls, obtain independent assurance, and provide for independent auditing.

Although the four domains define the scope of CobiT, they are only the beginning of CobiT. Beneath the four domains are 34 high-level control objectives, which Figure 6 also shows. Beneath these, in turn, are more than 300 detailed control objectives. CobiT also includes many documents that help organizations understand how to implement the framework.

Dominance in the United States

The IT Governance Institute was created by the Information Systems Audit and Control Association (ISACA). ISACA, in turn, is the primary professional association for IT audit professionals in the United States. The Association's certified information systems auditor (CISA) certification is the dominant certification for U.S. IS auditors, so it is not surprising that CobiT has become the dominant framework for auditing IT controls in the United States.

COSO and CobiT

Obviously, both COSO and CobiT pertain to information technology used in financial reporting.

Figure 6 shows how CobiT relates to COSO at a broad level. This figure illustrates that it is relatively simple to combine COSO with CobiT at a conceptual level, although the details are anything but simple.

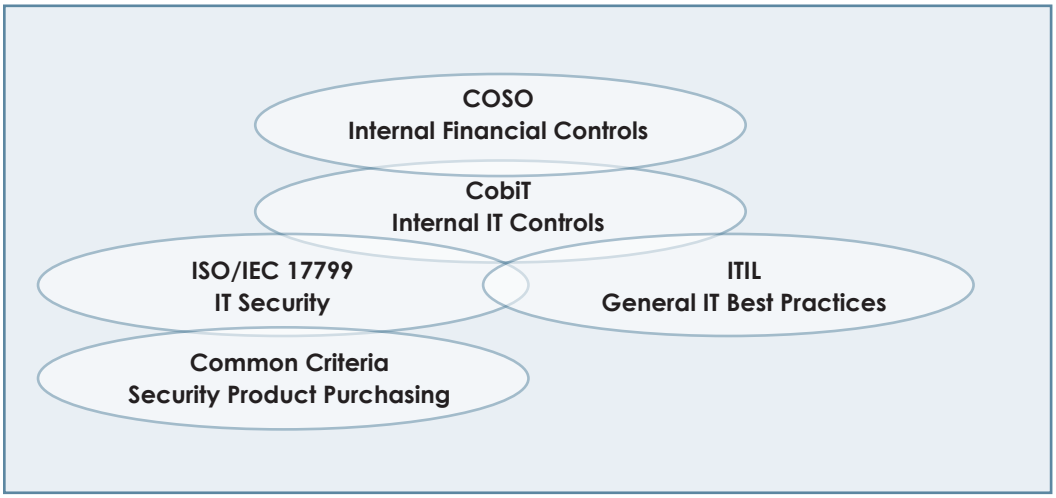
Actually, Figure 6, which was produced by the IT Governance Institute (2004), reflects the 2000 CobiT 3 framework rather than the new 2005 CobiT4 framework. At the level of top-level domains, there are only a few changes, and these could be easily mapped into Figure 6.

OTHER FRAMEWORKS

Although COSO and CobiT have dominated Sarbanes–Oxley planning in the United States, several other frameworks are also important.

Figure 7 shows the general relationship between COSO, CobiT and three other frameworks—ISO/IEC 17799, Common Criteria, and ITIL. The figure emphasizes that these frameworks overlap but focus on somewhat different areas. For example, CobiT, as its name implies, focuses specifically on controlling the entire IT process, while COSO focuses on internal financial reporting controls.

Figure 7: COSO, CobiT, ISO/IEC 17799, Common Criteria, and ITIL
Source: Panko (2005c).



ISO/IEC 17799

In contrast, ISO/IEC 17799, *Information technology—Security techniques—Code of practice for information security management* (ISO/IEC 17799:2005) focuses more narrowly on IT security. Security is part of IT controls, of course, so 17799 can help in creating IT controls.

ISO/IEC 17799 grew out of an earlier standards effort by the British Standards Institute. In 1995, the Institute produced BS 7799. This standard has two parts. ISO and the IEC adopted Part I as 17799 (adding a 1 before the BSI designation). This first part is a broad code of practice. ISO/IEC 17799 divides security into eleven broad areas, which is subdivides in many more specific elements:

- security policy;
- organization of information security;
- asset management;
- human resources security;
- physical and environmental security;
- communications and operations management;
- access control;
- information systems acquisition, development and maintenance;
- information security incident management;
- business continuity management; and
- compliance.

For organizations wishing certification of their standards effort, Part II of 7799 (Information Security Management System) has auditable controls. Consequently, many companies have chosen to be compliant with 17799 by being certified in Part II of 7799. In 2005, when in updated 17799, *ISO produced ISO/IEC 27001:2005, Information technology—Security techniques—Information security management systems— Requirements.* This standard is based on Part II of 7799. Implementation details are now being developed.

In other frameworks, including COSO and CobiT, companies certify themselves, sometimes with the concurrence of an external auditor. They lack 17799's third-party certification process, which external parties may value highly.

Common Criteria

Figure 7 shows that the Common Criteria (ISO/IEC 15408, Information technology–Security techniques–Evaluation criteria for IT security) standard is even more specific. Common Criteria focuses on the evaluation of security products, such as firewalls. It provides a way for purchasers to know specifically which security features a security product claims to offer and how rigorously the product was developed. However, the Common Criteria approach has somewhat limited use because it is difficult to apply and does not provide a high level of assurance that a product actually is secure.

ITIL

Another framework for IT is the Information Technology Infrastructure Library (ITIL). ITIL is a broad set of best practice guidelines for providing IT services. It is widely used in Europe and is becoming popular elsewhere. ITIL is highly process-oriented, specifying systematic approaches to implementing security and other IT services. ITIL best practices may be helpful in implementing other guidelines. However, ITIL does not have the detailed guidance necessary for developing and implementing IT financial reporting controls.

PRICEWATERHOUSECOOPERS ON SPREADSHEETS AND SARBANES–OXLEY

If one does an Internet search with the terms "spreadsheet" and "Sarbanes," there will be many hits. Nearly all of these, however, will be about spreadsheets used to document SOX compliance, not about how to control spreadsheets used in financial reporting.

PriceWaterhouseCoopers

The one major exception to this silence on spreadsheet control for Sarbanes-Oxley is a six-page report by PriceWaterhouseCoopers (2004). This report lists a large number of controls.

- The first control step is to inventory all of a firm's spreadsheets that are "in scope," that is, are used in financial reporting.

- The next steps are to evaluate the riskiness of these spreadsheets, determine necessary controls, evaluate the existing (as-is) controls on these spreadsheets, and develop action plans for remediating control deficiencies.

Other Controls

The report also lists a number of controls that should be considered to mitigate risks inherent in the spreadsheet environment.

- Change control: The authorization of change requests, testing the spreadsheet, and formal sign-off by another individual.

- Version control: Ensure that only the current and approved version of each spreadsheet is used. Naming conventions that include version numbers, dates, and the use of structured directories can help in this.

- Access control: Assign appropriate access rights to people who need to use the spreadsheet. Use a password to control access.

- Input: Whether manual or automatic data entry is used, there should be numerical reconciliations.

- Security and data integrity: Store spreadsheets in protected directories and lock formula cells to prevent logic changes.

- Documentation: Ensure that the business objective and functions of the spreadsheets are understandable.

- Development life cycle: Use a standard systems development life cycle. The report specifically says that testing is critical (although it does not discuss how to do testing).

- Backup and Archiving: Spreadsheets should be backed up because of their sensitivity. They should be archived in read-only format for later review.

- Logic inspection. The company should have logic inspection done by an independent person other than the developer. The report does not discuss how logic inspection is different from testing or discuss how it should be done.

- Segregation of duties/roles, and procedures: The company should define authorities, roles, and procedures for ownership, sign-off, and other matters. This item is so brief in the report that it provides little guidance.

- Analytics: The firm should require that calculations be built into spreadsheets (ratios, cross-checks, statistical patterns, etc.) that help detect errors and fraud.

The report says that firms should enforce these controls. For instance, changes should be made independently in two copies of each spreadsheet, and the two copies should be compared. In another example, a sample of cells can be tested to ensure that they are password-protected if they should be. In addition, names should include modification dates, and names should be compared with modification dates as recorded by the operating system.

Finally, the report gives a number of suggestions for remediation. Specific responsibilities should be assigned to specific people, remediation efforts should be prioritized, and remediation deadlines should be established.

ISSUES WITH THE PWC SPREADSHEET FRAMEWORK

Although the PriceWaterhouseCoopers report is extremely valuable, it has a number of weaknesses that need to be noted.

Lack of Detail

One obvious problem with the PWC advice is its lack of detail. It mentions many actions but does not explain what these actions mean or how to implement them. The PWC framework is a beginning but only a beginning.

A General Lack of Empirical Justification

A systematic problem with the PWC framework is that it does not seem to be aware of empirical spreadsheet research. This creates a number of potential problems that we will discuss in the following sections.

Modular Design

The PWC framework argues for a modular design—breaking the spreadsheet into a number of reasonably self-contained modules. However, modularization is a complex topic. For instance, some common advice on how to build modules emphasizes the desirability of placing all input data in one section, all processing in a second section, and all data output calculations in a final section. However, this IPO approach makes the processing section difficult to read, and it necessitates that formulas will require pointing to numbers far away. This can lead to an increased number of pointing errors. The alternative to input-processing-output (IPO) is to have a basic top-to-bottom design in which numbers are presented in the context of previous numbers and calculations. (If you use a tax preparation program, you will note that input is put in the context of a stream of calculations rather than all at the front.) The organization of modules is very much an open research question.

Assessing Risk: Length versus Complexity

One concern is the report's method for assessing riskiness. It lists nine factors to consider when evaluating the "risk and significance" of a spreadsheet:

- Complexity of the spreadsheet and calculations.
- Purpose and use of the spreadsheet.
- Number of spreadsheet users.
- Types of potential input, logic, and interface errors.
- Size of the spreadsheet.
- Degree of understanding and documentation of the spreadsheet requirements by the developer.
- Uses of the spreadsheet's output.
- Frequency and extent of changes and modifications to the spreadsheet.
- Development, including testing of the spreadsheet before it is utilized.

Although this is a good list generally, some of it reflects an incorrect view of spreadsheet errors. Research indicates that the largest indicator of the number of errors in a spreadsheet is simply the length of the spreadsheet. There will be errors in about 2% of all formula cells if there is no deep testing. This also is the case in programming. From programming, we know that more complex programs have more errors than simpler programs, although only by up to a factor of four (Zage and Zage, 1993). Consequently, long simple spreadsheets will have many more errors than short complex spreadsheets. Certainly, reporting complexity first and size half way down the list is a concern.

Testing

Even more of a concern is testing, including logic inspection. The framework gives almost no information on this topic. In fact, its advice on logic inspection—that logic should be done by a person (one person) other than the developer—flies in the face of what software developers have long known about code inspection in software development—that a single inspector will not find a high percentage of all errors. This inability of single testers to find a large percentage of errors also has been replicated in spreadsheet inspection experiments (Galletta, et al., 1993, 1997; Panko, 1999). Fagan (1976) first argued that multi-person code inspection is needed, and subsequent corporate experience has confirmed this.

In general, the PWC framework, like most other discussions of good spreadsheet practice today, spends almost no time on testing, treating it as just one of many considerations. However, software development experience has shown that testing is very difficult. As a consequence, a great deal of software development effort is spent in testing.

Fault rates in programming are very similar to error rates in spreadsheet models (cf. Panko, 2006a 2006b), as are detection rates in inspection (Galletta et al., 1993, 1997; Panko, 1999, cf. Panko 2006a). Programmers, appalled by actual fault rates (in programming, defects in the program are called faults), spend a great deal of time on testing. In a sample of 84 projects in 27 organizations, Jones (1998) found that the amount of time spent in testing to reduce errors ranged from 27% to 34%, depending on program difficulty. In every case, furthermore, subjects reported that insufficient time was allotted to testing. In another study, Kimberland (2004) found that Microsoft software development teams spent 40% to 60% of their total working time in testing.

Testing in programming is not simply one of many controls. It is the main control. Although good practice in defining, designing, and developing programs are all important, the residual error rate after good development still requires extensive testing.

Another problem with the PWC framework is that it assumes a system development life cycle in which testing comes after development is finished. In software development, however, testing is done after each module, not at the end of development. Although unit testing for modules is only one stage in testing, it is a critical stage. The longer a piece of tested code is, the more difficult testing it will be. For instance, in code inspection, in which a testing team pours over a module of code, there is strong empirical support for keeping modules very small—only 100 to 200 lines

of code. As module length increases, error detection rates fall precipitously. For example, Barnard and Price (1994) found that inspectors found 72% more errors when modules were smaller.

Testing, then, proceeds throughout development. It is not a separate stage in the systems development life cycle. In addition, it is important to test each module during development.

Another issue is that the PWC framework describes both testing and inspection but does not discuss how they are different. In testing, different sets of values for input variables are applied to the program (or spreadsheet), and the results are observed and checked. In inspection, the inspectors look over a program or spreadsheet line by line (or cell by cell) looking for errors. Both testing and inspection have strengths and weaknesses. However, although testing may seem like it would be simple to do, designing effective value sets for testing is surprisingly difficult (Glass, 2003). In contrast, code inspection does not require specific skills, although it is important that inspectors follow a step-by-step methodology (Fagan, 1976). If done correctly, both testing and inspection are capable of reducing error rates by 60% to 90%. However, if testing values are not selected very well, error reduction suffers greatly.

A third way to examine spreadsheets is auditing. In contrast to testing and inspection, which are designed to reduce errors heavily, auditing is primarily done to see if good practices were followed. Auditing only examines certain parts of the spreadsheet, so it is only likely to find a small fraction of all errors. For compliance and corporate use, which both require greatly reduced error rates, auditing is completely inadequate.

VII. CONCLUSION: THE SPREADSHEET COMPLIANCE PARADOX

Today, we have a puzzling situation. First, spreadsheets are widely used in financial reporting and many other business functions.

Second, given the compliance requirements of Sarbanes-Oxley and several other laws, having even a few spreadsheets that are incorrect to a material degree can have severe consequences for the firm.

Third, there is no doubt that human error rates in spreadsheet development are similar to those in software development and other human cognitive activities of comparable complexity. Even after careful development, we must expect to have errors in 1% to 5% of all lines of code. This means that all large spreadsheets have many errors. We also know that nearly all large spreadsheets have serious errors that reach the level of accounting materiality (Coopers and Lybrand, 1997) or decision impact (KPMG, 1998).

Fourth, as noted earlier in the text, a number of companies have already had to report control and error weaknesses associated with spreadsheets.

Given these facts, we would expect corporations and regulators to pay very close attention to at spreadsheet development, including implementing extensive testing. Paradoxically, however, corporations are not imposing extensive testing and other development requirements on their spreadsheets, and regulators are not insisting on such measures. What we appear to have is sham compliance in which spreadsheets are not effectively controlled yet are being treated by everyone as if they are.

However, this era of benign neglect in Sarbanes–Oxley may not continue for long. In the pharmaceuticals industry, 21 CFR 11 regulations have imposed control requirements on spreadsheets since 1997. In this realm, a period of neglect was followed by a testing requirement, albeit a fairly loose testing requirement. Quite a few companies have already received 21 CFR 11 warning letters from the Food and Drug Administration. Although the term "warning letter" may not sound serious, receiving a warning letter is a major concern for corporations in the industry and can have a devastating impact on stock prices.

Of course, error testing is not the only concern for spreadsheet compliance. Fraud controls are also crucial, and this is an area that has received even less attention. Operational procedures, auditing, documentation methods, and secure spreadsheet operations all need to be developed. Fortunately, some of the technological solutions developed for 21 CFR 11 compliance programs in pharmaceuticals may be of help in spreadsheet security. However,

operational procedures, documentation methods, and human controls are still badly in need of development.

VIII. ACKNOWLEDGEMENT

This paper is based in large part on Panko, Raymond R. (2005, July 7/8), "Sarbanes–Oxley: What about All the Spreadsheets? Controlling for Errors and Fraud in Financial Reporting," *EuSpRIG 2005*, University of Greenwich, London, UK. European Spreadsheet Research Information Group. http://www.eusprig.org.

IX. REFERENCES

ACFE (2004). *2004 Report to the Nation on Occupational Fraud and Abuse*. Association of Certified Fraud Examiners: Austin, Texas.

Allwood, C. M. (1984). "Error Detection Processes in Statistical Problem Solving." *Cognitive Science*, 8(4), 413-437.

Auditing Standards Board (2002). "Statement on Auditing Standards 99 (SAS 99)." *Consideration of Fraud in a Financial Statement Audit*. New York: The American Institute of Certified Public Accountants.

Barnard, J., & Price, A. (1994). "Managing Code Inspection Information." *IEEE Software,* 11(2), 55-68.

Basili, V. R. & Perricone, B. T. (1993). "*Software Errors and Complexity: An Empirical Investigation.*" Software Engineering Metrics, Volume I: Measures and Validation. Ed. M. Sheppard. Berkshire, England: McGraw-Hill International, 168-183.

Basili, V. R. & Selby, R. W., Jr. (1986). "Four Applications of a Software Data Collection and Analysis Methodology." *Software System Design Methods*. Ed. J. K. Skwirzynski. Berlin: Springer-Verlag. 3-33.

BBC (2002, October 24). " 'Rogue' AIB Trader Pleads Guilty to Fraud." http://news.bbc.co.uk/1/hi/business/2358463.stm.

Beizer, B. (1990). *Software Testing Techniques.* 2nd ed. New York: Van Nostrand.

Bereiter, C., & Scardamalia, M. (1993). *Surpassing Ourselves: An Inquiry into the Nature and Implications of Expertise*. Chicago: Open Court.

Boehm, B. & Basili, V. R. (2001, January). "Software Defect Reduction Top 10." *Computer,* 135-137.

Brown, P. S., & Gould, J. D. (1987). "An Experimental Study of People Creating Spreadsheets." *ACM Transactions on Office Information Systems*, 5(3), 258-272.

Bush, M. (1990, April 19). "Formal Inspection Processes—Do They Really Help?" *NSIA Sixth Annual National Joint Conference on Software Quality and Productivity*, Williamsburg, VA. Cited in Ebenau & Strauss (1994).

Butler, R. J. (2000, January 4-7). "Is this Spreadsheet a Tax Evader? How HM Customs & Excise Tax Test Spreadsheet Applications." *Proceedings of the Thirty-Third Hawaii International Conference on System Sciences*, Maui, Hawai`i.

Cale, E. G., Jr. (1994). "Quality Issues for End-User Developed Software." *Journal of Systems Management*, 45(1), 36-39.

CeBASE (2001, November 12). "eWorkshop 3." http://www.cebase.org/www/researchActivities/defectReduction/eworkshop3/item_6.htm.

Chan, Yolande E., & Storey, Veda C. (1996, December). "The Use of Spreadsheets in Organizations: Determinants and Consequences." *Information & Management,* 31(3), 119-134.

Clermont, M., Hanin, C., & Mittermeier, R. (2000, July). "A Spreadsheet Auditing Tool Evaluated in an Industrial Context." *Proceedings of the EuSpRIG 2000 Symposium, Spreadsheet Risks—the Hidden Corporate Gamble*. Greenwich, England: Greenwich University Press, 35-46.

Coopers & Lybrand in London. Description available at http://www.planningobjects.com/jungle1.htm. Contact information is available at that webpage.

COSO (Committee of Sponsoring Organizations of the Treadway Commission) (1994). *Internal Control—Internal Framework*. Available from www.aicpa.org.

CPS (2001). *Strategic Planning Survey: Results.* Corporate Performance Systems, Boston.

Cragg, P. G., & King, M. (1993). "Spreadsheet Modelling Abuse: An Opportunity for OR?" *Journal of the Operational Research Society*, 44(8), 743-752.

Daneman, M., & Stainton, M. (1993). "The Generation Effect in Reading and Proofreading: Is it Easier or Harder to Detect Errors in One's Own Writing?" *Reading and Writing: An Interdisciplinary Journal*, 5, 297-313.

Davies, N., & Ikin, C. (1987). "Auditing Spreadsheets." *Australian Account,* 54-56.

Dent, A. Personal communication with the author via electronic mail, April 2, 1995.

Durfee, Don (2004, July/August). "SPREADSHEET HELL?" *CFO Magazine*, CIO.com, http://www.cfoasia.com/archives/200409-07.htm.

Durfee, Don (2005, September 1). "The 411 on 404: Reporting Material Weaknesses in Control can Cost Shareholders Millions," CFO Magazine, CFO.com. http://www.cfo.com/article.cfm/4315498/c_4334841?f=magazine_alsoinside.

Ebenau, R. G., & Strauss, S. H. (1994). *Software Inspection Process.* New York: McGraw-Hill.

Endress, A. (1975). "An Analysis of Errors and Their Causes in System Programs." *IEEE Transactions on Software Engineering*, SE-1(2), 140-149.

Fagan, M. E. (1976). "Design and Code Inspections to Reduce Errors in Program Development." *IBM Systems Journal,* 15(3), 182-211.

Fernandez, K. (2002). "Investigation and Management of End User Computing Risk." Unpublished MSc thesis, University of Wales Institute Cardiff (UWIC) Business School.

Flower, L. A., & Hayes, J. R. (1980). "The Dynamics of Composing: Making Plans and Juggling Constraints." *Cognitive Processes in Writing*. Eds. L. W. Gregg & E. R. Steinberg. Hillsdale, NJ: Lawrence Erlbaum Associates. 31-50.

Floyd, B. D., Walls, J., & Marr, K. (1995). "Managing Spreadsheet Model Development." *Journal of Systems Management,* 46(1), 38-43, 68.

Gable, G., Yap, C. S., & Eng., M. N. (1991). "Spreadsheet Investment, Criticality, and Control." *Proceedings of the Twenty-Fourth Hawaii International Conference on System Sciences*, 3. Los Alomitos, CA: IEEE Computer Society Press. 153-162.

Galletta, D. F., & Hufnagel, E. M. (1992). "A Model of End-User Computing Policy: Context, Process, Content and Compliance." *Information and Management,* 22(1), 1-28.

Galletta, D. F., Abraham, D., El Louadi, M., Lekse, W., Pollailis, Y. A., & Sampler, J. L. (1993, April-June). "An Empirical Study of Spreadsheet Error-Finding Performance." *Journal of Accounting, Management, and Information Technology,* 3(2), 79-95.

Galletta, D. F.; Hartzel, K. S.; Johnson, S.; & Joseph, J. L. (1997, Winter) "Spreadsheet Presentation and Error Detec-

tion: An Experimental Study." *Journal of Management Information Systems*, 13(3).

Gosling, C. (2003). "To What Extent are Systems Design and Development Used in the Production of Non-Clinical Corporate Spreadsheets at a Large NHS Trust?" Unpublished MBA thesis, University of Wales Institute Cardiff (UWIC) Business School.

Grady, R. B. (1995). "Successfully Applying Software Metrics." *Communications of the ACM*, 38(3), 18-25.

Haley, T. J. (1996). "Software Process Improvement at Raytheon." *IEEE Software*, 13(4), 33-41.

Hall, A. (1996). "Using Formal Methods to Develop an ATC Information System." *IEEE Software*, 13(2), 66-76.

Hayes, J. R. & Flower, L. (1980). "Identifying the Organization of Writing Processes." *Cognitive Processes in Writing.* Eds. L. W. Gregg & E. R. Steinberg. Hillsdale NJ: Erlbaum. 31-50.

Healey, A. F. (1980). "Proofreading Errors on the Word The: New Evidence on Reading Units." *Journal of Experimental Psychology: Human Perception and Performance,* 6(1), 45-57.

Hendry, D. G. & Green, T. R. G. (1994). "Creating, Comprehending, and Explaining Spreadsheets: A Cognitive Interpretation of What Discretionary Users Think of the Spreadsheet Model." *International Journal of Human-Computer Studies*, 40(6), 1033-1065.

Hicks, L. (1995). NYNEX, personal communication via electronic mail.

Howarth, C. I. (1988). "The Relationship Between Objective Risk, Subjective Risk, and Behavior." *Ergonomics*, 31, 527-535.

International Organization for Standardization and the International Electrotechnical Commission (2005), ISO/IEC 17799, *Information technology— Security techniques—Code of practice for information security management, Geneva: ISO.*

International Organization for Standardization and the International Electrotechnical Commission (2005), ISO/IEC 27001:2005, *Information technology—Security techniques—Information security management systems— Requirements. Geneva: ISO.*

IT Governance Institute (2000). CobiT (*Control Objectives for Information and Related Technologies*). 3rd ed. 3701 Algonquin Road, Suite 1010, Rolling Meadows, IL, 60008. (www.itig.org).

Janvrin, D. & Morrison, J. (1996). "Factors Influencing Risks and Outcomes in End-User Development." *Proceedings of the Twenty-Ninth Hawaii International Conference on System Science*s, Maui, Hawai`i.

Johnson, P. & Tjahjono, D. (1997, May). "Exploring the Effectiveness of Formal Technical Review Factors with CSRS, A Collaborative Software Review System." *Proceedings of the 1977 International Conference on Software Engineering*, Boston, MA.

Jones, T. C. (1986a). *Programming Productivity*. New York: McGraw-Hill.

Jones, T. C. (1986b). "In-Process Inspections of Work Products at AT&T." *AT&T Technical Journal*, 106. Cited in Jones (1986a).

Jones, T. C. (1998). *Estimating Software Costs*. New York: McGraw-Hill.

Kellog, R. T. (1994). *The Psychology of Writing*. New York: Oxford University Press.

Kelly, Matt (2005, August 23). "Spreadsheet Blues: Few Controls Yield Many Weaknesses," *Compliance Week*. http://www.complianceweek.com.

Kimberland, K. (2004). "Microsoft's Pilot of TSP Yields Dramatic Results." news@sei, No. 2. http://www.sei.cmu.edu/news-at-sei/.

Klein, B. D., Goodhue, D. L. & Davis, G. B. (1997). "Can Humans Detect Errors in Data? Impact of Base Rates, Incentives, and Goals." *MIS Quarterly*, 21(2), 169-194.

KPMG Management Consulting (1998, July 30). "Supporting the Decision Maker—A Guide to the Value of Business Modeling." Press Release. http://www.kpmg.co.uk/uk/services/manage/press/970605a.html.

Lawrence, R. J. & Lee, J. (2004, August 26-27). "Financial Modelling of Project Financing Transactions." *Institute of Actuaries of Australia Financial Services Forum*.

Lerch, F. J. (1988). "Computerized Financial Planning: Discovering Cognitive Difficulties in Knowledge Building." Unpublished Ph.D. Dissertation, University of Michigan, Ann Arbor, MI.

Levy, B. A. & Begin, J. (1984). "Proofreading Familiar Text: Allocating Resources to Perceptual and Conceptual Processing." *Memory & Cognition*, 12(6), 621-632.

Levy, S. (1984, November). "People and Computers in Commerce: A Spreadsheet Way of Knowledge." *Harpers*, 18-26.

Lukasik, Todd, CPS. Personal communication via with the author, August 10, 1998.

Macdonald, Elizabeth, "Profit Pratfalls," *Forbes,* December 15, 2005. (Online. Available only to subscribers.)

Madachy, R. J. (1996, July). "Measuring Inspection at Litton." *Software Quality*, 2(4), 1-10.

Managing Office Technology (1994, February). "Enterprise-Wide Budgeting Demands Flexibility Beyond Spreadsheets." 40, 42.

McCormick, K. (1983, March). "Results of Code Inspection for the AT&T ICIS Project." Paper presented at the Second Annual Symposium on EDP Quality Assurance.

Myers, G. J. (1978, September). "A Controlled Experiment in Program Testing and Code Walkthroughs/Inspections." *Communications of the ACM*, 21(9), 760-768.

Nardi, B. A. (1993). *A Small Matter of Programming: Perspectives on End User Computing. Cambridge*, MA: MIT Press.

Nardi, B. A. & Miller, J. R. (1991). "Twinkling Lights and Nested Loops: Distributed Problem Solving and Spreadsheet Development." *International Journal of Man-Machine Studies*, 34(1), 161-168.

O'Neill, Don (1994, October). "National Software Quality Experiment." *Fourth International Conference on Software Quality Proceedings.*

Panko, R. R. (2006a). Human Error Website. http://panko.cba.hawaii.edu/HumanErr/. Honolulu, HI: University of Hawai`i.

Panko, R. R. (2006b). Spreadsheet Research (SSR) Website. http://panko.cba.hawaii.edu/ssr/. Honolulu, HI: University of Hawai`i.

Panko, R. R. (2006c). "Two Experiments in Reducing Overconfidence in Spreadsheet Development." Accepted for publication in the *Journal of Organizational and End User Computing*.

Panko, Raymond R. (2005, July 7/8), "Sarbanes–Oxley: What about All the Spreadsheets? Controlling for Errors and Fraud in Financial Reporting," *EuSpRIG* 2005, University of Greenwich, London, UK. European Spreadsheet Research Information Group. http://www.eusprig.org.

Panko, R. R. (2000, January). "Two Corpuses of Spreadsheet Errors." *Proceedings of the Thirty-Third Hawaii International Conference on System Sciences*, Maui, Hawai`i.

Panko, Raymond R. (1999, Fall). "Applying Code Inspection to Spreadsheet Testing." *Journal of Management Information Systems*, 16(2), 159-176.

Panko, R. R. (1988). End User Computing: Management, Applications, and Technology. New York: Wiley.

Panko, R. R. & Halverson, R. P., Jr. (1997). "Are Two Heads Better than One (At Reducing Errors in Spreadsheet Modeling?)" *Office Systems Research Journal*, 15(1), 21-32.

Public Company Accounting Oversight Board (2004, March 17). Standard No. 2: *An Audit of Internal Control Over Financial Reporting Performed in Conjunction with an Audit of Financial Statements.*

PriceWaterhouseCoopers (2004, July). "The Use of Spreadsheets: Considerations for Section 404 of the Sarbanes-Oxley." http://www.pwcglobal.com/extweb/service.nsf/8b9d788097dff3c9852565e00073c0ba/cd287e403c0aeb7185256f08007f8caa/$FILE/PwCwpSpreadsheet404Sarbox.pdf.

Putnam, L. H., & Myers, W. (1992). *Measures for Excellence: Reliable Software on Time, on Budget*. Englewood Cliffs, NJ: Yourdon.

Raffensperger, John F. (2000, 2003, July 15). "The New Guidelines for Writing Spreadsheets." http://www.mang.canterbury.ac.nz/people/jfraffen/spreadsheets/index.html.

Rankin, Ken. (2005, October 14). "PCAOB: 12% of Cos. Below Sox Standard," WebCPA™. http://www.webcpa.com/article.cfm?articleid=13699.

Rasmussen, J. (1974, May). "Mental Procedures in Real-Life Tasks: A Case Study of Electronic Troubleshooting." *Ergonomics*, 17(3), 293-307.

Reason, J. (1990). *Human Error*. Cambridge, U.K.: Cambridge University Press.

RevenueRecognition.com. "The Impact of Compliance on Finance Operations." *Financial Executive Benchmarking Survey: Compliance Edition*, 2004. (www.softtrax.com.)

Rickets, J. A. (1990, March). "Powers-of-Ten Information Biases." *MIS Quarterly*, 14(1), 63-77.

Russell, G. W. (1991). "Experience with Inspection in Ultralarge-Scale Developments." *IEEE Software*, 8(1), 25-31.

Saariluoma, P. (1989). "Visual Information Chunking in Spreadsheet Calculation." *International Journal of Man–Machine Studies*, 30, 475-488.

Schulmeyer, G. Gordon. Personal communication cited in Dobbins, James H., "CQA Inspection as an Up-Front Quality Technique." In G. Gordon Schulmeyer & James I. McManus, *Handbook of Software Quality Assurance* (217-253). Upper Saddle River, New Jersey: Prentice Hall.

Schultheis, R., & Sumner, M. (1994). "The Relationship of Application Risks to Application Controls: A Study of Microcomputer-Based Spreadsheet Applications." *Journal of End User Computing*, 6(2), 11-18.

Snyder, M. & Campbell, B. H. (1980). "Testing Hypotheses About Other People: The Role of the Hypothesis." *Personality and Social Psychology Bulletin*, 6, 421-426.

Speier, C. & Brown, C. V. (1996, January). "Perceived Risks and Management Actions: Differences in End-User Application Development Across Functional Groups." *Proceedings of the Twenty-Ninth Hawaii International Conference on System Science,* Maui, Hawai`i.

Spencer, B. (1993). "Software Inspection at Applicon." Software Inspection. Ed. T. G. D. Graham. Workingham, UK:

Addison-Wesley. 264-279.

Steiner, I. D. (1972). Group Processes and Productivity. New York: Academic Press.

Svenson, O. (1977). "Risks of Road Transportation from a Psychological Perspective: A Pilot Study." Report 3-77, *Project Risk Generation and Risk Assessment in a Social Perspective*. Committee for Future-Oriented Research, Stockholm, Sweden. Cited in Fuller, (1990).

Takaki, S. T. (2005). "Self-Efficacy and Overconfidence as Contributing Factors to Spreadsheet Development Errors." Working Paper. Information Technology Management Department, College of Business Administration, 2404 Maile Way, Honolulu, HI, 96822.

Teo, T. S. H., & Tan, M. (1997, January). "Quantitative and Qualitative Errors in Spreadsheet Development." *Proceedings of the Thirtieth Hawaii International Conference on System Sciences*, Maui, Hawai`i.

TMCnet.com (2004, September 20). "European Companies are taking a faltering approach to Sarbanes-Oxley." http://www.tmcnet.com/usubmit/2004/Sep/1074507.htm.

United States Department of Justice (2002). United States of America v. John M. Rusnak. SMS/SD/USAO #2002R02005. http://www.usdoj.gov/dag/cftf/chargingdocs/allfirst.pdf.

Vorhies, J. B (2005, May). "The New Importance of Materiality." *Journal of Accountancy (online)*. http://www.aicpa.org/pubs/jofa/may2005/vorhies.htm.

Weller, M. (1993). "Lessons from Three Years of Inspection Data." *IEEE Software*, 10(5), 38-45.

Zage, W. M. & Zage, D. M. (1993). "Evaluating Design Metrics in Large-Scale Software." *IEEE Software*, 10(4), 75-81.

Editor's Note: The following reference list contains hyperlinks to World Wide Web pages. Readers who have the ability to access the Web directly from their word processor or are reading the paper on the Web can gain direct access to these linked references. Readers are warned, however, that

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

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

3. the author(s) of the Web pages, not AIS, is (are) responsible for the accuracy of their content.

4. the author(s) of this article, not AIS, is (are) responsible for the accuracy of the URL and version information.

ABOUT THE AUTHOR

Dr. Raymond R. Panko is a professor of IT management in the College of Business Administration at the University of Hawai`i. He has been conducting research on spreadsheet development and testing since 1993. His home page is http://panko.cba.hawaii.edu. His Spreadsheet Research and Human Error websites are available through that home page.

*The low cell error rate probably reflects the fact that the methodology did not inspect all formulas in the spreadsheet but focused on higher-risk formulas. However, error has a strong random component, so the practice of not checking all formulas is likely to miss many errors.

**In 2003, the author spoke independently with experienced spreadsheet auditors in two different companies in the United Kingdom, where certain spreadsheets must be audited by law. Each company audited about three dozen spreadsheets per year. Both said that they had never seen a major spreadsheet that was free of errors. Both also indicated that about five percent of the spreadsheets they audited have very serious errors that would have had major ramifications had they not been caught. Audits were done by single auditors, so from the research on spreadsheet and software auditing, it is likely that half or fewer of the errors had been caught. In addition, virtually all of the spreadsheets had standard formats required for their specific legal purposes, so error rates may have been lower than they would be for purpose-built spreadsheet designs.