*Journal of Organizational and End User Computing 25(3), 2013[[1]](#footnote-1)*

End User Computing:  
The Dark Matter (and Dark Energy) of Corporate IT

Raymond R. Panko Daniel N. Port

University of Hawaii University of Hawaii

[panko@hawaii.edu](mailto:panko@hawaii.edu) [dport@hawaii.edu](mailto:dport@hawaii.edu)

Professionals Built the Titanic.  
Amateurs Built the Ark.

# Abstract

End user computing (EUC) is like dark matter in physics. EUC is enormous in quantity and importance yet has been largely invisible to corporate IT departments, information systems (IS) researchers, and corporate management. EUC applications, especially spreadsheet applications, are also “dark” in the sense that they pose a number of overlooked risks for organizations, including errors, privacy violations, trade secret extrusions, and compliance violations. On the positive side, EUC applications are also like the dark energy of physics. They are supporting critical gains in decision making, computing by scientists and engineers, operational systems, and other important processes in every corner of the firm. It is time to stop ignoring end user computing in general and spreadsheets in particular. The purpose of this paper is to document to the extent possible today then importance of end user computing relative to the concerns of corporate IT departments and IS researchers.

Keywords: End user computing, spreadsheet, spreadsheet error, human error, base error rate, cell error rate, mission-critical, descriptive research, primary descriptive research, use and user studies, user studies,3GL, 4GL.

# Introduction

In the 1930s, astrophysicists discovered that ordinary matter is only about a fifth of the total matter of the universe (NASA, 2010). The rest consists of *dark matter*, which is not simply too dim to see but actually is radically different from ordinary matter. In the last years of the 20th Century, astrophysicists received another shock. Something else, which physicists labeled *dark energy*, is pushing the universe to expand at an ever-increasing rate. This dark energy is almost three times as large as ordinary matter and dark matter combined (NASA, 2010). Traditional physics, instead of describing the universe, had only considered about five percent of it.

The contention of this paper is that end user computing (EUC) in general and spreadsheets in particular are the dark matter and dark energy of corporate information technology. EUC is like dark matter in the sense that it is enormous in size and impact yet seems to be invisible to the central corporate IT group, to general corporate management, and to information systems (IS) researchers.

EUC is also “dark” in another sense. We will see that research has shown that developers in departments outside the central IT departments develop many incorrect applications (Panko, 2012b). However, the problem does not appear to be end user computing tools *per se*. We will see that error rates in EUC spreadsheet development are no higher than error rates when professional programmers write applications in third generation languages (3GLs). Rather, the root problem appears to be poor end user development practices, especially an almost total lack of professional-grade testing. We need to teach new dogs old tricks learned painfully by programmers in the early years of software development.

However, nothing can be done about EUC risks until IS researchers decide to study them, corporate management assumes responsibility for identifying and reducing these risks, and corporate IT departments stop saying, “End user computing? That’s a business thing,” and, “Yes, we have a help desk. What’s your point?”

More excitingly, EUC also appears to be the *dark energy* of corporate IT. Applications developed by end users seem to be powering nearly every knowledge worker and department in corporations today. These applications go far beyond downloading and massaging data from central corporate databases. We will see that end user-developed-applications are enormous and complex. In addition, a large fraction of all knowledge workers and departments seem to have one or more EUC application that they characterize as mission-critical. Instead of focusing so heavily on traditional central corporate information systems and spin-offs, IS researchers need to understand how EUC is energizing the corporation’s myriad departments and employees

# Revising Beliefs about Spreadsheet Applications

In this paper, we focus heavily on spreadsheet applications because they are arguably the most studied aspect of end user computing. Spreadsheets are usually viewed as personal productivity applications used to download corporate data, quantify decision analyses, and do relatively straightforward computation tasks. In fact, spreadsheets are used in a broad spectrum of corporate activities. This difference between stereotypes and realities illustrates the dark matter nature of spreadsheets and of end user computing as a whole.

## Application Development Tools

Often, practitioners and researchers make a distinction between *programmed* applications developed with 3GLs on the one hand and EUC applications built with spreadsheets and other 4GLs on the other hand. We argue that this distinction is counterproductive.

In his early years, the first author wrote 2GL programs using assembly language. When programmers switched to 3GLs like COBOL or FORTRAN, they did not stop calling what they did *programming*. Certainly, many old diehards argued that 3GLs were toys that did not let you reach into individual registers. They also argued that 3GLs were too domain-specific. However, 3GL productivity gains were enormous, and rapidly increasing processing speeds made 3GL processing penalties unimportant. Most importantly, individual 3GLs proved to be broad-spectrum languages that could create a wide variety of applications. There were few application development needs that could not be met with 3GLs.

Beginning in the 1970s, another generation of development tools emerged. People inside and outside of the central IT department began to develop applications with 4GL tools, including spreadsheet programs, database applications, and other 4GL integrated development environments. Again, many 3GL proponents argued that this was not “programming.” But was that relevant? The goal is to develop effective applications productively. Arguments stating that using 4GLs is not programming again boiled down to the inability to do low-level things and the domain-specific nature of the tools. Again, however, 4GL productivity trumped most small losses in what users could do, and processing power growth quickly removed “insufficient power” problems.

More fundamentally, many 4GLs have revealed themselves to be extremely broad-spectrum tools. In particular, the spreadsheet’s grid-and-worksheets format has proven equally adept at handling scratchpad calculations and applications with thousands of root (non-copied) formulas. To call spreadsheets personal productivity applications, as many IT professionals, IS researchers, and IS courses do, makes very little sense given the enormous number of very large spreadsheet applications in business. Even without taking into account macros and Visual Basic for Applications, we have simply moved to yet another generation in application development tools. We should only talk about development, regardless of whether the developer user a 3GL or 4GL.

Even without taking into account macros and Visual Basic for Applications, we have simply moved to yet another generation in application development tools. We should only talk about development, regardless of whether the developer user a 3GL or 4GL.

When Grossman, Mehrotra, and Ozluk (2007) studied 18 mission-critical applications, they found that in three cases, applications were developed by experienced 3GL programmers. In two of these cases, the developers considered both spreadsheet programs and 3GLs and consciously chose spreadsheet programs. In the other 16 cases, spreadsheets were such an obvious choice that nothing else was considered. In one case, the application was later ported to a 3GL because of spreadsheet size limits at the time.

Accordingly, we will simply focus on application development without regard to the generation of the integrated development environment (IDE) used to create applications. While one can usefully argue whether to use 3GL IDEs or 4GL IDEs for specific applications, we argue that the term “programming” has become a distraction.

## Spreadsheets in Operational Systems

Spreadsheets are usually described as decision support tools. However, this is only one of the many uses of spreadsheets. For example, spreadsheets are often part of *operational systems* that are the traditional realm of data processing, including accounting, financial, and budgeting systems.

For example, research prompted by the Sarbanes–Oxley (SOX) Act of 2002, requirements showed that spreadsheets are widely used in financial reporting (Panko, 2006), which is normally viewed as an operational system in the realm of 3GLs and commercial software packages. SOX applies to all major American and foreign firms that trade on U.S. stock markets, and it requires extremely strong control over the firm’s financial reporting process. A single material control weakness that *might* result in a material misstatement is enough for a firm to receive a negative attestation from its auditors. Actual errors are not required for a negative attestation.

* In 2004, financial intelligence firm CODA reported that 95% of U.S. firms used spreadsheets for financial reporting (www.coda.com).
* RevenueRecognition.com (now Softtrax) surveyed 118 U.S. business leaders in 2004. The survey found that 85% were using spreadsheets in financial reporting and forecasting, and that these spreadsheets were central in the process.
* CFO.com interviewed 168 finance executives about information technology in finance departments. Out of 14 technologies discussed, only two were widely used—spreadsheets and basic budgeting and planning systems. *Every* interviewee said that his or her department used spreadsheets substantially.
* A.R.C. Morgan interviewed 376 individuals responsible for overseeing SOX compliance in multinational corporations. More than 80% said that their firms used spreadsheets for financial reporting.
* In a webcast for Deloitte on May 22, 2005, the first author was able to ask a series of questions to 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 only 7.1% said, “No.” As noted above, *every* part of the financial reporting process that is of material importance must be controlled under Sarbanes–Oxley compliance requirements for a firm to receive a correct positive attestation from its auditors.
* Furthermore, when companies use spreadsheets for financial reporting, they often use many spreadsheets. One firm used more than 200 spreadsheets in its financial planning process.
* Heavy spreadsheet use in financial reporting was even common in companies that used financial reporting packages; in such cases, companies often use spreadsheets to do such critical and risky work as computing end-of-period adjustments. At a 2005 Taxcorp presentation to 500 financial professionals whose companies were using financial reporting packages, the first author asked how many in the audience were in firms that used spreadsheets that way in financial reporting. Nearly every hand in the room went up.

Although the most evidence for the importance of operational spreadsheets comes from SOX-related studies, spreadsheets are often seen in other operational contexts (e.g., Hinh, Lewicki, & Wilkinson, 2009; Grossman, et al., 2007; Wagner, 2003).

## Spreadsheets in Tracking Applications

Spreadsheets are also used in corporate tracking applications that might normally be viewed as matters for database management systems. In interviews in one research institute, the authors found that spreadsheets are widely used for tracking. In one case, a large spreadsheet contained critical parameters used in many organizational units in several geographical locations. This was merely one example of tracking systems in the organization.

Large and important tracking systems have been seen frequently in other contexts. In Iraq, for example, information on suspected radio frequencies used by the Iraqis to detonate improvised explosive device danger zones was kept in an enormous central spreadsheet referred to as MOASS—the mother of all spreadsheets (Atkinson, 2007). This information was used to guide the use of radio jamming to prevent the detonation of these bombs.

It may seem that complex information should be kept in databases. However, tracking system users interviewed by the authors have said that the ability to see the actual data instead of just interacting with queries is vital to them. They also have said that they use spreadsheets because the structure of the information being tacked changes frequently and that spreadsheets are much better for rapid structural changes. In addition, the Tables capability introduced in Excel 2007 and refined since that time have made it far easier to do sorting, searching, and other actions previously available mostly in database management systems.

## Perspective

Overall, spreadsheet programs are extremely broad-spectrum integrated development environments. They are not limited to traditional decision support applications. They are used for operational systems, tracking systems, and many other large and important corporate processes.

Overall, spreadsheet programs are extremely broad-spectrum integrated development environments. They are not limited to traditional decision support applications. They are used for operational systems, tracking systems, and many other large and important corporate processes.

## Beyond Personal Productivity

A critical point is that many writers classify spreadsheets as *personal productivity* tools. However, while scratchpad personal applications are common, we will see that data since the 1980s has shown that many spreadsheets in organizations are extremely large and complex. Spreadsheets are indeed widely used as personal productivity tools, but they are also *organizational performance* tools.

Spreadsheets are used as personal productivity tools, but they are also organizational performance tools.

# Evidence for the Importance of Spreadsheets

We have just seen that spreadsheets are widely used and are used for many things beyond decision support. Of course, if spreadsheets are widely used but are not important, then there is no problem. However, there is strong convergent evidence suggesting that spreadsheet applications are extremely important in corporations today. We note again that we are focusing on spreadsheets because they are the most documented application in end user computing.

Of course, if spreadsheets are widely used but are not important, then there is no problem. However, there is strong convergent evidence suggesting that spreadsheet applications are extremely important in corporations today.

|  |
| --- |
| Research on spreadsheets driven by Sarbanes–Oxley |
| Computer and application penetration |
| Relative sizes of end user and IS application development |
| Censuses of organizational spreadsheets |
| Spreadsheet size and complexity |
| Prevalence of mission-critical spreadsheet applications |
| Spreadsheet dominance over domain-specific packaged applications |
| Other indication of importance |

Figure : Evidence for the Importance of Spreadsheets

## Research on spreadsheets driven by Sarbanes–Oxley

We saw in the previous section that corporations on U.S. stock exchanges are required to maintain strong controls over all material aspects of financial reporting. We also saw that the use of spreadsheets of material importance is very common in financial reporting. There is little control over these spreadsheets, so it is difficult to see how these corporations are in compliance with Sarbanes–Oxley.

## Computer and Application Penetration

The most basic data on the importance of end user computing is the extent to which end user computing tools are used. Several years ago, the Bureau of Labor Statistics conducted a series of surveys on computer penetration and application use at work. In 2001, the Bureau found that 72 million Americans used computers at work, with 60% using spreadsheets or databases and 15% saying that they “did programming.” Scaffidi, Myers, and Shaw (2005) projected these figures to 2012, using BLS population projections and a projection of use parameters based on historical data. They estimated that 90 million Americans would be using computers at work in 2012, that over 60% (55 million) would be using spreadsheets and databases, and that 15% (13 million) would be doing programming. In comparison, the BLS projected that there would be 3 million programmers in 2012 (Scaffidi et al., 2005). Certainly, the number of employees using application development tools will be far larger than the number of programmers.

## Relative Sizes of End User and IS Application Development

Although there is a large number of end users doing application development, end users almost certainly use these tools for application development less frequently than IT application developers use their tools. So how do end user and IT development compare?

Even during the 1980s, data indicated that end user development was significant compared to IT development and might soon be much larger.

* In 1982, Benjamin, Dickenson, and Rockart presented an analysis of mainframe usage in one corporation. At the time, both business end user computing were dominated by mainframe tools. They found that end user computing was already consuming 40% of CPU cycles. Furthermore, this number was growing rapidly, and they forecast that end user computing would be larger than centrally developed computing by 1995, consuming 75% of all CPU cycles. This analysis indicated that even thirty years ago, end user computing was not trivial compared to central IS systems and seemed poised to dominate corporate IT use.
* Panko (1988) listed two other sources that looked at end user computing in the 1980s. In 1986, Ronald Brezinski of Quaker Oats found that the IS department was paying all the bills but was only managing 40% of the CPU cycles. IBM Canada, about the same time, indicated that 25% of its computing was already coming from end user computing.

Although these numbers are not precisely the same, they all indicate that end user computing was very large before a large fraction of all today’s workers were born.

But how much time do users spend in application development? We will do an analysis based on Bureau of Labor Statistics population data, past use-of-time data for programmers, and estimates of how much time managers and professionals spend on spreadsheet analysis based on a survey by Baker, Foster-Johnson, Lawson, and Powell (2006), whose respondents indicated that they spent a quarter of their time doing spreadsheet *development* (as opposed to total spreadsheet *use*). Although respondents who spend more time than the average probably self-selected themselves to answer the survey, spreadsheet development seems to be at least of comparable size with IT application development and development in software firms.

|  |  |  |  |
| --- | --- | --- | --- |
| Category | IT Application Developers | Managers | Managers and Professionals |
| Employment (millions) | 2,158 | 12,642 | 27,147 |
| Percentage of time with application development tools | 60% | 25.4% | 25.4% |
| FTE development employment | 1,295 | 3,213 | 6,900 |
| **Ratio** | **1.0** | **2.5** | **5.3** |

Table 1: Application Development Time

The table indicates that although managers and professionals spend less time with application development tools than do professional application developers, their large numbers seem to mean that they collectively spend more time developing applications than do IT application developers. To equate application development with IT programmers seems ill advised. Even if the spreadsheet development time is far smaller than the table shows, users use computers for many other applications, including document-creation tools, presentation creation tools, website creation tools, and communication tools, among many others. The products they create are often interactive and have other characteristics we equate with the term “applications.”

## Censuses of Organizational Spreadsheets

Regulatory compliance requirements like the Sarbanes–Oxley Act of 2002 have forced companies to look at how they are using spreadsheets in several critical business functions. As we saw earlier, when organizations began to look at their financial reporting processes, they discovered a large number of spreadsheets.

Recently, vendors have developed tools to help companies manage their important spreadsheets. Most of these vendors recommend undertaking a series of steps, often beginning with doing a census of all spreadsheets available on servers and, if possible, on PCs and mobile devices. This usually results in companies realizing that they have huge numbers of spreadsheets on their servers.

In a 2010 presentation at the EuSpRiG conference, CIMCON gave sample data from a few of its recent censuses.

* A large financial services company with 48 departments had 60,000 spreadsheets. Of these, 280 were used in the financial reporting process.
* A large government agency had 630,000 spreadsheets and 2,500 Access databases.
* A large health insurance company had 42,000 spreadsheets on its servers. Before the census, the company had estimated that it had 2,000.
* A large global bank had 8 to 10 million spreadsheets, not the one or two million it believed it had.

After the presentation, a principal from another census tool vendor in the audience noted that his firm had one recent case in which it identified one million spreadsheets at a company, *counting only those accessed in the previous month*.

A principal from a census tool vendor noted that his firm had one recent case in which it identified one million spreadsheets at a company, counting only those accessed in the previous month.

Recently, McDaid et al. (2011) used the Luminous Map product to do an analysis of spreadsheets in the financial departments in two organizations. Although the total number of professional employees in the two financial departments was only 30, the census found 65,806 spreadsheets on the department servers—2,194 per professional employee.

## Spreadsheet Size and Complexity

There is a common belief that spreadsheet programs are used to develop small and simple models, many of which are one-time calculations. However, as noted earlier, spreadsheets are broad-spectrum calculation and data tools that can be used to develop extremely large and complex applications, especially now that PC processing power and spreadsheet size limits have become nonissues except in the most extreme cases.

There have long been indications that many spreadsheets are very large (e.g. Cale, 1994; Cragg & King, 1994; Floyd, Walls & Marr, 1995; Hall, 1996) and extremely complex (Hall, 1996). Census programs have given us the ability to characterize the complexity of spreadsheets based on very large samples. For example, the 65,806 workbooks that McDaid et al. (2011) analyzed contained over a billion cells. A quarter billion of these cells were formula cells. This was an average of 4,000 formulas per spreadsheet. Even with extensive formula copying, this indicates a large number of root (original) formulas per spreadsheet. Cragg and King (1993), Hicks (1995), Lukasic (1998), Lawrence and Lee (2004), and Powell, Baker, and Lawson (2008) also found large and complex spreadsheets when they identified operational spreadsheets to audit. In England, certain spreadsheets are required by law to be audited. A principal at one of the auditing firms reported that in the spreadsheets they audited, *unique* formulas were “in the range 1,000-10,000 and upwards (Croll, 2007).”

## Surveys of Spreadsheet Importance

An obvious way to gauge the importance of spreadsheets is to ask users how important their spreadsheets are to themselves and their organizations. Several studies have done just that.

In Hall’s (1996) study, only 7% of the spreadsheet developers sampled called the importance of their spreadsheets low; 39% rated their spreadsheets as being of high importance. In another early study, Gable, Yap, and Eng (1991) had spreadsheet developers and users rate the importance of all 402 spreadsheets saved on the corporate server. Respondents rated 86% as important or very important to themselves and 46% as important or very important to the organization. Furthermore, respondents said that 96% of the spreadsheets would have to be rebuilt if lost. Of the spreadsheets on the server, 13% were used daily, 6% weekly, and 40% monthly. At the other extreme, only 9% were rarely or never used.

In another early study, Wilkins (1992) asked the broadest level at which a spreadsheet application’s data was used. Table 2 shows that few spreadsheets were limited to personal productivity.

|  |  |
| --- | --- |
| Broadest Level of Data Use | Percentage |
| Other departments | 69% |
| User’s department | 26% |
| Only user | 5% |
| Other | 1% |
| Total (includes rounding error) | 101% |

Source: Wilkins (1992)

Table : Broadest Level of Data Use

Hall (1996) also asked the broadest use level for the spreadsheets her respondents rated. For the highest level of use, 17% selected *self only*, 31% selected *single department*, 23% selected *many departments*, and 29% selected *outside the organization*. This echoes the pattern seen by Wilkins (1992)

Chan and Storey (1996) asked a sample of spreadsheet users a slightly different question—“What was the *highest level* in the organization in which an application was used?” As Table 3 shows, the highest level of use was extremely high. Over half produced data used by the CEO or a vice president.

|  |  |
| --- | --- |
| Highest Level of Data Use | Percentage |
| CEO | 42% |
| VP | 13% |
| Department head | 20% |
| Self | 14% |
| Other | 10% |
| Total (includes rounding error) | 99% |

Source: Chan and Storey (1996)

Table 3: Highest Level of Data Use

Baker et al. (2006) reported on a survey of approximately 1,600 MBA graduates. In the survey, they asked how important spreadsheets were in the respondent’s job. Note that this question dealt with general importance, not the importance of individual spreadsheets. Only 17% said “unimportant or moderately important.” Another 34% said “very important,” and 49% said that spreadsheets were “critical” to their jobs.

Most of these studies were done some time ago. Spreadsheets are undoubtedly far more widespread today, and a smaller fraction may be as important as these data indicate. However, it is impossible to say more because little research like these studies has been done recently. In addition, even if the fraction of highly important spreadsheets is lower today, the enormous number of spreadsheets found in recent censuses indicates that there probably are very many important spreadsheets in corporations currently. It certainly would be difficult to envision spreadsheets averaging 4,000 formulas as being of low importance.

## Prevalence of Mission-Critical Spreadsheet Applications

Another indication of the importance of spreadsheets is research on *mission-critical* spreadsheets that would do considerable harm if they were lost or incorrect. Two studies have looked at mission-critical applications in depth. Both Grossman et al. (2007) and Wagner (2003) discovered a great deal about these applications. Perhaps the most important finding was that these researchers had no difficulty locating managers and professionals with mission-critical applications. This indicates that mission-critical applications are common.

## Spreadsheet Dominance over Domain-Specific Packaged Applications

Another indication that spreadsheet applications are important is that they dominate use in many application categories in which packaged application software is available. A number of package vendors have said that their target markets predominantly used spreadsheets (and in most cases were reluctant to change.) These target markets included sales force compensation, executive compensation management, fisheries management, planning and budgeting, oil and gas company revenue generating, metrology (measurement science), manufacturing and scheduling, data mining, and strategic planning. About a year before the financial crisis, a software vendor selling an application for creating derivatives called the first author saying that their research found that nearly everyone in financial services was creating derivatives using spreadsheets. In many cases, spreadsheet applications may be developed because finding and evaluating commercial applications is expensive compared to the creation of a spreadsheet application. In addition, many commercial packages are difficult or impossible to customize in ways that particular users need, leaving out features important to users or forcing users to work in ways they find unnatural. Of course, the “not invented here” syndrome is often at work. However, it seems presumptuous to assume that users who build their own spreadsheet applications instead of using computer tools are irrational.

## Other Indications of Importance

There are many other indications that spreadsheet applications are extremely important. For example, Croll (2005) studied spreadsheet use in the City of London (the financial district of London). He concluded that “the City of London runs on spreadsheets.” He also noted that in an environment that routinely handles transactions worth hundreds of millions to billions of dollars, “nothing of importance happens without passing through a spreadsheet.” Hinh, Lewicki, and Wilkinson (2009) at the Jet Propulsion Laboratory discussed the importance of spreadsheets at JPL in a paper titled, “How Spreadsheets Get Us to Mars and Beyond.”

## Perspective on Size, Complexity, and Importance

Overall, there are many indicators that a large number of spreadsheets are large, complex, and extremely important. The evidence is less than completely satisfying because it represents many individual studies without any study being completely convincing. However, the most important fact about this research is its high convergence. Every study, without exception, that has looked at size, complexity, and importance has found spreadsheets at the high ends of these concerns.

Every study, without exception, that has looked at size, complexity, and importance has found spreadsheets at the high ends of these concerns.

In addition, we have only been discussing spreadsheets. Although spreadsheets are critical in end user application development today, there are also end user database management systems (especially Access and MySQL) and other end user tools for developing applications. Our focus on spreadsheets has merely been a consequence of the fact that spreadsheets have received the most research attention, as sparse as that attention has been.

# Spreadsheet Errors

We noted earlier that end user computing is dark in two senses. First, it is invisible to corporate IT professionals, corporate managers, and IS researchers. Second, end user computing creates risks for corporations.

The most heavily studied aspects of end user computing risk have been spreadsheet errors. The first author’s spreadsheet research website (Panko 2012b) summarizes some of the research in this area.

## Spreadsheet Error Rates

Reading the results of spreadsheet error research is disquieting. Spreadsheets typically consist of multiple long chains of calculation. Consequently, almost any incorrect cell in a spreadsheet is likely to produce an incorrect bottom-line value. We will see that the experiment-based research on spreadsheet development indicates that spreadsheet developers make errors in about 2% to 5% of all formula cells at the end of module development, when they believe they have created a good module. Given the sizes of spreadsheets noted earlier, this cell error rate would suggest that most large spreadsheets have at least one incorrect bottom-line value and probably have several.

Table 4 bears out this concern. These studies were inspections of operational spreadsheets, and many of these spreadsheets had been in use for some time. The inspection methods fell short of those seen in software code inspection. Nevertheless, most of the spreadsheets were in fact found to have errors. Studies in boldface only reported errors if these errors were serious. They still found errors in most of the spreadsheets they studied.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Authors | Year | Number Inspected | Percent with Errors | Cell Error Rate | Comment |
| **Hicks** | **1995** | **1** | **100%** | **1.2%** | **One error would have cost over a billion dollars.** |
| **Coopers & Lybrand** | **1997** | **23** | **91%** |  | **Financial and accounting spreadsheets. Only counted errors for bottom line figures off by at least 5%--in accounting and finance, a material error.** |
| **KPMG** | **1998** | **22** | **91%** |  | **Decision-support spreadsheets. Only counted significant errors that could lead to a wrong decision.** |
| **Lukasic** | **1998** | **2** | **100%** | **2.2%, 2.5%** | **In Model 2, investment's value overstated by 16%. Quite serious.** |
| **Butler** | **2000** | **7** | **86%** | 0.4% | **Only reported errors serious enough to require additional tax payments. Did not examine all cells.** |
| Clermont, Hanin, & Mittermeier | 2002 | 3 | 100% | 1.3%, 6.7%, 0.1% | CER average is 2.7%. |
| Lawrence & Lee | 2004 | 30 | 100% | 6.9% |  |
| Powell, Baker & Lawson | 2008 | 50 | 86% | 0.9% |  |
| Powell, Baker & Lawson | 2009 | 25 | 44% |  | If zero-impact errors are removed, percentage containing errors is 44%. In 10 spreadsheets for which error size was listed, all had errors of $100,000 or more, and 7 had errors of $1,000,000 or more. Overall, 40% of the spreadsheets had significant errors, and many had very serious errors. |
| **After 1995** |  | **163** | **84%** |  |  |

\*The Butler (2000) cell error rate data is not included because it was not a complete inspection and did not examine all cells. At the high end, the Lawrence and Lee (2004) study counted “issues” rather than quantitative errors.

Table : Errors in Inspections of Operational Spreadsheets

Table 4 also shows cell error rates (CERs) found in these inspections. We will not look at the Butler (2000) cell error rate because that study was an audit, so it did not examine all cells. We will also not look at the Lawrence and Lee CER because it measured “issues,” which included qualitative concerns. Among the rest, Powell, Baker and Lawson (2008) had an average CER of 0.9%, while Hicks (1995) had a CER of 1.2%, Lukasic (1998) had CERs of 2.2% and 2.5% in the two spreadsheets he studied, and Clermont, Hanin, and Mittermeir (2002) had an average CER of 2.7% in the three spreadsheets they studied.

Corroborating the reasonableness of these CERs, human error researchers in many application domains have studied error rates in a broad variety of cognitive activities. For relatively non-trivial activities such as reading graph, error rates of 1% or higher are almost always found (Kirwan, 1994; Swain & Guttman, 1983). In software development in particular, unit inspection of software modules in three large corpuses of industrial code inspection found fault rates of 2.0% (O’Neill, 1994), 3.2% (Cohen, 2006) and 3.7% (Graden, Horsley, & Pingel, 1986). Given that error rates are always the same across various cognitive activities of common complexity, if spreadsheet error researchers had not found the CERs we have noted, their results would have been suspect.

Returning to the spreadsheet data, the CER range of 0.9% to 2.7% in inspections of operational spreadsheets is lower than the 2% to 5% seen in spreadsheet experiments (Panko, 2012b), although it is not much lower. One reason that CERs in experiments might be higher is that spreadsheet development experiments have known correct solutions, so all errors can be found. In contrast, the inspections of operational spreadsheets used methodologies far below the rigor of software code inspection.

## Testing

However, CERs in inspections of operational spreadsheets *should be much lower* than CERs in development experiments. As just noted earlier, software code inspections typically find faults in about 2% to 4% of the statements they inspect on average. These code inspections are done at the end of unit (module) development. The purpose of code inspection is to find these errors so that they can be fixed, and most are. After this initial unit testing, testing is done at multiple times after initial code inspection. As a result of this professional-grade testing, the fault rate in finished programs typically is 0.1% to 0.3% (Putnam & Myers, 1992). This is an order of magnitude lower than fault rates at the module level. The fact that CERs for operational spreadsheets are only about half as small as CERs in experiments suggests that very little testing is being done in spreadsheet development.

In fact, a lack of serious testing has been reported in *every* study that has looked at testing in spreadsheet development. Studies have shown that companies rarely test their spreadsheets extensively (Caulkins et al., 2007; Cragg & King, 1993; Davies & Ikin, 1987; Fernandez, 2002; Floyd, et al., 2005; Hendry & Green, 1994; Galletta & Hufnagel, 1992; Gosling, 2003; Hall, 1996; Nardi & Miller, 1991; Schultheis & Sumner, 1994; Wagner, 2003). Few companies have policies that require testing (Cale, 1994; Caulkins, Morrison, & Weidermann, 2007; Cragg & King, 1993; Fernandez, 2002; Gosling, 2003; Hall, 1996; Speier & Brown, 1996; Wagner, 2003), and among those that do, compliance tends to be low (Galletta & Hufnagel, 1992; Hall, 1996).

Furthermore, when spreadsheet users do what they call testing, what they do is typically a pale imitation of what programmers do when they test. In the Deloitte webinar mentioned earlier, nearly all of the 800 respondents said that they do testing. Yet only 12% tested all cells, and only 2% used multiple inspectors to examine all cells. In software development, inspecting *all* statements with multiple inspectors is mandatory (Fagan, 1976 1986; Gilb, 1993).

In contrast to spreadsheet developers, programmers are trained in the inevitability and seriousness of faults, so they 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 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. In fact, development teams at Microsoft typically have as many testers as developers.

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

## Why is Such Extensive Testing Necessary?

Why is such extensive testing necessary? The reason is that humans are worse at detecting errors than at producing accurate work. The cell error rates we have seen suggest that spreadsheet developers are 95% to 99% accurate when they work. However, research in spreadsheet inspection, which is frequently misnamed “auditing,” finds that individuals typically only find a little over half of all errors in incorrect spreadsheets (Panko, 2012b). This relative low detection rate for errors has been seen in other contexts, such as proofreading and software inspection (Panko 2012a). Given these relatively modest detection rates, multiperson testing and multiple rounds of testing are needed mathematically.

However, shouldn’t errors be obvious? Why not just depend on having people look at the results and seeing if the final numbers smell alright? The answer is that in situations like budgeting in which the approximate correct answer is known, small errors can be serious. In other situations, such as deciding how much to spend to purchase a new company, even large errors would be difficult to detect merely at looking at the final numbers.

# Other Concerns

Errors appear to be a serious issue in spreadsheet applications, and this is probably true of applications created with other end user development tools given base error rates found in human error research. In addition, end user applications raise several other concerns.

One is privacy. Too many application files contain personally identifiable information (PII) that companies have an obligation to protect. The leakage of even a single spreadsheet or data file containing such information can be a disaster for a firm. At the University of Hawai`i, a researcher left confidential information on a project file. This gaffe was discovered, and the University had to spend $550,000 to provide credit monitoring for the people whose personal information was compromised. It also endured the expense of analyzing the matter and notifying people whose personally identifiable information had been compromised.

Another concern is security. In recent years, hackers have increasingly focused on stealing trade secrets. Many organizational processes and extensive corporate knowledge are contained in spreadsheets and other end user application files. File theft or loss can seriously harm a firm’s competitive position.

A third risk is fraud. In 2002, Allfirst Corporation discovered that its currency trader, John Rusnak, had lost $691 million in trades over a period of several years. A spreadsheet was critical in keeping his growing losses hidden (Butler, 2009). Bernie Madoff also used a spreadsheet to maintain his Ponzi fraud (Arvedlund, 2009). In the Sellafield nuclear scandal, falsified data on safety critical nuclear fuel data was hidden in spreadsheets (Thorne, 2012).

More generally, corporations are finding themselves subject to an increasing number of compliance requirements. Some requirements, like 21 CFR 11 in the pharmaceuticals industry, explicitly mention spreadsheets. Others, such as Sarbanes–Oxley, do not mention spreadsheets explicitly but do require anything that might create material errors in financial reports to be controlled. In 2009, the PCAOB gave guidance that spreadsheets had been under-considered in previous years and needed to be considered explicitly in the future (Public Company Accounting and Oversight Board).

# Conclusion: Shedding Light on Dark Matter and Harnessing Dark Energy

This paper has argued that end user applications are often large, complex, and extremely important to firms. It has also argued that end user applications create corporate risks through inaccuracy and other concerns.

## Perpetuating the Darkness

Given that end user computing is many years old, how can these conclusions be true? The simple answer is that we have not looked at end user computing seriously in the past. Corporate management has not considered end user computing to be a pressing issue and so has ignored it. IT departments focus on the issues confronting IT departments—mostly central applications and databases. They view end user computing as a way for users to interact with these central databases despite there being no evidence that this is a major aspect of end user computing.

This attitude is understandable. When a CIO steps out his or her office door, there are programmers, systems analysts, networking professionals, security professionals, database administrators, and other personnel critical to centralized applications. Regarding the rest of the firm’s IT, the CIO may have strong feelings about what is going on but little direct knowledge. In one case, a CIO reported that no spreadsheets were in use after an Enterprise Resource Planning system was put into place. The researcher (Wagner, 2003) to which he said this had already interviewed employees and discovered that there were many spreadsheet “shadow system” in place, some of which users regarded as mission-critical. Even systems analysts only deal with non-IT departments on a limited and project-specific basis.

### Use and user studies.

If we are to really understand the risks and potential of user applications, we need to begin doing an extensive amount of research on end users, including who they are, what they do at work, and how they are using information technology. We call this research “use and user” studies. There is insufficient room in this paper to discuss all potential forms of use and user research, and the authors could only begin to do so if given room. The *Journal of Organizational and End User Computing* is the home journal of the field and illustrates the breadth of research methods in the field. Instead, we will argue that there should be an immediate focus on *primary descriptive research (PDR)*.

### The need for primary descriptive research (PDR).

In marketing, the consumer is often called “the familiar unknown.” This term emphasizes the fact that just because you interact with customers or think about them constantly does not mean that you understand them. It also emphasizes that this ignorance is dangerous.

In marketing, there is an important distinction between *market* research and *marketing* research. Market research analyzes the size and characteristics of a market and individual market segments. Marketing research, in turn, addresses specific questions, such as whether to introduce Product A or Product B. Marketers have long known that unless you understand the market broadly through market research, narrowly focused marketing research to answer a specific question is not likely to be productive because it may be aimed at the wrong question or even at a meaningless question. As we teach in our systems analysis classes, you cannot manage IT intelligently in a system you do not understand.

Market research is a form of what we call primary descriptive research. The goal of primary descriptive research (PDR) is to develop a quantitative understanding of a particular group of people, in this case, end user employees. The reason for doing so is to guide later, more specific research.

The goal of primary descriptive research (PDR) is to develop a quantitative understanding of a particular group of people, in this case, end user employees.

Typically, research fields begin with descriptive research studies and then move on to theories and theory testing research. However, this cannot work in end user computing because technology is constantly recreating itself. What was true even ten years ago is not likely to be true today. The use environment is constantly changing, and users are constantly changing the way they work. In information systems, we need ongoing preliminary descriptive research.

### Tearing down the pyramid.

To give an example of the types of PDR we need, our field often depicts corporations as pyramids, with a small number of senior executives, a substantial number of middle line managers, and even more operational managers and the workers they manage. This tends to focus attention on line managers. In fact, Anthony (1965) never intended the pyramid to be a model of how corporations are organized. He developed the three-layer model as a model about line management. In management, images of organizational structures are more complex (Mintzberg, 1979). IT departments must serve corporations, and to do so they must understand them in all their complexity. Of course, it can be argued that understanding organizations is a job for management researchers. However, as a practical point, while management researchers typically focus on broad issues, bringing IT into organizations is impossible without very detailed knowledge.

## Counting government beans.

One way to do primary descriptive research is to use government employment statistics. Table 5 gives basic 2010 U.S. Bureau of Labor Statistics employment data on what are probably the most IT-intensive workers: information workers. You may find the pattern surprising. There are many more knowledge workers than information support workers, a fact that has been true before the beginning of the last century. More importantly, line managers represent only a fifth of the knowledge workforce. There are almost as many staff managers and line managers, and there are far more professionals than line and staff managers combined. Mintzberg (1979) emphasized the size, importance, and complexity of staff work.

|  |  |  |  |
| --- | --- | --- | --- |
| **Employment Category** | **2010 Employment**  **(thousands)** | **Percentage of Knowledge and Information Support Employment** | **Percentage of Knowledge Worker Employment** |
| Total Employment | 143,068 |  |  |
| Knowledge Workers | 46,086 | 67% | 100% |
| Line Managers | 8,776 | 13% | 19% |
| Staff Managers | 6,789 | 10% | 15% |
| Professionals | 30,521 | 44% | 66% |
| Information Support Workers | 22,603 | 33% |  |
| Knowledge and Information Support Workers combined | 68,689 | 100% |  |

Table : Information Workers in the United States Economy

How much research has been done on the use of IT by staff managers and professionals? Relatively little. However, given the information presented in the table, we should focus more attention on the knowledge workforce in all of its messy complexity. One might argue that business school research should focus on business functions in the firm rather than professionals and other workers who will not graduate with business majors. Of course, corporate IT departments do not have that luxury, and they do not need graduates trained to serve a small part of the workforce.

### Close encounters.

Beyond government (and corporate) statistics, we need to spend much more time talking directly to users on a relatively open-ended basis. We need to engage in broad dialogues with individual end users to understand what is really going on in their worlds. The authors of this paper are currently engaged in such a study. Although data collection is ongoing, it is already clear that the diversity of end users and end user experiences and needs is enormous. Therefore, the amount of close-contact descriptive research that needs to be done is enormous.

### Primary descriptive research is not grounded theory.

Note that we are not talking about grounded theory or other methods that generate theories during qualitative studies. Primary descriptive research is more basic than this. We need a broad understanding before we can do anything else. Even grounded theory tends to focus on a specific aspect of broad situations. We will get to that eventually, but we need something broader now.

### Publishing: embracing the “D-Word.”

Publishing primary descriptive research is likely to be difficult. Descriptive research is often called, disparagingly, “the D-Word.” However, unless we do a great deal of primary descriptive research and publish it so that it is available to help other researchers, end user computing is likely to continue to be the dark matter of corporate IT.

## Perspectives

## Pressure to address the problem.

Today’s era of benign neglect toward errors may be coming to a close. As noted earlier, a number of regulatory compliance regimes, such as Sarbanes–Oxley, require that corporations increase their control over IT, and regulators and companies are beginning to understand how integral spreadsheets are in many operational processes.

In addition, we are beginning to see a rise in adversarial forensic testing, in which adversaries examine the spreadsheets of their opponents to find flaws. Even relatively small errors can discredit spreadsheets in negotiations. In recent years, the U.S. Army Corps of Engineers found that the 3GL and spreadsheet models it used to plan environmental projects were often challenged in court and that it could not stand up for its models. Since then, it has had all of its important project models audited by teams of spreadsheet or programming experts and content matter experts.

### The problem is not spreadsheets.

Spreadsheet error rates at the module level are about the same as statement error rates in 3GL programming, so spreadsheet development tools per se are not the problem. The problem appears to be poor application development, especially an almost complete lack of comprehensive testing.

Spreadsheet error rates at the unit module level are about the same as statement error rates in 3GL programming, so spreadsheet development tools per se are not the problem. The problem appears to be poor application development, especially an almost complete lack of comprehensive testing.

### The unanimity of findings.

We close with a note on the consistency of our data. In this paper, we have looked at data on the importance of spreadsheets and the size and complexity of end user applications. Although the data come from many studies using different broad methodologies and different specific protocols, they are in almost perfect agreement in their findings that spreadsheets and end user computing are extremely important to corporations. Looked at the other way, there is no data to support the popular stereotype that end user computing is a relatively unimportant matter.

Although the data come from many studies using different methodologies and different specific protocols, they are in almost perfect agreement in their findings that spreadsheets and end user computing are extremely important to corporations. Looked at the other way, there is no data to support the popular stereotype that end user computing is a relatively unimportant matter.

# References

Anthony, R. N. (1965). *Planning and Control Systems: A Framework for Analysis*. Cambridge, MA: Harvard Business School Division of Research.

Arvedlund, E. (2009, September 4). How Bernard Madoff escaped detection. *FinancialTimes.com*. Retrieved October 13, 2009. No longer available online.

Atkinson, R. (2007, October 3). If you don’t go after the network, you’re never going to stop these guys. Never. Washington, D.C.: *Washington Post*. http://www.washingtonpost.com/wp-dyn/content/article/2007/10/02/AR2007100202366.html.

Baker, K. R., Foster–Johnson, L., Lawson, B., & Powell, S. G. (2006). A survey of MBA spreadsheet users, Tuck School of Business, Dartmouth University. http://mba.tuck.dartmouth.edu/spreadsheet/product\_pubs\_files/SurveyPaper.doc.

Benjamin, R. I.; Dickinson, C.; and Rockart, J. F. (1982, June). Information technologies in the 1990s: A Long Range Planning Scenario. *MIS Quarterly*, *6*(2), 11-31.

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.

Butler, R. J. (2009). The role of spreadsheets in the Allied Irish Bank / Allfirst currency trading fraud. *Proceedings of the 2009 European Spreadsheet Risks Interest Group*.

Cale, E. G., Jr. (1994). Quality issues for end-user developed software. *Journal of Systems Management. 45*(1),36-39.

Caulkins, J. P., Morrison, E. L., & Weidermann, T. (2007). Spreadsheet errors and decision–making: Evidence from field interviews. *Journal of Organizational and End User Computing*, *19*(3), 1-23.

Chan, Y. E., & Storey, V. 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.

Cohen, J. (2006). *Best kept secrets of peer code review*. Austin Texas: Smart Bear.

Coopers & Lybrand in London (1997). Information is no longer online. Last read online June 7, 2005.

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

Croll, G. J. (2005). The importance and criticality of spreadsheets in the City of London,” *Proceedings of the 2005 European Spreadsheet Risks Interest Group*.

Croll, G. J. (2007). A typical model audit approach: Spreadsheet audit methodologies in the City of London,” *Proceedings of the 2007 European Spreadsheet Risks Interest Group*.

Davies, N., & Ikin, C. (December, 1987). Auditing spreadsheets. *Australian Accountant,* *57*(11), 54-56.

Fagan, M. E. (1976). Design and code inspections to reduce errors in program development. *IBM Systems Journal, 15*(3), 182–211.

Fagan, M. E. (1986, July). Advances in software Inspections. *IEEE Transactions on Software Engineering,* SE-12(7), Page 744-751

Fernandez, K. (2002). *Investigation and management of end user computing risk*. Unpublished MSc thesis, University of Wales Institute Cardiff (UWIC) Business School.

Floyd, B. D., Walls, J., & Marr, K. (1995). Managing spreadsheet model development. *Journal of Systems Management, 46*(1), 38-43, 68.

[Gable, G.](http://www.fit.qut.edu.au/cgi-bin/FITSIS/wwwstaff?Id=gable); Yap, C. S.; & Eng., M. N. (1991). [Spreadsheet investment, criticality, and control](file:///C:\Users\My%20Documents\My%20Webs\SSR\Gable\INVEST.HTM). *Proceedings of the Twenty-Fourth Hawaii International Conference on System Sciences*, *Vol.* *3*. Los Alamitos, CA: IEEE Computer Society Press, 153-162.

Galletta, D. F. & Hufnagel, E. M. (1992, January). A model of end-user computing policy: Context, process, content and compliance. *Information and Management.* *22*(1), 1-28.

Gilb, T. and Graham, D. (1993). *Software inspection*. Edinburgh Gate, England: Person Education.

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.

Graden, M., Horsley, P., & Pingel, T. (1986). The effects of software inspection on a major telecommunications project. *AT&T Technical Journal, 65*.

Grossman, T. A., Mehrotra, V., & Ozluk, O. (2007). Lessons from mission-critical spreadsheets*. Communications of the Association of Information Systems*, *20,* 1009.

Hall, M. J. J. (1996). A risk and control oriented study of the practices of spreadsheet application developers. *Proceedings of the Twenty-Ninth Hawaii International Conference on System Sciences*, Kihei, Maui.

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 with the first author via electronic mail.

Hinh, J.; Lewicki, S. A., & Wilkinson, W. B. (2009, January). How spreadsheets get us to Mars and beyond. *Proceedings of the Forty-Second Hawaii International Conference on System Science*, IEEE.

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

Kimberland, K. (2004). Microsoft’s pilot of TSP yields dramatic results. news@sei. Pittsburg: Carnegie Mellon University, Software Engineering Institute.

Kirwan, B. A. (1994). *A guide to practical human reliability assessment*, London: Taylor & Francis.

KPMG Management Consulting (1998, July 30). Supporting the decision maker: A guide to the value of business modeling. Press release.

Lawrence, R. J., & Lee, J. (2004, August). Financial modelling of project financing transactions. *Institute of Actuaries of Australia Financial Services Forum*, 26-27.

Lukasik, T., CPS. (1998, August 10). Personal communication with the first author,

McDaid, K., MacRuairi, R., Clynch, N., Logue, K., Clancy, C. & Hayes, S. (2011). Spreadsheets in financial departments: An automated analysis of 60,000 spreadsheets using the Luminous Map technology. *Proceedings of the 2011 European Spreadsheet Risks Interest Group*.

Mintzberg, H. (1979). *The structuring of organizations: A synthesis of the research.* Englewood Cliffs, NJ: Prentice-Hall.

Nardi, B. A., & Miller, J. R. (1991, January). Twinkling lights and nested loops: Distributed problem solving and spreadsheet development. *International Journal of Man–Machine Studies, 34*(1),161-168.

NASA, (2010, August 10). *Dark Matter*, *Dark Energy*. http://science.nasa.gov/astrophysics/focusareas/whatisdarkenergy/. Last viewed June 14, 2011.

O'Neill, D. (1994, October) "National Software Quality Experiment," *4th International Conference on Software Quality Proceedings*.

Panko, R. R. (1988). *End user computing: Management, applications, and technology.* New York, NY: Wiley.

Panko, R. R. (2006, May). Spreadsheets and Sarbanes–Oxley: Regulations, risks, and control frameworks. *Communications of the AIS*, 17(9).

Panko, R. R. (2012a). Human error website, http://panko.shidler.hawaii.edu/HumanErr.

Panko, R. R. (2012b). Spreadsheet error website. http://panko.shidler.hawaii.edu/ssr.

Powell, S. G., Baker, K. R., & Lawson, B. (2008). An auditing protocol for spreadsheet models. *Information & Management*, *45*(5), 312-320.

Powell, S. G., Baker, K. R., & Lawson, B. (2009). Impacts of errors in operational spreadsheets. *Decision Support Systems*, *47*(2), 126-132.

Public Company Accounting Oversight Board. (2009, September 24). Oversight Board, Report on the first-year implementation of Auditing Standard No. 5, an audit of internal control over financial reporting that is integrated with an audit of financial statements, PCAOB Release No. 2009-006, http://pcaobus.org/Inspections/Documents/09-24\_AS5\_4010\_Report.pdf.

Putnam, L. H., & Myers, W. (1992). *Measures for excellence: Reliable software on time, on budget*. Englewood Cliffs, NJ: Yourdon.

Scaffidi, C.; Myers, B., & Shaw, M. (2005, September). Estimating the number of end users and end user programmers. *Visual Languages/HumanCentric Computing Languages and Environments (VL/HCC ’05)*.

Schultheis, R. & Sumner, M. (1994, Spring). The relationship of application risks to application controls: A study of microcomputer-based spreadsheet applications. *Journal of End User Computing* *6*(2), 11-18.

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 Sciences, Vol. II,* Los Alamitos, CA: IEEE Computer Society Press, Kihei, Hawaii, January 1996, 374-384.

Swain, A. D., & Guttman, H. E. (1980). Handbook of human reliability analysis with emphasis on nuclear power plant applications (Technical Report NUREG/CR-1278). Washington, D.C.: U. S. Nuclear Regulatory Commission.

Thorne, S. (2012, January). The misuse of spreadsheets in the nuclear fuel industry: The falsification of safety critical nuclear fuel data using spreadsheets at British Nuclear Fuels Limited (BNFL) Sellafield, *Proceedings of HICSS 45*, the Hawaii International Conference on System Sciences, Maui, Hawaii.

Wagner, J. (2003). Mission critical spreadsheets in a larger urban university. *INFORMS*, Atlanta, GA, October 2003.

Wilkins, M. (1992, Spring). Spreadsheet usage of office systems and other business graduates. *Office Systems Research Journal,* *10*(2), 33-40.

Contents

[Introduction 3](#_Toc337049810)

[Revising Beliefs about Spreadsheet Applications 4](#_Toc337049811)

[Application Development Tools 4](#_Toc337049812)

[Spreadsheets in Operational Systems 5](#_Toc337049813)

[Spreadsheets in Tracking Applications 6](#_Toc337049814)

[Perspective 7](#_Toc337049815)

[Beyond Personal Productivity 7](#_Toc337049816)

[Evidence for the Importance of Spreadsheets 7](#_Toc337049817)

[Research on spreadsheets driven by Sarbanes–Oxley 8](#_Toc337049818)

[Computer and Application Penetration 8](#_Toc337049819)

[Relative Sizes of End User and IS Application Development 9](#_Toc337049820)

[Censuses of Organizational Spreadsheets 10](#_Toc337049821)

[Spreadsheet Size and Complexity 11](#_Toc337049822)

[Surveys of Spreadsheet Importance 11](#_Toc337049823)

[Prevalence of Mission-Critical Spreadsheet Applications 12](#_Toc337049824)

[Spreadsheet Dominance over Domain-Specific Packaged Applications 13](#_Toc337049825)

[Other Indications of Importance 13](#_Toc337049826)

[Perspective on Size, Complexity, and Importance 13](#_Toc337049827)

[Spreadsheet Errors 14](#_Toc337049828)

[Spreadsheet Error Rates 14](#_Toc337049829)

[Testing 16](#_Toc337049830)

[Why is Such Extensive Testing Necessary? 17](#_Toc337049831)

[Other Concerns 17](#_Toc337049832)

[Conclusion: Shedding Light on Dark Matter and Harnessing Dark Energy 18](#_Toc337049833)

[Perpetuating the Darkness 18](#_Toc337049834)

[Use and user studies. 19](#_Toc337049835)

[The need for primary descriptive research (PDR). 19](#_Toc337049836)

[Tearing down the pyramid. 20](#_Toc337049837)

[Counting government beans. 20](#_Toc337049838)

[Close encounters. 21](#_Toc337049839)

[Primary descriptive research is not grounded theory. 21](#_Toc337049840)

[Publishing: embracing the “D-Word.” 21](#_Toc337049841)

[Perspectives 22](#_Toc337049842)

[Pressure to address the problem. 22](#_Toc337049843)

[The problem is not spreadsheets. 22](#_Toc337049844)

[The unanimity of findings. 22](#_Toc337049845)

[References 23](#_Toc337049846)

[Figure 1: Evidence for the Importance of Spreadsheets 8](#_Toc337049942)

1. An earlier version of this paper was published in the Proceedings of the 45th Hawaii International Conference on System Sciences, January, 2012, [↑](#footnote-ref-1)