Spreadsheet Accuracy Theory

S. E. Kruck
Computer Information Systems, James Madison University
Harrisonburg, VA 22807
KruckSE@jmu.edu

and

Steven D. Sheetz
Accounting and Information Systems, Virginia Tech
Blacksburg, VA 24061
Sheetz@vt.edu

Abstract

Electronic spreadsheets have made a major contribution to financial analysis and problem solving processes. Decisions made using these tools often involve substantive consequences for the organizations. However, academic literature reveals that experienced professionals and students make many errors when developing spreadsheets. Practitioners recognize the importance of accuracy and have published many techniques they use for improving the accuracy of their spreadsheets. Systematic evaluation of this literature provides a basis for understanding practitioners' perceptions of how and why errors occur in spreadsheets and is also valuable source from which to identify a theory of spreadsheet accuracy and capture the knowledge of experienced spreadsheet developers. The analysis of this literature suggests three categories of issues spreadsheet developers must address to create more accurate spreadsheets including: planning and design; formula complexity; and testing/debugging. Spreadsheet accuracy theory explains and predicts how changes in development processes can be expected to impact spreadsheet accuracy.

Keywords: Spreadsheet Accuracy , Theory Building,, Spreadsheet Development

1. INTRODUCTION

Electronic spreadsheets have made a major contribution to financial analysis and problem solving. Decisions made using spreadsheets often involve billions of dollars. Several studies have demonstrated that business professionals use spreadsheets extensively to make decisions [Connors 1983, 1984; Davis 1997; Heagy & McMickle 1988; Heagy & Gallun; Lee 1986; Mingers 1991; Waller 1985].

Although many decisions are based on the analysis of a spreadsheet model, many spreadsheets have data quality problems, i.e. underlying formulas and resulting numbers are frequently wrong. A growing body of empirical evidence indicates these errors in spreadsheets are a pervasive problem both in laboratory and real-life settings [Brown & Gould 1987; Cragg & King 1993; Davis & Ikin 1987; Hassinen 1988; Janvrin & Morrison 1996; Panko 1995, 1996, 1999; Panko & Halverson 1994, 1995, 1997; Panko & Hicks 1995; Panko & Sprague 1997]. For example, a financial fund company analyst incorrectly entered a net capital loss in a spreadsheet thus causing a $2.6 billion swing in earnings. As a result of the fund's incorrect estimated earnings, the estimated *excess* year-end payout was $4.32 per share [Godfrey & Flatau 1995; Savitz 1994]. Existing academic literature clearly identifies the problem of high error rates in spreadsheets, but is lacking in terms of explanations or solutions.

Lack of theory limits advances in our understanding of the spreadsheet accuracy phenomenon. Theories are required because they enable communication among scientists. It is this communication about phenomena, concepts, and relationships among concepts that leads to progress in our ability to explain and predict our world. Refinement of concepts and relationships to form constructs and propositions that specify causal relationships among constructs is the language of science. In other words, theory identifies "what" is the topic or problem addressed along with the "how," "when," and "why" that explain the scientist's understanding of the world. Our goal is to use the academic and practitioner work related to spreadsheet accuracy to build a theory that explains this phenomenon. Theory building consists of creating or This paper proposes a theory where one did not previously exist. The goal of spreadsheet theory is to explain how spreadsheet errors are made. Investigating this will provide practitioners with the knowledge to improve spreadsheet quality and researchers with a framework for systematically evaluating the difficulties of developing spreadsheets. The following section presents a review of the empirical literature on spreadsheet accuracy. Next, the methodology section explains how a large number of practitioner articles were used to derive the theory of spreadsheet accuracy. The resulting theory section presents the constructs and propositions of the theory, and the discussion section evaluates the constructs and propositions in terms of the theoretical ideas of utility, validity, falsifiablity, and parsimony.

2. LITERATURE REVIEW

Spreadsheets have been one of the most highly used computer applications in business over last 20 years [Connors 1983, 1984; Lee 1986; Zhao 1997]. They are used for many different applications that are essential for business, such as budgets, forecasting production, financial modeling, workpaper generation, cost/benefit analysis, foreign exchange analysis, assets and liability management, determining rate of return on investments, mathematical modeling, analyzing scientific and engineering data, projecting market penetration, and evaluating the feasibility of divestitures, acquisitions, and mergers. Subsequently, spreadsheet development skills are among the most highly sought after skills by employers [Davis 1997; Heagy & McMickle 1988; Heagy & Gallun 1994; Waller & Gallun 1985; Zhao 1997].

Spreadsheets are clearly important due to the frequency of use and demand for spreadsheet skills. The problem is that spreadsheets contain errors, which leads to poor quality and costly decisions. The existence of substantive errors has been demonstrated empirically in many studies [Brown & Gould 1987; Cragg & King 1993; Davis & Ikin 1987; Hassinen 1988; Janvrin & Morrison 1996; Panko 1995, 1996, 1999; Panko & Halverson 1994, 1995, 1997; Panko & Hicks 1995; Panko & Sprague 1997]. However, no theory has emerged to provide for evaluating techniques purported to increase the accuracy of spreadsheets. This section provides a discussion of the academic literature demonstrating the high error rates both in laboratory and real-world settings.

Empirical Evidence of Spreadsheet Errors: Empirical evidence obtained from studies involving experienced professionals and students documents the existence of data quality problems in spreadsheets building new theories to explain known but previously unexplained empirical results [Godfrey & Flatau 1995; Savitz 1994].

[Brown & Gould 1987; Cragg & King 1993; Davis & Ikin 1987; Hassinen 1988; Janvrin & Morrison 1996; Panko 1995, 1996, 1999; Panko & Halverson 1994, 1995, 1997; Panko & Hicks 1995; Panko & Sprague 1997]. These studies reveal unacceptably high error rates in both in practice and the laboratory settings.

Brown and Gould (1987) found that 44% of the experimental spreadsheets developed by IBM professionals with 1 to 5 years of experience contained errors. The subjects perceived the three test problems as more simple than their typical spreadsheet developed at work. The development process followed by most participants included little time for planning the spreadsheet. Brown and Gould observed that the 44% error rate may understate the true error rate due to the well-defined nature of the experimental problems versus the often ill-defined and unstructured nature of real-world problems.

Davies and Ikin (1987) studied 19 working business spreadsheets from ten organizations. The spreadsheets supported essential business activities of project and product costing, budget, payroll, loan calculations, and investment analysis. Only five (26%) worksheets were considered error free and four (21%) of the spreadsheets contained “major” errors including: an error of seven million dollars; multiple exchange rates for the Australian dollar within the same time period; and negative units in the current stock account. The remaining spreadsheets (53%) were considered “inadequate and extremely prone to accidental errors” in actual real-world usage. For example, several spreadsheets contained no text or labels to aid the user, i.e., they consisted of only numbers and formulas.

Another study of 20 working spreadsheets from ten companies discovered a 25% error rate, even though the spreadsheets had gone through a formal testing process [Cragg & King 1993]. Although a 25% error rate is substantial, it may actually understate the true error rate of the sample spreadsheets because examination time was limited to two hours per spreadsheet. The actual size of the examined spreadsheets ranged from 150 to 10,000 cells.

Panko and Halverson (1994) found spreadsheet error rates of 81%, 71%, and 50% for individuals working alone, groups of two, and groups of four, respectively. The average number of cells with errors were 1.96, 1.24, and .50 respectively. Students working in groups of four made significantly fewer errors than students working alone ($p = .039$). However, there was no significant difference between working alone and working in groups of two, or between groups of two and groups of four. Panko and Halverson (1995) had similar results of high error rates and significant differences between individuals and groups with more experienced students and simplified test problems. Another study by Panko and Halverson (1997) also found similar results with more experienced students of differing majors. Students working alone had an error rate of 79%, groups of two had an error rate of 78% and groups of four had an error rate of 64%. Students working alone had significantly higher number of errors than those working in groups of four (p < .00). Cell error rates were 2.36, 1.61, .82 respectively. No differences due to students' major existed between treatment groups.

In practice, spreadsheet developers are not in the laboratory. Panko (1996) allowed some subjects to work at home. He found that 38% of the subjects who worked at home had errors in their spreadsheets: whereas, only 30% of the participants who worked in the lab had errors. It seems that substantive error rates exist in many situations.

Panko and Halverson (1996) also studied the effect of experience on spreadsheet error rates. A group of MBA students averaging 630 hours of experience developing and debugging spreadsheets were compared to two groups of novices, one group of MBA students with little experience and one group of undergraduates. The undergraduate novices had significantly higher error rates in their spreadsheets than the experienced MBA students, i.e., 79% with an average cell error rate of 5.6% versus 57% with an average cell error rate of .9%, respectively. Fifty-seven percent of the spreadsheets developed by the combined groups of all MBA students contained substantial errors. There was not a significant difference in error rates between novice and experienced MBA students.

Panko and Sprague (1997) studied the impact of explicit testing and debugging activities as part of spreadsheet development. The undergraduate students with the debugging treatment with 37% of spreadsheets with errors were not significantly different than the combined MBA group's error rate of 30% (p = .223). That is, statistical differences that existed in Panko and associates' previous studies were not found in this study, presumably due to the debugging activities. However, the error rates were substantively lower than those in previous studies that used the same simplified experimental task.

Janvrin and Morrison (1996) conducted experiments specifically looking at the impact of explicit testing and design activities associated with spreadsheet linking errors. The first experiment provided all the experimental groups of individual student developers with a template with which to test their solution. Janvrin and Morrison then examined the impact of design activities by requiring the treatment group to model their spreadsheets using data flow diagramming techniques. The treatment group had a 7% linking error rate, while the ad hoc development group had a 14% linking error rate. These rates are lower than in previous studies, but are high because the participants had an example paper solution showing all the correct numeric values and only linking errors were considered.

Janvrin and Morrison (1996) had another group of participants work from a single check figure to test their spreadsheets. The treatment group performed spreadsheet design activities by completing data flow diagrams before working on the computer. The percentage of linking errors increased to 18% for the control group and to 9% for the treatment group. These results suggest that design activities, e.g., using data flow diagramming techniques, can decrease spreadsheet errors. Similarly, testing activities may influence spreadsheet accuracy, e.g., comparing to examples or check figures may help developers reduce the number of spreadsheet errors.

The above studies on spreadsheet errors are summarized in Table 1. These studies demonstrate that spreadsheet error rates range from 7% to 81%. These high error rates are found under experimental and practical settings with both professionals and students.

TABLE 1 - Summary of Studies of Spreadsheet Errors

Author(s)	Year	Participants	% of Spreadsheets w/Errors
Brown & Gould	1987	IBM employees	44%
Davis & Ikin	1987	Live/real company spreadsheets	
		--major errors	21%
		--inadequate & extremely error prone	53%
Hassinen	1988	Novice students:	
		-- working on computer	48%
		-- working with paper & pencil	55%
Cragg & King	1993	Live/real company spreadsheets	25%
Panko & Halverson Jr.	1994	Business students:	
		-- working alone	81%
		-- working in groups of 2	71%
		-- working in groups of 4	50%
Panko & Halverson Jr.	1995	Accounting students	68%
		General business students working alone	82%
		General business students working in groups of 3	27%
Janvrin & Morrison	1996	Upper- & masters-level accounting & business administration students:	
		--ad hoc development group	14%*
		--structured systems group	7%*
Janvrin & Morrison	1996	Upper- & masters-level accounting & business administration students:	
		--ad hoc development group	18%**
		--structured systems group	9%**
Panko	1996	MIS upper-division undergraduates:	
		-- working at home	38%
		-- working in laboratory	30%
Panko & Halverson	1996	MBA students	57%
		Non-accounting & -finance upper-division undergraduates	79%
Panko & Halverson Jr.	1997	Business students:	
		-- working alone	79%
		-- working in groups of 2	78%
		-- working in groups of 4	64%
		Accounting & finance students	65%
Panko & Sprague Jr.	1997	Undergraduate students	37%
		Inexperienced MBA students	35%
		Experienced MBA students	24%

* paper template of solution provided
** check figure provided

These previous studies identify three key beliefs of researchers that are relevant to a theory of spreadsheet accuracy. First, Panko and associates demonstrate that as problem complexity is reduced, the number of spreadsheets with errors also is reduced. This may be due to the simplified nature of the resulting formulae required to complete the spreadsheet accurately. Second, both Panko and Sprague (1997) and Janvrin and Morrison (1996) conduct studies that provide students with explicit instructions for testing of their spreadsheets. The results of these studies suggest that this simple procedure could improve accuracy. Third, Janvrin and Morrison (1996) show that an explicit spreadsheet design process impacts accuracy. It is not clear how the persistent findings of the effects of group work might be incorporated into a theory of spreadsheet accuracy. Perhaps group members could complete multiple versions of spreadsheets then check them against each other. However, we are focused on the individual developer, as we believe it is appropriate unit of analysis for translating these findings into a theory that will be useful in practice.

Evidence of Spreadsheet Errors in Practice: Relatively few incidents of spreadsheet errors are made public and these are usually not revealed by choice. Some companies are hesitant to reveal errors due to embarrassment and concern for their corporate image. Others fear legal or financial repercussions of making errors public [Edge & Wilson 1990; Godfrey & Flatau 1995; How personal computers can trip up executives 1984; Knight 1992; Krull 1989]. However, several incidents of spreadsheet errors have reached the press.

One example included a very large capital budget planning spreadsheet from a major corporation. It contained almost 4,000 cells, divided among 19 sub-modules, with an average of 203 cells per module (ranging from 46 to 667 cells per module) [Panko & Hicks 1995]. The spreadsheet was developed by team of four and inspected by a team of three. During an audit for errors, a total of 45 errors were discovered in five of the 19 modules. This translates into a module error rate of 26%. Two of the modules consisting of 391 and 667 cells had 16 errors each, i.e., an average of 5% of the cells in those modules. This example demonstrates even spreadsheets that are developed by a team and then audited by other professionals may still contain errors when modeling complex business situations.

An incident highly publicized due to legal action involved a construction company. When preparing a bid the controller added a row to include additional overhead of $254,000 but failed to check whether or not this row was included in the formula that totaled the column. This caused the firm to underestimate the cost of the $3 million project, resulting in a large financial loss [Cragg & King 1993; Edge & Wilson 1990; Floyd Walls & Marr 1995; Hayden & Peters 1989; Kee & Mason 1988; Schultheis & Sumner 1994; Simkin 1987; Stone & Black 1989].

Another publicized case of spreadsheet errors involved the year-end distribution of the financial fund mentioned above. An employee entered a plus sign, rather than a minus sign, to cause a $2.6 billion swing in earnings. The net asset value of the fund fell 4.4% from the time of the mistake until it was corrected. It seems investor actions and confidence were impacted as many may have sold or waited to purchase the fund as a result of the estimates [Godfrey & Flatau 1995; Savitz 1994].

References to other spreadsheet errors exist but the companies involved have chosen to remain anonymous due to the negative publicity that arises from such errors. In one case, *The Wall Street Journal* reported that a Dallas oil and gas company fired several executives due to a spreadsheet error that caused the firm to lose millions of dollars in an acquisition [Cragg & King 1993; Davis & Ikin 1987]. In another case, a chief operating officer working with two related spreadsheets of 15,000 cells underestimated the market for computer-aided design equipment for the manufacturing industry by $36 million, due to rounding all spreadsheet figures to the nearest whole number. Thus, eliminating the inflation rate of six percent, i.e., 1.06 changed to 1 [How personal computers can trip up executives 1984; Krull 1989; Schultheis & Sumner 1994; Watt 1985]. In a final case, a senior consultant for a big five accounting firm identified 128 errors in four multi-billion dollar worksheets [Edge & Wilson 1990; Schultheis & Sumner 1994; Simkin 1987].

It is clear from this review of the literature that spreadsheets are essential to business and that problems exist in terms of spreadsheet accuracy. Researchers have endeavored to reveal and describe the phenomena of spreadsheet accuracy. Practitioners have published numerous articles describing their techniques for increasing spreadsheet accuracy, but there has been no aggregation of these techniques with the results of empirical research and no theory proposed to explain this phenomenon.

3. METHODOLOGY

The procedures used to develop a theory of spreadsheet accuracy should include analysis of the academic literature and the views of practitioners. The views of practitioners are valuable to ensure the consistency of the theory with developing and using of spreadsheets in practice. That is, we believe such an approach will result in a theory that has utility and validity.

The preceding section presented a summary of the academic literature on the topic of spreadsheet accuracy. We analyzed the results of those studies to determine where error rates were different from study to study. Then we identified what was different about the treatments in studies that could have caused the observed changes. This approach resulted in a set of treatments or issues that appear to impact error rates. This set should include the constructs appropriate for a theory of spreadsheet accuracy. The combined works of Panko and associates are an exemplar of focusing over time and maintaining consistency across studies to describe a phenomenon. The result of their dedication enables our attempt at theory building.

Practitioners that depend upon the spreadsheets they develop have published many articles on techniques to decrease errors in spreadsheets. This group of end-user developers considers these techniques important enough to take the time and effort to share this knowledge with others, and to ask others to take time to read, understand, and use their technique. They consider these techniques to be the ones that helped them increase spreadsheet accuracy in their own spreadsheets. As theorists we value this work as containing not only practical guidance, but also the essence of the constructs and propositions that constitute a theory of spreadsheet accuracy. Indeed, if a theory is not capable of addressing the concerns of these people, it will not be perceived as either valid or useful.

The articles published by the practitioners were examined to identify the techniques recommended. In the overwhelming number of cases the authors of this literature have a common belief that the adoption of their techniques will decrease errors in spreadsheet models. Their efforts resulted in a total of 262 error reduction techniques, recommended by 36 authors. Considerable redundancy existed in the recommendations. Some of these error reduction techniques were cited only a few times whereas others were mentioned in more than half of the papers. The issues identified and their corresponding techniques represented 32 unique recommendations. Examples of the unique categories included planning ahead, not using constants in formulas, using range names, and tracing circular or error messages. Each individual reference to a category was tallied and totaled. Examples included documentation and organized layout were each referenced in more than 20 of the 38 articles. Graphing to visually check the data and repeating the data display with the formula were each referenced once.

After each technique was categorized, the categories were grouped into related topics. To develop the theory, the categories of techniques were classified into like or similar activities . For example, using cross-footing techniques cannot be checked until the spreadsheet model is designed. Similarly, limit proofs cannot be checked until after spreadsheet numbers and formulas have been entered. Therefore, both of these techniques would usually be done after the model is created, often during some type of testing or error checking/testing phase, so they were placed into the same topic category.

The topic categories resulting from the merging process were used to define the constructs of the theory. The sorting and merging process resulted in three topic categories of similar techniques including: techniques for planning and designing spreadsheets; techniques for testing and debugging spreadsheets; and techniques for managing the complexity of formulas in spreadsheets. This analysis resulted in refining the 32 unique topics to define three constructs that represent the dominant themes present in the empirical and practitioner works. Table 2 shows the articles that provided the sample of techniques and whether the article recommended a technique associated with these constructs.

TABLE 2- Summary of Practitioners and Techniques Mentioned

Author(s)	Number of Techniques Mentioned	Design & Planning	Formula Complexity	Testing & Debugging
AICPA	15	X	X	X
Alexander, R. A.	7	X	X	X
Amoroso, D.	11	X	X	X
Anderson, K., & Bernard, A.	12	X	X	X
Bissell, J. L.	10	X		X
Bromley, R. G.	16	X	X	X
Carlberg, C.	1			X
Chan, W.	10	X	X	X
Dhebar, A.	5	X	X	X
Edge, W. R., & Wilson, E. J. G.	13	X		X
Fleenor, W. C., & Crain, J. L.	14	X	X	X
Floyd, B. D., Walls, J., & Marr, K.	3	X		X
Freeman, D.	8	X	X	X
Grupe, F.	7	X	X	X
Grushcow, J.	5		X	X
Hassinen, K.	5	X	X	X
Hassinen K., Sajaniemi, J., & Vaisanen, J.	3	X	X	
Hayden, R. L., & Peters, R. M.	10	X	X	X
Jones, J. M.	1	X		
Kee, R. C., & Mason, Jr, J. O.	9	X		X
Kiely, T.	4	X		X
Krull, A.	1			
Marcella, A., Jr.	6	X		X
Marchand, M. G.	1			X
Miller, S. E.	7	X	X	X
Pearson, R.	4			X
Pratt, M. J., & Coy, D.	10	X	X	X
Ronen, B., Palley, M. A., & Lucas, Jr. H. C.	7	X	X	X
Savage, H. M.	7	X	X	X
Savage, S.	3	X		X
Schultheis, R., & Sumner, M.	5	X		X
Simkin, M. G.	14	X		X
Stang, D.	14	X	X	
Watt, P.	8	X	X	X
Wittig, G. R.	1	X		
Yoon, Y.	5	X	X	
Total Techniques	**262**	**31**	**21**	**30**

Spreadsheet accuracy was assumed to be the final essential construct of the theory. The techniques placed in the categories resulting from the merging process were then evaluated to determine their impacts on spreadsheet accuracy. These impacts were assertions that were often stated in the following manner: using this technique will help developers reduce the number of errors in their spreadsheets. Such statements identify the relationships among the constructs that were used to define the propositions of the theory.

4. RESULTING THEORY

The comparison of the studies from the empirical literature revealed that treatments associated with design activity, problem complexity, and testing activity positively impacted error rates. That is, increases in design activity should increase accuracy, i.e., reduce errors as suggested by Janvrin and Morrison (1996). Similarly, Panko and Halverson (1996) suggest that simplifying the problem should reduce errors. This suggests three constructs that could be part of the theory and their potential impacts on the construct of spreadsheet accuracy, which could be propositions of the theory.

The similarity of the topic categories from sorting and merging and the empirical treatments that impacted error rates is striking. For example, data flow diagramming is a design technique that is consistent with planning and design category. The effect of the data flow diagramming treatment on error rates/spreadsheet accuracy also supports assertions about design and planning techniques recommended by the practitioners [Janvrin & Morrison 1996]. Testing and debugging also has been explicitly examined empirically with error rates impacted as expected [Galletta Abraham ElLouadi Leske Pollalis & Sampler 1993; Galletta Hartzel Johnson & Joseph 1996; Panko 1999; Panko & Sprague 1998]. Thus, these two constructs are consistent with both the empirical and practitioner literature.

The concepts of formula complexity identified by practitioners and the problem complexity manipulated empirically also are consistent. First, they have a similar inverse impact on accuracy. Decreasing formula complexity from the views of practitioners will increase accuracy. Similarly, decreasing the complexity of the problem has resulted in lower, but still substantive, error rates, i.e., increases in spreadsheet accuracy [Panko & Halverson 1994]. It seems reasonable that more complex problems result in more complex formulas, i.e., manipulating problem complexity also manipulates formula complexity, albeit not directly. However, the construct of problem complexity is problematic, because practitioners cannot control the complexity of the problems they must solve. Thus, a general theory that is applied across many problem domains should not include such a construct.

Constructs: The constructs included in the theory are planning and design organization, formula complexity, testing and debugging assessment, and spreadsheet accuracy. The relationship between spreadsheet accuracy and the other constructs are described by three propositions.

Planning and Design: The planning and design organization construct is defined as the degree to which the spreadsheet was laid out into an orderly and cohesive format. For example, net profit should follow income and expenses. In addition to the empirical academic research, 82% of the practitioners explicitly recommend engaging in spreadsheet design and planning to some degree. They identify a range of related activities from algorithm planning to module design to user interface planning techniques.

The most common design and planning recommendation was the need for developers to plan their spreadsheet development. Information systems developers have adopted a systems development life cycle (SDLC) approach to the development of information systems. Likewise, spreadsheet developers need an organized method for development of spreadsheets. Too many spreadsheets, including some that manipulate millions of dollars, are developed using ad hoc approaches. Such approaches have proven unreliable in the context of information systems development. Indeed the high error rates found in spreadsheets may be evidence of the failure of these ad hoc approaches. Almost every author recommends a variation of planning, analysis, design, implementation, and maintenance. Table 2 indicates the practitioners that advocate these methods. Amoroso (1992) states the following:

> Program development usually follows the systems development life cycle, which contains four steps: 1) planning, 2) analysis and design, 3) implementation, and 4) testing. … We all go through each of the development steps whether we realize it or not. It is better to explicitly plan for and effectively execute each activity. (Pg 222)

Hayden and Peters, as early as 1989, suggested using the SDLC to create spreadsheets and lists the steps as they relate to directly to spreadsheet development.

> And “perhaps the most fundamental of all the steps related to design methodology is planning. Adequate time must be given to planning the spreadsheet before the computer is ever turned on! Impatience is probably the greatest trap of all” [Edge & Wilson 1990].

Algorithm planning and design involves ensuring that the solution technique is correct from a business perspective. Ronen, Palley, and Lucas (1989) further this idea as related to spreadsheets:

> To minimize the probability and severity of a the problems…, the designer of a spreadsheet should be concerned with the following issues: 1) A spreadsheet should produce reliable results; the output it generates should be correct and consistent. 2) A spreadsheet should be capable of being audited… .

The goal of this activity is not so much related to design or planning of the spreadsheet, but that making the facts, numbers, and equations necessary to solve the problem have been properly organized to get an accurate solution. Furthermore, “the well-being of your company may depend on the accuracy of your spreadsheet figures” [Hayden & Peters 1989].

Another important component of design and planning is organizing how the data and processing components of the spreadsheet will be managed. This involves a data design that includes separating data used as input into calculations from the data that is created as a result of the calculations and defining data capture or export procedures [Yoon 1995]. Processing components require design to provide a logical fit with elements of the algorithm and utilize the data design. Many authors recommend addressing these aspects of the spreadsheet before working with the computer. This common theme was explicitly captured by 24 of the 38 authors.

Data and process design are essential to ensure that the required numbers are available to the elements of the algorithm when needed. Freeman (1996) comments that "it takes just one small error--a single misplaced code--to produce wildly erroneous results. Such errors can be devastating because the data often are the foundation on which many organizations base their key decisions."

The final component of the design and planning construct involves the user interface. "Data entry area should be formatted for ease of entry. For example, if a number is to be entered as a whole number with one decimal place, a template showing XXX.X can be used to help guide the user" [Anderson & Bernard 1988]. Steward and Flanagan (1987) suggest "any critical assumptions and numbers contained within the model be exposed, highlighted and easy to modify when not appropriate for the next model." Many authors recognize the need to carefully consider this aspect of the spreadsheet. By considering the goals of the prospective user and evaluating multiple possible interfaces, developers can explicitly design labels and input data validation to dramatically reduce the potential for data entry errors. Bringing all the necessary values together to minimize user paging and scrolling should also influence perceived ease of use of the spreadsheet. Many practitioners suggested not to create user interfaces that are too large. Yoon (1995) describes:

> It is easier and faster for human minds to digest a smaller block of information than a larger one…If you have to produce a big screen, regroup your data within the same screen and use subheadings. Adopt a divide-and-conquer approach in designing your model.

Predicting the goals of your user and designing the interfaces to help them accomplish these goals is essential for reducing data entry errors due to misunderstandings.

Data validation procedures also were highly recommended. For example, checking for zero values in cells used for division, checking for negative values in fields that should only have positive values [Anderson & Bernard 1988], comparing total purchase payments in the input area with the calculated purchased payments [Bissel 1986], checking range limits [Freeman 1996], and reviewing results that are impossible or unreasonable given the raw data [Pearson 1988] have all be recommended.

Another common theme of the recommendations for design and planning of spreadsheets is the need for documentation of the spreadsheet. Documentation can be placed in additional sheets within a workbook or in remote cells of a worksheet. This documentation should include the descriptions of all data values used in the spreadsheet and their sources, and should identify the parts of the spreadsheet that are used by other spreadsheets. Most importantly, this documentation must be maintained as the spreadsheet is changed to deal with new requirements.

Potential measures of design and planning include many items such as the boolean (yes - at least some design and planing occurred, no - an ad hoc approach was followed), the total time spent on design activities, the number of design components generated, the extent that an SDLC approach was followed, or the correctness of the design components.

Formula Complexity: Formula complexity is defined as the degree to which the formulae required in spreadsheet cells are difficult to understand. As shown in Table 2, 21 of the authors addressed this issue. Formulae can be difficult to understand due to high levels of coupling, complicated calculations, or due to naming conventions.

Coupling is linking a cell to cells in other areas of the current sheet, other sheets in the current workbook, or sheets in other workbooks. Controlling coupling enables the developer to encapsulate logical components of the algorithm to simplify development and testing. Simkin (1987) suggests that "smaller worksheets that link together make errors easier to detect." Uncontrolled coupling is a substantial cause of spreadsheet maintenance error. As cells become more highly interconnected, the developer spends inordinate time trying to remember the meaning of the number of the cell, and she/he is distracted from using it effectively. Naming of ranges [Miller 1989] and structure [Ronen Palley Lucas 1989] helps with this problem. Furthermore, "a well structured spreadsheet also clarifies the assumption to the model users" [Ronen Palley Lucas 1989].

Naming conventions also are important for creating understandable spreadsheets. Many spreadsheets are created to be used over time. It is likely that the original developers and users may be promoted or otherwise leave the organization with the knowledge of how the spreadsheet works. It is necessary to define and use standards for naming of cells and ranges to maintain the link between the problem domain and the spreadsheet solution. For example, using the formula SUM(NcSales + ScSales + GaSales + FlSales + AbSales + MiSales) in a cell named Southeastern-States-Sales-Total versus using SUM(A1..F1) or SUM(A1 + B1 + C1 + D1 + E1 + F1) in cell G1 will be easier to understand and maintain for future developers. Bromley (1985) captures the impact of cell naming common among the practitioners: "[cell naming …] reduces the probability of cell reference errors."

Managing the length of formulas also contributes to spreadsheet understandability. Fleenor (1989) and AICPA (1993) clearly state the views of practitioners: "Break down complex formulas into simple steps. Long formulas are difficult to edit, understand and review." Freeman (1996) concurs: "they (long formulas) are hard to understand and are susceptible to errors." By not following such conventions, developers increase the likelihood that others will not understand and consequently not use the results of their labors.

The formula complexity construct captures the recommendations of practitioners to control this essential factor that impacts spreadsheet accuracy. Potential measures of formula complexity are straightforward. Items such as average number of operators in the formula, the average number of cell references, the length of the longest formula, the boolean use of names (yes - cells are named, no - cells are referred to by row and column), number cells in other spreadsheets that are linked to the cell, or number of formulas in the spreadsheet are consistent with this construct.

Testing and Debugging: Testing and debugging assessment is the degree to which detecting and correcting errors took place. This issue was recommended by 30 (79%) of the authors as a method for reducing spreadsheet errors. Testing also was examined in the academic literature [Galletta Abraham ElLouadi Leske Pollalis & Sampler 1993; Galletta Hartzel Johnson & Joseph 1996; Janvrin & Morrison 1996; Panko 1999; Panko & Sprague 1998]. Interest in the issue by both academics and practitioners is an indicator that testing is perceived as an essential aspect of developing accurate spreadsheets.

Testing and debugging assessment is another aspect of the systems development model that seems appropriate for spreadsheet development. The time to test spreadsheets with sample data will be substantively less than the time required finding and fixing an error when discovered later.

Even the simplest tests can uncover errors. For example, looking at the reasonableness of results could catch the error of net income at $15 when sales are $10 and expenses are $10 or "why the percentage of total sales broken down by market segments totals 312 percent" [Pearson 1988], or testing to determine if the data entered is text or numeric [Floyd Walls & Marr 1995].

An important component of testing and debugging is the use of test data with known values to ensure spreadsheet accuracy. Similar to unit or system testing of computer programs, policies and procedures for comparing the spreadsheet to manual calculations, historical or sample data designed to test all aspects of the spreadsheet will uncover errors. Edge and Wilson (1990) stress the importance of an explicit testing plan. "Manual auditing of spreadsheets can rarely be done effectively by the person who developed the software. One option is for the spreadsheet to be tested by inputting actual historical data for which manual results are available."

Potential measures of testing and debugging include many items, such as the boolean (yes - at least some testing occurred, no - no testing occurred), the number of tests conducted, the percentage of the spreadsheet tested, the total time spent testing and debugging, the number of errors corrected, or the extent that a unit, system, and acceptance testing approach was followed.

Spreadsheet Accuracy: Spreadsheet accuracy is defined as the degree to which the spreadsheet is error-free or accurate. The three constructs presented above address a broad range of errors. For example, errors associated with understanding and using the spreadsheet are captured by the design and planning construct, errors associated with calculations and linking the spreadsheets are captured by the formula complexity construct, and errors resulting from a lack of attention to the problem are captured by the testing construct.

Potential measures of spreadsheet accuracy include the boolean (correct - no errors existed, incorrect - at least one error was found), the total number of errors, the number of errors of different types, e.g., linking, calculating, or user oriented errors, or degree the error changed results from correct results. Along with most practitioners, we refer to the idea of spreadsheet accuracy, thus the inverse of these measures are expected to track the construct.

Propositions: The expected influences derived from the practitioner and academic literature were used to define the spreadsheet theory. The propositions are designing and planning, formula complexity, testing and debugging, and spreadsheet accuracy.

Proposition 1 (P1): Planning and Design positively influences Spreadsheet Accuracy. The practitioners overwhelmingly agree that increasing the degree that the spreadsheet is planned and designed will increase accuracy of the spreadsheet. The empirical work also suggests that design activities positively influence spreadsheet accuracy [Janvrin & Morrison 1996]. Information systems professionals are commitment to analysis and design activities in the systems development as a field is dramatic. Many methodologies exist for aiding systems developers including data modeling, process modeling, and object modeling. Given the impact of these approaches on systems development, it is inconceivable that adopting similar approaches will not improve the spreadsheet development process and subsequently the accuracy of spreadsheets.

Proposition 2 (P2): Formula Complexity inversely influences Spreadsheet Accuracy. Decreasing the degree of complexity in formulas will increase the accuracy of a spreadsheet model. For example, splitting a long formula into smaller parts should make a formula less complex, and therefore it will be easier to ensure the correctness of each component of the formula. A formula that adds and subtracts 50 different cells could be split into three simple formulas: one to total the cells to be added, one to total the cells to be subtracted and the final formula simply subtracts the second from the first. Systematically applying such a process should increase the accuracy of a spreadsheet. This proposition captures the relationship represented by the idea that a decrease in average formula length will result in fewer errors, i.e., increase accuracy. Practitioners agree on the existence of this relationship. Fleenor (1989) and Feeman (1996) clearly assert the existence of this relationship in quotes used to support the formula complexity construct. Implicitly, Panko and associates test this relationship across studies by reducing problem complexity [Panko 1999; Panko & Halverson 1995]. Finally, a highly similar assertion is the major premise of the field of software complexity measurement, which has been actively researched for decades.

Proposition 3 (P3): Testing & Debugging positively influences Spreadsheet Accuracy. Increasing the extent or degree of testing and debugging will increase the accuracy of a spreadsheet. In other words, the more testing and debugging the developers do, the fewer errors should remain in the spreadsheet. Obvious as this may seem to anyone with a programming background, many end-user developers believe their spreadsheets are correct (indeed perfect), so they often ignore or skip testing. Practitioners that recommend testing are unanimous in their support for this proposition. The results of empirical studies that use different types of testing also imply a positive relationship between the testing and debugging construct and the spreadsheet accuracy construct [Janvrin & Morrison 1996; Panko & Sprague 1998].

The constructs and propositions are presented graphically in Figure 1. The design/planning and testing/debugging constructs have positive relationships with the spreadsheet accuracy construct, while the formula complexity construct has an inverse relationship with spreadsheet accuracy.

Other relationships may exist among the constructs. Logically, we expect the design and planning construct will inversely influence the formula complexity construct. That is, as more design and planning occurs, it is likely that problems will be decomposed into smaller components, and these smaller components will likely require simpler formulas. Similarly, we might expect the formula complexity construct would inversely influence the testing and debugging construct. As formulas become less complex they will require less testing to ensure they perform the correct calculations. We do not include these propositions in the theory due to the lack of both empirical support and explicit identification of these relationships by the practitioners.

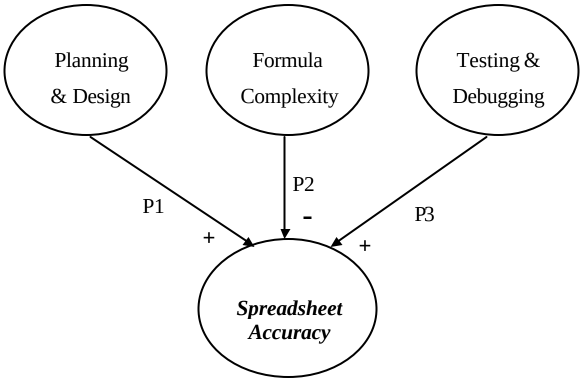

FIGURE 1: Spreadsheet Error Reduction Model

5. DISCUSSION

The goal of theory is to address questions of *how, when*, and *why* spreadsheet phenomena occur. Research that describes this phenomena address questions of *what* is occurring. In the case of spreadsheet accuracy theory, the phenomena has been identified and described by empirical studies and through the perceptions of expert spreadsheet developers that have published their views and methods. Theory quality is evaluated in terms of consistency with the theoretical ideals of validity, utility, falsifiability, and parsimony [Bacharach 1989]. In the following discussion, we examine how our spreadsheet accuracy theory addresses the how, when, and why questions associated with developing accurate spreadsheets. This is followed by an evaluation of the theory in terms of theoretical ideals.

The theory is complete from a spreadsheet development lifecycle perspective. It includes the need to both design and test spreadsheets. An order exits among the activities associated with constructs. Specifically, design and planning should occur before considering formula complexity, which should occur before testing and debugging. Indeed, if we as a field have learned anything in 30 years of computer programming, these would be the primary issues associated with the success of any programming activity. Thus, the theory addresses the question of when particular activities should be performed.

The techniques identified by the practitioners are activities to be performed "within" each construct. Together it is the techniques that are used implement or influence the construct, which represent the how of the theory, e.g., separating data and user interface areas in design and planning or using names in formulas.

The formula is the essential element of the spreadsheet. It may seem that the cell is the important concept; however, the formulas that use or are placed in that cell that accomplish work in the model. This element illustrates the when of the theory and is explicitly represented in the theory.

The question of why for the phenomena exists also is addressed by the theory in the views of the practitioners. Developers of spreadsheets are often very busy people in the organization. Their development activities occur outside and in addition to their day-to-day activities. Under these conditions the temptation to take shortcuts where possible, perhaps by skipping design or performing minimal testing, is real.

In addition to addressing the questions of how, when, and why errors occur in spreadsheets, our spreadsheet accuracy theory is consistent with the theoretical ideals of validity, utility, falsifiability, and parsimony. First, a theory must have validity. Validity is defined as "making sense" in terms of the problem. That is, the theory must accomplish what it purports to accomplish and apply to the problem from the perspective of the person that faces the problem in the world. The proposed theory has validity because it was derived from the views of people that develop and use spreadsheets in practice.

Second, theory must have utility. That is, it must be useful for explaining and predicting the phenomena. In this case, the theory must explain spreadsheet accuracy to be useful. The proposed theory has utility because it specifies direct actions that developers can use to influence each of the constructs and subsequently spreadsheet accuracy. In addition, our theory is useful because it provides for determining the relative influence of each of the constructs on spreadsheet accuracy. Thus, the theory we developed provides for evaluating a broad range of developer actions to understand and integrate their influence on spreadsheet accuracy.

Third, components of the theory must be falsifiable. The constructs and relationships must be sufficiently defined such that they can be demonstrated to be false. The theory is adequately defined because in addition to definitions of the constructs, sample measures have been suggested to clearly present the intent of the theory. For example, Proposition 1 will be shown to be false when developers that do not use any design and planning activities develop spreadsheets with fewer errors than developers that use design and planning activities. A similar approach can be applied to demonstrate the falsifiability of the other propositions. The constructs themselves could be falsified by determining that variations in their values do not contribute to explaining spreadsheet accuracy or by refutation of users in practice. However, this seems remote given the approach adopted for deriving the theory.

Finally, theory must have parsimony. It must have the minimum number of constructs and relationships required for explaining and predicting the phenomena. Spreadsheet accuracy theory fits this characteristic well with only four constructs and three relationships. The convergence from the 32 categories of issues into 3 constructs is more likely to result in too few rather than too many constructs. There may be more constructs to add to the model in the future, however, they are not apparent and we feel it is best to start with a simple manageable model. As the theory is tested and perhaps found lacking in certain respects, it is easier to expand from these core constructs to include new constructs than to redefine or eliminate constructs.

In addition to the research implications of spreadsheet accuracy theory, we expect the theory will be applicable in practice and teaching. In practice, specific tools could be developed to support activities associated with each construct and to provide coordination of activities across constructs. Indeed, simple policies requiring separation of data areas and user interface areas, requiring cell naming, limits on formula length, or documentation of tests conducted would result in more accurate spreadsheets.

Implications for teaching include the benefits of decomposing a large problem into a set of smaller problems and providing a process overview of spreadsheet development. Problem sets can be designed to demonstrate issues resulting from inadequately considering each construct and failing to coordinate across constructs in their development processes. Students should gain a deeper understanding of spreadsheet development from such an instructional approach.

The limitations of the proposed theory are that it is based on a small body of academic literature and is as yet untested. However, linking the theory closely to the views of practitioners should bias results toward utility, which we consider to be desirable. Now that the constructs and propositions have been presented, a more systematic evaluation of spreadsheet accuracy to integrate our current knowledge and to suggest additional issues to consider is possible.

6. CONCLUSION

This paper presents a spreadsheet accuracy theory which was derived from both empirical and practitioner literature. The theory addresses the questions how, when, and why problems with spreadsheet accuracy occur and provides a context within which progress can be made on this important issue. In addition, the theory is consistent with the theoretical ideals of validity, utility, falsifiability, and parsimony. Future research is required to test many aspects of the proposed theory. This includes tests that use multiple measures to converge on individual constructs as well as tests of propositions.

7. AUTHOR BIOGRAPHIES

S. E. Kruck is an Assistant Professor in the College of Business at James Madison University and teaches in the Computer Information Systems Program. Dr. Kruck received a Ph.D. from the Department of Accounting and Information Systems at Virginia Polytechnic and State University and has published articles in Journal of Computer Information System, Journal of End User Computing, and Journal of Accounting Education.

Steven D. Sheetz is an Assistant Professor of Accounting and Information Systems at the Pamplin College of Business at Virginia Polytechnic and State University. He received his Ph.D. in Information Systems from the University of Colorado. He has published articles in Decision Support Systems, International Journal of Human-Computer Studies, Journal of Management Information Systems, Journal of Systems and Software, and Object-Oriented Systems.

8. REFERENCES

AICPA, 1993, "Assisting clients in developing policies and procedures for electronic spreadsheet applications." American Institute of Certified Public Accountants.

Alexander, R. A., 1996, "Teaching good systems design for spreadsheet projects." Journal of Accounting Education, Spring, pp. 113-122.

Amoroso, D., 1994, Decision making using lotus 1-2-3 for windows. New York: McGraw-Hill, Inc..

Anderson, K., & Bernard, A., 1988, "Micros in accounting: Spreading mistakes." Journal of Accounting and EDP, Winter, pp. 42-45.

Bacharach, S. B., 1989, "Organizational Theories: Some Criteria for Evaluation." Academy of Management Review, pp. 496-515.

Bissell, J. L., 1986, "Spreadsheet planning and design." Journal of Accountancy, May, pp. 110-120.

Bromley, R. G., 1985, "Template design and review: How to prevent spreadsheet disasters." Journal of Accountancy, December, pp. 134-142.

Brown, P., & Gould, J., (1987, "An experimental study of people creating spreadsheets." ACM Transactions on Office Information Systems, July, pp. 258-272.

Carlberg, C., 1996, "Spreadsheet quirk keeps PC users going in circles." Denver Business Journal, December, 2B.

Chan, W., 1987, "Sorting out spreadsheets. " Australian Accountant, December, pp. 52-56.

Connors, S., 1984, "NAAA research." Management Accounting, November, pp. 16-65.

Connors. S., 1983, "NAAA research." Management Accounting, December, pp. 62-63.

Cragg, P. G., & King, M., 1993, "Spreadsheet modeling abuse: An opportunity for OR?" Journal of Operational Research Society, August, pp. 743-752.

Creeth, R., 1985, "Microcomputer Spreadsheets: Their uses and abuses." Journal of Accountancy, June, pp. 90-93.

Davies, N., & Ikin, C., 1987, "Auditing spreadsheets." Australian Accountant, December, pp. 54-56.

Davis, P., 1997, "What computer skills do employers expect from recent college graduates?" Technological Horizons in Education Journal, September, pp. 74-78.

Dhebar, A., 1993, "Managing the quality of quantitative analysis." Sloan Management, Winter, pp. 69-75.

Edge, W. R., & Wilson E. J. G., 1990, "Avoiding the hazards of microcomputer spreadsheets." Internal Auditor, April, pp. 35-39.

Fleenor, W. C., & Crain, J. L., 1989, "Standardizing spreadsheet designs." CPA Journal, October, pp. 81-84.

Floyd, B. D., Walls, J., & Marr, K., 1995, "Managing spreadsheet model development." Journal of Systems Management, May/June, pp. 43, 68.

Freeman, D., 1996, "How to make spreadsheets error-proof." Journal of Accountancy, May, pp. 75-77.

Freese, L., 1972, "Cumulative sociological knowledge." American Sociological Review, pp. 472-482.

Galletta, D. F., Abraham, D., El Louadi, M., Lekse, W., Pollalis, Y. A., & Sampler, J. L., 1993, "An empirical study of spreadsheet error-finding performance." Accounting Management Information Technology, pp. 79-95.

Galletta, D. F., Hartzel, K. S., Johnson, S., & Joseph, J. L., 1996-1997, "Spreadsheet presentation and error detection: an experimental study." Journal of Management Information Systems, Winter, pp. 45-63.

Glaser, B. G., & Strauss, A., 1967, The discovery of grounded theory. Chicago: Aldine.

Godfrey, K. & Flatau, A., 1995, “Computing error at Fidelities' Magellan Fund.” The Risks Digest, January.

Grupe, F., 1985, “Tips for better worksheet documentation.” Lotus, August, pp. 68-70.

Grushcow, J., 1985, “Avoid these common spreadsheet errors.” Lotus, July, pp. 59-62.

Hassinen K., Sajaniemi, J., & Vaisanen, J., 1988, “Structured spreadsheet calculation.” Proceedings of the IEEE Workshop on Languages for Automation, pp. 129-133.

Hassinen, K., 1988, “An experimental study of spreadsheet errors made by novice spreadsheet users.” Department of Computer Science, University of Joensuu, PO Box 111, SF-80101 Joensuu, Finland.

Hayden, R. L., & Peters, R. M., 1989, “How to ensure spreadsheet integrity.” Management Accounting, April, pp. 30-33.

Heagy, C. D., & McMickle, P. L., 1988, “An empirical investigation of the accounting systems course: Academic practice versus practitioner needs.” Issues in Accounting Education, Spring, pp. 96-107.

Heagy, C. D., & Gallun, R. A., 1994, Recommended microcomputer knowledge for accounting graduates: A survey. Journal of Accounting Education, pp. 205-210.

“How personal computers can trip up executives.” 1984, Business Week, September, pp. 94-102.

Janvrin, D., and Morrison, J., 1996, “Factors influencing risks and outcomes in end-user development.” Proceedings of the Twenty-Ninth Hawaii International Conference on System Sciences, January, pp. 346-355.

Jones, J. M., 1986, “Decision analysis using spreadsheets.” European Journal of Operational Research, pp. 385-400.

Kee, R. C., & Mason, Jr, J. O., 1988, “Preventing errors in spreadsheets.” Internal Auditor, February, pp. 42-47.

Kiely, T., 1996, “Spreadsheet risk: How and why to build a better spreadsheet.” Harvard Business Review, September, pp. 10-12.

Knight, B., 1992, “The data pollution problem.” ComputerWorld, pp. 81-83.

Krull, A., 1989, “Computer rip-offs and foul-ups: Is management to blame?” EDPACS, May, pp. 10-14.

Lee, D. M. S., 1986, “Usage pattern and sources of assistance for personal computer users.” MIS Quarterly, pp. 313-325.

Marcella, A., Jr., 1989, “Application reviews of end-user spreadsheet design.” EDPACS, October, pp. 10-14.

Marchand, M. G., 1995, “Successful spreadsheets.” Supervisory Management, October, pp. 3.

Miller, S. E., 1989, “8 ways to avoid worksheet errors.” Lotus, pp. 50-53.

Mingers, J., 1991, “The content of MSc operational research courses: results of a questionnaire of OR groups.” Journal of Operational Research Society, pp. 375-385.

Panko, R. R., 1995, “Finding spreadsheet errors; most spreadsheet models have design flaws that may lead to long-term miscalculations.” Informationweek, May, pp. 100.

Panko, R. R., 1996, “Hitting the wall: Errors in developing and debugging a "simple" spreadsheet model.” Proceedings of the Twenty-Ninth Hawaii International Conference on System Sciences, January, pp. 356-363.

Panko, R. R.,. 1999, “Applying Code Inspection to Spreadsheet Testing.” Journal of Management Information Systems, 159-176.

Panko, R. R., & Halverson, Jr., R. P., 1994, “Individual and group spreadsheet design: Pattern of errors.” Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences, January, pp. 4-10.

Panko, R. R., & Halverson, Jr., R. P., 1995, “Are three heads better than one? (at reducing errors in spreadsheet development).” working paper.

Panko, R. R., & Halverson, Jr., R. P., 1997, “Are two heads better than one? (at reducing errors in spreadsheet modeling).” Office Systems Research Journal, Spring.

Panko, R. R., & Hicks, L(1995). Capital budgeting spreadsheet code inspection at NYNEX.

Panko, R. R., & Sprague, Jr., R. H., 1998, “Hitting the wall: Errors in developing and code inspecting a "simple" spreadsheet model.” Decision Support System, pp. 337-353.

Pearson, R., 1988, “Lies, damned lies, and spreadsheets. “ Byte, December, pp. 299-304.

Pratt, M. J., & Coy, D., 1988, “'TIMADS' to avoid spreadsheet model grief.” Accountant's Journal, November, pp. 34-36.

Ronen, B., Palley, M. A., & Lucas, H. C., Jr., 1989, “Spreadsheet analysis and design.” Communications of the ACM, pp. 84-93.

Savage, H. M., 1991. “Document that spreadsheet!” Management Accounting, November, pp. 52-57.

Savage, S., 1997, “Weighing the pros and cons of decision technology in spreadsheets.” ORMS Today Online Edition, Feburary, pp 1+.

Savitz, E. J. (1994). Magellan loses its compass. Barron's, 84 (50), 35.

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), 1-18.

Simkin, M. G., 1987, “Micros in accounting: How to validate spreadsheets.” Journal of Accountancy, pp. 130-138.

Stang, D. 1987, “Spreadsheet disasters I have known. “ Information Center, (November, pp. 26-30.

Stewart, W. & Flanagan, J., 1987, “Spreadsheets design.” Australian Accountant, pp. 56-60.

Waller T. C., & Gallun, R. A., 1985, “Microcomputer competency requirements in the accounting industry: A pilot study. Journal of Accounting Education, pp., 31-40.

Watt, P., 1985, “The power - and pitfalls - of electronic spreadsheets.” Government Finance Review, August, pp. 32-33.

Wittig, G. R., 1990, “Spreadsheet design: Role of introductory documentation in creating spreadsheets.” Library Hi Tech News, pp. 61-67.

Yoon, Y. (1995). Applications-oriented spreadsheet instruction in financial model building: An extension. Financial Practice & Education, 143-147.

Zhao, J. J. (1997). Computer end-user skills needed by business professionals now and toward 2000. Journal of Computer Information Systems, 37 (4), 24-29.