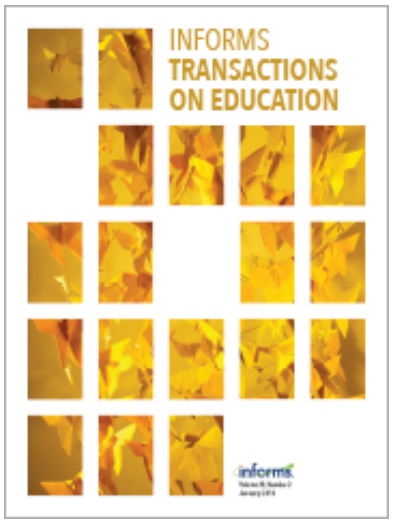

INFORMS Transactions on Education

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Integrating Spreadsheet Engineering in a Management Science Course: A Hierarchical Approach

Thomas A. Grossman,

To cite this article:
Thomas A. Grossman, (2006) Integrating Spreadsheet Engineering in a Management Science Course: A Hierarchical Approach. INFORMS Transactions on Education 7(1):18-36. https://doi.org/10.1287/ited.7.1.18

Integrating Spreadsheet Engineering in a Management Science Course: A Hierarchical Approach

Thomas A. Grossman
School of Business And Management
University of San Francisco
2130 Fulton Street, Malloy Hall
San Francisco, CA 94117-1045, USA
tagrossman@usfca.edu

Abstract

Business students who rely on spreadsheets can use management science only to the extent that they can build a spreadsheet model of an unstructured business situation. Spreadsheets are the primary vehicle for analytical work in business, are advantageous for modeling and model representation, and are used by management science practitioners as well as end-user modelers. There is a common misperception that spreadsheets are somehow "easy" to use. However, the spreadsheet is a powerful rapid development computer programming language that requires software engineering for serious work. People struggle to efficiently build effective, transferable spreadsheet models. Students need certain spreadsheet engineering principles that they do not know, but value highly when they learn them. Spreadsheet engineering can be taught using a hierarchical "Skills, Capabilities and Practices Model." Low-level "Skills" need to be evaluated and fixed early in a spreadsheet-oriented management science course. High-level "Capabilities" and "Practices" should be integrated throughout the course. Embracing spreadsheet engineering provides a magnificent opportunity for management science instructors to increase relevance and student satisfaction while increasing the likelihood of management science application by our graduates.

1. Introduction

In the working world, spreadsheets are the primary vehicle for analytical work. Because of this, spreadsheets are widely used in business school management science courses. The purpose of this paper is to help faculty deliver spreadsheet engineering abilities to improve student access to spreadsheet management science and increase student satisfaction.

In business school management science courses that rely on spreadsheets, students can harness the power of management science only after they create a useful spreadsheet model of a business situation: No spreadsheet model, no management science. However, in my experience, many students struggle to build even simple spreadsheet models. Instructors at conferences regularly complain about buggy spreadsheets and "spaghetti code" that is impossible to understand. Literature surveys (Panko, 1998,2006b) tell us that spreadsheet errors are a serious problem.

Business school management science students need education in spreadsheet engineering "Skills," "Capabilities," and "Practices." -These abilities allow them to create higher quality models in less time, enhancing their ability to genuinely learn and apply management science.

For the past ten years, I have been exploring and experimenting with the content and delivery of spreadsheet engineering to undergraduate, graduate, and executive business students at the University of Calgary in Canada (a large public university), the Tuck School of Business at Dartmouth College (a small private "top 10" school), and the University of San Francisco (a medium-sized private Jesuit university). In this paper, I share the key lessons of my experience, with the intent of motivating instructors to bring spreadsheet engineering into their classroom, and providing guidelines and recommendations to aid them.

  ISSN: 1532-0545

2. Why Spreadsheets Matter

Spreadsheets play a vital role in analytical work and are essential in the business world. It should therefore be no surprise that students are intensely interested in becoming highly effective spreadsheet users. Spreadsheets offer many advantages for modeling and are important to the users of management science, whether they are traditional OR practitioners or non-traditional end-user modelers.

2.1. Spreadsheets Are Advantageous for Modeling and Model Representation

Members of the management science community generally prefer to use algebraic notation to represent models. Algebraic notation is the norm in academic management science, is routinely used by OR/MS practitioners, and features in many papers in *Interfaces*. Although management scientists are fluent in the powerful, abstract "language" of algebraic notation, most people find that formulating and understanding a model in a spreadsheet is much easier.

Algebraic notation is at the heart of mathematics, and concerns about business student mathematical skills are well-known. Jordan et al. (1997) surveyed business school management science instructors and report that 77% of them view the "math background of students or fear of mathematics" as a principal source of teaching and learning problems, suggesting that difficulties with algebraic notation are widespread. Papageorgiou (1996 p. 232) echoes this when he states that "most business students lack the mathematical background and aptitude necessary to 1) understand the technical aspects of OR/MS, 2) appreciate their possibilities, and 3) develop a favorable attitude towards their uses."

In my experience, a large majority of students struggle to formulate an unstructured model using algebraic notation, and are much more effective using spreadsheets. The main exception is an MBA student with a first degree in mathematics, statistics, or a related technical field. It is possible that well-structured or small-scale "toy" problems are misleadingly easy for students to formulate in algebraic notation, but they struggle with less well-defined or larger-scale problems-which are more realistic. Savage (1997, 2003) explains the difficulties using the metaphor of an impenetrable "algebraic curtain" that prevents people from expressing their business knowledge in algebraic models, thereby blocking access to the power of management science.

Spreadsheets are advantageous for modeling unstructured problems. Grossman (2002) discusses the use of the spreadsheet for "exploratory modeling," where "the spreadsheet serves as a modeling tool to structure, explore, and understand a problem; it becomes a means for expressing one's ideas. Somewhere in the process of creating the spreadsheet, users realize something they didn't know before, recognize an insight that previously eluded them, articulate key issues to colleagues, reconceptualize their problem or even figure out how to solve it." Regan (2005,2006)discusses a decision modeling elective at a top-ranked business school where "Spreadsheet modeling of a business situation is central to the course. I want to expand the students' spreadsheet modeling tool kit and develop their ability to build models in a decision context that changes over time." Recent research by Willemain and Powell (2006) indicates that MBA students approach modeling of unstructured problems very differently than do OR/MS experts.

2.2. Spreadsheets Are the Primary Software for Custom Modeling and Analysis in Business

In industry, spreadsheets are so ubiquitous that it is difficult to find a knowledge worker whose computer lacks a spreadsheet. There is a small but powerful empirical literature on the importance of spreadsheet models. Croll (2005) finds that spreadsheet models are essential to the finance industry in the United Kingdom, and that spreadsheets are so valuable that "within certain large sectors, spreadsheets play a role of such critical importance that without them, companies and markets would not be able to operate as they do at present." Grossman, Mehrotra, and Özlük (2006) present empirical examples of spreadsheets used as application software and used for essential roles in financial risk management, executive information systems, business process infrastructure, and complex analytics. It is easy to obtain anecdotal evidence regarding the prevalence of spreadsheets by speaking with business school alumni or by walking down the aisle of an airliner and chatting with the spreadsheet users that one inevitably encounters.

In contrast, there is no empirical research suggesting general business users represent models with algebraic notation or use algebraic modeling software such as

GAMS or Matlab. I would welcome empirical research on this topic. While performing field interviews for Grossman, Mehrotra, and Özlük (2006), the researchers discovered that interviewees laughed when asked about representing their spreadsheet models in algebraic notation. This is not to say that algebraic model representation is not powerful or sometimes advantageous, but rather that its use pales in comparison to spreadsheet model representation for general business use.

2.3. There Are Substantial Barriers to Non-Spreadsheet Software in Industry

Even in a situation where an algebraic model implemented in algebraic software was deemed "better" than a spreadsheet model, the spreadsheet model could still be the better choice in practice. Non-spreadsheet management science software has all the problems of any specialized application software. Optimization software is not part of the "standard load" on desktop PC's in typical companies. Thus, the user must invest time and effort to discover and select appropriate software. This is substantially more difficult than the usual desktop software purchase because optimization software is not sold at popular retail outlets such as Office Depot or Amazon.com. Once the user determines the desired software, he must then have it approved and purchased, which can be time-consuming and costly. Furthermore, lengthy and expensive training is often required before the user can take full advantage of the software's capabilities. The organization must also make a commitment to ongoing training in anticipation of possible departures of trained staff. The small number of software licenses limits the access to the software, reducing its transparency and credibility. In contrast, due to the ubiquitous availability of spreadsheets and spreadsheet skills, spreadsheet management science software suffers less from the problems mentioned in this paragraph.

2.4. Traditional OR/MS Practitioners Use Spreadsheets

I regularly encounter OR/MS models that employ spreadsheets at the front or back of analysis for data entry, user interface, and reporting (Grossman, 1999). I believe that it is becoming common for OR/MS practitioners to use spreadsheets. However, this has not received much attention in the literature. To highlight the importance of this topic, *Interfaces* has called for a special issue on this topic (LeBlanc and Grossman,2006).

2.5. Non-Traditional Practitioners (End User Modelers) Use Spreadsheet Add-Ins

There are many spreadsheet management science add-ins, including Solver and What's Best! for optimization, Crystal Ball and @RISK for simulation, TreePlan and PrecisionTree for decision analysis, and various add-ins for neural networks, stochastic optimization, forecasting, and other techniques. (See OR/MS Today,2002, and Grossman,2004 for a listing of management science add-ins.) Crystal Ball and @RISK run annual user conferences. Oggier, Fragnière and Stuby (2005) describe the use of spreadsheet management science add-ins at a large multi-national company.

The total number of commercial users of management science add-ins is not readily available, but we have information from one vendor. Jim Franklin, CEO of Decisioneering, the maker of Crystal Ball, indicates that "I estimate our user base in the 200,000 range" (Franklin,2006). This number is enormous compared to INFORMS' non-academic membership of 4,000. Franklin states that one particularly large customer "estimates that they have at least 10,000 active users" of Crystal Ball, which is comparable to INFORMS' total membership of 12,000. It is evident that large numbers of end user modelers are using spreadsheet management science add-ins.

3. Spreadsheets Are Not Easy...for Serious Work

There seems to be a widespread perception that spreadsheets are somehow easy, and thus not worthy of sustained examination or study. A corollary is a belief that students should in some sense "know" spreadsheets, and therefore management science instructors need not "waste" limited class time on spreadsheets. I believe that the perception "spreadsheets are easy" is false. A more accurate description is "spreadsheets are easy at the fundamental level, but difficult at the advanced level." A corollary of this argument is that students should indeed be expected to "know" spreadsheets at a fundamental level, but they require education at the advanced level.

The perception that "spreadsheets are easy" is entirely understandable because of the ease with which almost anyone can acquire limited facility. This is evidenced by the widespread use of small spreadsheet models as relatively unimportant personal productivity tools. Indeed, basic spreadsheet use is now taught in many American high schools.

The ease with which spreadsheet users can obtain initial access, coupled with the difficulty of using spreadsheets well for complex tasks, is explained by ethnographers Nardi and Miller (1991). They indicate that spreadsheets have a "fundamental layer" that is easy to learn and an "advanced layer" that is harder to learn. The easy access to the fundamental layer, which contains facilities for "computation, presentation and modeling," means that users can "concentrate more fully on understanding and solving their problems, with much less cognitive overhead devoted to the distraction of coping with the mechanics of the software itself."

Nardi and Miller tell us that "users learn selected parts of the advanced layer as they need them, and as they feel ready to." That is, knowledge of the advanced layer is acquired only in the presence of a meaningful context, which is generally absent in spreadsheet training courses. Therefore, it is unreasonable to expect students to be effective at the level of the advanced layer.

3.1. The Spreadsheet is a Powerful Computer Programming Language

We must recognize that a spreadsheet is a computer programming language. In fact, according to McConnell's (1996) classic software development book, the spreadsheet is a powerful programming language that falls into the class of "rapid development languages," along with fourth generation languages (such as Focus and Ramis), database management systems (including Microsoft Access, Oracle, and Sybase), and visual programming languages (such as Borland Delphi, Microsoft Visual Basic, and Realizer). McConnell states that "Rapid development languages allow programs to be developed at a higher level of abstraction than they could with traditional languages." Nardi and Miller (1991, p. 176) echo this observation, stating that spreadsheets "shield users from the necessity of working with lower level programming primitives."

According to a comparative analysis of development speed using different computer programming languages, spreadsheets are the fastest. Spreadsheets on average need only 6 "statements" to implement a function point that would require 125 statements in C, 110 statements in FORTRAN, or 30 statements in Visual Basic (McConnell, 1996 Table 31-2, p. 519). To have a better feel for the differences in programming complexity, consider the code required for the simple task of adding a column of numbers. In Excel, this can be implemented using a single SUM formula. A third-generation language like FORTRAN or C requires the structure of a do-loop, a loop counter, a limit to stop the do-loop, plus variable definitions and other overhead. From the perspective of development speed, the spreadsheet is roughly 21 times as powerful as C and 5 times as powerful as Visual Basic. Or more simply, "spreadsheets boost productivity through the roof" (McConnell, 1996,p. 518).

This speed may be a key reason for the popularity of spreadsheets for analytic work. Spreadsheet analysts whom I interview or informally talk to frequently indicate that coping with time pressure is an important reason they work in spreadsheets. Investment bankers are particularly adamant regarding the importance of speed in the face of firm deadlines.

Development speed is particularly valuable when writing software without the benefit of a written software specification, as is usually the case for one-off analyses. The high level of abstraction in the spreadsheet allows users to more quickly convert their ideas into working computer programs, spend more time thinking about their ideas and refining their models, and less time coding. Management science instructors should be grateful that our students can escape the bonds of procedural programming languages and work in spreadsheets.

3.2. Computer Programming is Difficult and Requires Software Engineering

As every beginner programmer painfully learns, programming in a way that seems intuitive is neither effective nor efficient. That is why introductory computer science courses place so much emphasis on software engineering techniques such as structured programming, top-down design, and data structures, rather than on mere programming statements. Effective computer programming requires knowledge of a

programming language plus a set of software engineering techniques suitable for that language. Spreadsheets are no different, except that poor software engineering in a spreadsheet seems to get one a lot farther than poor software engineering in C or FORTRAN.

Software engineering is a large field. Key techniques include: designing before programming, using a modular structure (e.g., subroutines), referencing data rather than hard coding it, documenting the code, and so on. These important techniques were discovered by trial-and-error as programmers struggled to build complex software systems. They were carefully refined over years of empirical and academic work, and they are now universally taught in computer science departments and scores of trade books.

These techniques exist because computer programming is difficult. They are taught in computer science courses because they are subtle and not at all obvious. Many established software engineering techniques are directly applicable to spreadsheets. Spreadsheet users need to adapt these techniques to their spreadsheets, in the form of "spreadsheet engineering." Without spreadsheet engineering knowledge, users tend to build poor spreadsheets with confusing designs, spaghetti code (poor modularity), commingled data and calculations, inadequate documentation, and other flaws.

3.3. Spreadsheets Benefit from Thoughtful Spreadsheet Engineering

Rapid development languages such as spreadsheets are more amenable to poor coding habits than are third generation languages. McConnell (p. 521) explains why:

"With traditional programming languages, you tend to build infrastructure into your program to anticipate areas that aren't well-supported by the language itself. You develop highly modular code so that changes won't ripple through your program. You use coding standards to make code readable, and you avoid known worst practices such as using GOTOs and global variables.

One of the iconic weaknesses of rapid development languages is that in pushing back a lot of complexity, they lull you into a false sense of security-they make you believe they will do everything for you automatically. You get far into a project before you realize that you even need things like design and coding standards."

McConnell points out that "the problem is not the result of the rapid development languages' weaknesses as it is of the tendency to use sloppy design and coding practices on rapid development language projects." His solution is to apply software engineering techniques to "err on the side of over-designing your program and being too careful with your coding standards."

This discussion leads us to an essential insight:

Problems with spreadsheets are caused by poor development decisions made by people, *not by any limitation inherent in a spreadsheet*.

What this means for management science instructors is simple: Students need to be taught spreadsheet engineering techniques. Our students' ability to use the management science we teach them is tightly constrained by their ability to create effective spreadsheet computer programs. And right now, they have problems creating spreadsheets.

3.4. Spreadsheet Engineering Issues May Not Be Visible to Management Science Faculty

Problems with spreadsheets can be almost invisible at the scale of small spreadsheet models. They are highly visible at the scale of large homework problems, major case analyses, and in the spreadsheet error literature (summarized in Panko, 1998,2006b).

Management science faculty whose interactions with spreadsheets are confined to minor "toy" problems in the classroom are at a serious disadvantage at appreciating the challenges-and corresponding educational opportunities-that occur when people build sizable spreadsheets for important business problems.

4. The Need for Effective, Efficient, Transferable Spreadsheet Models

For business students (or anybody else) to benefit from spreadsheet management science, they need to be able to quickly create spreadsheet models that accurately express their business knowledge in a way that can be

understood by other people. In short, spreadsheet models require *effectiveness*, *efficiency*, and *transferability*.

- Effectiveness: The model must meaningfully capture the essence of the business situation, and it must be accurate. Effectiveness is closely related to the traditional modeling concepts of validity and verification.
- Efficiency: Business modelers have limited time and often work under deadline pressure. It is important they create spreadsheet models quickly, with minimal wasted effort, rework, and debugging.
- Transferability: Many essential spreadsheets are not simply personal productivity tools but important organizational assets. Spreadsheet models need to be understood by other people, whether they are students at school, colleagues at work, or a successor in a particular job.

There are serious concerns about effectiveness, efficiency, and transferability in the business world. There are widespread concerns about spreadsheet effectiveness, particularly accuracy (Panko, 1998,2000a, 2000b, 2006b), with an entire professional society, the European Spreadsheet Risks Interest Group (European Spreadsheet Risks Interest Group[(1)]), devoted to this topic. I have been unable to find in the literature any studies on the productivity of spreadsheet creation, but there is reason to expect that it is not high. In our empirical work (Grossman, Mehrotra and Özlük, 2006) we found many examples of spreadsheets with poor transferability. We have heard of many situations where multiple person-months of work were lost and had to be recreated when a spreadsheet owner changed jobs or left a company, and his successor could not understand his spreadsheet. In my experience, very few business students can efficiently create an effective, transferable spreadsheet model.

5. Spreadsheet Engineering Skills,Capabilities, and Practices

The problems experienced by spreadsheet users are not failures due to stupidity. They are failures due to ignorance. Fortunately, ignorance is curable with a stiff dose of education. In general, business students and business school graduates are ignorant of spreadsheet engineering. In order to educate them, we require 1) a set of principles that we can teach, and 2) a framework that distinguishes between low-level training-that should be assumed as a prerequisite to a management science course-and high-level education that cannot easily be assumed as a prerequisite.

It is useful to distinguish among spreadsheet *Skills*, *Capabilities*, and *Practices*. *Skills* are essential low-level abilities. *Capabilities* are required to use a spreadsheet to perform meaningful analytical work. *Practices* are sophisticated issues of concern to experienced users which depend upon the context of the work; therefore they are (or should be) of managerial interest and require context-specific education.

I use a 13-element Spreadsheet *Skills, Capabilities, and Practices Model* to categorize and prioritize spreadsheet abilities.

Skills to Use a Spreadsheet (Low-level training)

1. Familiarity. Entry level basics such as opening Excel, opening existing spreadsheets, cells, selecting regions, navigating the worksheet, sheet tabs, and cell referencing.
2. Fundamental Programming. Creating cell formulas; editing cell formulas; using the most common functions; absolute and relative references; simple formatting; cutting, copying, and pasting cell formulas.

Capabilities for Meaningful Analysis (High-level education)

1. Modeling. How to convert business knowledge into a spreadsheet model, prototyping in a spreadsheet, distinguishing between exploratory modeling and purposeful programming (Grossman 2002). (See Powell, 2001 and Powell and Baker,>2004 for more on this important topic.)
2. Design. Principle of modularity; use separate data, computation, and report/output modules; thoughtful row and column structure; visual formatting; row and column labels; and avoidance of spaghetti code.

(1) http://www.eusprig.org/

3. Intermediate Programming. Write once and copy, follow a disciplined approach, separate design from programming, reference data and do not hard code data in formulas, compute once and reference, write for the reader, avoid complicated formulas, the summation button, the Insert Function dialog box, analytical functions such as SUMPRODUCT.
4. Documentation. Indicate what the spreadsheet does and how it does it, row and column labels, notes column, printing, other documentation techniques inside and outside the spreadsheet. (Pryor, 2006 provides a valuable conceptual overview.)
5. Usage. The distinction between *general analytical principles* that apply to any model in any software, and *effective implementation* of analytical principles using Excel's powerful tools to extract insight from models and data sets, including Data Table for sensitivity analysis, the Scenario Manager for scenario analysis, Chart Wizard, Histogram tool, Regression tool, Descriptive Statistics tool, Auto Filter for "drilling" into datasets, and Pivot Table for cross-tabulations.

Practices for Professional Effectiveness (Contextual/Managerial education)

1. Development Parameters. Planning, predicting usage (who, when, how often, how many people), the distinction between spreadsheets used one time by one person and spreadsheets that see repeated use or have multiple users, the challenges of establishing good planning habits (effective planning requires discipline and is difficult to learn until after the user has personally experienced a debacle caused by poor planning), resources to construct the spreadsheet model (time, people, money), anticipated future modifications, and any standards to be followed.
2. Quality Control. Verifying that the spreadsheet is accurate, validating that the model is appropriate, and coping with the intrinsic difficulty of verifying software without a known test case. (Panko,2006a provides a clear and accessible overview.)
3. Debugging. How to locate the cause of an error in a spreadsheet or in the underlying model, how to fix it efficiently without introducing new problems, principles and application of Excel audit toolbar, third-party auditing tools, and Excel error-checking. Much research is needed here.
4. Modification. How to modify a spreadsheet after it has been used, particularly if it is being used by many people in an organization.
5. Advanced Programming. Range names, the auditing toolbar, lookup functions, the summation button, string and character handling, handling numbers formatted as text (a common problem with downloads from ERP systems), conditional formatting, adding buttons and sliders, data cleansing (see Isken, 2003 for an excellent tutorial), and perhaps even simple VBA. (Note that there are two common threads to these techniques: 1) they don't seem to "stick" in a spreadsheet training class, but are retained only if there is a meaningful context-essentially, that they are genuinely useful for a particular spreadsheet model that a student cares about-and 2) students are grateful for learning them.)
6. Management of Spreadsheet Model Assets. Managerial issues related to treating spreadsheets as valuable organizational assets rather than mere personal productivity tools, managing spreadsheet users, how to insure efficiency, effectiveness and transferability.

The *Skills* correspond to Nardi and Miller's "fundamental layer." Faculty can reasonably expect students to possess the *Skills* upon enrollment in a management science course (but as discussed in the next section, faculty should not take the *Skills* for granted). The *Capabilities* and *Practices* seem to be acquired through personal experience, informal on-the-job learning, or (rarely) through organizational standards. It is unreasonable to expect students to possess the *Capabilities* and *Practices* upon entry to a management science course.

This model provides a useful hierarchical framework for thinking about what students might learn. It provides an important prioritization: *Skills* before *Capabilities* before *Practices*. A foundation of *Skills* must be in place before we can expect a student to learn *Capabilities*. Experience with *Capabilities* is required before a student can fully grasp the *Practices*.

Note that the details presented for each *Skill, Capability*, and *Practices* are only sketches. This is intentional. There has not been sufficient research to enable a

complete list of details for each topic. The key is for each instructor to recognize that 1) there is a hierarchy of abilities, 2) Excel learning is part of a larger intellectual framework of spreadsheet engineering (Grossman, Özlük, 2004), not mere "Excel tips and tricks", and 3) they must carefully identify the *Skills, Capabilities* and *Practices* that will benefit their students in their course at their school. Hardin and Ellington (2005), Regan (2005), Powell and Baker (2004), Grinde and Kammermeyer (2003), Albright, Winston and Zappe (2001), and Bell (2000) provide useful ideas and guidance.

6. The Problem of Low-Level Skills

Gaps in student spreadsheet *Skills* are an ongoing challenge for instructors. Several schools send a spreadsheet book to prospective students and demand, as a condition of admission, that students "know spreadsheets," or that they pass a prerequisite spreadsheet course. However, these measures do not guarantee that students will have the necessary *Skills*. At every school I have taught, some students are unskilled even at the level of Familiarity. This can be a particular issue with executive MBA students.

Students who lack the Skill of Familiarity are in very serious trouble in a spreadsheet-intensive class. Students who lack the Skill of Fundamental Programming will experience frustration. They will learn and retain little management science. These students are liable to express harsh judgments in teaching evaluations. We summarize this in the form of an inconvenient fact:

Spreadsheet *Skills* are ***not your responsibility*** but they are ***your problem***.

In a spreadsheet-intensive management science course, you are in some sense betting the course on your students' spreadsheet *Skills*. Whether you like it or not, you need to pay attention to them.

The instructor has three options:

1. Do Nothing. Assume the students have the requisite spreadsheet *Skills* or will somehow acquire them.
2. Evaluate and Fix. Evaluate students' spreadsheet *Skills* and insure remedial training to fix any deficiencies.
3. Review. Devote class time to teaching low-level spreadsheet *Skills*.

The instructor must carefully weigh the costs and benefits of these three options. Option 1 has the advantage of being easy, but carries the risks of students failing to learn management science, becoming frustrated, and giving low teaching ratings. Option 2 can be implemented with very little use of class time, as I explain in section 7. Option 3, although more time-consuming, can lay the foundation for better student learning of management science concepts. This can be the best choice in some situations such as when a vocal minority of students has virtually no spreadsheet *Skills*, or when the majority of students have major gaps in their *Skills*.

Low-level spreadsheet *Skills* are too important to risk deficiencies. I believe that instructors should eschew option 1, which I will no longer discuss. I recommend that instructors Evaluate-and-Fix spreadsheet *Skills* (section 7), or if necessary bite the bullet and perform a Review (section 8).

7. Evaluate and Fix Low-Level Spreadsheet Skills

At the start of the course, the instructor should evaluate students' spreadsheet *Skills*. In my experience, the majority of students have deficiencies that require work.

I have used three techniques for evaluating and fixing student spreadsheet *Skills*: 1) a diagnostic quiz, 2) a spreadsheet modeling exercise, and 3) a web-based evaluation and remediation tool. These techniques can be used alone or in concert. I currently use a diagnostic quiz coupled with a spreadsheet modeling exercise. I am experimenting with a web-based evaluation and remediation tool. I note that some faculty tell me they have good success from providing a tutorial or other learning resource and asking students to self-evaluate and remedy any deficiencies.

7.1. Student-Led Evaluation and Remediation Using a Diagnostic Quiz

The diagnostic quiz is a written test on spreadsheet *Skills*. I use one in every course, at every level (undergraduate, MBA, Executive MBA). I use a short, closed-

computer, multiple-choice spreadsheet diagnostic quiz that was developed by Stephen Powell at the Tuck School at Dartmouth College. He indicates that the quiz has serious limitations, and they have transitioned to a web-based tool. As I explain below, I find the quiz to be sufficient for my purposes. A copy of the quiz and the answers is attached as Appendix 1.

My preferred approach is to place the responsibility for *Skills* remediation upon the students. I announce on the first day of class that students are expected to know spreadsheets at the level of their spreadsheet training course. I make it clear they will be doing lots of spreadsheet modeling, and if they do not get their fundamentals up to snuff immediately, they will not succeed in the course. I emphasize that the responsibility is upon the student to identify and fix any problems. At the end of the first class, I instruct the students to review and practice their spreadsheet *Skills* on their own. I pass out the quiz and have them do it at home.

In the second class, I distribute the diagnostic quiz answer key but not a detailed solution. I do not collect nor grade the quiz-this signals that these *Skills* are expected and learning them will not be rewarded. I tell the students they should use the diagnostic quiz as guidance for self-study. Their job is to figure out why the correct answers are correct. I insist that they do this *outside of class*. After all, this is review of material they are supposed to know, not coursework. I encourage them to collaborate with their fellow students.

I make myself available during office hours and by appointment to discuss spreadsheet concerns. I steadfastly refuse to explain the answers to quiz questions. I show the students how *they* can discover the answer by experimenting with the spreadsheet, guide them in approaching a fellow student for help, and sometimes counsel them to drop the course and take a spreadsheet training class. This has three powerful effects. First, it indicates my interest in their learning. Second, it helps to establish a culture that students are responsible for their own learning. Third, it quiets complaints from students with significant *Skill* deficiencies. These effects are particularly important for undergraduates.

I do ask the students to report to me their self-assessed diagnostic quiz scores, which I keep confidential. This helps me gauge the aggregate level of preparation and enables comparison of scores across semesters. I can identify students who are at risk due to very poor spreadsheet preparation. I counsel these individuals on the importance of quickly honing their *Skills*, confirm they have an adequate plan, and follow up with them to track their progress.

I don't think it is appropriate for me to invest much energy on remedial *Skills*. Because I don't grade it, I can use the same diagnostic quiz every year and not worry about its confidentiality. The benefit of the quiz is not due to the quality of the questions. Instead, the quiz reinforces that the students are responsible for their learning during my course-and in later life.

7.2. Observing Programming and Modeling Skills Using a Spreadsheet Modeling Exercise

In the second or third class, I give the students an easy individual spreadsheet modeling exercise. I use the Wall Exercise (Panko and Sprague, 1998), which is a one-paragraph problem statement of a very simple analysis. (A copy is in Appendix 2.). Unlike the quiz, it evaluates the students' ability to apply the *Skills* in a model-building context. The quiz quickly establishes who can and cannot create a simple spreadsheet model. This can be done inside or outside of class. I prefer to do this exercise during class time because it is very informative to observe students' individual techniques for spreadsheet modeling and programming. I marvel at the myriad approaches they take. I take notes on what they do for later use when we explore the spreadsheet engineering topics of modeling and design.

I tell the students the correct answer to the exercise in class. We spend a few minutes discussing how they approached the exercise. I tell them that they need to understand any errors that they made and how to avoid them in the future.

This spreadsheet modeling exercise has benefits beyond evaluating *Skills*. It provides feedback to the students and me regarding students' ability to create a useful model, which is essential to the use of management science. To motivate *Capabilities* and *Practices* later in the course, I display selected students' Wall spreadsheets during class. This establishes accountability and motivates students to work hard because a student likes it when his or her spreadsheet is held up as an exemplar of good technique-and cringes when it is the opposite. I sometimes assign students to re-

design their Wall spreadsheet for homework during the spreadsheet design topic.

7.3. Web-Based Evaluation and Remediation

In May 2006, Responsive Learning Technologies made available a web-based tool (Wood, 2006) called the "MBA Prerequisite Spreadsheet Skills Assessor and Instructor" which was developed in cooperation with the Stanford Graduate School of Business. The product has a customizable 100-question evaluation. The questions are linked to instructional information to help students learn what they don't know. This product can be used before students arrive on campus, which may allow for *Skills* to be sorted out prior to a management science course. I explored the tool with my colleague Steve Huxley. We were favorably impressed and are testing it with students in the fall of 2006. This tool, and competitors that may emerge if it is successful, have high potential for establishing a solid foundation of spreadsheet *Skills* prior to a management science course.

8. Instructor-Led Remediation of Skills in a Review Session

An alternative to the Evaluate-and-Fix approach is to run one or more Review sessions. With a Review, the instructor prepares training materials on spreadsheet *Skills* and delivers the materials in an appropriate way. When I do this, I don't worry about individual deficiencies and self-remediation and simply deliver remedial material to the entire class. Some schools have an optional pre-enrolment course that covers Excel before the students enter a management science course.

I dislike doing a spreadsheet *Skills* Review session, since I believe it is an unfortunate use of class time. However, when enough students are very badly prepared, this may be necessary. After all, the students can't learn what I really want to teach them until they have an adequate foundation of spreadsheet *Skills*.

A Review can be scheduled in the course syllabus or it can be added during the course if the diagnostic quiz, modeling exercise, or other information indicate unusually poor aggregate preparation. When I do a Review, I perform it in the first or second week of class. We do hands-on active learning, which requires that every student has their own computer.

I start with Familiarity *Skills* and move on to Fundamental Programming *Skills*. Students spend most of their time typing into Excel, sometimes starting with a blank spreadsheet, sometimes using simple spreadsheet templates that I provide.

9. High Level Capabilities and Practices

The high-level *Capabilities* and *Practices* should be delivered after all students have a solid foundation of *Skills*. I believe the *Capabilities* and *Practices* should be woven into the fabric of the course to establish that a professional is always interested in strengthening his spreadsheet engineering abilities, just as a professional is always interested in strengthening his analytical and communications abilities.

In my experience, spreadsheet *Capabilities* and *Practices* can be delivered in three ways. First, I deliver some of the *Capabilities* and *Practices* as standalone topics in the course. I deliver principles using readings, demonstrations, and sometimes mini-lectures. These principles are always accompanied by hands-on active learning application, where students perform a task in the spreadsheet either alone or in small groups. I do this for Modeling, Design, Programming, Documentation, and Development Parameters. The Management of Spreadsheet Model Assets is an exception, in that it has no hands-on content; instead I deliver it as a guided class discussion.

Second, I deliver some of the elements in mini-modules throughout the course. A full class session on programming can be overwhelming, and I find that students do not retain very much of what they did. I organize some content (for example Intermediate Programming and the Advanced Programming) as notes in their own sections of the course pack. I deliver them as active learning in 10-30 minute doses throughout the course. When teaching night classes, I find this particularly useful during the last half hour when students are tired and can't handle more cognitively-challenging material.

Third, I integrate the *Capabilities* and *Practices* throughout the course. I role model them whenever I do demonstrations in class, and illustrate them in all example models in the course notes. I point out good and bad examples in the textbook. We discuss the *Capabilities* and *Practices* when students build models in

front of the class or demonstrate their models to classmates.

The *Capabilities* and *Practices* are reinforced by regular practice in homework and in class. I require that all homework models demonstrate good spreadsheet engineering, and we discuss different designs and programming choices for these models in class. For example, when we consider the classic transportation problem during the optimization module of the course, I have the students build from scratch a what-if model in advance of class, and we explore different designs and programming choices during class. A matrix design highlights the inherent structure of the problem better than a columnar design and is more transferable to other people. However, a columnar design is more consistent with a "traditional" management science optimization implementation. The students can easily see the tradeoffs between design for transferability (matrix) and design for traditional optimization (columnar). The SUMPRODUCT function provides significant benefits in terms of programming accuracy and time when compared to cell-by-cell calculations.

I have students explore their model manually to evaluate alternative decision variable values. This provides multiple input/output combinations for them to evaluate. This approach is a key technique for spreadsheet quality control (Powell and Baker, 2004; Panko, 2006a). Thus, students obtain active learning in modeling, design, and quality control. In addition, the model exploration process prepares their minds for the insights that they will need to articulate when we perform optimization using the Solver add-in.

I note that spreadsheet modeling, design and programming are in some sense more important than the details of using Solver for optimization. This notion may seem strange at first. However, an accurate, transferable what-if model (without optimization) can add real value to an organization. In contrast, optimization of a flawed what-if model is a waste of time. Students need solid spreadsheet engineering *Capabilities* before they can meaningfully learn and apply management science.

10. Conclusion

In my opinion, the essence of management science is the application of algorithms to models to obtain business insight. Without a model, management science algorithms are meaningless mathematics. For business students and business analysts, models are virtually always built in spreadsheets. We need to educate students to efficiently build effective, transferable spreadsheet models.

Business School Management Science instructors have a marvelous opportunity. By claiming ownership of spreadsheet engineering we can dramatically increase relevance and student satisfaction with a business school management science course, while strengthening the foundation for the actual use of management science by our graduates.

People aren't getting what they need from mass-market spreadsheet training books or in their introductory spreadsheet training class because those sources don't provide the context needed to learn spreadsheet engineering *Capabilities* and *Practices*. Management science instructors have the context that is essential to learning these vital spreadsheet engineering abilities.

In my experience, students greatly value learning the *Capabilities* and *Practices*. Mastery of these abilities seems to make students more effective at tackling difficult cases during the course. The focus on spreadsheet engineering has greatly increased students' perception of the relevance, importance, and usefulness of my management science course. This makes them eager, willing participants in what I teach and dramatically increases their desire to devote limited out-of-class time to my course rather than to competing "sexy" courses such as finance, marketing, or leadership. This lets me teach a more demanding course with very high expectations of my student's performance.

High-level spreadsheet *Capabilities* and *Practices* can only be learned in the context of solving a problem. They can not be acquired in a standalone "introduction to spreadsheets" course. They are best delivered in a course with analytical content, such as a management science course. Perhaps one day introductory spreadsheet courses will have sufficient problem-solving context that students will arrive with good *Capabilities* and *Practices*, but until then we can aid our students by delivering these abilities in a management science course.

Faculty must thoughtfully choose which spreadsheet *Skills*, *Capabilities* and *Practices* they incorporate into

their course. They may choose to reduce the quantity of traditional management science topics in order to develop the essential spreadsheet engineering foundation for actual usage of management science.

Instructors have many choices to make regarding spreadsheet *Skills*, *Capabilities* and *Practices*. These choices depend on the goals of the school, program, and students, as well as the type of work the graduates will engage in. Different choices will be appropriate in different settings. Key questions to ask when designing a course include:

- How do you ensure that your students have an adequate foundation of low-level spreadsheet *Skills*?
- What is the appropriate mix of traditional management science content, versus spreadsheet *Capabilities* and *Practices*?
- How can you deliver spreadsheet *Skills*, *Capabilities*, and *Practices* appropriately and effectively?
- How can you assess student learning?

Issues to consider when writing a detailed syllabus include the following:

- Early evaluation of low-level spreadsheet *Skills*
- Early remediation of *Skills* deficiencies
- Choice of high-level spreadsheet *Capabilities* and *Practices*
- Integration of *Capabilities* and *Practices* throughout the course
- Class modules on selected *Capabilities* and *Practices*

Enhancing our students' spreadsheet abilities is good for us because it enhances student support and interest for management science. It's good for our profession because spreadsheets are an essential prerequisite to business students and business graduates using management science. And best of all, it's fun.

11. Acknowledgments

I am grateful to the many contributors and attendees at the annual European Spreadsheet Risks Interest Group meetings (European Spreadsheet Risks Interest Group[2]), where a unique combination of business people, academics and students provided powerful inspiration and insight, and to the members of the Spreadsheet Productivity Research Interest Group (Spreadsheet Productivity Research Interest Group[3]), which is working to become as successful as EuSpRIG. I want to thank the Associate Editor for her tireless efforts to improve the quality of the exposition, and two referees for their useful comments. Any errors or omissions are my responsibility alone.

References

Albright, S. C., W. L. Winston and C. J. Zappe (2001), "Excel Tutorial to Improve Your Efficiency," available at http://www.kelley.iu.edu/albrightbooks/#Tutorial , accessed May 30, 2006.

Bell, P.C. (2000), "Teaching Business Statistics with Microsoft Excel," *INFORMS Transactions on Education*, Vol. 1, No. 1, http://ite.pubs.informs.org/Vol1No1/Bell/

Chadwick D., L. Strous, D. Ward, P. Cleary, and G. Croll eds. (2005) "European Spreadsheet Risks Interest Group (EuSpRIG) Combined Proceedings 2000-2004," European Spreadsheet Risks Interest Group, Ipswich, UK, ISBN 1-902724-02-X, http://www.eusprig.org/

Croll, G. (2005), "The Importance and Criticality of Spreadsheets in the City of London," European Spreadsheet Risks Interest Group 6th Annual Symposium, London, UK.

Franklin, J. (2006), Personal Communication, September 8.

Grinde, R. B. and J. A. Kammermeyer (2003), "Experiences Using Thematic Assignments in an Undergraduate Management Science Course,"

[2] http://www.eusprig.org/
[3] http://sprig.section.informs.org/

INFORMS Transactions on Education, Vol 4., No. 1,
http://ite.pubs.informs.org/Vol4No1/GrindeKammermeyer/

Grossman, T. A. (1999), "Why Spreadsheets Should Be in OR/MS Practitioners' Tool Kits," *OR/MS Today* Vol. 26, No. 2, pp. 20-21, April.

Grossman, T. A. (2002), "Spreadsheet Engineering: A Research Framework," Proceedings of the European Spreadsheet Risks Interest Group 3rd Annual Symposium, Cardiff, Wales. Available in Chadwick et al 2005.

Grossman T. A. (2004), "Spreadsheet Analytics," http://www.spreadsheetanalytics.com/ (accessed May 9, 2006)

Grossman, T. A., V. Mehrotra and Ö. Özlük (2006), "Lessons from Mission Critical Spreadsheets," University of San Francisco School of Business and Management Working Paper.

Grossman, T. A. and Ö. Özlük (2004), "A Paradigm for Spreadsheet Engineering Methodologies," European Spreadsheet Risks Interest Group 5th Annual Symposium, Klagenfurt, Austria. Available in Chadwick et al 2005.

Hardin, J. R. and A. J. Ellington (2005), "Using Multimedia to Facilitate Software Instruction in an Introductory Modeling Course," *INFORMS Transactions on Education,* Vol. 5, No. 2, http://ite.pubs.informs.org/Vol5No2/HardinEllington/

Isken M. W. (2003), "Data Cleansing and Analysis as a Prelude to Model Based Decision Support," *INFORMS Transactions on Education*, Vol. 3, No 3,available at http://ite.pubs.informs.org/Vol3No3/Isken/

Jordan, E., L. Lasdon, M. Lenard, J. Moore, S. Powell, and T. Willemain (1997), "OR/MS and MBAs," *OR/MS Today*, Vol. 24, No. 1, pp. 36-41.

Leblanc, L. and T. Grossman (2006), "Call for Papers: Interfaces Special Issue: Spreadsheet Applications of Management Science and Operations Research," *Interfaces,* Vol. 36, No. 2, p. 108, April.

McConnell, S. (1996), "Rapid Development: Taming Wild Software Schedules," Microsoft Press: Redmond, Washington.

Nardi, B. and J. R. Miller (1991), "Twinkling lights and nested loops: Distributed problem solving and spreadsheet development," *International Journal of Man-Machine Studies*, Vol. 34, pp. 161-184.

Oggier, C., E. Fragnière, and J. Stuby (2005), "Nestlé Improves Its Financial Reporting with Management Science,", *Interfaces,* Vol. 35, No. 4, pp. 271-280, July-August.

OR/MS Today (2002), "Spreadsheet Add-Ins for OR/MS Software Survey," *OR/MS Today,* Vol. 29, No. 4, August, available at http://www.lionhrtpub.com/orms/orms-8-02/frsurvey.html , accessed May 30, 2006.

Panko, R. R. (1998), "What We Know About Spreadsheet Errors," Journal of End User Computing, 10(2), pp. 15-21. Available at http://panko.cba.hawaii.edu/ssr/Mypapers/whatknow.htm

Panko, R. (2000a) "Errors in Spreadsheet Auditing Experiments," http://panko.cba.hawaii.edu/ssr/auditexp.htm , accessed November 6, 2005.

Panko, R. (2000b) "Errors During Spreadsheet Development Experiments," http://panko.cba.hawaii.edu/ssr/devexpt.htm , accessed November 6, 2005.

Panko, R. R. (2006a), "Recommended Practices for Spreadsheet Testing," European Spreadsheet Risks Interest Group 7th Annual Symposium, pp. 73-84, Cambridge, England, July.

Panko, R. R. (2006b), "Spreadsheet Research Website," http://panko.cba.hawaii.edu/ssr/ , accessed September 14, 2006.

Panko R. R. and R. H. Sprague (1998), "Hitting the wall:errors in developing and code inspecting a 'simple' spreadsheet model," *Decision Support Systems*, Vol. 22, No. 4, pp. 337-353.

Papageorgiou, J. C. (1996), "Are We Promoting OR/MS to Our Future Clients?" *International Journal of Operations and Quantitative Management*, Vol. 2, No. 3, pp. 231-38.

Powell, S. G. (2001), "Teaching Modeling in Management Science," *INFORMS Transactions on Education,* Vol. 1, No. 2, http://ite.pubs.informs.org/Vol1No2/Powell/

Powell, S. P. and Baker, K. R. (2004), "The Art of Modeling With Spreadsheets: Management Science, Spreadsheet Engineering and Modeling Craft," Wiley.

Pryor, L. (2006), "What's the Point of Documentation," European Spreadsheet Risks Interest Group 7th Annual Symposium, pp. 159-162, Cambridge, England, July.

Regan P. J., (2005), "Professional Decision Modeling: Details of a Short MBA Practice Course," *INFORMS Transactions on Education,* Vol. 6, No. 1, http://ite.pubs.informs.org/Vol6No1/Regan/

Regan P. J., (2006), "Professional Decision Modeling: Practitioner as Professor," *Interfaces*, Vol. 36, No. 2,pp. 142-149, March-April.

Savage, S. (1997), "Weighing the Pros and Cons of Decision Technology in Spreadsheets," *OR/MS Today,* Vol. 24, No. 1, February.

Savage, S. (2003), "Weapons of Mass Instruction," *OR/MS Today*, Vol. 30, No. 4, August.

Willemain, T. R., and S. P. Powell (2006) "How Novices Formulate Models Part I: A Quantitative Description of Behavior," Working Paper, Department of Decision Sciences and Engineering Systems, Rensselaer Polytechnic Institute.

Wood, S. (2006), Personal Communication, Responsive Learning Technologies, http://www.responsive.net/

Appendix

Appendix I - Diagnostic Tests - Excel

These questions are designed to test your understanding of *basic* Excel techniques. They are meant to be answered without using Excel itself. Some questions may have more than one correct answer, and each question is independent of the others. Answers are available at the end of the quiz.

1. If the number 829.24 appears in a cell, which of the following could be the number as actually stored in Excel?

a. 829.3
b. 829.24567
c. 829.2389
d. 829.2

2. True or false: All column widths in a single worksheet must be identical.

3. Excel workbooks have the filename extension

a. .doc
b. .exe
c. .xls
d. .wbk

4. Which of the following worksheet entries is considered text?

a. 127
b. -85
c. 555-1212
d. .098

5. Which of the following keys can you use to fix typos before you press the Enter key?

a. Page up
b. Delete
c. Home
d. Backspace

6. When you first open a new workbook, the tab label for the first worksheet is

a. A1
b. Sheet1
c. The same as the workbook name
d. The same as the filename

7. All Excel formulas begin with

a. F
b. A1
c. =

d. SUM

8. Which of the following is equivalent to the formula B17+B18+B19+B20?

a. =B17:B20
b. =B17*B20
c. =SUM(B17:B20)
d. =TOT(B17:B20)

9. If you replicate a formula and some of the resulting cells contain #VALUE!, you probably:

a. Need to use an absolute reference in the formula you're trying to replicate.
b. Forgot to begin the formula with an equal sign.
c. Tried to copy the formula to nonadjacent cells.
d. Used a ? instead of a $ for the absolute reference.

10. Which of the following are *not* relative addresses?

a. C$1
b. AA649
c. $A$94
d. J42

11. Which of the following IF-statements *always* give a numeric answer?

a. IF(D7>=99,1,0)
b. IF(D7>99,1)
c. IF($D7<=99,1,0)
d. IF(D7>99,1,"NA")

12. Which of the following formulas can be used to add numeric values in cells B2:G4?

a. SUM(B2:G4)
b. SUM(SUM(B2:G2), SUM(B3:G3), SUM(B4:G4))
c. SUM(B2, G4)
d. SUM(B2:B4) + SUM(C2:C4) + SUM(D2:D4) + SUM(E2:E4) + SUM(F2:F4) + SUM(G2:G4)

13. If you enter the formula "=A2*(1+$A$1)" in cell B2 and then copy cell B2 to C2, the numerical result in cell C2 is:

	A	B	C
1	0.10		
2	100	=A2*(1+$A$1)	???

a. 109
b. 200
c. 110
d. 121

14. Suppose you have the formula =$D$5*E5 in cell F5. When you copy the formula into cell F6 what will the new formula be?

a. $D$6*E6
b. D5*E6
c. $D$5*E6
d. $D$5*E5

15. You type the number 110 in cell B2 and press enter. However, the values in other cells involving formulas that depend on B2 do not change. What could be wrong?

a. Nothing. Just press F9.
b. You need to save the spreadsheet and reopen it.
c. Automatic calculation is turned off.
d. There is a bug in Excel. Just try entering the number again.

16. Suppose you want to enter a formula to calculate a standard deviation, but you can't remember the formula from your statistics course. You can use Excel's

a. Sum button
b. Absolute references
c. Function wizard
d. Statistics AutoFormat

17. When Excel displays ####### in a cell, it means that

a. The formula in the cell contains an error.
b. The cell is not wide enough to display the number it contains.
c. The text in the cell cannot spill over into the next cell.
d. The disk is full.

18. When you insert a row in a worksheet, what happens to the formulas in the rows that are renumbered?

a. The formulas display incorrect results unless they contain absolute references.
b. Excel deletes any formulas in those rows.
c. Excel puts ###### in any cells that contain formulas you'll need to change.
d. Excel automatically changes any references in formulas that refer to renumbered cells.

19. An example of Excel's Fill Series command is:

a. Duplicating a formula from one cell into many other cells.
b. Making sure the cells are exactly wide enough to fit all the numbers in a series.
c. Completing a series of numbers, dates, or years, such as 1995, 1996, 1997, and 1998.
d. Creating a series of similar worksheets, but each with a unique tab label.

Appendix II - Diagnostic Tests - Answers

1. If the number 829.24 appears in a cell, which of the following could be the number as actually stored in Excel?

C. 829.2389

2. True or false: All column widths in a single worksheet must be identical.

False

3. Excel workbooks have the filename extension

C. .xls

4. Which of the following worksheet entries is considered text?

C. 555-1212

5. Which of the following keys can you use to fix typos before you press the Enter key?

D. Backspace

6. When you first open a new workbook, the tab label for the first worksheet is

B. Sheet1

7. All Excel formulas begin with

C. =

8. Which of the following is equivalent to the formula B17+B18+B19+B20?

C.=SUM(B17:B20)

9. If you replicate a formula and some of the resulting cells contain #VALUE!, you probably:

A. Need to use an absolute reference in the formula you're trying to replicate.

10. Which of the following are *not* relative addresses?

A. C$1

C. A$94

11. Which of the following IF-statements *always* give a numeric answer?

A. IF(D7>=99,1,0)

C.IF($D7<=99,1,0)

12. Which of the following formulas can be used to add numeric values in cells B2:G4?

A. SUM(B2:G4)

B. SUM(SUM(B2:G2), SUM(B3:G3), SUM(B4:G4))

D. SUM(B2:B4) + SUM(C2:C4) + SUM(D2:D4) + SUM(E2:E4) + SUM(F2:F4) + SUM(G2:G4)

13. If you enter the formula "=A2*(1+$A$1)" in cell B2 and then copy cell B2 to C2, the numerical result in cell C2 is:

	A	B	C
1	0.10		
2	100	=A2*(1+$A$1)	???

D. 121

14. Suppose you have the formula =$D$5*E5 in cell F5. When you copy the formula into cell F6 what will the new formula be?

C. $D$5*E6

15. You type the number 110 in cell B2 and press enter. However, the values in other cells involving formulas that depend on B2 do not change. What could be wrong?

A. Nothing. Just press F9.

C. Automatic calculation is turned off.

16. Suppose you want to enter a formula to calculate a standard deviation, but you can't remember the formula from your statistics course. You can use Excel's

C. Function wizard

17. When Excel displays ####### in a cell, it means that

B. The cell is not wide enough to display the number it contains.

18. When you insert a row in a worksheet, what happens to the formulas in the rows that are renumbered?

D. Excel automatically changes any references in formulas that refer to renumbered cells.

19. An example of Excel's Fill Series command is:

C. Completing a series of numbers, dates, or years, such as 1995, 1996, 1997, and 1998.

Appendix III - Spreadsheet Modeling Exercise

The paragraph below is the Wall Exercise, adapted from Panko and Sprague (1998). This exercise was used in fascinating research on spreadsheet errors and the poor ability of people to predict the accuracy of their spreadsheet model. By giving this exercise to students, they obtain a direct connection to the research literature.

"You are to build a spreadsheet model to help you create a bid to build a wall. You will offer two options-lava rock or brick. Both walls will be built by crews of two. Crews will work three eight-hour days to build either type of wall. The wall will be 20 feet long, 6 feet tall, and 2 feet thick. Wages will be $10 per hour per person. You will have to add 20% to wages to cover fringe benefits. Lava rock will cost $3 per cubic foot. Brick will cost $2 per cubic foot. Your bid must add a profit margin of 30% to your expected cost."