SOCIETY OF ACTUARIES

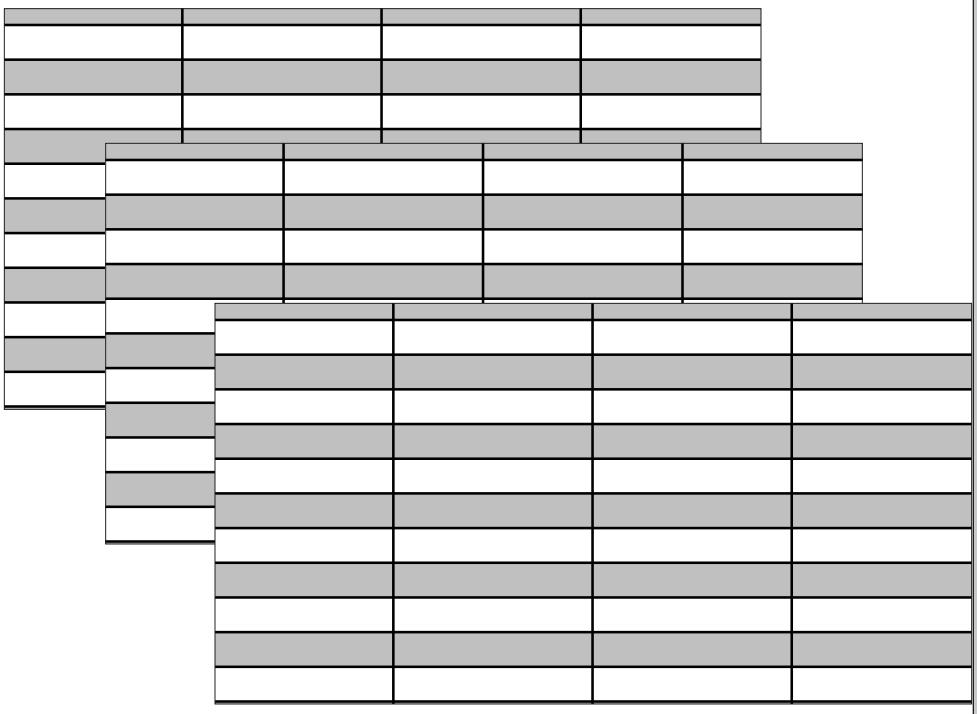
Article from:

The Financial Reporter

June 2006 – Issue No. 65

Spreadsheets – Yay or Nay?

by Bob Crompton



When I was a young actuary working in product development, we used an in-house system written in APL to perform our pricing work. Although the system was quite good for its day, it suffered from the usual shortcomings—limited documentation, inability to incorporate new features without significant coding and subtle bugs that would appear at inopportune moments—usually when I was working on a tight deadline.

One day the head of the department brought in a representative from a brokerage house in an attempt to set up another distribution channel. In his naïveté, the rep asked how long it would take to develop a new product with this system. The department head answered, "With this system, we can do it in a day."

Of course this statement was an egregious exaggeration, and could only be true if referring to a minor tweak to an existing product. I later told the department head that he was fortunate that the system didn't crash when the brokerage house rep was looking. Since that time, the proliferation of electronic spreadsheets has significantly improved actuaries' ability to respond in a timely manner to both market place developments and financial reporting requirements. Although a new product in a day may still be unrealistic, spreadsheets have improved response time over the bad old days of APL pricing systems.

However, spreadsheets have a dark side. Because of their nearly infinite flexibility, they can be used in many situations, including situations where they should not be used. Spreadsheets should be used when a quick solution is needed or when there is no existing programmatic solution. Spreadsheets should not be used for *production*—that is, for ongoing, periodic calculations or compilations of amounts for financial or management reporting.

Sarbanes-Oxley has put a spotlight on how errors can creep into spreadsheets. The horror stories relating to spreadsheet errors were so sufficiently circulated during the initial phases of Sarbanes-Oxley implementation that no repetition is necessary. These errors are, however, a red herring. It is possible to manage spreadsheets so that there are no material errors.

The real problem with spreadsheets is deeper and more endemic than intimated by Sarbanes-Oxley. Any spreadsheet used for production work is a nexus of inefficiency and a sinkhole of opportunity costs. Spreadsheets used for production work convert actuaries from risk analyzers into spreadsheet managers. The constant attention that these spreadsheets need drains resources that could be more profitably employed elsewhere. Actuaries, with their can-do attitude and facility with spreadsheets, have become enablers of poor management.

This problem is not amenable to correction, the way common errors are. Any spreadsheet that is complex enough to require an actuary is complex enough to require ongoing actuarial support. This ongoing support, combined with the notorious lack of documentation skills found in most actuarial shops results in business processes that devour valuable resources.

In the table on page 16, I compare some of the strengths of spreadsheets to some of the strengths of

continued on page 16 >>

formal programs. In this context, a formal program is used in the general sense of a program written in a formal programming language with a specific syntax and semantics, rather than in a more specialized sense of a program written or analyzed using formal methods.

Strengths

Spreadsheets	Formal Programs
Ubiquity	Scarcity
Facility	Complexity
Rapidity	Deliberateness

Ubiquity versus Scarcity

Spreadsheets are everywhere in the electronic ecology. Almost every single business computer has some form of spreadsheet installed. This widespread availability means that spreadsheets are ideal for dealing with urgent tasks.

The ubiquity of spreadsheets allows for the implementation of urgent processes by those who are directly concerned with the process—those who best understand the "what" and "how" of the *res* of the process.

Spreadsheets are everywhere in the electronic ecology. Almost every single business computer has some form of spreadsheet installed. This widespread availability means that spreadsheets are ideal for dealing with urgent tasks.

Formal programs on the other hand, at least those used for insurance company production purposes, have a limited existence. Such programs exist in one place, or perhaps in a few places. This scarcity is appropriate for business processes that are mature and well defined. Scarcity prevents mutant versions of the program. Changes are not made willy-nilly, but are made in a controlled manner, complete with testing.

Facility versus Complexity

While it may take a village to raise a child, it only takes a village idiot to create a spreadsheet. Spreadsheets are easy to learn. In fact, where I live, mothers routinely enroll their newborns in spreadsheet training classes so they will have a leg up for enrollment in private kindergarten.

Spreadsheets' ease of use makes it useful to people with no programming skills. In fact, this ease of use sometimes seems to unleash a great deal of creativity. I have seen spreadsheets of fascinating sophistication and breathtaking scope.

The facility of spreadsheets is closely connected to their ubiquity. Their ease of use is what makes them so useful on a widespread basis.

Formal programs, on the other hand, are written in a programming language with strict syntax and semantics that requires education and practice to use properly. This formalized structure is designed to deal with complex data structures in a precise manner.

Writing good code is a skill that requires both some theoretical knowledge as well as practical experience. In fact, writing good code is somewhat like actuarial science in the sense that most individual tasks are not very difficult—but there are a lot of individual tasks and both the programmer and the actuary have to exercise due care in making sure that all of the individual pieces fit together properly. One such area of fitting is in data structures.

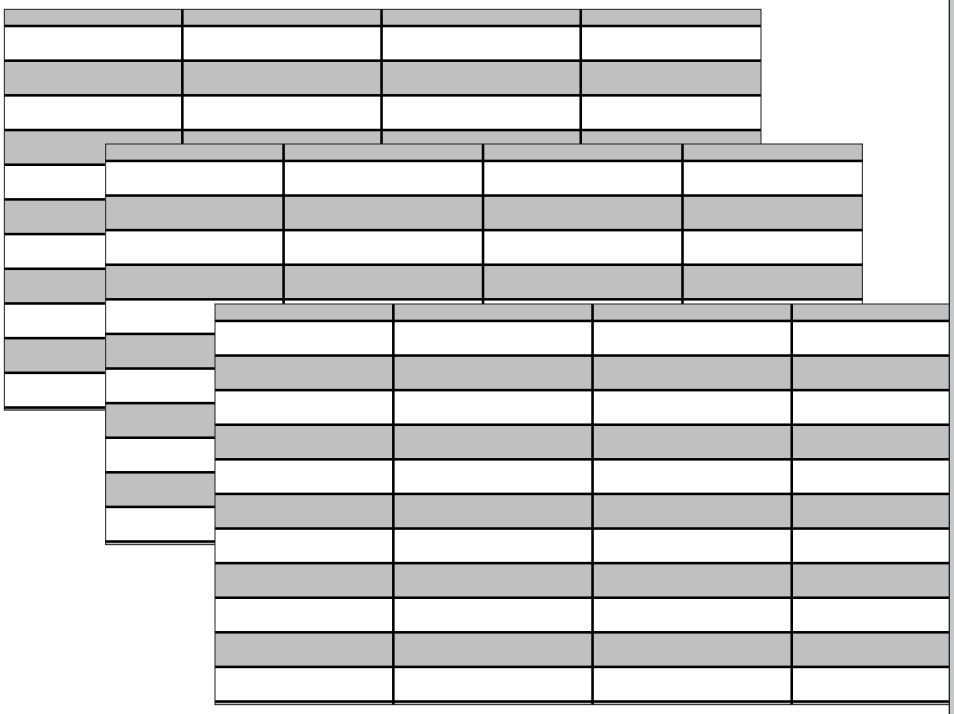
Data structures used for insurance production work are often more complex than is appropriate, or even possible, to use in a spreadsheet, where data and processing are mixed together in an electronic goulash. The complexity of formal programs allows for separation of data from processing, and for each to be designed for efficiency and effectiveness.

Rapidity versus Deliberateness

Rapidity here refers to the time between conceptualization and calculation. Spreadsheets can be created quickly—even spreadsheets of great complexity. And while developing a completely new product in a day may still be a fairy tale, the use of spreadsheets for development of new products or features can create significant efficiencies.

This speed advantage is the greatest benefit that spreadsheets bring to any enterprise. The ability to complete a project in weeks rather than the months or years required with formal programs is of inestimable value.

Formal programs, on the other hand, are developed with deliberateness. Although quicker is always better, the strength of formal programs is the process used to consider the business rules and possible circumstances likely to occur in practice and to design code to deal with these circumstances. Spreadsheets, on the other hand, are often operated on a hindsight basis, where results are certified by a review for reasonableness after the fact, with little or no before the fact review.



In addition, a formal program makes provision for situations that "don't compute." Rather than the dreaded #VALUE! that a spreadsheet will generate, a formal program has a specified procedure to deal with exceptions—even if it is only to print out the exception list so someone can perform processing manually (that is, on a spreadsheet!).

Formal programs also have extensive testing procedures to ensure that what the program does is what it was intended to do. This points out the distinction between spreadsheets and programs. Spreadsheets are ideal for testing (among other things), while programs are ideal for ongoing and extensive calculations. In addition, formal documentation rules for formal programs encapsulate the thought processes and business rules that lie behind the items being processed.

The deliberateness of formal programs is an investment and becomes part of the operating leverage of the enterprise.

Some Examples

Three examples are given below. One is a good example of the use of spreadsheets, one is a bad example and the third is somewhere between good and bad.

Example 1: The Good—SOP 03-1

When SOP 03-1 became a requirement, some (perhaps many) companies had no formalized valuation process in place for this GAAP requirement. Because there was no off-the-shelf solution available, and because lead time for in-house coding was insufficient for many companies to implement formalized programs to handle SOP 03-1 (given the amount of time for actuaries to become aware of the pronouncement, digest the requirements and then write programming specifications, followed by the normal Code-Test-Revise cycle of the programmers), the first time through SOP 03-1 for some companies was performed on spreadsheets.

This use of spreadsheets as a stopgap valuation measure fits well with spreadsheets' strengths (ubiquity, facility and rapidity). SOP 03-1 knowledge was not typically widespread at year-end 2004, but spreadsheets were available to actuaries who were knowledgeable. These actuaries, often working under tight deadlines, were able to craft temporary solutions to the SOP 03-1 requirement.

In addition, the use of spreadsheets to prototype SOP 03-1 calculations, or to prototype any type of involved and extensive set of calculations, seems to fit better with our "natural" way of thinking than does the stylized "Plan–Implement–Test" way of thinking that is used in formalized systems. Working out prototypes on a spreadsheet seems to enhance the process of intellectual discovery, to assist in thinking through issues and to give a more complete definition to the problem being analyzed.

Example 2: The Bad—VOBA Amortization

One bad example I have seen is VOBA amortization performed on a gigantic spreadsheet (multi-gigabyte size). This spreadsheet is updated quarterly, requiring several days of actuarial involvement at each update. The personnel involved were experienced ASAs or higher, rather than lower-level actuaries. Checking and sign-off of the results was performed at an even higher level of actuarial experience.

Although this process has produced results that are acceptable for financial reporting purposes, there is a business issue lurking behind the spreadsheet. Why has so much time and effort been invested in such an inherently inefficient process? Sure, it keeps a few actuaries off the streets who might otherwise be involved in mortality table fraud or deferred premium shell games, but it is a drain on the enterprise's resources. With both a global oversupply of programmers and the availability of reasonable off-the-shelf valuation products, it makes no sense to tie up actuarial talent on this process. None of the strengths of spreadsheets apply here, while the strengths of

continued on page 18 >>

formalized code (scarcity, complexity and deliberateness) since the process is stable and well defined.

Example 3: The In-Between—DAC Amortization
The following example illustrates that an enterprise can recognize an issue and take at least some corrective action, even if there was no implementation of formalized code. Low hanging fruit is everywhere—you just have to look!

For this enterprise, DAC amortization schedules (both FAS 60 and FAS 97) are created by actuaries, but are passed to the financial area for periodic financial reporting. Application and updates of existing schedules are performed almost entirely by accountants rather than actuaries. Ongoing actuarial involvement is limited to final review and sign-off and any FAS 97 unlocking.

This solution relieves the enterprise of the ongoing opportunity cost attributable to maintenance of spreadsheets by actuaries. Even though there are still improvements that can be made, this is a better solution than leaving the spreadsheets in the hands of the actuaries.

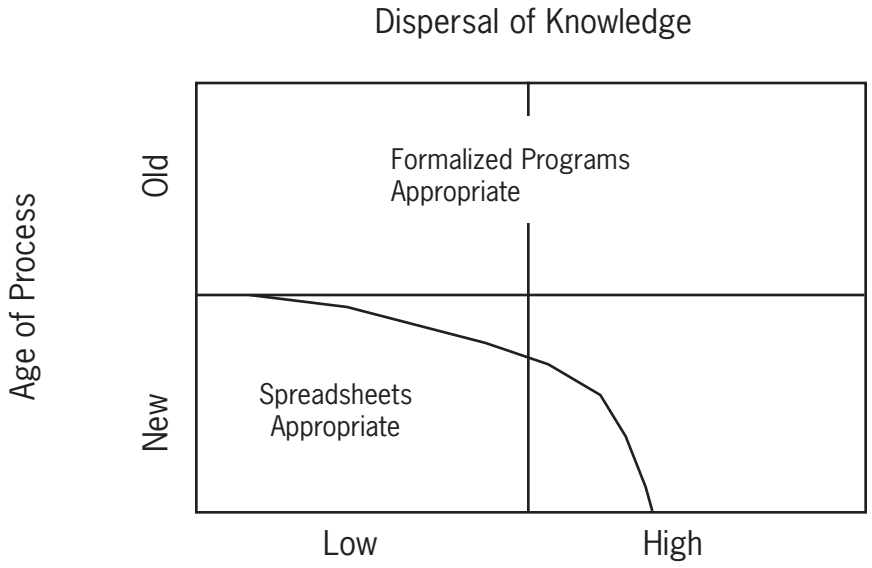
These examples further suggest when spreadsheets are appropriate and when they are not appropriate. The chart below captures the idea that spreadsheets are appropriate to use for processes that are new and for which understanding of the process is not widely dispersed through the enterprise.

This chart below also points out an interesting sidelight—that is, that one of the ways in which actuaries add value to an enterprise is through broadening the knowledge base so that normal processes can be accomplished without constant high-priced intervention.

Conclusion

Spreadsheets are valuable for urgent tasks or exploratory tasks. The ability to quickly generate results in spreadsheets has enabled insurance companies to be more nimble and responsive.

However, spreadsheets have also been used in situations where their use is inappropriate, sometimes resulting in considerable inefficiency. Actuaries who are too in love with their numbers to relinquish their spreadsheets do a disservice to their organizations. §



Robert B. Crompton, FSA, MAAA, is a consulting actuary with Actuarial Resources Corporation in Alpharetta, Ga. He may be reached at bob.crompton@arcga.com.