Article from

The Modeling Platform

March 2017
Issue 5

Excel and Modeling Governance: What Can We Do Better?

By Tim Heng

More than four years ago, I wrote an article in the *CompAct* newsletter responding to calls for actuaries to move away from Excel and onto more specialized actuarial software such as MoSeS and Prophet.[1]

Looking around at the start of 2017, a few things are apparent:

- Everyone is still using Excel.
- Spreadsheets still contain errors.
- Graduates still don't have the requisite spreadsheet skills upon entering the workforce.
- Companies still aren't regularly enforcing good model governance.

In short, not much has changed!

EXCEL ISN'T GOING ANYWHERE

Every so often, it becomes more fashionable to predict the death of Excel, in favor of other systems that are either (a) more user-friendly and/or faster to use, (b) less prone to error or (c) more powerful and/or capable. It's a bit of a pipe dream though—it's like having the car that has great fuel economy, has lots of power and is cheap to purchase. Chances are, it's a case that you choose one and compromise on the others.

There is an inherent contradiction in the earlier requests. If something is more capable and gives you greater flexibility, then by definition, you open yourself up to a greater range of errors as more things can go wrong. If you lock down parameters that would cause errors, then you give up the user control that may be required to solve problems outside the narrow operating scope. If you create something that is faster to develop, it's likely at the expense of implementing features that you may want to use.

Excel is by no means perfect. However, it represents a balance between usability and accessibility, modeling integrity features (without requiring their use) and the flexibility to be a jack-of-all-trades—a piece of software that you can use for budgeting, pricing, reporting, data transformation, valuations and a range of other day-to-day tasks that upward of one billion users worldwide use Excel to solve (depending on what estimates you look at). For that reason, the question of power and capability will be ignored throughout the rest of this article—Excel is more than capable enough to solve key modeling issues.

It's generally safe to say at this point that Excel isn't going anywhere, at least not anytime soon. So rather than dream up impossible software solutions to the issues at hand, we can look at company policy and process to address the issues and make spreadsheets more user-friendly and faster to use, less prone to error and more capable of completing the tasks required of them.

EASY TO USE OR BETTER TRAINED?

Excel is already one of the easiest pieces of analytical software on the market. Simply install your chosen version of Office, click the green X, and start doing your calculations. There is little to learn by way of syntax and coding. Despite this, spreadsheets often take a long time to produce, partly because the vast majority of users use only about 1 percent of Excel's capabilities and are unaware of tools and features in it that can improve the efficiency of their work. It would be fair to say that even 99.9 percent of Excel users use no more than 10 percent of its capabilities. If you don't believe this, then ask yourself—how many different types of functions and features in Excel would you use on a day-to-day basis, out of the 500 or so functions available? There are also 46 buttons on just the Home tab of the Ribbon, not counting all the submenus and options available under those buttons.

> a) easy to use; b) less prone to error; c) more powerful—choose one and compromise on the others.

It's not just the knowledge of the tools available, however. It's interesting to note how the treatment of Excel usage (and Microsoft Office more generally) in companies differs from the use of other software tools. I recall being introduced to SAS early in my career at a major bank, being sent on specific training courses on how to use SAS to solve broad types of problems, and receiving user guides and manuals that I could reference if I ran into problems. Regular retraining was scheduled when new software updates became available and new tools were unlocked. In short, the business made an investment to ensure that skills were up-to-date and that new staff were properly trained to use the tools available, to achieve specific outcomes using the tools.

Now we can contrast this with how a typical company treats Excel training. I have never been fortunate enough to be sent

to any sort of formal Excel training course, other than on the teaching side of the classroom. It's assumed that people start off with an inherent ability to use basic Excel, and that they can simply pick up what they need to know from colleagues and by observing more advanced Excel in action. Occasionally, staff may attend general training courses (often labeled Beginner, Intermediate and Advanced), rather than targeted courses that relate to the sorts of business problems that the Excel user will be dealing with in the office.

There's a general understanding that because Excel is easy to use, if you have the tools in the toolbox (you've attended courses to learn the functions), you're capable of building complex spreadsheets that use those tools. That's just as true as the understanding that as long as you know how to use the tools, you can build a house (i.e., *not true at all!*). It's one thing to know how the functions work that are used in a financial model. It's completely different to have been trained how to build a financial model. Here's another analogy—just because a person has a great vocabulary and has read a lot of books doesn't mean that he or she will be any good at *writing* a book.

So to answer the first criticism that other software vendors level at Excel, you don't need a more user-friendly program or one that's faster to use. Instead, what you need is an appropriate level of training to use the software you already have.

INHERENTLY PRONE TO ERRORS, OR ERRORS OF PROCESS?

In my previous article in *CompAct*, I highlighted that a large portion of the fault around the number of errors in Excel models can be attributed to the process of Excel file development when compared to the process of more specialized software development. For people interested, it's worth a quick read—for the time constrained, here is the summarized version: Excel models have more errors because we (collectively, across any and all industries) do not adequately scope, review, test or document Excel models—at least, not to the same rigor that we might apply to models built in specialized software, or to the development of specialized software in the first instance.

It seems a simple matter to say "Do these things better!" and then suddenly errors will start to disappear. However, we then need to ask how we can do these things better, what it will cost and whether the costs sufficiently outweigh the benefits. Let's break it down into sections.

Scoping

Too often, scoping comes in the form of a discussion across a meeting room table, or a brief email containing high-level model requirements. Very rarely is a scoping document prepared for an Excel-based model. If you're looking for a checklist of things that you might want to know before building an Excel model, it obviously changes by industry and model type, but here's a few to get you going:

- Statement of the model's purpose
- Identifying the model end user and the format of presentation
- Delivery time frame
- Determining who is responsible for signing off on the scoping document
- Determining who is responsible for signing off on completion of the model
- Determining who is responsible for ongoing updates and changes to the model
- Highlighting both explicit assumptions (e.g., numeric assumptions entered in the model) and implicit assumptions (e.g., relationships between variables)
- How the model is to be structured: time series going down or across the page? Deep sheets with lots of rows or many shallow sheets?
- What will the review and/or quality assurance (if any) involve?

This list can go on for a very long time! Of course, contractors and consultants will often use lists like this, but it's rare for internal staff to be this thorough in documenting model requirements. More likely, the documentation is an email along the lines of

> *"Hi, Bill! Can you help me work out what's a fair price for ABC Enterprises? I need it for the next board meeting on Tuesday. Thanks."*

The act of writing down a set of requirements inherently reduces the risk that the model produced is not going to achieve the desired outcomes. If we scoped out Excel models with the same rigor that we scope other models, many errors and issues that normally arise would be dealt with long before any formulas hit the cells.

Review/Testing

I'm going to lump the two of these together, even though they're slightly different. Testing refers to an internal process where the model is provided to either a dedicated test team or to the end users, and the opportunity is there to use the model and provide feedback to the modeling team on issues relating to functionality, usability and accuracy. Issues raised may be in or out of scope—it's common for new issues to arise after a first draft has been created and the implications of scoped items

realized. This is an important part of any modeling process: to ensure the satisfaction of the end user.

The review component is perhaps even more critical, although it is often overlooked. Review processes in companies can vary from the following:

- It's the responsibility of the model builder to check his or her work before delivering it.
- A colleague must peer review a model before it is delivered, by looking at outputs and "high-risk" calculations.
- Whenever a model is built, an unrelated third-party will check the work before it is signed off, by looking at outputs and "high-risk" calculations.
- Whenever a model is built, an unrelated third-party will check the work before it is signed off, by inspecting every single unique formula and checking for logical and mathematical accuracy.
- … Wait, what review process?

While the last is laughably common in companies around the world, the implicit understanding is that the minimum level of responsibility requires model builders to do their own sense checks and ensure that they are not being negligent in their work.

The other forms of review processes require an investment from the company to receive assurance that the model contain no material errors. Often, it will be seen that a peer review is a "free" (read: sunk cost) form of assurance, since no cash is being paid for the service (explicitly). Realistically, though, if this is part of the culture of a modeling team, it should be perfectly clear that there is no such thing as a "free lunch," and any time spent peer reviewing is time that needs to be paid for, generally by having a slightly larger team than they would use without such a policy. Generally, the benefits of doing so outweigh the costs!

A key problem with the peer review process is that there can still be interference of a political nature, as well as an inherent bias in the way of thinking of different members of the same team. There is also the bias of expectation when outputs are reviewed by members within a company—if a sales target is set at $100m, and the model yields a result close to that number, fewer questions are asked. Even though probability would dictate that, on average, a modeling error would have an equal chance of sending an output up or down, in practice, most errors result in a negative impact on a company's forecast, indicating a persistent bias for companies to identify primarily positive errors upon internal review.

My personal favorite story in this regard is of a company in Australia whose models all pointed to secondary product revenue of $130M, which matched previous estimates (about 9

percent of the total forecast company revenue). However, all previous estimates were based on a calculation that converted dollars into millions of dollars, but divided by 100,000 instead of 1,000,000, meaning the true revenue forecast should have been $13M. Shortly after this was identified, this company went into administration (roughly equivalent to a Chapter 7 bankruptcy for U.S. readers).

This is where a third-party reviewer can add value. Moreover, third-party reviewers will often sign off on a model, putting their professional liability insurance coverage on the line, ensuring that they will perform a thorough inspection. The process usually involves an initial review of the model by the external reviewer, who highlights any errors, issues and questions, and passes it back to the company. Modelers will then update the model and address those issues, and pass it back to the external reviewer to see if the changes satisfactorily solve the problems. This is generally referred to as one "iteration" in the review process, and most reviews will usually take three to five iterations before all of the issues are ironed out, because fixing some issues inevitably cause or shine light on others.

This provides security for all stakeholders—both inside the company and external entities such as banks, shareholders and other involved parties. However, this comes at a cost: model reviews for transactions, for example, are primarily done on a unique formula basis and result in costs ranging from $30K to $100K by the time the model is finalized. As such, this sort of review process might best be left for particularly sensitive internal models or for those models that are being relied upon by a range of stakeholders who are all collectively seeking formal assurance. For less sensitive models that just need an independent set of eyes, reviews can usually be done by consultants on an hourly rate basis.

For other spreadsheets in an organization where there isn't the budget to throw thousands of dollars at a review process, it is important to set up standardized procedures and checklists to make any sort of peer review more effective and overcome the bias effects. These might include the following:

- Ensuring that the peer review is not conducted within the same team, or at least, not by a subordinate (to reduce political risk—would you tell your manager that they can't model?).
- Choosing peer reviewers who are capable of thinking "outside the box" and who don't necessarily do things the same way. This is more likely to catch issues that might otherwise be considered "standard" processes.
- Performing key high-level tests such as:
  - Ratio analyses
  - Sensitivity analyses
  - Chart inspection

  These items are likely to highlight unusual or unlikely results in outcomes that might not be immediately evident in the primary output.

Documentation

The final step of the modeling process that is often poorly implemented revolves around documentation. Different types of relevant documentation need to be considered:

- Notes describing the functions/workings of the model
- Style guides and formats for models
- Modeling guidelines or policies describing the process of model development within a business

The first is self-explanatory—documentation should contain the necessary information for a reasonable user to understand the model's purpose, how it achieves that purpose, and any assumptions, restrictions or other concerns that may relate to the model and how it is to be used.

The second type almost falls more into a marketing-type category, where styles, colors and formats are chosen around what the company brand represents. However, this form of documentation is far from trivial—by having a standardized color scheme, for example, users throughout a business can pick up any model with confidence, knowing that if yellow cells represent assumptions, then any yellow cell they see will contain an assumption. This helps to create an intuitive understanding throughout a business of how models are to be used—a form of implicit documentation, if you like.

The final type is one that is commonly found lacking in companies, particularly regarding Excel models. Several "best practice" modeling frameworks have been put forward globally, ranging from highly technical papers that dictate how models should work, right down to the functions being used, to high-level guidelines that seek to help clarify and provide guidance on what to do when developing a model.

At our company, we follow four main guidelines—that models should be consistent, robust, flexible and transparent. Personally, I find a modeling policy based on guidelines more useful from a practical perspective, because it gives you the flexibility to adapt rules around the specific requirements that a model might have, if you keep the high-level goal posts in line. However, there is merit in having a more clearly defined, rules-based approach, where you can enforce strict modeling standards to apply consistency across a company's models. Some might also say it's more useful from a practical perspective, because you have specific rules and standards to apply to each modeling situation. I'll leave you to decide which approach you find more suited to your business!

WHAT CAN WE DO BETTER MOVING FORWARD?

If you're still reading this, then you're probably already moving in the right direction! Excel models, like any models, are prone to error, and it's important to have the appropriate policies around scoping, testing and review, and documentation to reduce or mitigate the risk.

There have been many horror stories around Excel errors, perhaps the highest profile in the last few years going to Harvard researchers who inadvertently excluded several countries from a research paper that had been used as the basis for fiscal austerity around the world.[2] It's unfair to suggest that they are the first who have had formulas that did not encompass the entire data range, and it's highly likely that they won't be the last. However, with a bit of oversight and better appreciation for the risks, as reflected through improved corporate policies, we can work to reduce the likelihood of Excel errors slipping through the cracks in the future. ■

Tim Heng , BAppFin, MCom-ActStd, is a director at Australian consulting firm SumProduct. He regularly speaks at conferences and training events on Excel-based modeling and best practices. He can be reached at *tim.heng@sumproduct.com*.

ENDNOTES

1 Tim Heng, "Excel as an IT System," *CompAct*, October 2012, *https://www.soa.org/News-and-Publications/Newsletters/Compact/2012/october/Excel-As-An-IT-System.aspx*.

2 Peter Coy, "FAQ: Reinhart, Rogoff, and the Excel Error That Changed History," *Businessweek*, April 18, 2013. *https://www.bloomberg.com/news/articles/2013-04-18/faq-reinhart-rogoff-and-the-excel-error-that-changed-history.*