SPREADSHEET ERRORS and techniques for finding them

by David Whittaker

Getting a grip on a large and complex model is a real challenge, particularly when time is of the essence. When under pressure, the techniques illustrated here can swiftly reduce, but not fully eliminate, modelling risk. However, being focused on what is important and being creative with testing techniques can ensure that the greatest value is obtained from the time spent reviewing the model.

Research recently undertaken in 1998 by Pricewaterhouse Coopers shows a trend of increasing model size and complexity –models appear to be doubling in size and formula content every three years. The trend towards larger model size and complexity is clear, and I can see no reason why this trend should not continue. It seems that given increasingly powerful spreadsheet tools, users wish to model at greater levels of detail and are using ever more advanced formulas and techniques, to manage complex data sets, and manage a greater range of scenarios (see Figure 1).

There is a wealth of research regarding the incidence of spreadsheet errors, for example those cited by Berglas and Hoare in the July/August edition. I work in PricewaterhouseCooper's Spreadsheet Assurance Service, which specialises in the review of spreadsheet-based models, for example to support project finance lending and major acquisitions. Given the trend towards greater model size/complexity, and my own observations of errors based on testing client models, in my view the risk of modelling errors is likely to grow.

The challenge is therefore to find ways of preventing errors affecting the quality of management decisions. In my view the best technique is prevention by using a structured development approach such as that recommended by Berglas and Hoare or as set out in the ICAEW/PricewaterhouseCoopers publication, *Spreadsheet Modelling Best Practice*. When supported by good independent quality assurance, these techniques should result in quality, reliable models.

However, I recognise that due to the often tight timescales inherent in business/financial decisions such best practice design standards may be difficult to follow. In real life, models underpinning headline-making deals are often put together by several people, working long hours under time pressure and, not surprisingly, with little regard to modelling best practice. So what can be done when the board has a matter of days to close a major transaction and demands some assurance that the model is reliable?

The purpose of this article is to offer a perspective based on my extensive experience of practical situations involving the need to eliminate potential errors when testing time is short, through being risk-focused and using alternative testing techniques.

Risk analysis

When given a complex model it is only natural to want to dive into checking the detail. However, this can become overwhelming, and may achieve very little in terms of gaining significant assurance.

The first steps should be to:

- Understand the nature and dynamics of the problem being modelled by undertaking a discussion with the modeller and the key users who will be placing reliance upon the results.
- Understand the structure and flow of the model through a discussion with the modeller. It is useful if the modeller can provide documentation which includes data flow diagrams and descriptions of key areas of the model's logic. This process could be further supplemented with a diagrammatic representation of the flows between the model's worksheets produced from a spreadsheet based interrogation tool or by hand.
- Agree with the modeller and users what the key outputs are and whether any areas are low risk or could even be ignored.

Consequently, it is important to reach a consensus on potential risks with the model builder/decision-maker before commencing the review. It may be that certain components of the model carry a higher risk in terms of making or breaking the deal or indeed the complexity/risk of calculation. Other areas may be low risk, or could even be ignored. It is from such a discussion that a risk-based testing plan can be structured. The following paragraphs outline the available tools and techniques that potentially could be used when there is no time for a full code review.

Tools and techniques for reducing model risk

Model design review

Once the model and the problem are understood it is necessary to make a rapid assessment of whether the model appears to be fit for the purpose intended and is built to an adequate standard.

A model design review is useful for a quick 'fit for purpose' test and should be done before addressing any other areas because if the model is poorly designed it may need significant rework– in other words spot the dogs quickly! My approach involves the following tasks, which are intended to provide a basis for comparison to good practice build standards. A Spreadsheet Audit tool such as Spreadsheet Professional, Operis Analysis Kit or Spreadsheet Detective will help to identify:

- assumptions embedded within formulas. It is important to distinguish between constants and embedded assumptions. Constants are required in order to perform the calculations from the input assumptions, e.g. dividing annual cash flows by 52 in order to calculate the weekly result. A risk exists with embedded assumptions because they will not be updated as scenarios change;
- the degree of hard coded cells, i.e. those which represent mere numerical inputs—obviously these will also not change when the assumptions are varied;
- the degree of separation of inputs, calculations and outputs;
- any inconsistency in formula copying.

Top level/analytical review

This technique involves reviewing the model's 'big picture'. It is good for detecting potentially large errors for one model run, typically the base case, but can be applied to sensitivity cases. This is similar to a technique used in financial audits for high level reviews of financial statements. My approach involves:

- the computation of key ratios over the forecast period. Emphasis should typically be placed, for example, on revenue structures, cost structures, profitability, working capital

and sources of long-term funding. Where possible the results should be correlated back to the model's input assumptions, e.g. trade debtor days are capable of being recalculated from the statement forecasts and matched back to the assumptions;
- key areas should be graphed, because this facilitates interpretation and shows patterns and 'blips' not visible from the numbers alone and these could indicate errors;
- obviously the trends and ratios should be reviewed to ensure overall reasonableness in the model's projections and investigations should be made into any irregularities arising from this review.

The degree of integration and reconciliation of financial statement forecasts

This issue is important because the failure properly to integrate P&L, balance sheet and cashflow is a common error. Financial statement forecasts should follow double-entry principles and reconciliation in terms of the cash balance in the balance sheet and the cashflow movement over the forecast period from the cashflow. My approach involves:
- a walk through review of the financial statement forecast's code, checking where the balance sheet cash figure comes from—if it is not from the cash flow, be on guard for a fudged balance sheet!
- varying the model's key input assumptions and checking that the balance sheet still balances and the cash balance reconciles to the cashflow.

Parallel modelling

Parallel modelling is a re-performance technique which can be used either for the model as a whole or for areas that are perceived as the key risk areas either due to the materiality of output or it's complexity. This technique usually aims to achieve a high level of comfort. The technique is good for reviewing complex, black box type calculations and functions, as well as for the testing of links between workbooks and consolidations. My approach involves:
- the re-performance of the area under review;
- the comparison of the key results derived from the re-performance to the original model, given that the input assumptions are the same;
- the differences arising from the comparison should be rationalised and investigated as appropriate.

Flex and sensitivity testing

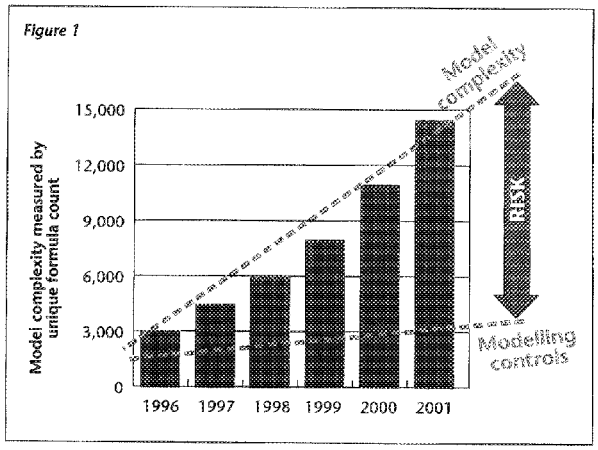
This technique is used for reviewing the reasonableness of the model's sensitivity runs. It is important at this stage to differentiate flex testing from sensitivity testing. Sensitivity testing is where a stated sensitivity is reviewed, e.g. a six-month delay in a construction project's completion. However, there will be occasions where there is a need to test whether the model can handle a *range* of input values, typically the decision's key value drivers and risks which are likely to be varied by the model's user. Therefore in the case of flex testing it is not known exactly what the values of the input assumptions are at the time of testing. My approach is as follows:
- Regardless whether a range of flex tests or a defined sensitivity is to be reviewed there is a need for a transparent audit trail from the inputs' to the model's key outputs, thus removing the 'black box' risk. This is achieved by freezing the model's results in a reference sheet and extracting the variance and % variance between this and the model's current results. We have automated such a structured approach through developing a macro interrogation routine.
- The assumptions would be varied for each flex test/sensitivity case.
- The effect on the calculations and results of each test should be reviewed for reasonableness given the scenario. Here we are looking for reasonable changes where we expect to see them and no changes where we do not.
- The variances/% variances which do not appear to be logical given the scenario should be investigated as appropriate.

Macro review

This technique is useful when the models key calculations are reliant on macro driven code. As pointed out in the introduction, modellers are increasingly using more complex macros and to a large degree this was due to the introduction of Excel's increased programmability through Visual Basic For Applications (VBA). We need to differentiate between low- and high-risk macros. Low risk typically describes a macro or piece of Excel VBA code that is non-complex, relatively small with no program control structure probably recorded with the objective of undertaking repetitive key strokes. At the other end of the spectrum lies the high-risk case which typically describes a piece of Excel VBA code that is complex, relatively large and includes program control structures (i.e. If Then, Do Until, For Next etc).

The correct operation of low-risk macros can usually be established by simple means, so for the purpose of this article we will concentrate on the approach for a high risk case. This can be outlined as follows:
- Understand the purpose of the macro. The ideal situation would be to flowchart the intended logic (or to obtain this from a specification if one exists).
- Perform a walk through of the code, auditing against the 'specification' obtained above.
- The code should be annotated every two to three lines by placing an apostrophe at the end of the relevant line to record your interpretation of the code as appropriate.
- Where the actual logic differs from the 'specification' clearly this needs investigating.
- Once the intentions and actual operation are understood, test runs should be designed, the macro run and the results reviewed by reference to the test data. This is important because the review of the macro's code in isolation may not be completely reliable, and so corroboration with test data provides additional comfort.

Figure 1



David Whittaker BA(HONS), ACMA is the Technical Manager in PricewaterhouseCooper's Spreadsheet Assurance Services. He can be contacted on 0171-212-5957 or by e-mail on david.whittaker@uk.pwcglobal.com

Conclusion

Getting a grip on a large and complex model is a real challenge, particularly when time is of the essence. When under pressure, the techniques illustrated above can swiftly reduce, but not fully eliminate, modelling risk. However, being focused on what is important and being creative with testing techniques can ensure that the greatest value is obtained from the time spent reviewing the model.

Bibliography

ANTHONY BERGLAS and PETER HOARE, *Management Accounting*, July/August 1999.

NICK READ and JONATHAN BATSON, *Spreadsheet Modelling Best Practice*, Pricewaterhouse Coopers, April 1999.