Believing your results

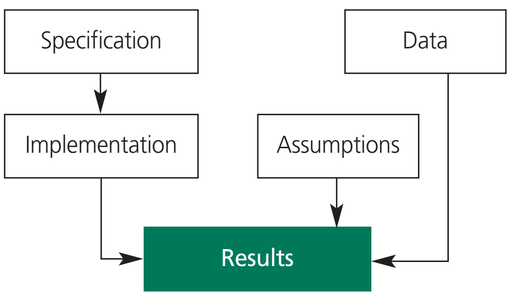
Louise Pryor gives us some tips on keeping actuarial models under control.

COMPLEX ACTUARIAL MODELS are increasingly common in life and general insurance companies. However, it is not enough simply to have these models and to use their results: you must also have confidence in the results, and be able to justify your confidence. The FSA's emphasis on systems and controls and the effects of Sarbanes-Oxley are making themselves felt.

Actuarial models, as well as being used to model risks, are themselves subject to risk. The following problems can arise.

- The mathematical abstractions and business descriptions used may be inappropriate.
- The actual piece of software used may not implement the theoretical model specified.
- The data and assumptions fed into the model may be inappropriate.
- The results used must actually come from the software, data, and assumptions that have been adjudged to be correct.

Figure 1 Example of an actuarial model



To have confidence in your model, you must therefore be confident in its specification, implementation, data, and assumptions, and the processes used to run it. Moreover, you must be able to demonstrate to others why they should have confidence in the model too.

What is the 'right thing'?

Clearly, to be confident the abstract mathematical model is appropriate, you need to know what it is. In other words, you need an explicit specification of the model. The specification must be at a detailed enough level that it can be used to direct the implementation.

Specifications often go through many drafts as they are refined and elaborated. You must always be clear which is the current version and who, if anyone, has signed it off.

Doing the right thing

The concrete model, or piece of software, should be an instance of the theoretical model or specification. The model must be thoroughly reviewed and tested to make sure this is the case.

Reviewing consists of inspecting the code for conformity to the specification. Testing consists of running the model on known inputs and checking that the outputs are as predicted by the specification. Reviewing and testing are both necessary, as they tend to detect different kinds of errors. Both require a good development process with effective version control, so you can tell what it is you have tested or reviewed.

Testing should take place at different levels of detail, looking at individual calculations and also the overall model. Although regression testing, which is testing against a previous version of the same model or against a previous model, is useful, it is not in itself sufficient as it doesn't check the model directly against the specification.

Any reviews and tests performed must be fully recorded, so that other people can tell what has been done, what the results were, and whether there are any gaps.

Garbage in, garbage out

As we all know, the results depend on the data. Again, as the data and assumptions are prepared you need to be clear on the versions to use and who has signed them off. Getting the data right is often a matter of using the correct files; assumptions may be specified on paper, in which case you need to check that the actual model inputs are the same as the paper version.

At the end of the day...

The end result of the modelling process is a set of numbers. You need to be sure the numbers you use derive directly from the actual implementation, data and assumptions in which you have confidence. For this to happen there should be a good process for running the model, with supporting documentation and records.

Systems and controls

Overall, good development practices make it much easier to have confidence in your model. For example, the alert reader will have spotted by now that an essential prerequisite is effective version control and tracking. This is needed for nearly everything: specifications and other documentation, implementations, data, and assumption sets. It is absolutely vital to keep track of which version was used for what, and especially of which version was accepted as being fit for purpose, and by whom.

In practice, another vital ingredient is automation. Manual procedures, for example copying and pasting information from one place to another, are prone to error, and usually slower. Time spent on automating processes is usually a good investment from a pure efficiency point of view, and has other advantages as well. Automated testing, for example, may actually happen! ❑

Louise Pryor is a freelance actuary specialising in risk management, especially operational risk and software risk; she can be contacted through www.louisepryor.com