

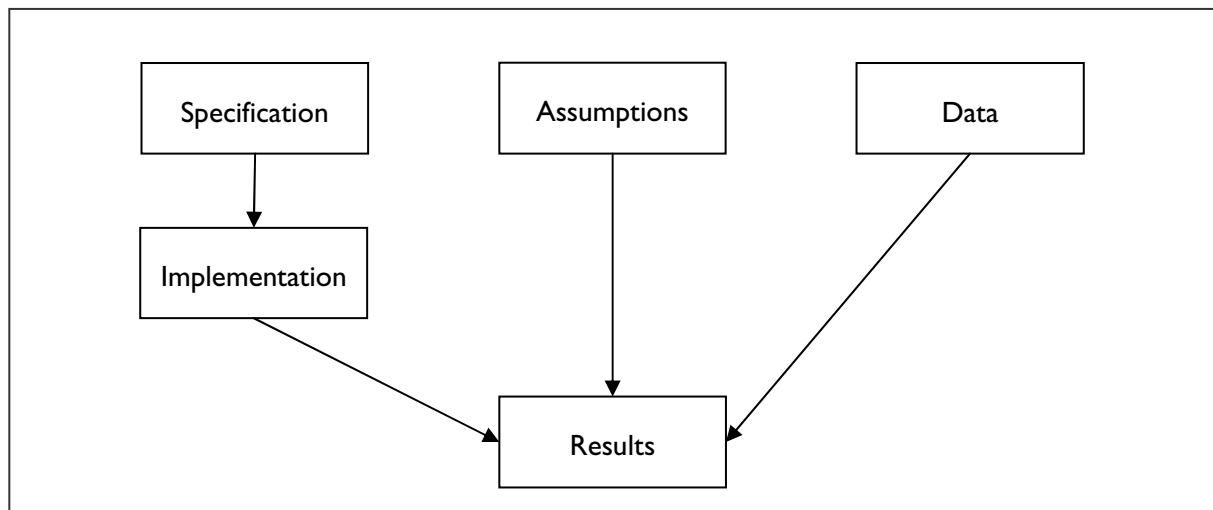
Believing your results

Complex actuarial models are increasingly prevalent in both life and general insurance companies. However, it is not enough simply to have these models and 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 that you are using may be inappropriate.
- The actual piece of software that is used may not implement the theoretical model that has been specified.
- The data and assumptions that are fed into the model may be inappropriate.
- The results that are being used must actually come from the software, data and assumptions that have been adjudged to be correct.

In order to have confidence in your model, you must therefore be confident in its specification, implementation, data and assumptions, and the processes that are 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, in order to be confident that 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 off on it.

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 that 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 that you can tell what it is that you have tested or reviewed.

Testing should take place at different levels of detail, looking at both individual calculations and 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 that are 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 which are the versions to use and who has signed off on them. 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 that the ones that you use do actually come 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 that you are able 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