Techniques for Taming Model Risk

By Bill Cember, Scott Houghton and Dylan Strother

By the end of 2022, many major accounting regimes commonly used by actuaries will have significantly changed. Most of these changes are material and will increase the complexity of models needed to calculate actuarial-related balances. New regimes like Principle-Based Reserving (PBR), GAAP LDTI, and IFRS 17 require detailed cash flow projection models, often with multiple assumption sets that need to be updated frequently, increasing model complexity and risk.

As actuaries begin to use more-complex models, it's not enough just to have the right risk governance in place. The right infrastructure will also help reduce risk by making models easier to maintain and manage. The following techniques will help build a reliable infrastructure and increase model management:

- Modular design
- Model consolidation
- Model documentation
- Modeling roles & model change management

BUT AREN'T OUR MODELS ALREADY REALLY COMPLICATED?

For products like variable annuities, this type of modeling and the associated model risk has been part of reserve calculations for years, and models supporting these products are the focus of existing model governance framework. For many other products, especially traditional life and health products, reserve calculations have traditionally been classified as low-risk models, as the calculations are generally formulaic, and assumptions are prescribed or locked in. While more-complex models for these products typically exist within insurance companies, they are generally used for applications such as pricing, forecasting and pass/fail-type tests such as cash flow testing or loss recognition testing. Using more-complex models to calculate reserves directly extends the model risk inherent in these models to the financial statements.

MODULAR DESIGN

Actuarial models are made of many parts. There are inputs, such as liability in-force extracts, asset investment accounting extracts, scenario data, and assumptions. Once the model has its inputs, the next step is coding the methodology for calculations such as reserve calculations, cash flow projections, and capital calculations and projections. A model run processes these inputs and calculation methodologies and produces outputs, which are then taken by an end user or utilized as input by another model (See Table 1).

Table 1
Model Components by Type

Component	Component type
Liability in force	Input
Asset portfolio	Input
Liability assumptions	Input
Asset assumptions	Input
Statutory & tax reserves	Calculation Engine
Embedded value	Calculation Engine
GAAP / IFRS	Calculation Engine
Cash flow projections	Calculation Engine
Capital	Calculation Engine
Reports	Output

Designing models as a set of model components, also known as modular design, offers several advantages:

- **Reusability**: Components can be reused across models and can be developed and tested once rather than multiple times.
- **Change Management**: Management of models is easier when model components are modular. Having distinct and well-defined components streamlines development and testing, allowing model changes to be done once and then leveraged in multiple ways.
- **Model Releases**: It's easier to show progress to users of the model when they are designed out of smaller components, which can be changed and released more quickly. More frequent releases allow the user of the model to more quickly use it and provide feedback and also decrease the probability of projects going overtime and over budget.

As an example, Figure 1 contains modules needed for a cash flow testing model. A model needed for VM20 deterministic/stochastic projections may use the same modules (with different assumptions) but not require a formulaic reserve projection.

In addition to the advantages listed above, modular design can increase the ability to consolidate, document and manage changes within models.

MODEL CONSOLIDATION

Using components across models leads to the idea of model consolidation. While it's tempting to consolidate models as much as possible—after all, who doesn't want to minimize work and maximize sharing?—there can be challenges with sharing components. These are summarized in Table 2 (Pg. 11).

MODEL DOCUMENTATION

Documentation is a very effective tool to manage risk from models as not all team members are involved with the technical aspects of a model. Model documentation helps stakeholders and other business partners understand what the model does, what it doesn't do, and what the input, output and calculations are.

Key items to include in model documentation are shown in Table 3 (Pg. 11).

In addition to items in the Model Documentation chart, it is also helpful to have a "Day 2 list" of potential future improvements to the model. What goes on this list? Everything that someone might want that isn't there now. This list can include:

- Functionality desired at the time the model was built that is not currently present, perhaps due to complexity or software limitations
- New business requirements due to new regulations and new products
- New experience studies to improve model assumptions
- New and improved data elements and data feeds
- Approximations that are in the model now that could be removed

Figure 1
Example Cash Flow Testing Model as Viewed as a Set of Components

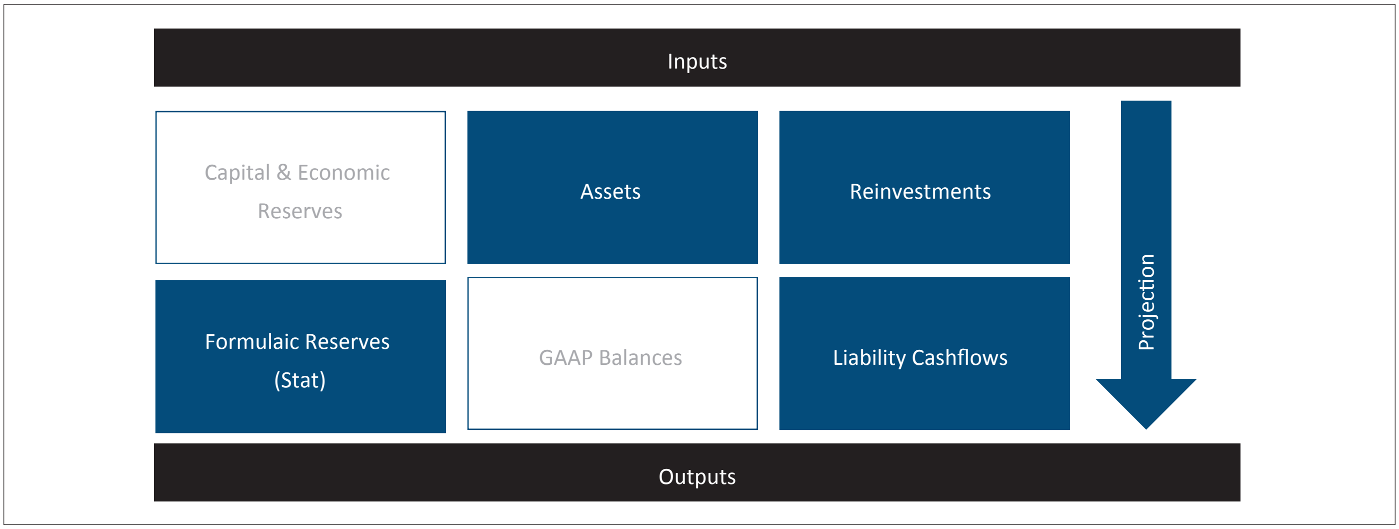

Table 2
Advantages and Disadvantages of Sharing Model Components

Design Consideration	Advantages of Model Consolidation	Disadvantages of Model Consolidation
One source of truth	One model or model component can force consistency	Getting buy-in from multiple stakeholders Consistency not always desirable Technological limitations
Avoid repeating work	One model avoids situation where multiple teams doing redundant work	Getting buy-in from multiple stakeholders
Flexibility	Fewer models	Different use cases have different requirements for flexibility (e.g., pricing vs. valuation)
Controls	Fewer controls	Different use cases have different requirements for controls (e.g., pricing vs. valuation)

Table 3
Model Documentation—Key Items

Key item	Description	Comments
Model business requirements	• What the model is supposed to do • What the model is used for; intended uses • Functional and other requirements • Uses of model output • Other downstream processes supported by the model	• Examples: – Produce reserves on business Day 4 – Produce asset and liability cash flows to support VM20
Key elements of model design	• Choices and trade-offs made when model was being constructed • Rationale for choices • Limitations of model due to design decisions	• Include rationale • Cost savings, time savings
Model input	• Where input comes from and how it's used	• Include any limitations of data or other input
Model output	• What the output is • What the output means • What the limitations of the output are	• Documenting output and limitations of output limits the risks that the model output is misused or misunderstood
Procedures for running the model	• The "nuts and bolts" process for updating the model and running projections	• Include enough detail so that another person with correct qualifications could run the model
Model approximations	• Tracking of current approximations in the model	• Documenting these reduces the risk that approximations impact results inappropriately • Approximations with financial statement impact may need to be tracked separately
Model specifications	• Model technical specifications	• Include in appendix or reference another document
Roles	• Model owner • Model steward • Stakeholders	• Include descriptions of owners of model and users of output

MY DAY 2 LIST IS AN EXCEL FILE …

The Day 2 list is a helpful tool to set priorities at a company level, not the level of a single stakeholder—it ensures the company's priorities for the model are set correctly.

To help ensure there is a single source of truth—i.e., one Day 2 list—enterprise project management software such as Jira or Trello are much better tools than multiple Excel files being emailed back and forth.

MODELING ROLES AND MODEL CHANGE MANAGEMENT

Model development, including design, documentation and consolidation, takes a lot of effort. It is not practical for a single actuary to design, implement and validate a model. Generally, a model development and maintenance process involve roles borrowed from IT application development. Roles can include:

- **Developer**—Responsible for coding the model according to specifications
- **Tester**—Unique from the developer and responsible for testing model development
- **User**—Specifies model requirements and uses model results
- **Steward**—Responsible for governance and change management of the model

Many actuaries may be working on a model at a given time, which can make the model steward function challenging. Many IT application developers use programs such as GitHub and Subversion to manage changes made to source code. These programs allow control and documentation over the model development process to help reduce model risk. For example, a developer or tester can check out a copy of the code, simultaneously make changes or perform testing, and check the model in with documentation. The management section of the program allows the model steward to review sequential changes to the model and assess whether the proper change management steps were followed and then decide whether to accept changes into a master version.

This type of change management functionality is starting to gain traction in the actuarial world but has not yet gained widespread adoption. Actuaries can learn from tools typically used in IT settings and advocate for integration of similar tools to their model development process.

CONCLUSION

In this article we walked through techniques that can help manage model risk and complexity. As regulatory change increases the complexity of our models, utilizing best practices and tools from software development such as modular design and software-assisted change management reduces the risk of making complicated changes to our model. Our models don't have to take us to the moon (at least not yet)—but let's build them as if they should. ■

Bill Cember, FSA, MAAA, is a director and actuary at Prudential. He can be reached at *william.cember@prudential.com*.

Scott Houghton is a vice president at Valani Global. He can be reached at *shoughton@valaniglobal.com*.

Dylan Strother is a senior consultant at the Actuarial Practice of Oliver Wyman. He can be reached at *dylan.strother@oliverwyman.com*.