SOCIETY OF ACTUARIES

Article from:

The Actuary Magazine

August/September 2014 – Volume 11, Issue 4

WHAT IS THE SCOPE OF MODEL VALIDATION?

The first thing that comes to mind when one thinks about validating a model is whether the coding is correct. But what are the other important parts that should be included in the scope of model validation? By Jim McClure

I work for a European company with a U.S. Life and Annuity division, which means we fall under the requirements of Solvency II (SII) and its model validation requirements. Our company has taken this requirement and expanded it to all our models, not just the ones specifically targeted for SII. That means in addition to our economic capital and market-consistent embedded value models, we are validating models for pricing, valuation and hedging, to name a few.

What is the first thing that comes to mind when you think about validating a model? Most people think about whether the coding in the model is correct. Some of the questions one needs to answer include:

- Does the code meet requirements set by any regulatory authority?
- Does the code follow best practices?
- Can the code/results be replicated in Excel or other software packages to ensure confidence?
- Are there limitations in the code? A short example is introducing simplifying code due to long run time.
- Does testing include stress and unit testing?
- Does the portion of model require static or dynamic validation?

These bullet points are just a taste of all the questions that should be asked as part of reviewing the *application code*.

Other than the code, what are other important parts that should be in the scope of model validation? Clearly *assumptions* are an important part of any model. How the products/plan codes are mapped in the model is equally important for achieving efficient and reasonable results. I will refer to that as *plan code mapping* henceforth. Given that you mapped plan codes, are the *product features* coded in the model correctly? Lastly, one of the key items that many actuaries overlook is *controls* on the model.

ASSUMPTIONS

What are we concerned about with assumptions? The assumptions in the model need to be loaded or input correctly. The assumptions should be reviewed for datedness. Has an experience study been completed in the last couple of years? If not, the validation should point this out.

Larger companies may have dedicated staff that complete experience studies to develop assumptions. Assumptions should be reviewed and formally approved by some type of committee; we refer to ours as the Underwriting Committee. Some assumptions will lack credible experience, so expert judgment among the committee members is essential.

PLAN CODE MAPPING

Plan code mapping is a part of modeling where the model developer is attempting to speed up run time, and does this by mapping similar administration plan codes to a single model plan. Here is a sample of some of the questions to expect from a model validation reviewer:

- Are product features preserved by the mapping?
- Is the mapping periodically reviewed? Sometimes modelers map a new plan code to older plans until sales hit a threshold to justify coding an additional model plan. Is there tracking in place to alert the modeler

What is the first thing that comes to mind when you think about validating a model?

when the threshold has been met?
- If modeling assets, is the asset mapping reviewed with the same vigor as plan code mapping?

PRODUCT FEATURES

Review of the product features is crucial to confirm your model is representing some form of reality of how products actually work. When looking at product features it is important to consider policyholder behavior as well. Some of the questions that should be asked are:

- Is there an understanding of what product features are not modeled? Why?
- Are simplifications in the model for features documented?
- Does the model cover features in the products like guarantees?
- For annuities with guarantees, there should be testing of in- and out-of-the-moneyness.

It is important to have organized folders that store all the policy contracts and other material for easy reference. In addition, it can be very helpful to create a matrix of benefits and policy features that lay out the differences between variations of policies. This matrix can also be referenced for plan code mapping when mapping state variations to the standard policy.

CONTROLS

Some models are used for financial results and fall under Sarbanes-Oxley (SOX) or Model Audit Rule (MAR), and likely have the needed controls in place. For other models like pricing and hedging, most of the time controls are overlooked. What kind of controls should be in place?

- Is data accurate going in and coming out of the model?
- Is access to models limited to the appropriate people?
- Are results independently reviewed? (separation of doer vs. reviewer)

- Is there a model change management process in place?

This is just a short list of controls that should be in place. Controls should be periodically reviewed. As your business grows or retracts, some new controls are needed, and some become stale and can be retired.

From my actuarial audit background I cannot stress enough the importance of creating a process map. Taking the process map to a sit-down with internal audit and/or risk team (second line of defense at some companies) can be beneficial in locating control gaps.

In the end the scope of model validation should include:

1. Application code
2. Assumptions
3. Plan code mapping
4. Products features
5. Controls.

Now that I have described the scope of model validation, where should one begin validating the model?

Let's walk through an example model and cover the five sections of model validation. My example is just a generic valuation model that could be for annuities, life or health insurance. Starting with *application code* you have to determine where your model starts and ends. For a valuation model you start with assimilating administrative system data, loading into valuation modeling software, running valuation, and summarizing the results in a spreadsheet where the final report or result is summarized. In a process like this, I would consider it a must to review the code from end to end. A helpful tip: Start by drawing a process map. See page 27 for a process map for a generic valuation model in the suppliers, inputs, process, outputs and customers (SIPOC) format auditors are familiar with. The process map will help for two purposes. One, you can make sure you cover the whole model in your validation. It is very easy to overlook inputs to your model other than the in-force data. Secondly, it is a great piece of documentation for you, for the validation file, and for auditors. For the coding that pulls data from an administration system you may use an IT specialist or actuarial programmer to review the program coding for pulling in-force data from these systems. Next, the most significant part of the application code review would be in the model itself. To do this type of analysis, a validator either reviews all the coding in the model manually or, better yet, replicates the model in a spreadsheet. If using a spreadsheet, you will have to break complicated models into manageable pieces to test. Replicating the model in a spreadsheet is another great piece of documentation to save.

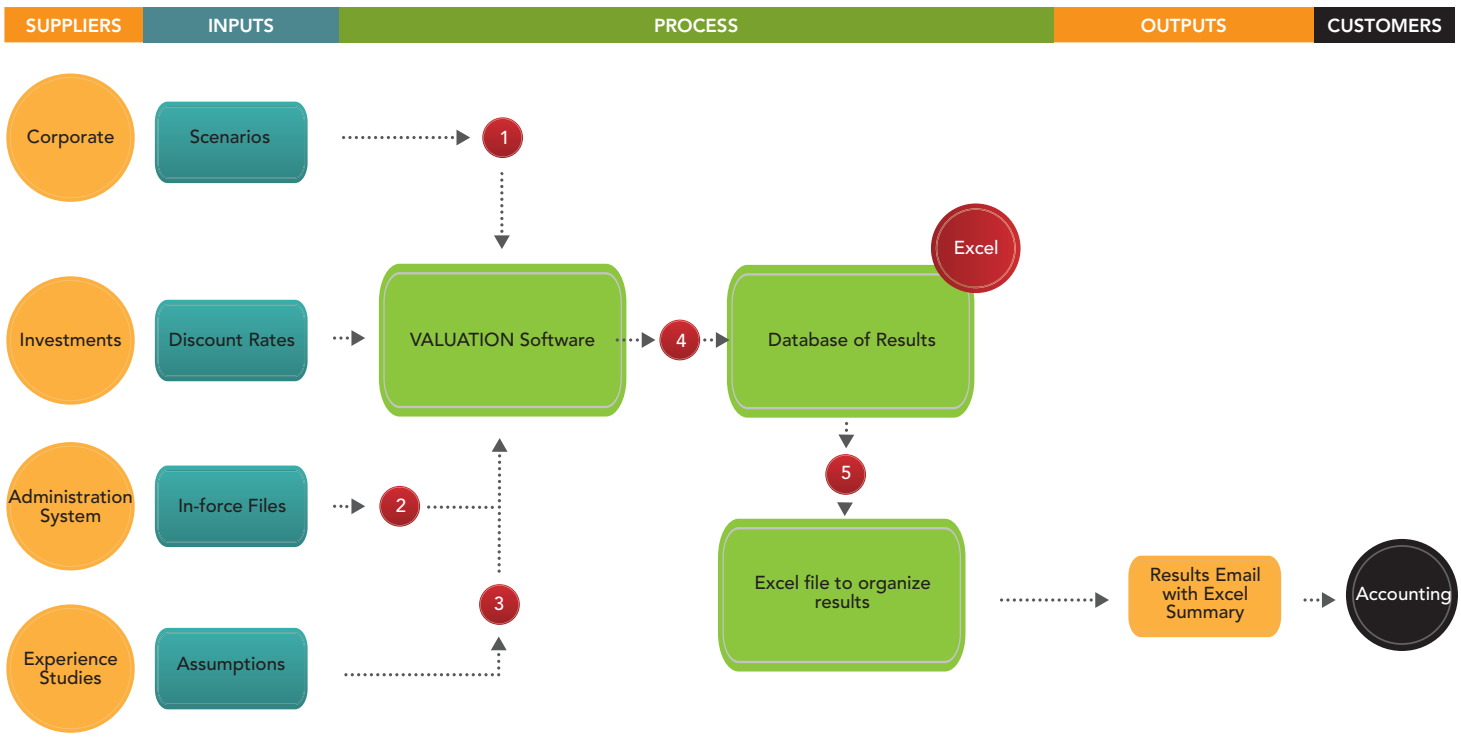
Replicating the model in a speadsheet is another great piece of documentation to save.

What types of testing should be considered? Below is a series of testing that is common for testing actuarial models.

- **Unit testing**—Testing code at a policy or cell level. A model reviewer completes this testing before running larger product groups, legal entities or full blocks. One example is testing a formula that uses in-the-moneyness to determine utilization of a benefit for valuation. When a policy is in the money does it actually exercise the benefit successfully and accurately?

- **Regression testing**—Usually completed after unit testing, consists of running larger product groups, legal entities, or full blocks of business. Test to make sure the results are in expected reasonable ranges. A common mistake is to not have set prior expectations for reasonableness before running and reviewing the results.
- **Static validation**—A comparison of the in-force produced by the model at the time of valuation against the actual in-force for the modeled plans at that same date. Some of the items to check include number of policies, cash values, account values, premiums, face amounts, etc. A preferred source for control reports is from the administration system. An alternative source for actuals could be obtained from your accounting area from what they use as control reports.
- **Sensitivity testing**—In the valuation model example, if you run a sensitivity on interest rates rising/falling 100 basis points in a single year, do the results still look reasonable? This is a typical example of sensitivity testing. Other sensitivity examples are having mortality swings 5 percent up/down, lapse rates up/down 10 percent, and the market movements are up/down 20 percent. It is recommended to combine sensitivities as well. If the market

Valuation Model Process Map



CONTROLS:

1. Review scenarios, complete battery of tests on new scenarios.
2. Check policy count, account value/face amount, against administration. System control reports.
3. Confirm experience study report results on new/updated assumptions are in model correctly.
4. Check results by legal entity and compare to prior quarter for reasonableness. (consider direction and magnitude of change)
5. Check policy count, account value, face amount, whatever makes you comfortable that all policies/blocks of business are covered against system control reports.

drops 20 percent, also try the lapses moving in an inverse relationship and increase 20 percent. Combinations of sensitivities should make sense with what happens in reality.

- **Stress testing**—Some actuaries call this trying to break the model using extreme assumptions or scenarios. An example is running the valuation model with 0 percent lapses and 100 percent lapses. Do the results seem reasonable, or do you get unfathomable returns? The logic behind this type of testing is if the model still produces reasonable results for extreme cases, the model will likely produce reasonable results for all the variations between the extreme stress points.
- **Back testing**—I have seen many definitions of back testing, but I prefer the one where you load in actual current experience for assumptions, re-run a prior period model and see if the projections match current period actuals. Some refer to this as dynamic validation interchangeably.

Most models have some sort of coding simplifications. An example for modeling annuities: In current modeling software it may be prohibitive to implement stochastic-on-stochastic modeling for hedging assets/liabilities. To test this you may have to let the model run the real stochastic-on-stochastic scenarios for a week or more and compare results to your hedging simplifications. Validating a simplification comes down to the materiality of the difference versus the time constraints to report results. Let's say you are doing valuation work and it takes a week to run a model. You find an error and it takes another week to run your model; you are obviously not going to meet reporting deadlines. It is not wrong to have simplifications in your model, as long as you understand the materiality of the

difference from coding perfectly. It is not acceptable to have simplifications and not have any inkling if they are reasonable. You will find actuarial judgment is a key part of this analysis.

How should an actuary go about reviewing *assumptions*? First, the actuary should check the assumptions for reasonableness. The actuary could compare the company assumption versus Society of Actuaries studies. If the company takes part in surveys, the survey results are generally shared. Some of these surveys develop industry assumptions by compiling studies across companies. This is a good basis to use for reasonableness checks of assumptions. Second, actuaries can review the experience studies themselves. If you have a dedicated staff working on experience studies, scheduling a walk-through of the process to perform the study can build confidence that the resulting assumptions are adequate to use. Sometimes, though, you may find that the studies may not have enough credibility, and you may find that you have to use more actuarial judgment on the reasonableness on the blending of company experience with industry experience. I cannot stress enough the importance of validating the credibility behind your assumptions. Be sure to ask questions of the experience study staff when assumptions change drastically from study to study. There can be other influences at work that drive assumption changes. For example, the swing in the economy can affect lapse rates in a material way. You may have to suggest revamping an assumption structure to make assumptions dynamic. This is especially important when you run thousands of scenarios through a model and the effects of the interplay between assumptions and scenarios can produce unexpected, extreme results. For example, dynamic lapse assumptions can be formulaic to adjust to inputs from scenarios, like market movements in the S&P or bond markets. In our valuation example, a reviewer should check the assumption tables that are input in the process map to the model against the assumptions produced in the experience study reports that were approved for use.

The best way I have found to test *plan code* mapping is to view all the plans in a spreadsheet and create a mapping grid. You can quickly tell how all your administration plan codes align with the modeled plan codes. Then load in all the specifications for the products and make sure that the mapping makes sense. In another annuity example, you want to check that return-of-premium (ROP) riders are mapped to other similar plans with the same or similar ROP design. Mapping a product with an ROP to a product with a nursing home rider or any other unrelated benefit rider would obviously be wrong and you would visibly see that in the grid created. Be sure to include the amount of account value for annuities, or premium/face amount for life products with each administrative plan code. This will be used not only for checking the size of the plan mapping, but also as a control to confirm all account value is mapped within the model. It is common to map new plan codes to other similar plan codes in your model. Be careful—once the new plan grows to a substantial size, it may no longer be a nice simplification, but a material risk difference. It can look pretty silly when

How should an actuary go about reviewing assumptions? First, the actuary should check the assumptions for resonableness.

Checklists Can Be A Plus

Says Atul Gawande in his book, *The Checklist Manifesto*, "The Contribution of a checklist is not the checklist itself but the discipline that the checklist brings to the management of any operation." Read Jay Jaffe's book review in the April/May 2014 issue of *The Actuary* at *SOA.org/actuarymag/*.

you are mapping a plan with more account value than the plan you are mapping into. A good practice is to have a quarterly production report to use to track the growth of new plans. Once the new plans hit a threshold for material size, then it makes sense to code the model plan code in the model.

To review the *product features* it is really helpful to have the product specifications. Working with the actuarial pricing team or representatives from the administrative system teams can save lots of time gathering the product information. If you are lucky enough, someone may have a product grid that lines product families together, which can really expedite your review. Otherwise, you may have to get samples of all the policy contracts. Take advantage of the plan code mapping grid mentioned earlier to include the product features; it will keep the product information organized. Then start testing the product features beginning with the larger plan codes by premium or account value.

A third use of the process map is to review where all the *controls* are, or in some cases are not, in the model process. There should be control points at the beginning of the process when pulling the data from the administration system, at the point they are loaded into the model, and when the output leaves the model. Items like policy counts, account value, face amount and premium should match exactly at each control point. Other controls should be in place for checking the reasonableness of results. In the process map on page 27, the red circles represent controls. At the bottom of the map there is a description of each control. For some products, if the market goes down, the reserve goes down in a similar manner. A quarter-over-quarter trend control should be in place to monitor the model for inconsistent trends in results. When I was a valuation actuary I liked having rule-of-thumb ranges. If the relationship between the market change and the reserve change is in the rule-of-thumb range, then control earns a success; otherwise a fail. Failures then require investigation of the model for potential errors.

After reviewing your model for all these sections (and making any fixes), a model change process should be in place to ensure the integrity of your model going forward. An article on model change processes and best practices sounds like a good idea for a follow-up article. A

Jim McClure, FSA, MAAA, is manager, Model Valuation, at Transamerica Capital Management. He can be reached at *Jim.MCClure@transamerica.com*.