Mo.net
End-to-End Governance with the Mo.net Platform

November 2020
Revision 7

Background

In today's highly regulatory environment, ensuring that the correct models, data and assumptions have been used to generate actionable insight, management information and regulatory submissions is a critical requirement of any insurance undertaking. Traditionally the task of answering the fundamental question "Where did these results come from?" can create a significant volume of additional work for roles already subject to significant resource and time constraints.

The Mo.net platform includes a range of features & functionality designed to help ensure that the right artefacts are used as part of any end-to-end modelling activity – from model design, development and testing through to operational use. Mo.net is specifically designed to address the lack of transparency & lineage from results through the often opaque layers of modelling & data transformation and as such offers best in class end-to-end modelling governance & clarity.

The Challenges of End-to-End Modelling Governance

Historically, the focus of financial modelling was simply to develop and operate models with sufficient levels of functionality and performance, to deliver key business & regulatory metrics within available operational windows. This basic requirement has not really changed, although the number of metrics required has increased, and the time available to produce them has reduced with each new regulatory regime. However, in today's heavily regulated environment, there is an increased focus on governance of the overall modelling process. This covers the two fundamental dimensions of:

- Model design, development, testing and change
- Model consumption and operational use

Including the specific segregation of those functions, plus the particularly challenging aspect of demonstrating that the correct models and associated artefacts of data and assumptions are the ones intended for that purpose, and no unexpected / unapproved changes have crept in. Furthermore, ensuring that models, data & assumptions have been prepared, approved, changed and used by the appropriate personnel is now an added complication.

These governance challenges are summarised in the figure below:

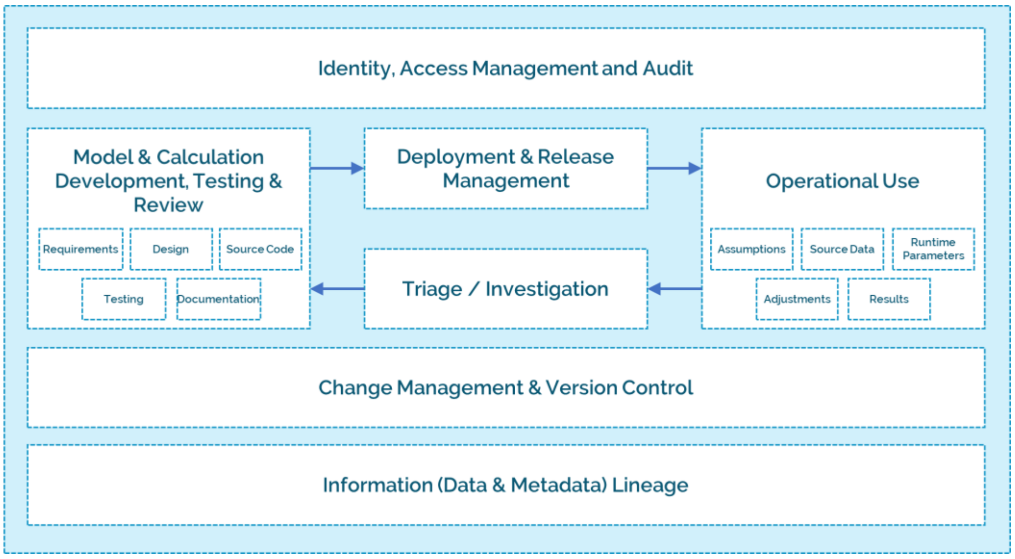
Identity, Access Management and Audit
Model & Calculation Development, Testing & Review
Requirements
Design
Source Code
Testing
Documentation
Deployment & Release Management
Triage / Investigation
Operational Use
Assumptions
Source Data
Runtime Parameters
Adjustments
Results
Change Management & Version Control
Information (Data & Metadata) Lineage

Figure 1 - End-to-End Modelling Topology

Development Governance

The fundamental objective of development governance is to ensure models are developed & tested in line with agreed designs, specifications and other corporate standards, and that any changes to these aspects and the resulting models themselves are appropriately controlled. Some of the key requirements of a robust model development environment are set out in Table 1 below.

- Source code control	- Code (logic) & structural change control
- Version control	- Design documentation & specifications
- Access management	

Table 1 - Development Governance

Release Governance

Once models have been developed, tested and approved for operational use, consideration should be given to the mechanism by which models are made available to any live environment(s). Historically the line between development & operational environments was very blurred, and in some cases non-existent, with models being developed and run within the same environment and sometimes by the same personnel. As a result, model developers could conceivably run what appeared to be production / live models to produce production-like results, and likewise, operational users could potentially change models in the production environment.

Any release management process should clearly evidence where models, data, assumptions or other modelling artefacts originated, ideally with a path back to any golden sources. Similarly, the ability to promote models and other artefacts to the live environment should be restricted to specific roles or personnel, otherwise the segregation of responsibilities is undermined.

Operational Governance

The majority of end-to-end modelling governance requirements lie within the operational domain, since this is where most models are actually consumed to produce business insight & regulatory metrics.

Perhaps the most fundamental requirement of the operational environment is actually the explicit prevention of changes to model logic, since these should be managed through strictly controlled configuration, change and version management in the model development environment. Artefacts such as assumptions and data are likely to be far more volatile than model logic / code, and therefore changes to these must be permitted in the operational world, albeit under strict controls, with appropriate levels of approval and with a comprehensive audit history in place.

Some additional operational governance considerations are detailed in Table 2 below.

- Artefact versions & compatibility
- Audit & change history
- Access history
- Management & access to results
- Operational resilience
- Alerting & notifications
- Downstream reporting & analytics
- Regeneration of historic results, using previous versions of models & artefacts
- Workflow and approvals, and evidence thereof
- Data quality & completeness

Table 2 – Operational Governance Considerations

Obviously specific users & organisations will have their own explicit requirements, based perhaps on their size, the nature of their business or their risk management processes.

Additional Overarching Requirements

In addition to the primary areas of governance detailed above, it worth also thinking about those requirements which straddle the development, release and operational domains. Aspects such as centralised role & identity management, metadata lineage (i.e. did this operational model come from this source code?), or the management of highly volatile artefacts (e.g. has my data changed since I last used it?) place an additional burden on any end-to-end governance framework.

Providing End-to-End Governance with Mo.net

The Mo.net platform has been specifically designed with end-to-end governance in mind, and as such includes a range of features for each community to make the implementation of such frameworks as simple as possible. Furthermore, we acknowledge that not all customers will face an identical set of challenges or have exactly the same requirements, so we have developed functionality that can be configured to meet the specific needs of the organisation and without forcing customers to adopt a purely vendor-defined approach.

Development Governance Features

All Mo.net modelling projects begin life in the Mo.net Model Development Studio, which is a comprehensive integrated development environment ("IDE") specifically designed to support the process of designing, developing and testing financial models. Some of the principal governance & control features of Model Development Studio are detailed below.

- Integration with industry standard source code control platforms, e.g. Git, Azure DevOps, Team Foundation Server
- Atomic level control of Model Development Studio features through roles, permissions, and optional integration with Active Directory
- Visibility of all changes whether using source code control or not
- Explicit versioning of all compiled Mo.net models, regardless of their form
- Ability to generate comprehensive documentation of any model or the differences between any pair of models or versions of models
- Unique project, task and binary level identifiers to explicitly highlight the version of the model

Table 3 – Governance Features of Mo.net Model Development Studio

Controlling the Model Release Process

One of the cornerstones of the Mo.net platform is the deliberate segregation of product components between development and operational domains. While production users can use the Model Development Studio to perform operational tasks, we would only suggest this if customers have locked down the model development environment using the Mo.net Identity Service. The Mo.net Identity Service ensures that development and operational users have different permissions and that all activities are closely monitored / audited to ensure no contamination of production models occurs, either deliberately or accidentally.

Our recommendation is to limit the use of the Model Development Studio to development & testing activities, and instead use one of the operational components of the Mo.net platform to support production / live modelling activity. Components such as the Operational Modelling Centre or the Mo.net Quotations Service have no facility to change model code / logic, and only allow models (in the form of service packages) published from the Model Development Studio to be uploaded into the operational environment, and to be combined with live data & assumptions.

The process of uploading or pushing models & assumptions to the live environment is tightly controlled & audited, with clear lineage of modelling assets back to empirical sources.

Operational Governance Features

The Mo.net platform provides a number of operational modelling & calculation components, each with their own specific features & primary use cases. These range from the Operational Modelling Centre, which is focused on managing & scheduling regular pricing, valuation & reserving activities, to the Mo.net Quotations Service, which is focused on providing real time calculations for front or back office systems. Each operational component of the Mo.net platform inherits & extends the governance features baked into the original models & calculations developed in Mo.net Model Development Studio.

For example, the version of source code used to provide the calculations in the live environment is clearly visible to the users, and users can download the source of any operational model back into the development environment should this be required for investigation or triage work. Furthermore, the operational modelling components enable customers to create approval workflows to minimise the likelihood of production runs using incorrect versions of artefacts. Some of the other of operational governance features of the Mo.net platform are listed below.

- Clear visibility of artefact versions, ownership, dates and sources
- Atomic level history of artefacts from creation through changes to use and disposal
- Filtering visibility of artefacts to specific groups of users
- 4 or 6-eyes to check review process where necessary
- Built-in reporting & analytics avoiding the need to take results off the platform and potentially opening up the possibility of contamination / uncontrolled changes
- Complete history of results and component inputs for regeneration of historic results

Table 4 – Operational Governance Features of the Mo.net Platform

Conclusion

This short paper is only designed to provide a brief summary of how the Mo.net platform delivers best in class end-to-end modelling governance requirements, and answers with confidence the fundamental question of "Where did a specific set of internal or regulatory results originate?" With its inherent flexibility and market-leading integration potential, these are only some of the ways that a modern insurance undertaking can meet its governance obligations, while at the same time delivering other organisational benefits.

Contact Us

To find out more about the Mo.net Financial Modelling Platform and its range of end-to-end governance features, please get in touch:

Software Alliance Limited
30 Stamford Street, London, SE1 9LQ
Tel: +44 (0) 20 3964 2755
www.softwarealliance.net

Author: Guy Shepherd