Mo.net Financial Modelling Platform
SOX Controls with the Mo.net Platform

February 2026
Revision 4

Purpose

This datasheet explores how the various features of the Mo.net platform can be used to deliver the controls required as part the Sarbanes–Oxley Act (SOX).

Background

For life insurers operating under the Sarbanes-Oxley Act (SOX), internal controls over financial reporting extend far beyond finance systems and the general ledger. Increasingly, the actuarial modelling environment sits directly within the financial reporting control framework.

Modern actuarial models produce some of the most important numbers in an insurance company's financial statements. They calculate reserves, expected future cash flows, profit emergence, and under IFRS 17 the fulfilment cash flows and Contractual Service Margin (CSM). When these outputs feed into financial disclosures, they effectively become part of the financial reporting process.

That means actuarial modelling platforms, whether built in specialist systems such as Mo.net, or developed internally in spreadsheets, Python, or other languages, must operate within a structured control environment. For most insurers today, actuarial models are firmly within SOX scope.

The challenge is ensuring that these models are governed, controlled, and auditable in a way that supports reliable financial reporting.

Why Actuarial Models Fall Within SOX Scope

SOX Section 404 requires organisations to demonstrate that effective internal controls exist around financial reporting. Any system that materially contributes to reported financial results may fall within this scope.

Actuarial models clearly meet that threshold. They often determine the insurance liabilities reported on the balance sheet and the profit recognised in the income

statement. Changes to model logic, assumptions, or data can therefore have a direct and material impact on financial outcomes.

From an auditor's perspective, the risk is straightforward. If actuarial models can be modified without appropriate oversight, or if their inputs and assumptions are poorly controlled, there is a risk that financial statements could be misstated. As a result, insurers are increasingly expected to implement formal control frameworks around their actuarial modelling environments.

Model Governance and Accountability

A foundational element of SOX compliance in actuarial modelling is clear governance and accountability.

Organisations should maintain a formal inventory of the models that contribute to financial reporting. This inventory should identify the purpose of each model, the financial outputs it generates, and the individuals responsible for maintaining it. Assigning clear ownership ensures that every model has someone accountable for its ongoing integrity and maintenance.

Documentation also plays a critical role. Models should be supported by clear descriptions of their methodology, key assumptions, and intended use. This allows both internal reviewers and external auditors to understand how results are generated and whether the methodology remains appropriate.

Independent validation is typically expected as part of this governance framework. Validation teams or independent actuaries review model design, calculations, and assumptions to ensure that the model performs as intended. These reviews are often conducted periodically or when major model changes occur.

The overall goal is transparency. The organisation should be able to clearly explain what models exist, what they do, and how they are governed.

How Mo.net Supports This Requirement

- Centralised Model Inventory [Mo.net Model Development Studio Source Control Connectors]
- Formal Model Lifecycle Management [CICD via Model Development Studio Source Control Connectors]
- Integrated Version Control [Mo.net Model Development Studio Source Control Connectors]
- Full Audit Trail and Activity Logging [Mo.net Identity Service]
- Structured Model Documentation [Mo.net Documentation Service]
- Role-Based Access Control (RBAC) [Mo.net Identity Service with Active Directory / Entra Connector]
- Model Dependency and Impact Analysis [Precedents / Dependents in Mo.net Model Development Studio]

Change Control and Version Management

Actuarial models evolve continuously. Assumptions change, regulatory requirements shift, and new products are introduced. Without a structured change management process, however, these updates can introduce significant risk.

SOX therefore places considerable emphasis on formal change control.

Any modification to model logic, assumptions, or configuration should be documented through a change request process. This documentation should explain the purpose of the change, the specific components affected, and the expected impact on model results.

Before changes are introduced into production models, they should be thoroughly tested in a development or testing environment. Testing confirms that the change behaves as intended and does not introduce unintended side effects. In many organisations, testing also includes reconciliation checks against prior model results or benchmark calculations.

Equally important is independent review. Changes should not be deployed solely by the individual who implemented them. Instead, they should be reviewed and approved by a senior actuary, a model governance function, or another appropriate reviewer.

Version control is another essential element of change management. All model code and configuration should be maintained in a controlled repository that records the history of changes over time. This allows organisations to trace exactly which version of a model produced a particular set of financial results. For auditors, this traceability is essential.

How Mo.net Supports This Requirement

- Structured Change Request Workflow [Source control + CICD with Mo.net Model Development Studio]
- - Environment Separation [Development - Mo.net Model Development Studio (Enterprise Edition), Testing – Mo.net Model Development Studio (Runtime Edition), Production – Operational Modelling Centre]
- Integrated Version Control [Mo.net Model Development Studio]
- Automated Testing and Validation [Source control + CICD with Mo.net Model Development Studio]
- Release Management and Deployment Controls [Source control + CICD with Mo.net Model Development Studio]
- Full Change Audit Trail [Mo.net Identity Service + Source control with Mo.net Model Development Studio]

Access Control and Segregation of Duties

Another important SOX principle is restricting access to systems that influence financial reporting.

Within actuarial modelling environments, access should be limited to authorised users, with permissions granted according to role. Not every user should be able to modify model logic, change assumptions, or run production calculations.

Segregation of duties helps reduce operational risk. For example, developers responsible for building model logic may not be authorised to approve model changes for production use. Similarly, those running production calculations may not have the ability to modify the underlying model code.

By separating these responsibilities, organisations reduce the likelihood that a single individual could introduce an unreviewed change that affects financial reporting.

How Mo.net Supports This Requirement

- Role-Based Access Control (RBAC) [Mo.net Identity Service with Active Directory or Entra Integration]

- Segregation of Development and Production Environments
- Separation of Duties Across Key Roles [Mo.net Identity Service with Active Directory or Entra Integration]
- Integration with Enterprise Identity and Security Systems [Mo.net Identity Service with Active Directory or Entra Integration]
- Controlled Access to Assumptions and Data [Mo.net Model Development Studio / Operational Modelling Centre Data Connectors]
- Audit Logging of User Activity [Mo.net Identity Service / Operational Modelling Centre]

Assumption Management

Assumptions are central to actuarial modelling. Mortality rates, lapse behaviour, expense assumptions, discount rates, and economic scenarios all influence the financial outputs produced by models.

Because of their importance, assumptions require their own governance framework.

Many insurers manage assumptions through formal committees that review proposed changes and approve new assumption sets. The rationale behind each assumption should be documented, including the data sources and analysis used to support it.

Version control is equally important for assumptions. When financial results are produced, it should be possible to identify exactly which set of assumptions was used in the calculations. Maintaining clear records of assumption changes helps ensure transparency and supports the reproducibility of results.

How Mo.net Supports This Requirement

- Centralised Assumption Repository [Data Connectors to Industry-Standard Data Stores in Mo.net Model Development Studio / Operational Modelling Centre]
- Version Control for Assumptions [Operational Modelling Centre / Source Control Connectors with Mo.net Model Development Studio]
- Assumption Approval Workflows [Operational Modelling Centre or CICD Workflows with Mo.net Model Development Studio]
- Assumption Traceability in Model Runs [Operational Modelling Centre]
- Controlled Access to Assumption Updates [Operational Modelling Centre]
- Documentation and Metadata for Assumptions [Operational Modelling Centre Reporting Services]

Data Integrity and Input Controls

Actuarial models rely on large volumes of input data drawn from policy administration systems, economic scenario generators, and other internal systems. Ensuring the integrity of this data is therefore a critical component of SOX control frameworks.

Data validation checks are commonly used to detect missing, inconsistent, or unexpected values before models are run. Reconciliations between source systems and model inputs can confirm that policy counts, premiums, and other key data elements match expected totals.

These controls help ensure that the data feeding the actuarial models is complete and accurate before calculations begin.

How Mo.net Supports This Requirement

- Controlled Data Ingestion and Integration [Data Connectors to Industry-Standard Data Stores in Mo.net Model Development Studio / Operational Modelling Centre]
- Data Validation and Quality Checks [Data Validation Tasks in Mo.net Model Development Studio / Operational Modelling Centre]
- Data Reconciliation and Completeness Controls [Data Comparison in Mo.net Model Development Studio / Operational Modelling Centre]
- Versioning of Input Data [Operational Modelling Centre]
- Data Lineage and Traceability [Operational Modelling Centre]
- Exception Management and Issue Tracking [Mo.net Model Development Studio / Operational Modelling Centre]
- Secure Storage and Access Controls for Data [Data Connectors to Industry-Standard Data Stores in Mo.net Model Development Studio / Operational Modelling Centre]

Production Runs and Output Validation

Even when models are well governed and data inputs are controlled, organisations still need to verify that the results generated by production runs are reasonable.

Production processes often include formal run logs capturing the model version used, the assumptions applied, and the data inputs included in the calculation. These records provide a clear audit trail linking reported financial results back to the specific model run that produced them.

Actuaries typically perform analytical reviews of outputs as part of the run process. Variance analysis comparing results against prior periods can highlight unexpected movements that may indicate data issues or model errors.

These reviews act as a final safeguard before results enter the financial reporting process.

How Mo.net Supports This Requirement

- Controlled Execution of Production Runs [Operational Modelling Centre]
- Run Configuration Management [Operational Modelling Centre]
- Automated Run Logging and Audit Trails [Operational Modelling Centre]
- Output Reconciliation and Validation Checks [Mo.net Model Development Studio / Operational Modelling Centre]
- Reproducibility of Production Runs [Operational Modelling Centre]
- Run Approval and Sign-Off Workflows [Operational Modelling Centre]

The Importance of Reproducibility

A key expectation under SOX is that financial results must be reproducible.

Auditors should be able to trace reported numbers back to the exact model version, assumptions, and data that produced them. Without structured

governance, version control, and documentation, reproducing historical results becomes extremely difficult.

This is one reason many insurers are modernising their actuarial modelling environments by incorporating software engineering practices such as version-controlled code repositories, automated testing, and structured deployment pipelines.

These practices strengthen control frameworks while also improving development efficiency.

How Mo.net Supports This Requirement

- Complete Run Snapshots [Operational Modelling Centre]
- Integrated Versioning Across Models, Data, and Assumptions [Operational Modelling Centre]
- Reproducible Run Re-execution [Operational Modelling Centre]
- Traceability from Financial Results to Model Runs [Operational Modelling Centre]
- Long-Term Archiving of Model Artefacts [Operational Modelling Centre]

Conclusion

Actuarial models now sit at the core of insurance financial reporting. As their importance has grown, so too has the need for strong governance and control.

A well-designed SOX control framework ensures that actuarial models are not only technically robust but also transparent, traceable, and auditable. By implementing structured model governance, disciplined change management, strong access controls, and rigorous data validation processes, insurers can significantly reduce the risk of financial misstatement.

Ultimately, the goal is not to slow down actuarial innovation but to ensure that the models underpinning financial reporting operate within a reliable and well-governed environment.

Contact Us

For more information regarding the Mo.net platform and achieving compliance with SOX controls, please get in touch:

Software Alliance Limited
54 Charlotte Street, London, W1T 2NS
Tel: +44 (0) 20 3964 2755
www.softwarealliance.net

Author: Guy Shepherd