WHITE PAPER

Effective Model Risk Management with MATLAB

MathWorks®

Introduction: Rising Awareness of Model Risk

Financial institutions are increasingly reliant on statistical risk models. The number of models being developed—whether to manage risk, price different instruments, or calculate regulatory capital—has increased rapidly as banks and other institutions automate decision-making processes. By some estimates, the number of models being developed is increasing at a rate of 10–25% annually in larger firms [1].

Two major factors are driving this trend. The rise of machine learning and artificial intelligence is enabling banks to build more models and automate a widening array of complex decisions. At the same time, financial institutions are deploying models in large numbers to respond to tighter capital requirements and local regulatory stress-testing exercises.

But automation and increased complexity create the danger of heightened model risk. With the fallout from the 2008 financial crisis still firmly top of mind, banks and financial regulators alike are all too aware of the potential consequences should a model be misused or prove defective, ranging from heavy losses to bankruptcy or even liquidation.

Financial regulators around the world have been intensifying their oversight of model risk management in recent years. The aim is to ensure that banks and other institutions are paying greater attention to understanding and mitigating the risks associated with their models by establishing a robust model risk management (MRM) framework.

As the regulatory burden grows, financial institutions face higher costs associated with developing, validating, and reviewing models to ensure that they continue to be fit for purpose. Global banks face risks across a broad spectrum of activities, including their client relationships, the products they offer, and the transactions they undertake, as well as the markets where they operate. To operate effectively, they need to identify risks and then measure, monitor, and manage them so they remain within the organization's risk appetite and comply with regulatory requirements.

Given the scale and complexity of model risk management, the process can be lengthy and labor intensive. This paper details how institutions are using MATLAB® to improve productivity and reduce costs by accelerating model development and validation and provide full traceability of every step of the model life cycle.

Definition of Model Risk

In 2011, the Federal Reserve published guidance on model risk management (SR Letter 11-7 [2]). In providing a framework for effective model governance, it included an early definition of models and model risk, stating that models "are simplified representations of real-world relationships among observed characteristics, values, and events." The guidance further states:

> *"The use of models invariably presents model risk, which is the potential for adverse consequences from decisions based on incorrect or misused model outputs and reports."*
>
> — SR Letter 11-7

Crucially, the guidance outlined that model risk occurs because a model may have fundamental errors that cause inaccurate results, or because it may be used improperly or inappropriately. Because of this, the regulator stipulated that model risk should be managed like other types of risk and that financial institutions should conduct a critical analysis to identify model limitations and assumptions and produce appropriate changes. The guidance also stated that risk cannot be eliminated, so companies should establish limits on model use, monitor model performance, adjust or revise models over time, and supplement model results with other analysis and information.

The Life Cycle of a Model

To mitigate model risk, financial institutions are developing MRM processes that ensure that models are developed and implemented in a robust and appropriate manner. A typical model life cycle comprises six stages:

1. Definition of the model, including the risk classes it measures and its scope
2. Model development, where the source data is manipulated and analyzed, and quant specialists and analysts carry out backtesting to describe how the model implements the definition
3. Model review, where an independent model validation team verifies that the model is fit for purpose and does what it intends to do
4. Pre-implementation validation of the model performance against validation and production data
5. Deployment of the model for implementation, for scoring credit applicants, regulatory reporting, or other purposes
6. Ongoing monitoring of the model against its purpose, definition, and limitations

A model spends most of its life cycle in the development, review, and validation stages (Figure 1). By the time the model has been approved for use, many months may have passed and the assumptions underpinning the model may be out of date.

MRM is also an ongoing process, which means that implemented models need to be constantly reviewed and monitored to ensure that they continue to work as designed and meet business needs. Given the complexity of the risk model life cycle, this leads to stretched resources, high costs, and slow implementation times.

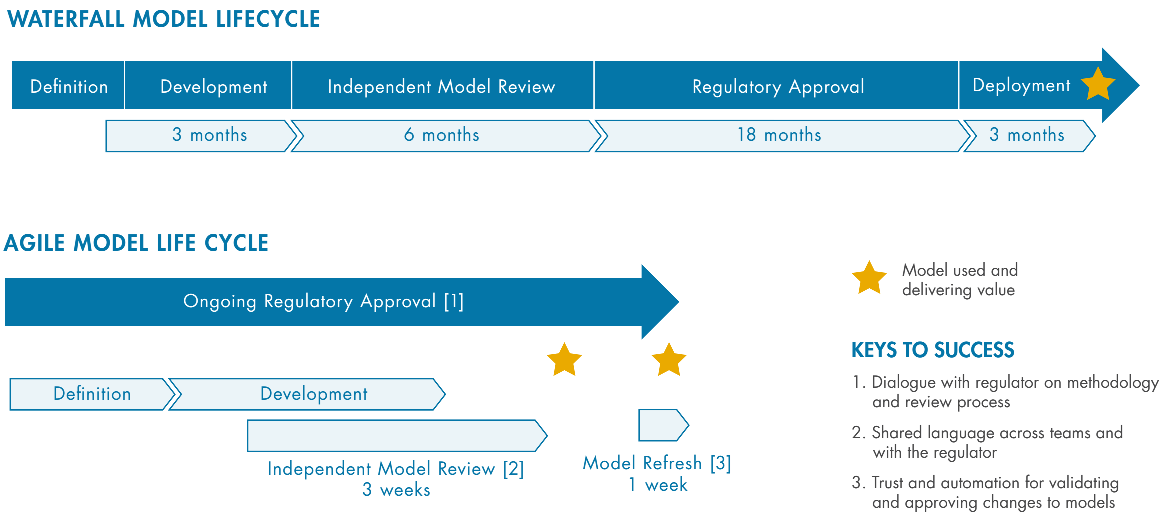
WATERFALL MODEL LIFECYCLE
Definition
Development
Independent Model Review
Regulatory Approval
Deployment
3 months
6 months
18 months
3 months
AGILE MODEL LIFE CYCLE
Ongoing Regulatory Approval [1]
Definition
Development
Independent Model Review [2]
3 weeks
Model Refresh [3]
1 week
Model used and delivering value
KEYS TO SUCCESS
1. Dialogue with regulator on methodology and review process
2. Shared language across teams and with the regulator
3. Trust and automation for validating and approving changes to models

Figure 1. Reducing model time-to-market and refresh by using an agile approach rather than waterfall.

Case Study: Streamlining Model Development and Deployment at HSBC

The Global Risk Analytics department at HSBC provides the bank with technology to manage risk and compliance by providing a platform for end-to-end management of the life cycle of risk models. To innovate, executives and senior management found opportunities to reduce costs in the model development and deployment process by streamlining and automating manual processes and to make use of advances in artificial intelligence and machine learning.

Working with MathWorks, HSBC looked at each stage of the model development and deployment process and identified areas that could be improved:

- Time spent processing data versus modeling and analysis
- Reproducibility and traceability of results
- Consistency of modeling approaches across teams
- Documentation of models for validation and regulatory approval
- Recoding of models from documentation to execute in production

MathWorks partnered with HSBC to develop an integrated solution that optimizes the above areas (Figure 2). The solution used by HSBC comprises two key environments developed in MATLAB that correspond with different stages of the model life cycle: the Model Development Environment (MDE) and the Model Execution Environment (MEE).

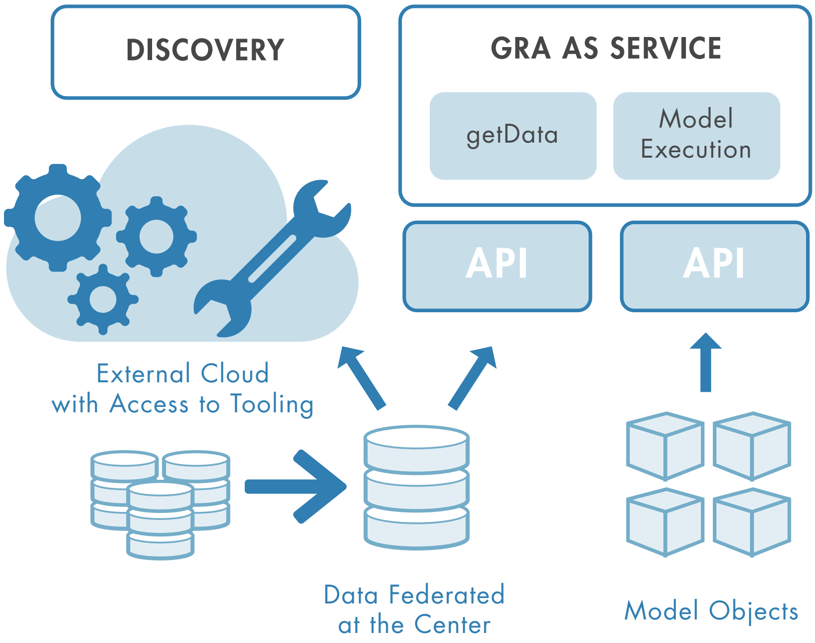
DISCOVERY
GRA AS SERVICE
getData
Model Execution
API
API
External Cloud with Access to Tooling
Data Federated at the Center
Model Objects

Figure 2. HSBC's vision of an agile model development and deployment environment.

> *"We are trying to build an environment where we have consistent data and run standard tools against it in terms of modeling."*
>
> — Ray O'Brien, HSBC

With MDE, HSBC increases the pace, transparency, and reproducibility of the model development and review process. The user-friendly MATLAB environment supports all stages of the workflow and encourages a consistent approach to model development.

With MEE, HSBC reduces the amount of time needed to recode approved models to run in a production environment. Models are automatically deployed to MATLAB Production Server™, and users can call and monitor models in real time using the built-in APIs to ensure that the models are behaving as expected and meeting business objectives.

MATLAB enables HSBC to develop models consistently according to best practices and standards and meet evolving regulatory requirements and timelines. Models developed using MDE are comprehensively documented for model validation teams to review, and can be deployed automatically to the MEE once approved, delivering immediate benefits.

Pain Points in Model Risk Management

Data Handling and Documentation Writing in Model Development

The traditional approach to building risk models is data intensive and document centric, requiring significant resources and investment. Quantitative researchers must collect large amounts of data, manipulate it to build a risk model, and produce a lengthy justification document that is presented to the model review team and regulators.

Reproducing, Interpreting, and Validating Model Results

The validation process is time-consuming because models are not developed consistently and use different methodologies. The process of manually building models from lengthy definition documents means that they are not easily replicated for independent model review.

Accelerating and Scaling Implementation

Deploying models frequently requires translation and recoding by an IT department. This process can introduce errors if coding mistakes are made. Errors cause further delays in implementing models and increase model risk.

Monitoring in Real Time

Manual monitoring and review processes take place at set intervals; therefore, a model's performance is only measured at a specific point of time. If models are only reviewed on a quarterly or annual basis, defects may go unnoticed for long periods of time, raising the risk of financial loss. Changes to economic and market conditions, the company's strategy, or even the personnel using the model can create a risk where the model is being used for unintended purposes.

Case Study: The PRA's Four Principles for Model Risk Management

The UK's Prudential Regulation Authority (PRA) and the European Banking Authority (EBA) have introduced their own requirements for MRM. In 2018, the PRA introduced supervisory statement SS3/18 [3], which provided guidance on model risk management principles for stress testing exercises, and the EBA published its Supervisory Review and Evaluation Process (SREP), which requires model risk to be identified, mapped, tested, and reviewed. Under the EBA regulation, model risk is treated as a risk to capital, and institutions are required to quantify it accordingly.

The PRA supervisory statement sets out four principles that require firms to put in place definitions and controls over governance, usage, and monitoring. The four principles state that banks must:

- Have an established definition of a model and maintain a model inventory
- Have implemented an effective governance framework, policies, procedures, and controls to manage their model risk
- Have implemented a robust model development and implementation process and ensure appropriate use of models
- Undertake appropriate model validation and independent review activities to ensure sound model performance and greater understanding of model uncertainties

These principles clearly show that the PRA views model risk management as covering the entire model life cycle.

Model risk management with MATLAB provides a transparent platform that streamlines the process and helps companies meet regulatory expectations of how institutions manage their models. The platform addresses the PRA's requirements through a series of integrated environments that address the needs of different users of models at each stage of the model life cycle (Figure 3).

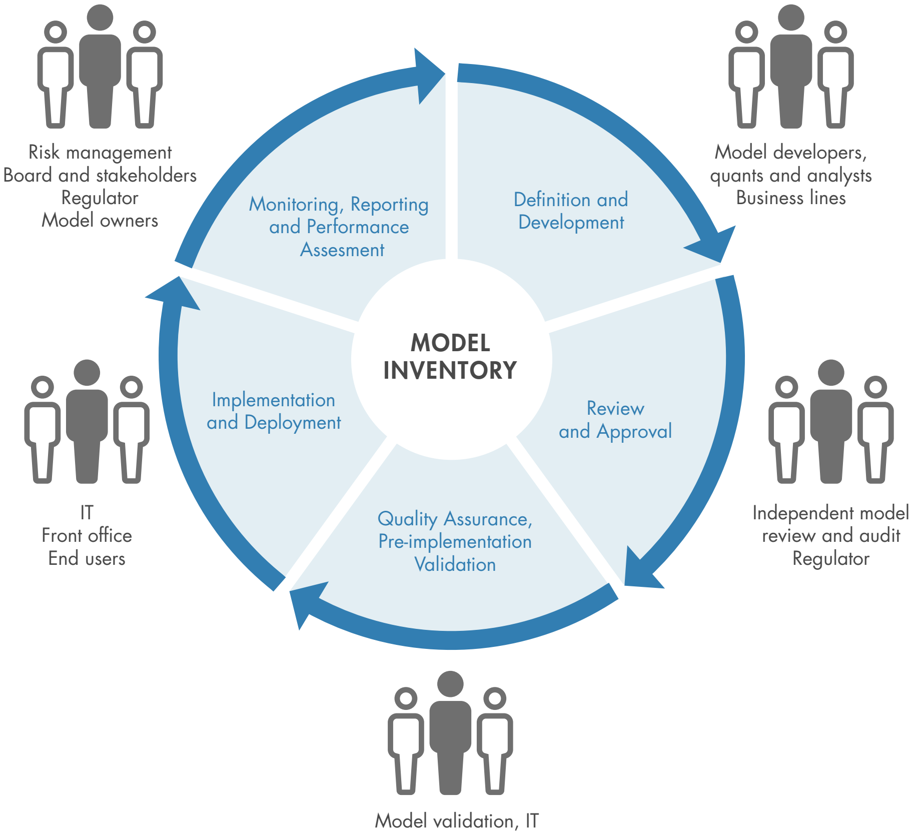
Risk management
Board and stakeholders
Regulator
Model owners
Monitoring, Reporting and Performance Assesment
Definition and Development
Model developers, quants and analysts
Business lines
MODEL INVENTORY
Implementation and Deployment
Review and Approval
IT
Front office
End users
Quality Assurance, Pre-implementation Validation
Independent model review and audit
Regulator
Model validation, IT

Figure 3. Meeting the needs of users throughout the model life cycle.

Complete Model Inventory

The Model Inventory is a comprehensive inventory that details who developed each model and how it is used. It ensures that models use data that has been checked for quality and relevance, and that there is detailed documentation that enables users to understand how the model operates.

Once models have been designed and developed, they are stored in the Model Inventory, where users can access them through other environments. Users can quickly review a model's status, view its development history, and place access controls on models and artifacts.

Effective Governance Framework

The PRA holds the view that while firms' boards of directors do not need to understand the model and the equations within it, they do need to understand the limitations of the model: when it will work and when it will not. MATLAB uses automation to monitor models in real time to reduce the risk of a financial institution using a defective or incorrect model. The Model Monitoring Dashboard provides an overview and monitoring capability so that model owners and risk functions can track the status and performance of models hosted in the Model Execution Environment. Along with a configurable web dashboard that summarizes execution results measured against time, client, and model inputs, it also has fully configurable alerts that notify users of unusual model activity or changing input conditions. Unlike intermittent monitoring, which can take place quarterly or annually, real-time monitoring prevents models from drifting over time.

Every step of the model life cycle in MATLAB is logged and documented, ensuring a transparent audit trail, and users can access a full model inventory that provides access to their status, documentation, and test reports. MATLAB ensures that all model development and validation activities are implemented in a transparent, accurate, and consistent manner that reduces model risk, builds trust in process and methodologies, and helps to ensure compliance with regulatory requirements.

Robust Model Development and Implementation

Using the Model Development Environment built in MATLAB, financial institutions benefit from a single environment that enables them to quickly develop and implement robust models. MATLAB supports the creation of robust model templates at the beginning of the development process. It features automated documentation generation so that these templates can easily be reviewed and updated as required. Automation not only accelerates the model creation process, but also provides an audit trail that ensures that the process is repeatable and consistent.

As an integrated model risk management tool, MATLAB makes it possible to deploy models quickly and without the need to go through the slow process of translation. The Model Execution Environment allows users to automatically deploy models into production without going through the time-consuming process of translation. It is a secure, controlled environment for hosting production models that can scale for large numbers of users and for increasing amounts of data and simulations.

Appropriate Model Validation and Independent Review

MATLAB improves and accelerates the validation process by providing comprehensive documentation covering the model's journey through the development process. The Model Review Environment (MRE) provides validation teams with a desktop and browser-based environment where they can access a complete set of artifacts generated during the development process.

In MRE, users can run the end-to-end workflow, inspect the model, analyze results, and view the full audit trail produced in MDE. They can quickly backtest models with alternative data sources and evaluate the choices the developer made when creating the model. Because all of the information related to developing models is traceable and easily accessible, reviewers can inspect every aspect of a model so that issues can be quickly identified, diagnosed, and traced back to the source.

Conclusion

In a world of fast-moving financial markets, increasing regulatory burdens, and larger datasets, the practice of managing model risk becomes critically important. Regulators have put firms under greater pressure to understand the risks associated with their models, while management teams are seeking ways to create efficiencies and reduce costs. By implementing robust MRM frameworks, financial institutions are improving the quality and transparency of their models, while also demonstrating effective oversight of the tools they use for making critical decisions.

Good model risk management is essential for ensuring that models are clearly defined, properly validated, used as intended, and continually monitored and updated. The complex financial systems of the modern world require ironclad processes that reduce risk and support regulatory compliance. Therefore, using an integrated model risk management solution that incorporates validation and testing, and promotes accuracy and transparency, is paramount.

MATLAB provides financial institutions with an interactive environment and a rich feature set to enable the development, review, and implementation of risk models. Using the rich documentation and a full lineage of the model life cycle provided by MATLAB, MRM professionals can gain a greater understanding of the model risk and articulate this to management and boards of directors. Moreover, by processing large amounts of data from multiple sources automatically, they can develop and implement models quickly while using less time and resources than in the past.

References

[1] McKinsey & Company, *The evolution of model risk management*, 2017

[2] Board of Governors of the Federal Reserve System Office of the Comptroller of the Currency, *SR Letter 11-7: Guidance on Model Risk Management*, 2011

[3] Prudential Regulation Authority, *Model risk management principles for stress testing – SS3/18*, 2018

Learn More

Two Worlds Coincide: Financial Risk Management and Model-Based Design (18:45) - Video

MATLAB for Quantitative Finance and Risk Management - Overview