

Model Risk: Implementation Testing Challenges

By

P S S Moorthy & Venkata Krishna R

Model development, implementation and use phases of the model life-cycle are usually handled by different teams within a bank or financial services institution. So, the risk associated with the implementation and use of the models is very high due to the handing-off of the model to different stakeholders at specific phases of model life-cycle.

Models are usually embedded in large information systems that manage the flow of data from various sources into the model and handle the aggregation and reporting of model outcomes. Model calculations should be properly coordinated with the capabilities and requirements of information systems. Consequently, this whitepaper talks about the challenges a bank would face while testing the implementation of model in its complex systems.

Common checks that need to be performed while testing the implementation of a model are discussed in Table # 1.

TABLE #1:

Check	What to look for?
1. Model implementation planning and execution	<ul style="list-style-type: none">• Existence of a process of migrating model from development environment into a production environment for ongoing execution and use in business processes.• Whether practices are in place to ensure that implementation planning, testing, and execution procedures are sufficient to support proper migration of the model into production.
2. Model implementation controls	<ul style="list-style-type: none">• Robust controls in place to ensure that all the components of the model in production continue to function as intended.• Whether practices are in place to ensure that sufficient control infrastructure is in place to support proper model execution post-implementation.

Typical challenges faced by banks while testing the implementation of models in its systems are discussed in Table # 2.

TABLE #2:

Implementation testing task	Challenge
1. Technical specifications of model	<ul style="list-style-type: none"> • Lack of documentation on the following: <ul style="list-style-type: none"> ○ Description of model structure and process/data flow ○ Input data specifications (data dictionary) and processing rules ○ Details of the model calculation algorithms and formulas ○ List of all assumptions and limitations of model ○ A user's guide to provide model users with detailed instructions on how to use the model, especially if the model interface is fairly complex. ○ Model development code (with annotation) written as per set/approved coding standards, acceptance criteria etc. ○ Tests conducted before moving the model from development to production
2. Model implementation environment	<ul style="list-style-type: none"> • Models are developed and implemented in different environments without continuous system integration between them. • Standard IT practices not being followed for full unit and tie-out testing.
3. Technical architecture	<ul style="list-style-type: none"> • Models are implemented in different platforms that do not share a standard language and technical architecture, along with a standard set of Application Programming Interfaces (APIs) and user interfaces.
4. System integration testing	<ul style="list-style-type: none"> • Automated integration testing of key downstream systems is not performed to analyze full impact during development. • Impact on business systems is not evaluated and accepted by relevant stakeholders. • Full regression testing of code is not performed for any change in systems in order to ensure proper working of desired functionality.
5. Model implementation code review	<ul style="list-style-type: none"> • Automated code review tools are not used to measure and control changes to key components of code and interfaces. • Manual reviews are not performed when there are major changes in code.
6. Verification & User Acceptance Testing (UAT)	<ul style="list-style-type: none"> • Test plans designed by developer, including developer-specified test scenarios are not executed by an independent 'Quality Assurance' Team. • User acceptance criteria are not properly defined and implemented.
7. Model implementation controls	<ul style="list-style-type: none"> • Lack of following controls: <ul style="list-style-type: none"> ○ Security and access ○ Robust release/deployment ○ Version
8. Ongoing monitoring	<ul style="list-style-type: none"> • Lack of robust ongoing monitoring mechanism to constantly track the model's performance through specific model diagnostics. • Business continuity plan (BCP) and Disaster recovery plan (DRP)

Parting thought

Robust model risk management depends on substantial investment in supporting systems to ensure data and reporting integrity, together with controls and testing to ensure proper implementation of models, effective systems integration, and appropriate model use.

About authors

P S S Moorthy works as Assistant Vice President – Financial Services Analytics at GENPACT. He is a subject matter expert in risk and regulatory space and currently manages the model risk management teams of Genpact's US- based financial services clients.

Venkata Krishna R works as Senior Manager – Financial Services Analytics at Genpact. He is a subject matter expert in 'ongoing monitoring and implementation testing' aspect of model risk management and currently leads a team in that space for Genpact's US - based financial services client.