INSIGHTS

Model Validation Is Everyone's Business

A robust and efficient model validation process is in the interest of all stakeholders; from the model development group, through business lines and IT, and all the way up to senior management and the regulators.

- What is the model validation team really concerned about?

- How does model development sometimes obstruct this process?

- How can you reinforce a collaborative culture with a platform that takes you seamlessly from research to production?

MathWorks®

Introduction

SR11-7 defines model validation as "the set of processes and activities intended to verify that models are performing as expected, in line with their design objectives and business uses." This definition really doesn't do it justice. An efficient, industrialized, model validation function is key to:

- Staying competitive
- Keeping costs under control
- Getting the most from your model developers
- Managing risk
- Securing a partnership with regulators
- Gaining senior management confidence

Model validation is sometimes perceived as the less glamourous and perhaps even obstructive bit of the model risk management (MRM) workflow. Certainly, done wrong, it can feel that way to the model development team. The validation team are the ones who keep finding problems with the models and telling everyone about it, including management and sometimes even the regulator.

But the model development team needs to ask themselves, who are being the obstructive ones? With the right institutional, cultural, and technical framework in model development, and across the entire model lifecycle, model validation can have an enormous added value.

Model Validation

Models get built by the model development team, and the validation team is usually responsible for validating, among other things:

- The business case for building the model
- The assumptions that went into the model
- The data selection and cleaning processes
- Variable selection / feature engineering
- Statistical methods applied
- Results
- Robustness to different / extreme inputs

Development passes over a load of documentation, code, and data to the validation team. The quality, consistency, and efficiency of this process make an enormous difference to the validation team's ability to get the model validated quickly and accurately.

Traditional Model Development and the Model Validation Relationship

A poor framework for the interaction between development and validation is one where the developers perceive the quality of the validation process as "not their problem" and behave accordingly.

One like this:

Code / Data / (maybe) Docs
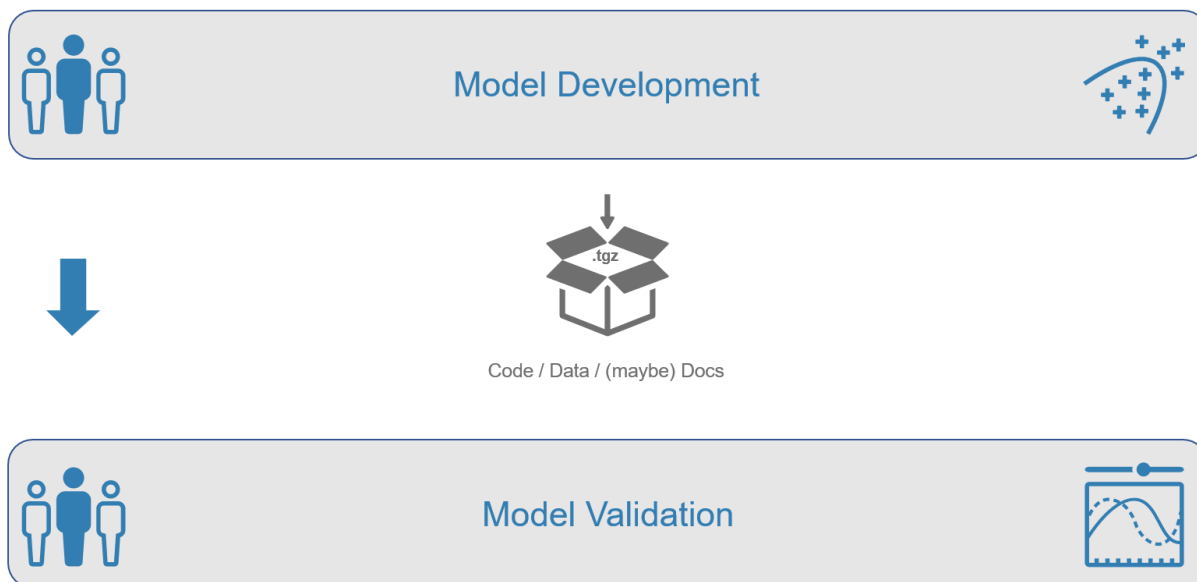
Poorly structured, uncommented, bug-ridden code that's inconsistent with the documentation is fun for nobody. Inconsistent or vague data selection hobbles reproducibility. Apparently arbitrary feature choice and statistical techniques confuse the picture further. The validation process immediately gets bogged down in an iterative attempt to just get the development team's code to run, understand what data it's supposed to be run on, and work out what it's doing.

As a model developer, up against a tight business deadline, you do not want to have someone from model validation having you step through your code in the debugger and shouting:

> *"HOW HAVE YOU GOT -42.67234129 IN CELL AF44952 OF THE THIRD TAB IN THE SPREADSHEET?!?!"*

No, this approach to model validation helps nobody.

Constructive Partnership Between Development and Validation

While model validation is strictly independent of development, that doesn't mean it needs to be one-way traffic. The right culture, enabled by software, can make for a collaborative, positive interaction.

One like this:
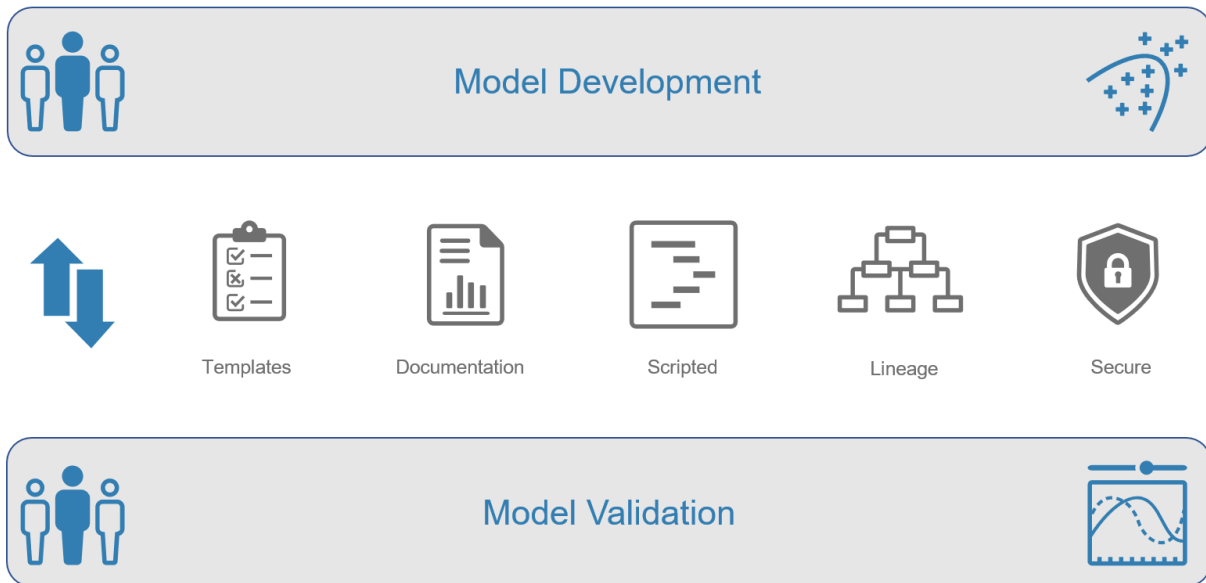
Software tools and workflows reinforce a culture of collaboration and partnership between development and validation, while keeping the discrete separation of responsibilities intact. The model development team knows what the validation team expects to receive, and how they expect to receive it. This facilitates an organized, iterative, and efficient validation process.

- **Model templates** help developers build models in a consistent style using well-structured code.
- **Documentation templates** ensure that every aspect of the model is well documented.
- **Structured, scripted, executable models** allow for an iterative, agile development process.
- **Full model lineage** through model inventory eliminates confusion about the versioning of code and data.
- **A secure, controlled environment** allows appropriate access to sensitive data.

The software infrastructure that makes this possible would include the following:

Feature	Capability	Validation Benefit
Centralized model inventory and repository	Manage validation workflow and artifacts Inspect models, intermediate results, and audit trails	Avoid: "Are you sure you're running the right version?" Understand model and data dependencies immediately
Template-driven model development workflows	Encourage a consistent model development approach Include common validation issues in the template, can be addressed proactively	Dealing with the same framework every time Common issues never get to validation
Graphical, interactive, executable model documentation	Understand model usage, limitations, and reproducibility	The code is consistent with the documentation Diagnostics, like sensitivity analysis and residuals, available out-of-the-box
Complete model and data lineage	Report on development, usage, and dependencies	Understand the full development lifecycle, including when and why decisions got made
Highly scalable compute and big data capabilities	Fast iteration of code runs	Less time waiting for code to run

These are all features, among others, of the Development, Review, and Validation stages of the MathWorks Model-Risk-Management solution. It's a complete, comprehensive, and modular platform that equips the first and second lines of defense with robust **analytics** and integrated **model lifecycle** management.

Summary

From the model developer's perspective – the easier they make life for model validation, the less time they spend answering queries and the more time they spend building models – which is what model developers love to do!

From the business line's perspective – the more efficient the validation process, the faster models can get from inception to production, and the more responsive the business can be to competitive threats and opportunities.

From the risk manager's perspective - validation is responsible for ensuring that models are not prone to 'blowing-up' either financially, or reputationally (when found to be using some protected attribute, or a proxy thereof).

From the IT department's perspective - improved coding practices, an ancillary benefit of good validation practices, feed through to more reliable production model code, reducing the deployment and support load.

From the senior manager's perspective – an efficient validation function makes for lower costs, not just within the validation team but in development and across the model lifecycle.

From the regulator's perspective – robust validation reduces systemic risk, increases innovation, and allows the regulated entity more headroom for growth.

Model validation is everyone's business.

> Learn about the MathWorks Model Risk Management solution:
>
> mathworks.com/mrm