International Journal of Management, IT & Engineering
Vol. 16 Issue 4, April 2026
ISSN: 2249-0558 Impact Factor: 7.119
Journal Homepage: http://www.ijmra.us, Email: editorijmie@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gate as well as in Cabell's Directories of Publishing Opportunities, U.S.A

Evaluating Model-Based Products in Enterprise Program Management: A Governance and Decision Framework for Technical Program and Product Managers

Prakash Achuthan
Senior Manager, Delivery, Amazon Services LLC

Abstract

Model-based products powered by artificial intelligence and large language models generate probabilistic outputs that introduce reliability risk, cost variability, performance drift, explainability limitations, data dependency, and organizational adoption challenges [1][2]. Traditional deterministic software evaluation mechanisms are insufficient for assessing such systems in enterprise environments [3]. This study proposes an expanded Model-Based Product Evaluation Framework (MBPEF), a structured governance architecture designed for Technical Program Managers or Product Managers overseeing cross-functional AI deployments. The framework integrates seven evaluation pillars: reliability assessment, economic viability modeling, performance measurement, continuous lifecycle governance, explainability and transparency, data governance and quality management, and human-in-the-loop adoption oversight. Drawing upon decision support systems theory [4], AI governance standards [5], data management frameworks [6], and organizational change principles [7], this study synthesizes a practical and scalable approach for balancing innovation with accountability.

Keywords:

Technical Program Management;
Technical Product Management;
Generative AI;
AI Governance;
Model Evaluation;
Enterprise Risk.

1. Introduction

Artificial intelligence systems differ from deterministic software in that they generate probabilistic outputs based on statistical inference rather than fixed rules [1]. While this enables pattern recognition and automation, it also introduces uncertainty and potential hallucination risk [2]. Enterprise leaders must evaluate AI systems through structured governance mechanisms before large-scale deployment [3].

Technical Program Managers (TPMs) and Product Managers (PMs) coordinate engineering, product, legal, and finance stakeholders. However, structured evaluation models tailored to AI-driven products remain limited in existing literature [4]. The MBPEF addresses this gap by integrating multi-dimensional governance pillars.

2. Research Method

Decision Support Systems research emphasizes structured managerial oversight combining analytics and governance [4]. AI governance literature highlights hidden technical debt, bias propagation, and lifecycle drift in machine learning systems [2][5]. Data governance research underscores lineage tracking, quality monitoring, and regulatory compliance [6]. Organizational change research stresses trust and adoption readiness as determinants of technology success [7].

Although these fields provide foundational insights, they often assume technical expertise. There remains a need for practical, program-level frameworks that translate these academic insights into clear execution guidance for cross-functional leaders.

This study applies conceptual synthesis across peer-reviewed AI governance literature [2][5], enterprise deployment best practices, and program management theory. Recurring evaluation themes were consolidated into seven integrated governance pillars.

3. Framework

3.1 Reliability and Hallucination Risk Assessment

The concept of reliability pertains to the consistency of an AI system in producing correct outputs. One of the most common challenges in AI model consistency pertains to Hallucinations which are a documented and persistent risk in AI systems [1][2]. Large Language Models (LLMs) choose the most likely sequence of words, and the resultant hallucination may result due to either gaps in the training data, or being required to force a

specific pattern ("overfitting") or have limited context windows causing the model to extrapolate beyond its sources of information or "attention span".

A common means of measuring whether a model hallucinates is to measure the Hallucination rate percentage (% incorrect outputs) or Faithfulness (adherence to training data or retrieved context) by asking the model to respond repeatedly to similar queries while incorporating minor changes in the input parameters. Changes in the product's output for multiple input queries that are the same indicate a lack of model consistency. Another means of inspecting model consistency is by inspecting the model's Temperature setting where lower temperatures tend to make the model's outputs more deterministic and limited to the knowledge base, while higher temperature settings encourage creative (and likely) erroneous outputs.

For Model based Product development, it is critical to understand the domain and risk acceptability. Regulatory, Healthcare and Financial sectors lend themselves to outputs that are more deterministic, grounded in knowledge bases and trusted sources. Creative ventures such as marketing or establish predefined acceptable error thresholds and apply adversarial testing methods before deployment [5].

3.2 Economic Viability Modeling

AI-enabled systems consume computing resources that incur operational cost, particularly in cloud-based environments [8]. A few key metrics used to quantify the "cost of operations" for model enabled products include (i) total cost of model deployment (TCMD); (ii)marginal cost per successful outcome; and the (iii) scaling sensitivity analysis. A key part of any product development must therefore include careful calculation of the Cost per Inference (the price of a single question/answer) and the Total Cost of Model Deployment (TCMD), which includes everything including Infrastructure and Inference costs, Data pipeline, Governance and Operational and Human capital. By performing a scaling sensitivity analysis, a TPM or PM can predict if a sudden surge in users will bankrupt the project or if it remains profitable as it grows. An often-missed element of calculating the scaling sensitivity analysis is the inclusion of forecasting/assuming retraining costs (cost to train and test the model when new data/knowledge is available) to address any model response drifts.

This pillar is critical for all organizations but especially for startups where model-based products (such as an AI enabled photo editing app) may be subject to rapid scaling when the

product enjoys popularity with a large number of users, and an insufficient scaling plan can lead to the product failing to generate revenue due to burgeoning costs.

3.3 Performance Metrics Architecture

In model-based products, the "performance" of a model isn't just a single number; it is a balance between technical precision and how a human actually feels using the tool. Technical metrics like precision and recall inform the TPM and the team, if the model is finding the right information, while non-functional metrics such as latency measure if the AI is too slow to be useful. This is an area where experience in building model scorecards and intuition play equally important roles. A common approach used by TPMs tasked with delivering successful model-enabled products is the usage of a balanced scorecard to look at all these factors at once. For example, they might decide to use a faster, less accurate model if the user's primary need is speed, or a slower, highly accurate model if the task is critical, such as a legal review. User satisfaction in the model's response is a key factor in the success of any model-enabled products, while latency, consistency and other metrics that define a successful product "experience", play equally important roles.

3.4 Continuous Improvement and Drift Governance

It's a generally well-known fact that model performance degrades over time due to concept drift and evolving data distributions [9]. AI models are trained on a "snapshot" of data from the past. As the real world patterns change, the model's performance begins to "drift", meaning it becomes less accurate because it doesn't understand new trends or information. Another reason for drift is that the data it's retrained on, might be biased toward a specific pattern, causing a drift in the model's responses relative to expected outputs. Some of the more common mechanisms used to address model drift are the (i) establishment of a drift detection threshold; (ii) establishing a retraining frequency with data analysis; (iii) regression test pass rates informing the team how consistently the model-enabled product generates successful, consistent outcomes and (iv) anomaly rate. Last but not least , a metrics dashboard measuring these metrics on an ongoing basis provide the TPM or PM and development teams with a clear, objective measure of whether the product's output is beginning to drift from intended/expected outcomes or not.

3.5 Explainability and Transparency

"Explainability" is the ability to look inside the AI's "brain" and understand why it made a specific choice. This is vital for building trust with stakeholders when there is a question about why the product responded in a particular way, and in meeting legal requirements. Unlike traditional software engineering, , where model outputs could be traced to code blocks or actions, models are “probabilistic” meaning they make decisions based on statistical probability. However, despite the challenge in model explanability, the ability to demonstrate model explainability enhances stakeholder trust and regulatory defensibility [5][10].

During product development, the TPM generally tracks the (i) explanation coverage rate to ensure the system can provide a reason for most of its actions. They also ensure (ii) version tracking and audit logs are complete, creating a permanent record of the AI's logic, if the need to inspect an output arises. By implementing (iii) trace retrieval, a TPM makes it possible to go back in time and "audit" why the AI gave a specific (and perhaps controversial) answer. However, these mechanisms depend generally on the kinds of models used for the product. LLMs in general, tend to be highly complex and harder to explain since “explainability” is not a native feature of LLMs. TPMs generally adopt version tracking and explanation coverage rates (how many of the outputs can be traced back to training data or logic path) to address LLM explainability when required to do so.

3.6 Data Governance and Data Quality

As indicated in prior sections, an AI model is only as smart as the data it consumes. If the data is "stale" (old) or contains "bias" (unfair patterns), the AI will be flawed. Model reliability therefore depends on data integrity, freshness, and bias mitigation [6][11].

The TPM or PM monitors (i) data freshness to ensure the info isn't outdated and uses a (ii) bias disparity index to check if the AI is treating different groups of people unfairly. They also track (iii) data lineage, which is a "family tree" for information that shows exactly where data came from and how it was changed. Automated pipelines help validate this data constantly, ensuring the AI is built on a clean foundation.

3.7 Human-in-the-Loop and Organizational Adoption

Successful adoption of a model-enabled product depends on user trust and structured oversight mechanisms [7]. All production models in general adopt sample testing before and once the model-enabled product has been deployed into Production, through Human-in-the-

loop (HITL) inspection, allowing for a structured inspection of a select set of the product's outputs and interactions. The wide adoption of a model-enabled product is also a key factor in the success of the product since wider adoption and therefore increased feedback data allow for more signals to evaluate the product (and model's) performance and gather retraining data for subsequent cycles.

For the HITL processes, a TPM or PM measures override frequency—how often a human has to correct the AI's work—to see if the system is truly helping or just creating more work. They also track trust scores and adoption rates to see if employees or external users are using the tool. By designing formal review workflows, the TPM or PM ensures that the AI acts as a helpful assistant rather than an unmonitored system.

4. Conclusion

The 7 pillars of the MBPEF integrate reliability, cost modeling, performance evaluation, lifecycle governance, explainability, data integrity, and human adoption into a unified enterprise framework. By structuring AI evaluation across these seven pillars, TPMs, PMs and in general any teams working on model-based product development can responsibly scale model-based products while maintaining transparency, compliance, and sustainable business value.

References

[1] Bender, E.M., et al., On the Dangers of Stochastic Parrots, FAccT, 2021.

[2] Sculley, D., et al., Hidden Technical Debt in Machine Learning Systems, NIPS, 2015.

[3] Floridi, L., AI Ethics and Governance, Nature Machine Intelligence, 2019.

[4] Power, D.J., Decision Support Systems: Concepts and Resources, 2007.

[5] ISO/IEC 23894, Artificial Intelligence Risk Management, 2023.

[6] Khatri, V., & Brown, C., Designing Data Governance, Communications of the ACM, 2010.

[7] Venkatesh, V., et al., User Acceptance of Information Technology, MIS Quarterly, 2003.

[8] Armbrust, M., et al., Above the Clouds: A Berkeley View of Cloud Computing, 2010.

[9] Gama, J., et al., A Survey on Concept Drift Adaptation, ACM Computing Surveys, 2014.

[10] European Commission, Ethics Guidelines for Trustworthy AI, 2019.

[11] Barocas, S., & Selbst, A., Big Data's Disparate Impact, California Law Review, 2016.