

In the fast-evolving landscape of AI-powered SaaS products, one of the most critical—and often misunderstood—decisions product teams face is: **which model should power my feature or workflow?** More specifically, when does it make sense to use a reasoning model (e.g., large language models with deep contextual capabilities) instead of a smaller, leaner model? This question is central to building AI features that are reliable, scalable, and aligned with business goals.

In this post, I'll share practical frameworks and considerations drawn from years of shipping AI-assisted workflows, incorporating insights from tools like PM Toolkit and lessons learned at companies like PayPal and H2O.ai. We'll also discuss how you can transition smoothly from spreadsheet-based lightweight evals to automated pipelines without losing control over quality.



The Model is Not the Product: Start With User Workflow

A fundamental mistake I often see is teams fixating on the model itself—as if the model is the product. The reality is that the model is a component, typically one step in a multi-stage user workflow. Before you even consider which model to use, you must answer:

- **What are users doing today without AI? What pain points exist?**
- **What parts of the task could be automated?**
- **Is the AI intended to augment users' capabilities or fully automate a task?**
- **Which failure modes are tolerable—and which are deal-breakers?**

For example, at PayPal, models <https://pmtoolkit.ai/learn/ai-modern-pm/ai-literacy-for-pms> supporting fraud detection workflows must be highly reliable with clear kill switches, whereas in a creative content generation feature, tolerance for some hallucinations might be higher if users can quickly validate and edit output.



Augmentation vs Automation vs Innovation

Another crucial framing is: are you building an AI that augments human decision-making, aims to automate an end-to-end process, or creating an innovative new product category? This determines the complexity and type of model required.

- **Augmentation:** Smaller models often suffice to surface helpful suggestions or extract structured insights that humans validate.
- **Automation:** Requires more robust reasoning models with *defined kill switches* and monitoring to handle corner cases.
- **Innovation:** May warrant experimenting with new reasoning model architectures or multi-model ensembles, but with careful pilot evaluations before scaling.

Defining an Evaluation Before Release: The Non-Negotiable Step

One of my personal quirks is that I always write evaluation examples like user stories—realistic scenarios that reflect expected and edge-case user behaviors. But here's the catch:. This comes before picking or training models.

Tools such as PM Toolkit calculators become invaluable here. For example:

- **A/B test sample sizes:** Before deploying a new model, ensure you can statistically detect performance differences with PM Toolkit's *sample size calculator*.
- **Retention and churn:** Model impact on retention is a key eval metric, tying to Business Case calculators for ROI forecasts.
- **NPS and PMF scores:** Use these after initial deployment to gauge user satisfaction and product-market fit.

In my previous projects at H2O.ai, we tracked metrics like cycle time and velocity in delivering model improvements, iterating rapidly on failures caught in evaluation spreadsheets. This lightweight, transparent vetting helped avoid costly model errors in production.

Task-Dependent Model Behavior and Failure Modes

Choice of model size and type isn't only about performance metrics like perplexity or token accuracy. It's about understanding how different models behave on *specific reasoning tasks* and their failure modes.

Model Type Strengths Common Failure Modes Best Use Cases Smaller Models

- Faster inference & lower cost
- More predictable and interpretable
- Good for extraction and classification
- Limited context window
- Reduced reasoning capacity
- More brittle on less-structured inputs

Structured data extraction, simple classification, rule-based augmentations Reasoning Models (Large Language Models)

- Strong contextual reasoning
- Flexible multi-step problem solving
- Generative creativity and summarization
- Hallucination and fabricated facts
- High compute cost and latency
- Opacity and difficult debugging

Complex reasoning tasks, natural language understanding, generation, multi-turn dialogue

In practice, hybrid models or fallback heuristics are often necessary. At PayPal, for instance, smaller classification models trigger workflows that invoke larger reasoning models selectively for difficult cases, optimizing cost and reliability.

From Spreadsheets to Automated Pipelines: Managing Evaluations

Before automation pipelines became widely accessible, I relied heavily on spreadsheet-based eval suites. These suites captured user scenarios, model outputs, error categorizations, and allowed sorting and filtering by failure type. This lightweight approach brings enormous clarity to model choice and tuning.

Combining PM Toolkit's calculators for business metrics with spreadsheet evals of reasoning tasks creates a holistic view for decision-making. For example, if a reasoning model's incremental accuracy gain translates to only marginal ROI increase due to much higher cost and latency, the spreadsheet evaluation can stop you from premature scaling.

Now, as automated pipelines become easier to implement, maintaining the rigor of defined evaluations—feature-aware inputs, clear pass/fail criteria, versioned test datasets—is non-negotiable to avoid regression and unanticipated failure modes.

How to Decide: Reasoning Model or Smaller Model?

1. **Map the user workflow and pain points:** Use the PM Toolkit's business case and TAM/SAM/SOM calculators to quantify opportunity size and prioritize features.
2. **Define task-specific evaluation criteria:** Write user story-style eval examples capturing success and failure cases before model selection.

3. **Analyze model behavior on these tasks:** Use spreadsheets or lightweight pipelines to assess reasoning capabilities vs failure risks.
4. **Weigh augmentation vs automation:** For augmentation, smaller models usually suffice; for automation or innovation, consider reasoning models with kill switches.
5. **Calculate cost vs benefit:** Use PM Toolkit's ROI and pricing calculators to evaluate feasibility at scale.
6. **Plan rollout with kill switches and monitoring:** Ensure you can roll back quickly if failure modes emerge in production.

Summary Table: Model Choice Decision Checklist

Question	Use Smaller Model if...	Use Reasoning Model if...
What is the complexity of the reasoning task?	Simple extraction/classification, predictable inputs	Requires multi-step reasoning, abstraction, or synthesis
What is the acceptable error tolerance?	Low tolerance for errors and hallucinations	Higher tolerance with user validation or iterative feedback
Are latency and cost constraints critical?	Yes — prioritize efficiency and lower compute cost	No — can absorb higher cost for improved performance
Is augmentation or full automation the goal?	Primarily augmentation and decision support	Partial or full automation with fallbacks
Do you have clear eval data and business metrics?	Comprehensive evals showing smaller model meets thresholds	Reasoning model's improvements justify additional complexity

Final Thoughts: Avoid Model Wars and Focus on Workflow

I can't emphasize enough that **model choice without workflow context leads to model wars**—competitive debates over model size and architecture that ignore the real-world task and user needs. As product leaders, our job is to balance AI capabilities with business realities and user experience.

Companies like PayPal and H2O.ai succeed by defining clear evaluation frameworks and measured deployments. PM Toolkit offers a robust set of calculators that can guide your business case and experimentation design. And never forget the humble spreadsheet—it's still a powerful ally in stitching together user scenarios, model outputs, and failure modes into actionable insights before you automate.

So the next time you ask, *"Should I use a reasoning model or a smaller model?"*—pause, zoom out to the user journey, define your evals, use appropriate tooling, and pick the model that fits your task, failure tolerance, and business impact.

I'll be honest with you: author note: this post reflects my 12 years in b2b saas product management, focused on ai-assisted workflows and eval best practices developed at companies including paypal and h2o.ai.