

In the world of AI-assisted workflows, setting the right latency target for delivering the *first useful output* is a deceptively complex challenge. As product managers and AI leads, we often hear phrases like “faster is better,” but what does that mean in practice? How fast is fast enough? To answer these questions, we need to think beyond raw model speed and ground latency targets in real user workflows, task dependencies, and UX expectations.

In this blog post, we explore practical latency targets through the lens of modern B2B SaaS, drawing insights from companies like **PayPal**, **H2O.ai**, and tools like **PM Toolkit’s calculators and eval spreadsheets**. We’ll dissect why “the model is not the product,” illustrate how to define solid evals before release, and unpack how latency expectations depend heavily on automation, augmentation, or innovation use cases.

Why Latency Matters — More than Just Speed

Latency, the delay between user input and “first useful output,” is often treated as a raw engineering KPI. But for AI products, latency directly impacts:

- **UX Expectations:** How users perceive the responsiveness and usefulness of a tool.
- **Workflow Fit:** Whether the AI output aligns with the pace and flow of existing human tasks.

Consider a fraud detection product at **PayPal**. The system might produce model predictions in under 300ms, but the *first useful output* includes other steps like signal aggregation, contextualization, and risk scoring. For a fraud investigator, a 2-3 second delay for a well-contextualized flag might feel instantaneous and actually improve trust —whereas a too-fast, raw model output might feel noisy or incomplete.

Latency Isn’t One-Size-Fits-All

Think about it: it’s critical to tailor your latency target based on the nature of the task and user behavior. For instance, at H2O.ai, automating data preprocessing workflows requires a different latency expectation than a product augmenting live sales calls. In the former, batch processing with minutes-long latency may be acceptable, while in the latter sub-second responses might be critical.



Before fixing on a target number, ask yourself:

1. What does the user do today without AI?
2. How often do they pause to read or interpret results?
3. Is the AI output replacing a manual step or enabling a novel capability?

The Model Is Not The Product: Context Matters

It's a pmtoolkit.ai pet peeve of mine when teams focus too much on raw model benchmarks without anchoring in user context. A model's response time is necessary but never sufficient to define a product's latency target.

- **Post-processing and Integration Layers:** Real-world products layer business logic, split-second user interface updates, and safety checks on top of models. These layers add critical latency that shapes UX.
- **Failure Modes:** When models fail or hallucinate, the system must detect, mitigate, or degrade gracefully — introducing varying latency penalties.
- **Eval-Defined Thresholds:** A robust evaluation ahead of release—ideally using PM Toolkit spreadsheets or business case calculators—helps quantify acceptable latency vs. user value tradeoffs.

Simply put, shipping a model without a “kill switch” or fallback increases risk and erodes trust when latency spikes or failure modes appear. As a rule, your *ship/no-ship thresholds* should include latency bounds geared to user task tolerance, not just engineering quotas.

Defining an Eval for Latency Before Release

Instead of brainstorming abstract speed goals or chasing “lower latency,” define your eval as user stories and workflows. Examples might include:

- “When a claims agent uploads a document, the system should surface first suggestions within 2 seconds to keep pace with their review rhythm.”
- “Loan officers must see risk classification in under 5 seconds to avoid disrupting the 10-minute call cadence.”
- “Automated content tagging can tolerate 30 seconds latency since users batch-upload and don't wait for results interactively.”

Use PM Toolkit calculators like RICE (Reach, Impact, Confidence, Effort) or ICE to prioritize feature releases that optimize latency improvements in the highest value workflows. Combine this with business case models for ROI and NPS impact to quantify how latency shifts affect adoption and retention.

Spreadsheets remain an unsung hero in AI eval cycles: before investing in complex pipelines, lightweight scorecards capturing latency vs success rate vs user satisfaction can guide investment decisions. Leveraging an eval as user stories enables you to avoid “model wars” where competing teams argue raw benchmarks lacking workflow context.

Example Eval Table: Latency Impact on NPS and Retention

Latency Target	User Task	Expected NPS Impact	Retention Change	Effort (Story Points)
1 second	Live sales call augment	+15%	+7%	20
5 seconds	Claim review suggestions	+8%	+3%	13
30 seconds	Batch tagging	+3%	+1%	5

Task-Dependent Model Behavior and Failure Modes

Latency targets can't be divorced from understanding how models behave in real contexts. Different tasks introduce different failure modes that affect responsiveness:

- **Automation:** In fully automated pipelines (e.g., fraud detection at **PayPal**), latency must be low enough to avoid blocking multi-step processes. Failure modes typically include outright system failure, false positives, or bottlenecks, requiring quick fallbacks or kill switches.
- **Augmentation:** When AI augments user decision-making (e.g., customer support workflows), slight delays may be acceptable if they improve accuracy or confidence. UI states to gently indicate loading versus failure are essential.
- **Innovation:** Novel use cases (like generative design tools) may redefine user expectations around latency altogether, accepting longer waits if the value derived is significant or unique.

Each pattern demands distinct latency tolerances, and your evals should explicitly capture both speed and failure tolerance criteria to inform tradeoffs during model and infrastructure selection.

Automation vs Augmentation vs Innovation: Patterns for Latency

Understanding your AI product's role in the user workflow clarifies latency goals:

1. **Automation:** Tasks previously manual, automated end-to-end. *Latency target:* Near-real-time or bounded batch (milliseconds to seconds).



Example: Fraud scoring at PayPal — latency must fit within a 1-3 second window to not delay transactions.

2. **Augmentation:** AI assists human decision-makers. *Latency target:* Seconds to low tens of seconds, matching natural cognitive task pacing. *Example:* AI-assisted underwriting at H2O.ai where outputs supplement, not replace, expert review.
3. **Innovation:** New capabilities or exploratory workflows. *Latency target:* Flexible — user patience depends on user value. *Example:* Generative content tooling with latency spanning 10s of seconds to minutes tolerated due to novel benefits.

This classification helps tailor your *ship/no-ship thresholds* — a personal checklist I keep — ensuring latency commitments align with product success criteria and user acceptance.

How PM Toolkit Tools Help Set Latency Targets

PM Toolkit — widely used by leading product teams — offers calculators for TAM/SAM/SOM, ROI, churn, retention, RICE, ICE, Kano, and velocity metrics that help product leads quantify the business impact of latency improvements. Some ways to use their tools in latency planning include:

- **RICE/ICE Prioritization:** Quantify impact of latency improvements to identify high-leverage engineering investments.
- **Business Case Calculator:** Model how latency-induced frustration correlates with churn or reduced MRR/ARR.
- **Eval Spreadsheets:** Capture how latency variations change user satisfaction (NPS), task completion rates, and activation rates.
- **Switch Metrics:** Track DAU/MAU, conversion, and velocity metrics to detect when latency breaches cause productivity drag.

Grounding latency tradeoffs in these tangible business and user metrics beats abstract model benchmark chases and supports better cross-functional alignment.

Final Thoughts: Your Latency Target Is a User-Centered Commitment

The practical latency target for “first useful output” is not a fixed number; it’s a function of:

- User workflows and rhythms (what do users do today?)
- Task automation, augmentation, or innovation pattern
- Model behavior and failure characteristics
- Business outcomes such as retention, churn, and conversion
- Robust pre-release evals encapsulated in user story-based spreadsheets

To achieve real impact, product teams must always treat *the model as a tool embedded into the whole product* and never an end goal itself. Leveraging PM Toolkit calculators and defining evals with a workflow-first mindset lets you calibrate latency targets that truly delight users and move business needles.

In the words of AI product leads at companies like **PayPal** and **H2O.ai**, it’s not about relentlessly chasing milliseconds—it’s about weaving latency commitments into trustworthy, delightful user experiences that transform workflows for good.