

Adding an augmentation feature to your B2B SaaS product can unlock significant value, but it's not as simple as just plugging in a new AI model. Over my 12 years in SaaS product management — and specifically shipping AI-assisted workflows — I've learned that the model is not the product. Without the right evals, workflows, and user controls in place, even the most advanced algorithms fall short of improving actual user outcomes.

In this post, I'll share a quarter-by-quarter, hands-on roadmap for product teams aiming to add a solid augmentation feature. I'll reference practical tools such as the **PM Toolkit calculators** used by product managers at companies like *PayPal* and *H2O.ai*, and explain the critical concept that defining an **eval + kill switch** is as important as model accuracy. Throughout, I'll emphasize that understanding your **top workflow**, modeling tasks specifically, and giving users control are non-negotiable foundations for success.

Quarterly Roadmap to Ship a Successful Augmentation Feature

Step 1: Understand the User's Current Workflow (Weeks 1-3)

Before you even look under the hood of the AI model, start by mapping the user journey in your top workflow where augmentation could create impact. Here's what you should focus on:

- **Identify the task:** What specific problem is the user trying to solve? What are the steps they take today, and what frustrates them?
- **Quantify impact opportunities:** Use PM Toolkit calculators for *TAM/SAM/SOM* to understand the market size, and *ROI* calculators and *business case* models to justify investment.
- **Map performance levers:** Look at metrics like *DAU/MAU*, *activation rate*, *conversion rate*, and *retention*.

For example, PayPal's product teams rigorously analyze transaction flows to identify where small automation or guidance nudges could reduce friction or errors. At H2O.ai, product managers run Kano analyses to prioritize features that blend innovation with ease of adoption.

Step 2: Define an Eval Before You Build (Weeks 4-6)

Far too often teams rush to select a model before defining success criteria. But your product's augmentation feature isn't just about model numbers — it's about measurable impact in your user workflow.

Use lightweight evals — I prefer simple spreadsheets for early examples — to codify user stories that the model must support. These become your acceptance tests, defining precisely when you would **ship vs no-ship**.

- Write examples as *user stories*, not lab prompts.
- Define task-dependent behavior expectations and failure modes explicitly.
- Estimate sample sizes for A/B testing, and run early feasibility metrics such as *NPS* and *PMF scores*.

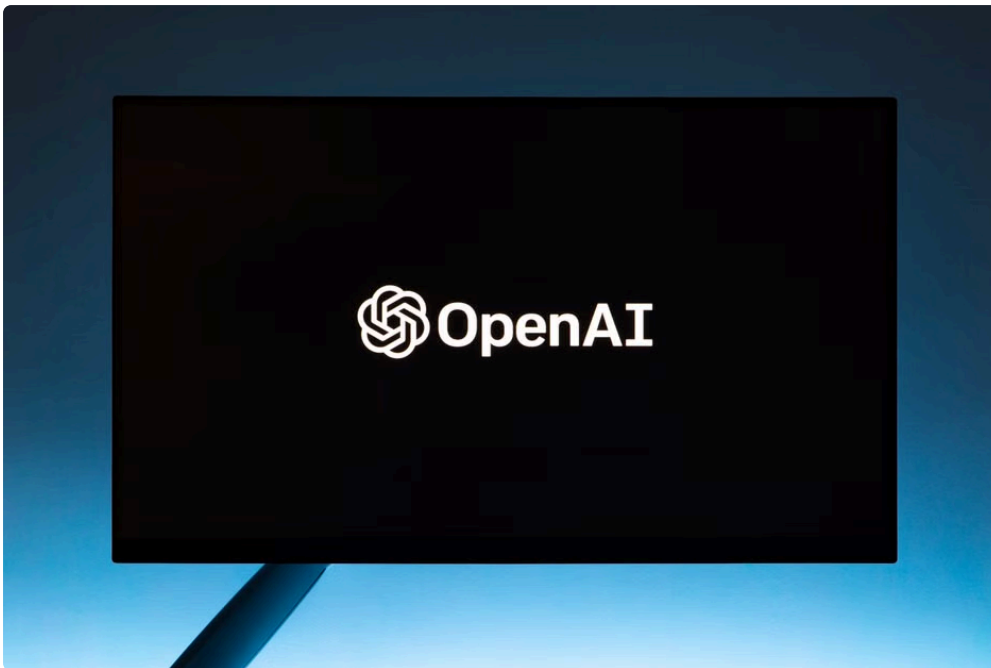
This step is crucial. If the augmentation feature cannot clearly demonstrate value on these evals, you should pause before pmtoolkit.ai building. Tools from the PM Toolkit like *RICE*, *ICE*, and impact/effort matrices help prioritize which augmentation ideas are worth pursuing.

Step 3: Build with Automation vs Augmentation vs Innovation Patterns in Mind (Weeks 7-10)

Not all AI enhancements are equal. Keep these patterns front and center:

1. **Automation:** Replace manual tasks end-to-end. Great for bottlenecks but high risk if failure modes aren't guarded.
2. **Augmentation:** Assist the user with suggestions, recommendations, or auto-completions – user keeps the final control. This is typically lower risk and higher trust.
3. **Innovation:** Create entirely new capabilities impossible before AI. These require robust user education and change management.

Augmentation features should always preserve **user control**. For instance, PayPal's fraud detection augmentation alerts but never makes final blocking decisions without human override. Similarly, augmentation in H2O.ai tools helps data scientists with feature suggestions but lets them choose final selections.



Step 4: Implement an Eval + Kill Switch (Weeks 11-12)

This is my mandatory personal checklist item for any AI-powered release:



- Embed real-time evaluations into your deployment — monitor task-specific success and error rates.

- Define concrete kill-switch criteria before rollout — if model behavior drifts, or error rate exceeds threshold, disable the augmentation immediately.
- Ensure users can manually override or disable augmentation features seamlessly.

Remember, benchmarks like “accuracy improved by 5%” mean nothing without context about failure modes in your actual product usage. A kill switch is your emergency brake and a trust anchor for users.

Step 5: Measure and Iterate (Weeks 13-14+)

After release, focus closely on the metrics that matter for augmentation:

- User adoption of the augmentation feature
- Changes in *cycle time* and *velocity* of the workflow
- Impact on *conversion rates*, *retention*, and *churn*
- Customer feedback via *NPS* and qualitative channels

Use A/B testing frameworks and the PM Toolkit’s calculators to measure both expected and unexpected effects. The augmentation feature should be a leap forward in your product’s value, not a black box experiment that confuses users or adds friction.

Key Takeaways: Avoid Common Pitfalls

Problem	Recommended Fix	Why It Matters
Focusing on model accuracy metrics without task context	Define task-specific evals and user stories first	Models need to prove value in real workflows
No user control over augmentation behavior	Always allow manual override and transparency	Builds trust, prevents catastrophic errors
Skipping kill switches and automated monitoring	Implement eval + kill switch early	Essential for safe and responsible rollout
Confusing automation with augmentation or innovation	Be clear which pattern fits your use case	Sets realistic expectations and design goals
Ignoring market sizing and business case	Use PM Toolkit TAM/SAM/SOM and ROI calculators upfront	Ensures strategic alignment and resource prioritization

How PM Toolkit and Companies Like PayPal and H2O.ai Inspire Best Practices

Throughout this process, the PM Toolkit’s comprehensive suite of calculators is invaluable. Whether calculating market potential (TAM/SAM/SOM), project prioritization (RICE, ICE, Kano), or success measurement (MRR/ARR, LTV, CAC), these tools help you make evidence-driven decisions rather than guesswork. Teams at *PayPal* rely heavily on these frameworks to ground their AI features in user problems and financial impact. Similarly, *H2O.ai*’s disciplined approach to augmentation emphasizes clear evals, task-dependent model assessments, and maintaining user control — principles you can emulate.

For early-stage evaluation, I’ve long favored spreadsheet-based eval suites that capture real user scenarios before even training models. This approach allows rapid iteration on acceptance criteria and reveals failure modes faster than black-box metrics.

Conclusion

To add an augmentation feature this quarter and make it genuinely useful and safe, start with a deep understanding of your users’ top workflow, build strong task-dependent evals, and prioritize user control with fail-safe kill switches. Balance automation and innovation with augmentation to meet user needs effectively.

Leverage PM Toolkit calculators throughout to ground your prioritization and measure impact. And follow the best practice examples set by leaders like PayPal and H2O.ai who integrate augmentation features with clear evals and user trust mechanisms.

With this approach, your augmentation feature won't just be an AI backend; it will be a powerful, trusted extension of your product that moves the needle for your users.