Metrics Framework Builder

An AI powered tool that generates rigorous, product specific metrics frameworks

Meghna Bhagwat | Product Manager | LinkedIn | Live tool: Click here

What I built

Choosing the right metrics is one of the hardest things a PM does. Most teams default to vanity metrics because they are easy to measure and easy to celebrate. But they rarely tell you whether users are actually getting value.

I built a browser-based tool that generates a complete, product specific metrics framework for any product. You enter six inputs describing your product context and the tool uses Claude to return a structured framework covering your north star metric, three leading input metrics ranked by criticality, a guardrail metric, a secondary success metric for edge case users, and a metric constellation that ties everything together.

The tool is live at https://metrics-framework-builder.meghna.app/ and the full source is on GitHub.

Why I built it

Metrics was an area where I knew I had a gap. I could explain what a north star metric was. But when asked to define one for a product I had never seen before, I would often default to the obvious, active metric rather than the one that actually captured value delivered.

I studied three core concepts deliberately: north star metrics and why activity metrics mislead, the metrics tree and how input metrics connect to and predict the north star, and guardrail metrics and how they prevent teams from optimizing one number at the expense of real value.

Then I built a tool that forces me to apply all three every time I use it.

How it works

Input

The tool requires six inputs. Each one captures a dimension of product context that changes which metrics are meaningful. Stage matters because an early-stage product optimizing for retention is optimizing for the wrong thing. Business model matters because a free tool measures different values than a subscription platform.

Input	Purpose
Product name	Anchors the metrics framework to a specific product.
One sentence description	Defines the core value.
Primary user	Role and context and not a persona.
Product stage	Early, Growth or Mature. This changes which metrics matter most.
Business model	Free, Subscription, Platform or Enterprise.
Problem it solves	What the user struggles with if this product did not exist.

Output

The framework Claude generates covers five layers, each serving a distinct purpose in telling the complete story of product health.

North Star	Input Metrics	Guardrail	Secondary	Constellation
Value delivered to user	Three leading indicators ranked by criticality	Prevents gaming the north star	Captures edge case users	How all five tell the complete story

What I learned

The prompts do not just generate a framework. They force you to be precise about what you are building before you get the answer. You cannot get a meaningful north star without first being clear about who your user is, what problem they have, and what value delivery actually looks like for them.

The vanity metric contrast was the most valuable part to build. Take a project management tool. The obvious metric is ‘tasks created’. It goes up every time someone uses the product. But tasks created does not tell you whether work actually got done. ‘Tasks completed by their due date’ is harder to measure but it captures whether the product is delivering real value. Building the tool required being precise about that distinction for every product type, which sharpened my own instincts in the process.

How I tested it

I tested the tool on 8 products, 4 well known ones (Stripe Connect, Asana, Strava, Etsy), 2 of my own product ideas, and 2 from testers. Across every run, the North Star consistently landed on a value metric, never an activity metric, this held for all 8.

The exact wording varied between runs of the same product with the same inputs. Running Stripe Connect twice returned "platforms with a connected account reaching first successful payout, per platform, cumulative" in one run, and "platforms reaching first successful payout to a connected seller within 30 days of integration start" in another. Both are genuinely value metrics, but they're not the same metric, one is an all-time count, the other is time-boxed. The same pattern showed up in the input, guardrail, and secondary metrics too.

Four PMs from my network tested the tool directly, checking two things, did it return a North Star at all, which it always did, and would they trust or use the output as-is. They noticed the same wording inconsistency. What they consistently valued was that the framework always centered value over activity, and listing the vanity metric to avoid alongside the value metric made that distinction concrete rather than abstract. Their overall take, the tool needs more consistency across repeated runs to fully trust as-is, but the metrics it generates are good each time, and it's genuinely useful for showing how easy it is to default to an activity metric without something forcing the value framing.

After identifying this, I added caching so identical inputs return the same result instantly instead of generating fresh each time. This fixes exact repeats, but it doesn't fix the deeper issue, a slightly reworded input describing the same product can still return differently worded metrics, since that reflects variance in the model itself, not just redundant computation.

Conclusion

I used to start with the metric that was easiest to measure and work backwards to justify it. Now I start with the question ‘what does it look like when this user gets real value?’ The metric comes from that answer, not the other way around. Getting that judgment right consistently, every time, in the exact same words, is the part I'm still refining.