

Turning a design bottleneck into an organizational capability.

When a recurring workflow problem was draining design capacity, I treated it as a build-vs-buy decision rather than a request queue to manage. The result was an internal AI-powered tool that shifted branded-graphics production from a designer service desk to org-wide self-serve, while protecting a licensing constraint no external vendor could meet.

3 days

The cost of validating the solution before asking for any resources **16**

Internal users

Adopted across UX and Graphics on merit

1–3 hours saved

Per request, across up to 6 daily requests per designer

THE BUSINESS PROBLEM

The design team was operating as a production service desk. Stakeholders across the organization needed branded graphics built on licensed Getty imagery, and each request consumed one to three hours of designer time, often more with feedback rounds. At up to six requests per designer per day, a meaningful share of design capacity was going to production work instead of design work. The cost wasn't just hours. It was what those hours displaced.

There was also a risk dimension. Every workaround teams might reach for, including external AI tools, created licensing exposure. Image rights are non-negotiable in an enterprise context, and the existing process was the only thing standing between the org and compliance risk.

THE DECISION: BUY VS BUILD

I evaluated the market first. External AI tools failed the core constraint: they wouldn't honor the licensed Getty source images, either ignoring them or generating new people entirely, which is exactly the legal exposure we needed to avoid. Getty's own AI product had the same gap. It generates new imagery; it doesn't compose branded graphics from licensed material a company already owns.

That evaluation is what justified building. Not preference, not capability for its own sake: the market couldn't meet a constraint we couldn't move.

THE STRATEGIC CALLS

1 • Governance first: designers own the system, the org gets speed.

The tool operates only on uploaded, owned Getty images. Designers configure the layouts, brand palettes (Gallup and client-specific), and reusable assets that live inside it. Everyone else self-serves within those guardrails. That separation, brand control centralized and production distributed, is what lets the capability scale without scaling headcount.

2 • Compliance as architecture, not policy.

Rather than telling teams which tools not to use, I built the constraint into the tool itself. It cannot generate unlicensed imagery, because it only composes from the owned library. The safest process is the one where the wrong move isn't possible.

3 • One product, two altitudes of user.

Casual users generate, adjust, and download a PNG. Advanced users get creative control and SVG export to continue in their own tools. Designed into the file architecture from day one, so neither audience compromises the other.

4 • Ownership economics.

Building internally avoided recurring seat and API costs for external tools that failed our constraint anyway, and it means we own the roadmap. The next capability we need is a build decision, not a vendor negotiation.

5 • Validate cheap, then resource deliberately.

I built the MVP in three days, which made the validation nearly free. Adoption across UX and Graphics proved the value. Now I'm writing the spec for the next phase, deeper Adobe integration and an AI layer constrained to the owned library, and handing implementation to engineering. My job in phase two is focused on direction.

WHERE IT'S HEADED

The current work with the Graphics team is about turning a designer-facing tool into an organizational capability: pre-built layouts for the teams who request most often, so the org self-serves and design capacity goes back to design. That's the actual outcome being managed, not the tool's feature list.

THE PATTERN UNDERNEATH IT

Evaluate the market before building. Protect the constraint that can't move. Validate at the lowest possible cost. Institutionalize what works, then resource the next phase deliberately. That's how I approach product decisions at any scale, from founder work to capability-building inside large organizations.