

SAHIL SAPRA

Charlotte, NC 28273 | (216) 848-7236 | sahilsapra391@gmail.com | sahilsapra.com | [LinkedIn](#) | [GitHub](#)

PRODUCT MANAGER | AI & AUTOMATION FOR AGENT WORKFLOWS AT SCALE

August 20, 2026

[Affirm Hiring Team](#)

I am applying for the Senior Product Manager, Agent Enablement role. The line in your posting about the internal tools agents rely on is the one I keep re-reading, because the internal tooling rollout I am proudest of is also the one I got badly wrong the first time.

At CAMP Systems I drove org-wide adoption of LaunchDarkly to decouple deployment from release, and feature release frequency went up 400%. What the number hides is that the first rollout stalled. I had designed it around my own mental model of the workflow, and the feedback I got was blunt: I had shipped my tool, not their tool. So I sat and watched engineers actually try the task, rebuilt it around the friction I saw rather than the friction I assumed, and adoption followed. Then I noticed the vendor dashboard was flag-centric while our workflow was release-centric and the seats were expensive, so I built a JavaScript web application on their API in Cursor for centralized flag monitoring and release control. Seats consolidated and team OPEX dropped 7%. Nobody asked for either piece.

Designing for scale through automation is the other half of what you are describing. Onboarding one enterprise customer onto TFBO Next-Gen took over two weeks of manual ETL, and I owned a 75-customer migration program, which sequentially is roughly three years of work. Rather than asking for headcount I drove automation of the data injection ETL and took capacity to three customers per day. Alongside that I ran a feature parity program through influence without authority over engineering, translating workflow gaps into prioritized requirements and migration sequencing, because some customers had gaps that would have broken live aviation operations if we moved them on schedule. All 75 moved with zero operational downtime and zero churn on a platform change they did not choose, and ARR grew 104% year over year two years running.

On AI, I build rather than review. I founded SpecHawk, a deploy-aware no-code agentic QA platform where the agent maps a staging environment, generates and executes test cases, validates Jira tickets and returns screenshot evidence. I grounded its test case generation in retrieval-augmented generation over roughly 2,000 of our own ticket-to-test conversions, so output followed house conventions and improved with each added example instead of requiring retraining. Four enterprise design partners, weekly loops measured on real scheduled usage rather than demo enthusiasm, and \$100K raised pre-seed on a SAFE. I can argue the tradeoffs of embeddings and prompt iteration with engineers and then explain the same decision to a leadership team without changing the facts.

Where I am thinner is servicing itself. I have not owned a support organization, chat or phone channels, or the deflection and handle-time metrics that go with them, and I would be learning that vocabulary from your Servicing and Operations partners rather than bringing it. What I would bring on day one is the internal tooling half: the habit of watching a real agent attempt a real task before shipping anything, and the automation instinct that treats an impossible workload as a systems problem rather than a staffing request.

Sincerely,
[Sahil Sapra](#)