

# SAHIL SAPRA

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

August 20, 2026

[Anthropic Hiring Team](#)

Re: Technical Program Manager, Revenue Operations

The line in this posting about a change having a process side and a systems side that rarely stay separate is the most accurate description of my last two years that I have read in a job description. At CAMP Systems I owned a program to move 75 enterprise customers off competing platforms onto a next-generation one, and it was never a systems project or a process project. Migrating a customer touched the data model, downstream reporting, and how operations teams spent their week, so I mapped the whole chain, sequenced the work across engineering, QA, and go-to-market, and translated workflow gaps into prioritized requirements, acceptance criteria and a migration order engineering could build against. When I found gaps that would have broken live operations for specific customers, I said so to leadership and to those customers before anyone asked and re-sequenced around them. All 75 moved with zero operational downtime and zero churn on a forced platform change, and ARR grew 104% year over year across two consecutive years.

Finding where people lose time and quantifying it is the part of this I would do whether or not it was in the job description. Onboarding one customer onto that platform took over two weeks of manual work. Run sequentially across 75 customers that is roughly three years, so the program as scoped was arithmetically impossible. Rather than ask for headcount I walked the onboarding process end to end, found the data injection step doing the damage, and drove the automation that took capacity from over two weeks per customer to three customers a day. The same reflex produced a LaunchDarkly API-based web application I built when the vendor's flag-centric view did not match how we actually released, which consolidated seats and cut team OPEX 7%.

Making a change land with the people who have to live with it is a separate skill from designing it, and I learned that the hard way. I drove org-wide adoption of LaunchDarkly, decoupling deployment from release and lifting feature release frequency 400%, but the first rollout stalled. I had built it around my own mental model, and the honest feedback was that I had shipped my tool rather than theirs. I watched people attempt the real task, rebuilt around the friction I saw, and adoption followed. I have not put an internal system in front of a team since without doing that first. All of it ran on influence rather than authority, which is the only way that work gets done when the teams involved report elsewhere.

Anthropic specifically, because the AI work has been my own hands rather than a spec I handed off: I grounded a test case generator in retrieval-augmented generation over roughly 2,000 of our own ticket-to-test conversions so output followed house conventions without retraining, and I founded SpecHawk, an agentic QA platform, with four enterprise design partners and pre-seed capital raised. Two gaps stated plainly rather than left for you to find. I have not worked inside a Salesforce or GTM systems estate, so quoting, forecasting and territory work would be new ground I would be learning on your stack, and my four years sit below what a role at this level usually assumes. What I bring on day one is program plans people can actually execute, requirements that translate cleanly in both directions between business stakeholders and the engineers who build the systems, Snowflake SQL and PowerBI in daily use so I check where a program stands myself, and a habit of watching a real user attempt the real task before calling anything done. I am authorized to work in the United States and happy to meet the in-office expectation. Thank you for your consideration.

Sincerely,

[Sahil Sapra](#)