

SAHIL SAPRA

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

TECHNICAL PRODUCT MANAGER | DEVELOPER TOOLING & AI AGENT PRODUCTS

August 19, 2026

[Xsolla Hiring Team](#)

I am applying for the Product Owner role on Shop Builder. The product is the layer other developers build on, and the roadmap runs toward agents that stand up a working storefront from a prompt. Both halves of that are what I actually do. I have spent the last two years on API design and integrations with engineering, and the last year building an agentic product where the agents have to work on somebody else's real system, not a demo.

At SpecHawk I built an AI-powered agentic QA platform: agents map a staging environment, generate and execute test cases, validate Jira tickets and attach screenshot evidence. I scoped the MVP to the smallest slice that could produce a test run a design partner would trust on their own application, then ran weekly loops with four enterprise design partners and their developers, measuring scheduled usage rather than demo enthusiasm. V2 and V3 added a developer CLI, GitHub integration, Security Hub for dependency risk analysis, and a credit-based pricing architecture. I write the code as well as the specs, in TypeScript and Python, which is mostly why the API conversations go faster.

The API and sequencing work came from CAMP Systems. Enterprise customers were timing out on a legacy monolith and the consensus fix was more infrastructure. I traced request-level timing, found synchronous retrieval paths and no caching, and led the move to domain-driven services in Java, Spring Boot, TypeScript and Angular with reusable RESTful APIs, caching and lazy loading. Data retrieval latency dropped 68% and the timeouts stopped. Alongside that I owned roadmap sequencing for a feature parity program that moved 75 enterprise customers onto TFBO Next-Gen with zero operational downtime, which was almost entirely a question of what shipped in what order and being able to defend it to customers who did not want to move at all.

One lesson I would bring in on day one. When I rolled LaunchDarkly out across the org, the first attempt stalled because I had built it around my own mental model of the workflow instead of the engineers' workflow. I watched real users attempt the real task, rebuilt around where they got stuck, and adoption followed; release frequency ended up 400% higher. That is the same failure mode a documentation and developer experience surface has, and it is why I treat docs as part of shipping rather than as follow-up work.

Where I am thinner: I have not owned a storefront, payments or games platform, and my API work has been first-party rather than a published surface with outside partners and agencies depending on it. I would want to talk about how you draw the line today between what a publisher can wire up themselves and what stays managed, and about the posted salary band, which reads low for this scope and is worth clearing up early.

Sincerely,
[Sahil Sapra](#)