Maybe going around CIRCLES is worth it

Varun Choraria · 2026-06-26 · https://www.varunchoraria.com/maybe-going-around-circles-is-worth-it/

Frameworks get a bad rep. But early in your problem-solving journey, they're scaffolding — not a crutch. Over time, as you get acquainted with more real-life scenarios, your intuition gets powerful enough to spot-check and work bottoms-up.

For the purpose of this field note, let's stick to some 101s.

CIRCLES is a helpful framework to bring a rounded thinking hat to the table and actually ask good, focused questions. I first read this in cracking the PM interview.

	Step	Description
C	comprehend the situation	clarify what's within scope and out of scope for the problem statement. eg: business goal, constraints
I	identify the customer	figure out how to segment the customers. low hanging fruit will always be the biggest customer segment with the largest gap. list down firmographics, demographics, technographics - get your fact sheet in place about the customer's day in the life.
R	report the customer's needs	list all their needs (not wants) in the order of priority. I wrote a helpful mental model here.
C	cut through prioritization	narrow the problem surface. from the prioritized needs, pick the highest-impact, most-urgent ones to address — don't try to solve everything at once.
L	list down solutions	generate 3-5 solutions per prioritized need
E	evaluate tradeoffs	this needs to happen across 3 axes - implementation complexity, user impact, cost / benefit
S	summarize	this is where you need to COMMIT. it's fair to be opinionated here, because you need to pick a solution and defend it. In the age of agentic AI, we're all compensated for context and judgement.

The same thing works in RCAs

1. A costly lesson I learnt in my career: check the damn dashboard itself for telemetry errors, sanity of the data lake and how a metric is computed. You'd be surprised how often that occurs. Plumbing issues like hygiene and data pipeline breakages often misreport signals.
2. Second, check for seasonality. In B2C, holiday season has an impact on business. In B2B, it's a factor of sales cycle length, targeting the right buying committee, competitor takedowns etc (it is complex).

3. Another dumb error I've made in the past is not to check for things that might've gone wrong on our end in production. Was there a server downtime? Did API response times fall? Was the customer's plan not grandfathered after a price change?
4. Segment the surface area that was affected: check traffic sources, user segments like mobile vs. desktop, feature adoption data, past usage history, past support tickets etc.
5. Most of the above 4 pointers are internal-facing. Externally - you can look at things like competitor launches, industry trend causing the dip, customer awareness about your product / features (enablement).

Other helpful things you can do:

1. Setup observability. There's options like new relic (freemium); signoz (open-source). Or, you can also vibe-code a dashboard or even build one on google sheets and have that updated real-time via a tool like coefficient. You can then hook it up to an LLM and run a /routine or cloud agent where it pings you daily updates on slack.
2. For the love of god, track conversion funnels on both google analytics as well as your internal product analytics platform like heap.
3. You can also setup an RSS or a custom competitor scraping agent via firecrawl or context.dev to scrape product pages, newsroom / PR, product hunt, app exchange listings etc via an LLM and connect it to slack for real-time updates. Costs super less.

I've done both - for PLG as well as sales-led motions. For PLG, I was able to figure out positioning and plays to target customers in-product for plan upgrades and upsell motions based on their usage patterns, and verifying that with regional sales folks to understand what works.

For sales-led motions, I was able to identify if an increase in traffic = ICP vs. non-ICP visits via SwanAI led to more demo form fills or not. SwanAI helped us profile, enrich and classify web visitors and ping us on slack with an outreach drafted. It was a good way for us to validate our SEO / AEO bets. Additionally, having access to data helped us reverse-engineer our ideal deal / account and customer profile based on past-won deals data to better target them for future pipeline.

Direction still needs judgment

The premise is simple: being insight-led vs. data-driven is a lot better because it helps you drive cross-functional influence and evaluate tradeoffs better. Everyone talks about velocity, few understand it. Velocity is speed + direction. Speed is mostly solved for by AI. Direction is still very judgment and context-heavy, as most insights still need to pass the human smell-test.

Being insight-led also helps you ruthlessly prioritize closest to revenue tasks. I've said this before too organizations are now becoming flatter by the day, and so every project invariably becomes cross-functional because skill gap has collapsed due to AI.

The playing field is quite leveled.