Why Slight Failed: A Slight Post-Mortem

My best guesses as to why our early stage data startup failed. This isn't a retrospective, but something closer to advice for past me. Or maybe just exorcism.

MON OCT 28 2024

BACK in late 2020, my good friend Raiden and I co-founded Slight. Over the next two and a half years, we got it off the ground, raised a pre-seed round, and got some customers on-board, but in the end we didn't have enough traction to raise our seed round. We got close with a few investors (after I talked to many…), but getting close still means the company dies.

pre-seed ⟹ seed

I wish I could point to some obvious fuck-up, so in lieu of that I'll give my best guesses.

Before we dive in — Slight was a data "switchboard" that turned SQL queries into user-friendly interfaces for anyone who needed data. Click the details below for more.

▶ **Details** | More info on what the product was

Did We Make Something People Want?

I could answer this with a million "Yes, but" paragraphs. Which means the answer is no. Or if I'm being generous: not enough.

~~Yes, but~~ Large companies didn't want it enough to deal with our lack of "big company features" (enterprise SSO, compliance certifications, support for legacy databases, or just the long tail of databases). They saw value, but not enough to justify the internal friction.

well put

~~Yes, but~~ Startups didn't want it enough, because we were a solution to organizational friction they hadn't hit or just wasn't top of mind. Slacking around SQL snippets, schlepping around CSVs, devs having minor inconsistencies when running a bunch of ad-hoc queries: problems for sure, but not major ones for a small company.

~~Yes, but~~ Not enough in a crowded space. ~~Yes, but~~ Not enough in tough start-up environment. ~~Yes, but~~ Not enough to deal with UX papercuts.

not enough

We were nearly in no-man's land. Our ideal customer was probably a fast-growing Series B company hitting these pain points for the first time — too narrow a target to systematically find before they settle for good-enough alternatives.

series B fast growing being the ideal customer isn't great

Maybe we were just a few needed tweaks or features away. Maybe we got the distribution wrong (pretty important in B2B). Maybe we just needed another six months. But at some point you have to admit you either didn't build something people wanted enough, or at least not enough to buy time to figure out the rest.

so true

Distribution

→ most people do

I think this is where I most clearly dropped the ball. We never fully solved a very important question: **how will your product go from "cool software" etc. to actually being in people's hands?** I don't just mean a demo someone can play with, I'm assuming here your product is ready enough to onboard customers. What I mean is: what's the end-to-end process for a company to get to a place where they're using your product in a serious way?

the real use case

I know I got this badly wrong, because in truth I don't even know the answer now. If you squint, we looked a little bit like a BI tool, so we mostly followed the BI distribution playbook. This was a mistake: we intentionally avoided some "core" BI features, e.g. dashboards, and so companies couldn't replace their BI tool with Slight. That would be fine, but we didn't really offer a vision of how to go from pre-Slight to using Slight in production. No clear wedge, no very obvious first use case, no natural expansion path. I didn't properly understand which stakeholder needed to see what value.

BI Tool
⇒ BI distribution

need a vision
1. Your world before using a certain tool
2. Your world after using a certain tool.

Building a solution isn't enough – it needs an obvious path forward

Gotta fix this

When someone asked "how do we get started?", we had a technical answer ("connect your database, write some queries, data for all!") but no story about which team should champion it first, or which problem to solve first. Data teams? Product teams? Analysts? We had some answers, but not *the* answer. We had pitches for individual teams that worked well, but we never nailed down the way companies should adopt Slight.

Instructions!
1. Connect DB
2. Write queries
3. Data for all

Need a story
- which team?
- which problem?
- which product?

If you don't know how your product will be adopted, how would your customers know?

When things were sort of but not quite working, I made the stupid mistake of just working harder and harder to on-board companies. Instead, we should have sat down and mapped out ways to properly experiment with our approach. Maybe simplifying to a single clear use-case, or finding a completely different initial wedge, or focusing on specific verticals. Near the end I actually did try some of this (specifically around sharing data with customers/externally), but made a hilarious error: I focused the changes on my fundraising efforts instead of actual customer acquisition.

Mistake:
- on-boarding new users w/ half-sold product
- Fundraising

Learning:
- One use case
- Different wedge
- Focus
- vertical
- customer acquisition

OPEN SOURCE?

Slight could have been open source. One analogy for what we were making is "dbt, but on the other side of the data warehouse": dbt gets you from raw data to nice datasets, while Slight is a data switchboard that routes datasets to all types of users (technical or not, internal and external).

Open source might have been a better path to adoption — particularly since we were building (data-)engineer-focused infrastructure. But it would have meant a completely different business and go-to-market strategy.

I expect something like Slight will eventually exist as an open source tool. The concept is solid — we're already seeing many data tools expand into this space. After the company shut down, I initially planned to open source Slight, but the reality of post-startup life (including a baby no less!) meant this didn't happen.

startup + baby

> **Note** | Is It Too Late?
>
> No... Maybe during a future sabbatical. But don't bet on it.

Money...

I think at this stage, there's enough written about this that many people have been disabused of the notion that "oh shucks, we just happened to run out of money" is a standard risk for startups. It's a symptom.

> *The #1 cause of startup death is making something no one wants. The #2 cause is spending too much. Those two account for so many deaths that I'm not even sure what #3 is.*
>
> *If you merely make something people want and don't spend too much, you're way ahead.* — @paulg

1. make something no one wants
2. spending too much

Whenever I tell people about Slight, they always mention what a dreadful market it was to raise into. This is 100% true: at the time of raising (early 2023) the market was a disaster, and the limited amount of money in play was chasing AI. But the market is what it is, and that's exactly why you need runway: to survive tough conditions. We did not!

> *"Well, It Was a Tough Market!"*

The market downturn also meant companies were cutting SaaS spend and consolidating tools (very reasonably). Slight was cheap to run, but that's not a great selling point.

no one cares about cheap
=> everyone will pay up for expensive

Blaming the market is cope. Yes, it was tough, but we just didn't have enough customers. Everything else is context.

More Guesswork: Quickfire

I doubt anything outside the above makes a huge difference. But you never know.

MOVING TOO SLOWLY?

→ the fact this wasn't an issue is 100%.

Eh, probably wasn't this.

⇒ thats cool

We had a great team, but they hadn't all worked in an early-stage startup before, and weren't necessarily used to "startup trade-offs" — e.g. between quick iterations and robust/polished features. The transition wasn't too hard though; everyone was very receptive to feedback and adapted quickly.

} that's amazing

Of course aligning the team is one of your roles as founder, but the hard part is coming up with the prioritisation to begin with, not so much in transferring it, especially when the team is small.

OVER-ENGINEERING?

→ the fact this wasn't an issue is 100%

We probably could have put more features on the backburner until after launch, to tighten the feedback loop.

⇒ classic

We spent a bunch of time on multi-tenant infrastructure when we could have just gone with single tenant for a long time. But this is mostly downstream of getting caught in no man's land with who we were trying to serve.

We also built a public version of Slight with hosted datasets and queries. I have mixed feelings about this: fantastic for feedback and hiring, but likely more effort than it was worth for customer acquisition.

Lesson:
customer acquisition
⇒ hiring
⇒ feedback

CO-FOUNDER OVERLAP?

Working with Raiden was fantastic, but we both have the same background: stats guys who went into data; solid engineers, but more specialist than generalist. This overlap was great during Slight's genesis, but it meant we felt compelled to hire both frontend and backend engineers very early on — maybe too early.

} problem

still worth it

Still, I'd do it again: having the right partner who you can trust matters way more than having the perfect skill mix.

COMPETITORS

We did OK here, by which I mean we weren't consumed by competitor focus (investors on the other hand...).

} honestly, they did everything right & just weren't lucky

However, many potential customers did bring up competitors they used, or ones they were considering — they essentially wanted a comparison. That's of course reasonable, they're trying to understand where your product fits in, and what their needs are. So it's not something you can afford to ignore, unless your product is solving such a unique problem that comparisons are meaningless.

Making a comparison just works

very unlikely

Was It All So Bad?

Oh not at all. Running a start-up was hard work, stressful, but rewarding. We did onboard some actual customers, so clearly we weren't completely wasting our time. There were plenty of things that went well: hiring (maybe a future post!), the team we built, our customer conversations and understanding of their problems, and technical execution.

Bam

retrospective != postmortem

But this post isn't a Slight retrospective. I'm trying to isolate what went wrong for a business that could execute well — a strong team, solid tech, and good customer engagement. Those weren't our problems.

We tried, it didn't work. If the right situation comes along, I'll probably try again.

Avoid the Absorbing Barrier

When startups fail, the problems they aimed to solve might still exist. Annoying!

I wonder why they couldn't raise again?

Startup founders will probably get a lot of shit advice along their journey, or advice that's not relevant or suitable for their type of startup. But that's a shite excuse for anything: we are all surely responsible for our own company's success.

so true

also true

Everyone who says to move fast is basically right. It's not about cutting corners, but about giving yourself as many chances as possible to learn, to tweak, to "pivot", to iterate your way to something people actually want.

this post is feeling amazing

Really know what you're selling. Then actually sell it.

TOPICS

SLIGHT **STARTUPS**

colmanhumphrey@gmail.com
colmanhumphrey