Product Build: Onboarding for Growth

When I was at Algorithmia, we had a basic issue: we didn't know much about our signups, which made it hard to help and direct them properly. I led a new feature build to achieve that goal – with a couple of iterations based on data and testing – that helped us increase sales efficiency and improve our user experience.

North Star and Vision

PRODUCT VISION

Algorithmia faced (and continues to face) a core information asymmetry: we don't know enough about our users, and our users don't know enough about us. In practice, that plays out like this:

- We don't know enough about our users: more information like company, role, and deployment goal will help us target and help the right potential customers
- Our users don't know enough about us: we had multiple product lines, but only one signup: it was totally unclear to users how to get the most value out of our platform

Our vision for onboarding: improve the user experience on our platform by asking for intent without hurting our signup numbers. Our major constraint was keeping our signup numbers healthy and growing.

USER NEEDS

Our end users are all developers, which introduces a unique set of product constraints (and even more marketing ones, but that's for a different time). Developers are notoriously conservative about giving out information like where they work and what their role is: and benchmarks show that asking for those fields can drop signups by 40% or more.

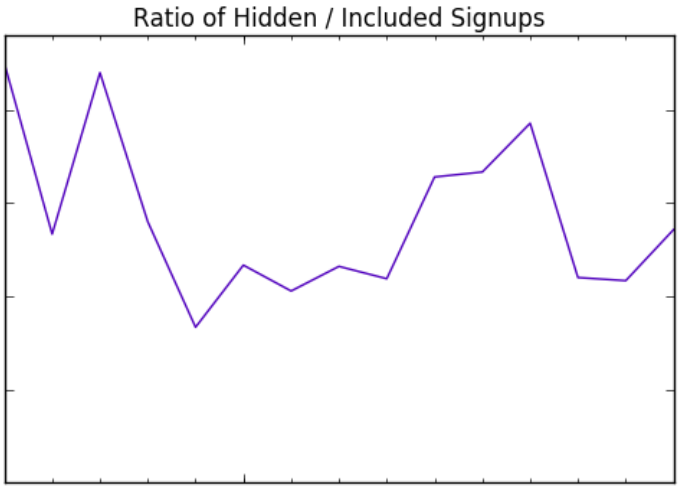
Phase 1: Testing

Before building a full new onboarding process, we wanted to understand the impact that information asks would have on our signups. In that vein, I worked together with our frontend team to design and run our company's first A/B test in a tightly scoped iteration.

The process was pretty straightforward:

1) Create a new signup screen that asks for "company" and "role" (both optional)
2) Add columns to our user database to record (a) version shown, and (b) answers
3) Randomly generate which version to show to a user, and use cookies to make sure they see the same version if they ever leave in the middle
4) Run the test and analyze results

We ran the test for a few weeks, and I did an analysis of our results in a Jupyter Notebook.



Our results ended being very similar to what we assumed: more than a 40% drop off in signups when users were asked for the extra information, *even though it was optional*. This experiment taught us a few things:

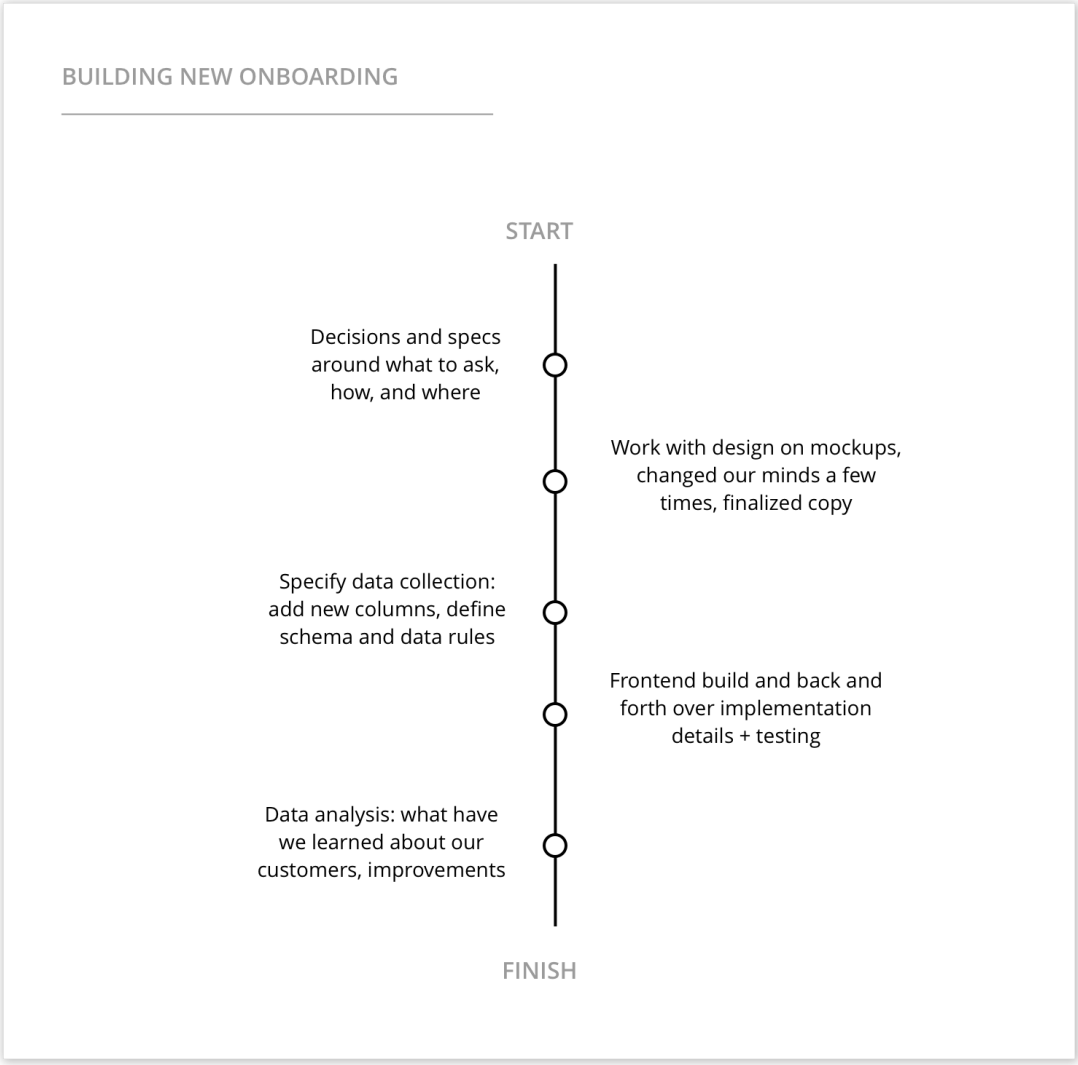
- We need to make any information asks *very, very optional*
- Information asks will likely hurt our signups, and that's a tradeoff we need to navigate

Phase 2: Main Product Build

My team put together a spec of exactly what we wanted this onboarding process to look like, and worked with design to create mockups that were navigable. Some of the questions we considered:

- Should this process happen on its own screen or as a modal?
- What exact information do we need to help users find what they're looking for?
- What order to we ask things in?

Here's what the process looked like:



Problems and Iteration

PROBLEM: SIGNUP DROP OFFS

The first problem we ran into wasn't unexpected, it was tested for: we quickly found that asking for information directly is kind of a dead end. To keep in line with our original vision, we made what I think was a clever adjustment:

- Keep the signup process the same, and ask for info after the fact
- Communicate to users *why* we're asking for information (show them the value proposition)

We built the process with these in mind, and ended up successfully avoiding major decreases in signups.

PROBLEM: DATA COLLECTION

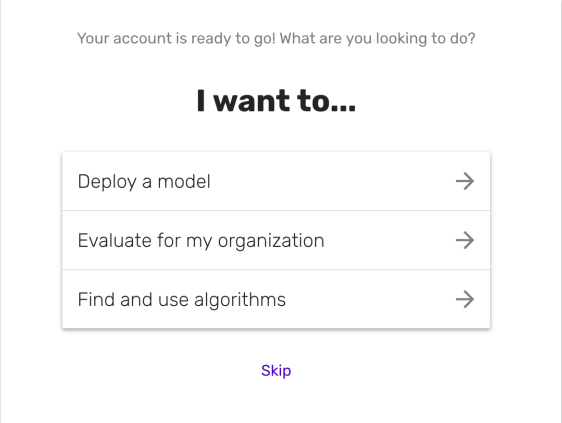
We used Segment for managing our customer data, and there were a bunch of internal constraints that made it tricky to store the data in the way that we wanted. We ended up designing each question as categorical, and storing the results as text in our data warehouse (while using Segment to send it to Hubspot).

PROBLEM: DIRECTION

One of the major goals we had was to direct users to the right part of our product: our marketplace if they wanted to use algorithms, and to publish if they wanted to deploy a model. But we had a major product backlog stuck on backend stuff, so there weren't clear dividers between parts of our product: we needed to improvise.

As a temporary solution, we were able to create landing pages tailored for the 3 use cases we gave as options. Those pages directed users towards how to accomplish their goals, and are acting as placeholders until the team can build better product landings.

Successes and Failures



Overall, we made concrete and positive steps towards our vision, but there's a lot more work to be done, both in building and in testing. The process increased the meaningful information that we had on customers by >5x, and almost 50% of signups now fill out at least part of this flow after the sign up.

My team brought in 3 new enterprise POCs through onboarding in the first few weeks: since we knew what company they were from and what they needed, we were able to pass them fast to sales.

It was very difficult for us to measure this feature's impact on users finding their way properly: we defined the metric in advance as API Calls, but as an enterprise company it's difficult to peg that complex outcome to a particular set of metrics.

The whole project got put on hold for resource reasons, but we were able to spec out the next phase of improvements:

- A new product landing UI that communicated our value and how to use the product in a much clearer way
- A series of deeper A/B tests for specific onboarding flows and what we ask them for
- Testing encouraging users to enter more information by providing free credits

I didn't get a chance to work on any of these iterations, but overall I think the project was successful!