**Build or Buy**

Companies looking to invest in their data infrastructure often find themselves debating whether to buy or build the relevant tooling.

Both options bring unique benefits along with potential drawbacks. Building gives companies complete control and flexibility, but the project can also quickly become a drain on employee time, energy, and department budgets. Buying, on the other hand, is convenient, but there’s often still an integration cost; moreover, the solution likely doesn’t even completely solve your problem.

Over the past few years, the proliferation of data infrastructure startups – data warehouses, data transformation and data observability tools – means that buyers increasingly have a wealth of great options to choose from. Many of these tools were first built as in-house solutions at Uber, Airbnb, or Netflix, open-sourced, and eventually spun out into stand-alone hosted products.

Bigeye’s recommendation is that data tools should generally be bought rather than built because the problems they are solving are not core to most businesses. Catching data outages, for instance, is a commoditized service; a fintech company gains no extra plaudits for doing it well.

However, each buy or build situation is unique, and it’s possible that your company, if it has unique data needs or requirements, *should* build. In this blog post, we’ll walk through the major considerations.

**Prework**

Mark Grover, a former data engineer at Lyft, suggests that prior to any cost/benefit evaluation, you should first determine how the company will measure if the implementation of the solution is successful.There are two metrics typically used:

* adoption – a certain number of people are using the product
* NPS – a certain number of people are loving the product

Now that you know what your goal is, you can take a look at what you have at your disposal to get there, whether that’s time (need this done by Q2), or engineers (have a team of 4 engineers). You can then proceed to the major considerations below. Buying and building offer different tradeoffs.

**Major Considerations**

1. Is the technology core to your business?

Is the technology something that is core to your business/critical to your business success? If not, go towards a buy approach. If, on the other hand, you’re a large business with special data needs or if what you’re building is relevant to what your company is selling, then go towards a build approach.

1. Does the philosophy of the tool fit the organization or does the organization have to adapt?

When buying a data tool off-the-shelf, it often comes with a certain philosophy, for instance, that all your data is going into a cloud data warehouse. If your data *isn’t* all going into a cloud data warehouse, then tools that run on top of data warehouses might not be a fit for you.

1. What are the ongoing costs?

Calculating initial costs is easy but successful cost-benefit analysis should also consider long term spending, including ongoing server and engineer costs.

1. Are you in a highly regulated industry, where vendors need to be vetted carefully?

In certain regulated industries like healthcare and finance, it may take 6 months for a bought product to get approval and pass security checks. In this case, it may not make sense for the org to remain bottlenecked on that purchase.

1. Is it onerous for you to be bound to a vendor's roadmap?

While vendors listen to their customers’ feedback, they likely won’t listen to your feedback alone. Any feature requests or changes that you make will have to be balanced against their own goals and promises they’ve made to other clients. On the other hand, if you build, you can build exactly the features and integrations that you need, and no more.

1. How easy is it to hire engineers who will be able to work with either your built or bought solution?

With engineers moving from company to company more often, one benefit to buying a popular third-party solution is that you’re likely to hire someone who has used the tool in a previous job. In contrast, if you build a solution in-house, you have to educate new engineers on how the system works and how to use it.

**Considering Open Source Solutions**

The option to utilize open source solutions has often been billed as an “in-between” option between buy and build, but it isn’t necessarily cheaper than buying or provide more control than building. Some considerations to keep in mind include:

1. Open source solutions are not free

While the code itself is free, open source solutions need to be deployed and managed – this means AWS server costs and engineering time. Keeping your open source solution up to date, secure, and working well can quickly become a very expensive endeavor.

1. Open source solutions don’t offer complete control or flexibility

While you can certainly choose to fork the code at any point and build off the open-source solution, in reality, you’re often dependent on the road map of contributors, and can be left out in the cold if key contributors decide to leave the project. Maintainers of even popular open source projects are paid nothing, which means you can’t expect prompt customer support or bug fixes.

**Case Studies**

**Perpay**

Perpay, a smaller fintech company of only fifty employees, needed a reverse ETL tool to move its data from its online services into its marketing tool.

The team initially decided to build, but integrating with the various data source API’s – maintaining the infrastructure that would do back-offs, respect rate-limits, etc, ended up being too cumbersome.

The answer ultimately led them to buy a reverse ELT tool, Census, which had a built-in integration with the marketing tool used.

**Lyft**

Lyft was looking for a data catalog and data governance product to more effectively organize the massive amounts of data that the organization was generating. There were three options under evaluation:

* buying off-the-shelf data catalog solutions from Atlassian or Calibra
* adopting Atlas, an open source solution
* building their own data catalog tool

Ultimately, Lyft decided to build their [own tool](https://eng.lyft.com/amundsen-lyfts-data-discovery-metadata-engine-62d27254fbb9), because the existing off-the-shelf solutions required a certain philosophy about organizing data that did not fit with Lyft’s at the time.

**Changing Course**

Regular re-evaluation can help keep companies nimble and allow them to let go of data tooling that has become outdated. A common scenario, for example, is that in an initial evaluation, the company decides that off-the-shelf solutions are not sufficiently mature or are missing certain features. But over the course of the next few years, the third-party solution improves, to the point where it eventually makes sense to switch over.

Cost/benefit analyses should not only happen at the start of the project but should continue throughout implementation and maintenance. Once a year, companies should update their list of requirements, and determine whether the tooling they have, either bought or built, is fulfilling them. While pivoting away from legacy built solutions can be a difficult, even emotional process, this sort of objective performance evaluation makes it part of the natural growth process of a company.