Stop being precious about building data infrastructure tools

Most startup founders will agree that having a first-mover's advantage by reducing time to market is an essential part of their long-term success strategy.

Temps de lecture : minute

11 May 2022

One of the best ways to reduce the time to market is to buy, not build, the tools needed to augment the productivity of your team.

In theory, most founders and CTOs understand this concept. The advent of GitHub eliminated the need for internal version control tools. Stripe made building internal payment infrastructure a thing of the past. And no one constructs internal messaging tools or email lists anymore– why bother, when you can purchase Slack?

And yet, many AI-focused startups continue to build their data infrastructure in-house. They think that their use case is special and specific, so it requires in-house tools, and they must hire a team of software engineers to build them.

> *Trust me, it isn't, and you don't.*

Building an internal data system takes months. It lengthens the time it takes to get to market, and the result won't give you an edge over competitors.

With the increase in the accuracy of machine learning models and the decrease in computing power, an AI-focused startup's main competitive advantage comes from having access to, and training their model on, an abundance of real-world data. Getting to market quickly means getting access to a greater amount of data. Access to that data will improve your model, and that's what gives your product a competitive edge.

In just the past few years, the data infrastructure software market has expanded tremendously, making it rarely necessary for software engineers to build tools in-house.

AI-focused startups still have a tendency to want to build tools in-house, but, ultimately, using in-house tools will waste valuable engineering resources that can be better spent getting your machine learning model ready for the market.

The attachment to in-house tools

Not long ago, everything had to be built from scratch, and purchasing data infrastructure solutions from an external vendor wasn't really an option.

As a result, many AI application companies founded in the late 2010s run on legacy infrastructure – not because it's a good business decision but because they've already paid for the infrastructure's development. The thinking goes that they built the tools, the tools work for their current use case, and to discontinue use of the tools would be a waste of money.

Welcome to the sunk cost fallacy.

This short-term thinking fails to factor in the spending required to maintain a custom infrastructure and the costs of extending that infrastructure should a new use case arise. From a business standpoint,

building in-house tools comes with the additional risk that the team members who built the tools often have the institutional knowledge needed for their upkeep. If they leave, that knowledge leaves with them.

Building internal systems is challenging, their upkeep is costly, and often they don’t meet the needs of the end users. Believe me, I would know. For years, I worked at large financial services institutions. These companies hired top quality engineering teams and spent billions of dollars each year on their internal technology systems. And you know what? Their internal systems still suck.

These days, there’s often a product on the market that is cheaper and orders of magnitudes better than an in-house tool. Just because something was done a certain way, doesn’t mean we should continue to do it that way. In-house tools were the best solution of their time, but that time has passed.

Put your focus on the product

At their core, most AI-focused companies focus on building machine learning models that can derive valuable insights from data. To get their products to market, these companies must train the model on data until it can make accurate predictions about never-before-seen data. All of the preparation required to train a model– cleaning data, making raw data readable, managing the data pipelines– falls within the remit of a data engineer.

A good data engineer is a costly but worthwhile investment. However, to get the most for their money, companies need to ensure that engineers have the best tools available so that they can work effectively and efficiently.

Often, software engineers don’t build in-house tools with the data

engineers in mind, so these tools don’t meet their day-to-day needs. Working with suboptimal tools means that data engineers have to waste time on low value activities (such as writing parsers to transform one data format to another or learning Javascript and Django for web development tooling) and troubleshooting infrastructure problems. These distractions limit their ability to focus on product improvements.

As a startup founder, I want my engineers to focus on our product, and I aim to allocate our engineering resources to making our product better. I have no interest in our team spending their time building in-house tools when we can purchase solutions to problems we encounter.

Recently, our developers need to figure out a way to replicate errors in our front-end code base. It’s a complex problem, and solving it would enable us to better serve our customers. After a few months, we decided that we’d have to build some in-house infrastructure for this task.

Fortunately, around that time, we also hired a new front end engineer, who casually told us that his previous employer had used a software called Rollbar to solve the same problem.

> *Purchase made. Problem solved.*

Building that tool would have been a catastrophic waste of time because it would have taken significant resources and limited our capacity for focusing on more important product-related issues.

Only a few years ago, external solutions weren’t available for most data engineering issues. Now, software stacks are popping up every day, offering specialized solutions, so make the smart decision and don’t be precious about building and using in-house tools.

Ulrik Stig Hansen is the co-founder of Encord. He has been coding since he was 14. He started his career in emerging market rates and FX at JP Morgan. He holds a M.S. in Computer Science from Imperial College London.

Article écrit par Ulrik Stig Hansen