Open in app ↗ Sign up Sign In

Published in Towards Data Science

Micha Kunze Follow

Oct 31, 2020 · 7 min read · Listen

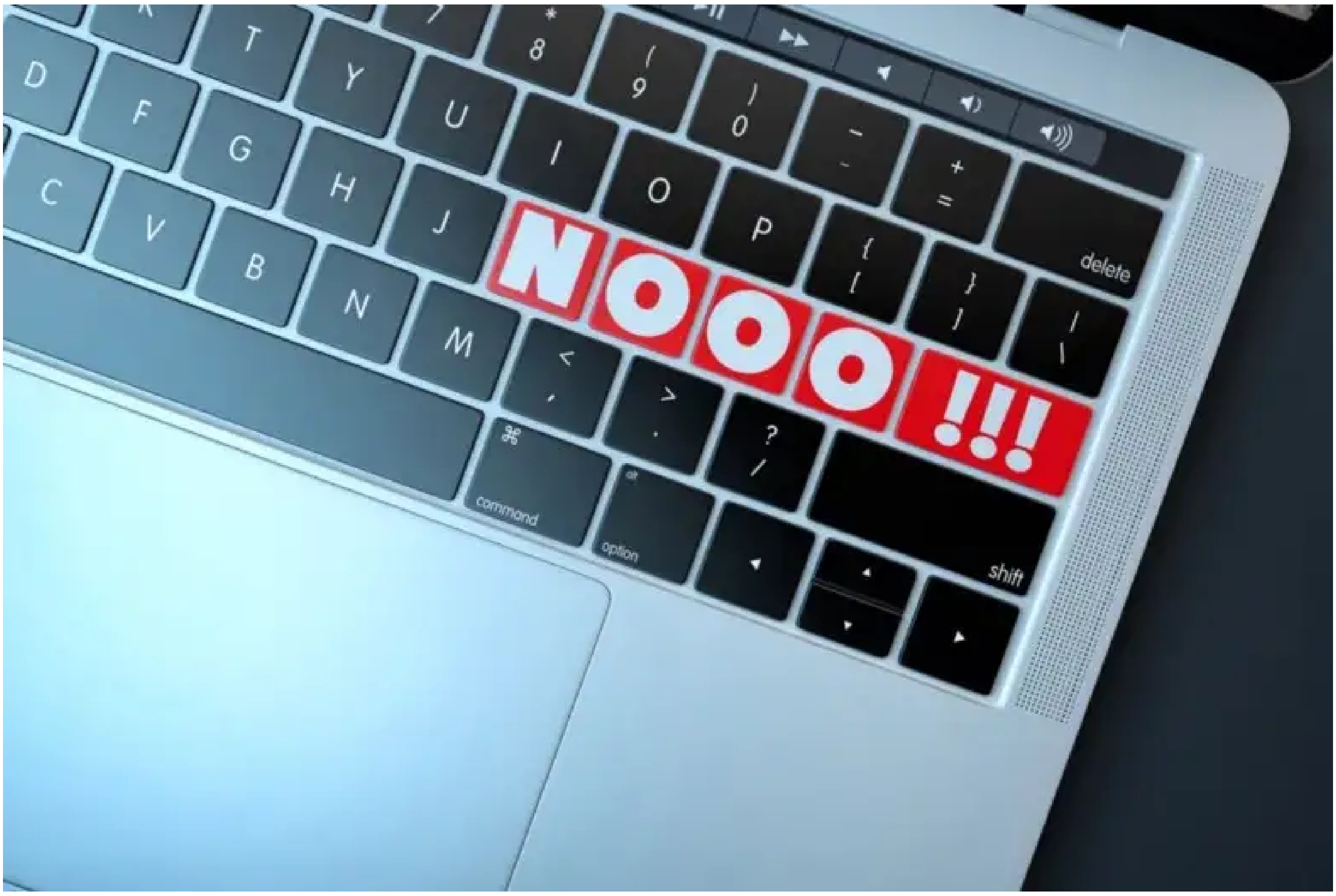
Save

MAKING SENSE OF BIG DATA

Develop Your Data as a Product

Drafting data engineering practices

TL:DR — KISS ❤️ DevOps!



Picture from slon.pics

269 1

Sparked by a discussion at work, I am trying to write down my personal perspective on what best-practices for data engineering look like. To do that I thought it would be best to first define the goal of data engineering by answering the question:

> *What is the **outcome** that we, the Data Engineers, are trying to drive?*

And then, after we are clear on the outcome, let us detail which principles and practices can help us drive that.

A disclaimer: I have working experience in what I would call big enterprises (> 50.000 employees), working closely with scientists or directly in a product/analytics team. If you are working in a small company as a one-man army, or in a start-up, some of the things might not be relatable — though I think the premise and principles still apply.

Premise: Data is a Product

Data itself sitting idle and dormant is somewhat worthless, it is the outcome derived from this data that is generating value for you (and the company). This means data is a product. Without customers that actually use it/buy it, it is not generating anything.
If we follow this simple premise, we can quickly draw up some naive metrics that tell us how successful our data product is:

- Usage of the product that generates value
- Speed-to-market and speed of change
- Quality and reliability
- Availability and interoperability

Build Good Products = Keep it Simple

In my experience, a product thinking mindset really helps my work to be impactful. Every time we develop something the product dimensions of value of the solution, viability, desirability, and feasibility should be what we are thinking about.

Most of the time we will spend on how we build and operate solutions and that will dictate how *feasible* and *viable* they are. Hence, it is crucial to understand the business problem and then design sustainable solutions. And here I see a strong relationship between the product mindset and KISS (keep it simple, stupid). The latter is key to build sustainable (read long-term feasible and viable) products.

A (too) simple example: Assume we are Starbucks and need to make decisions every day on how many paper cups we ship out to each and every store. Now, would it not be cool to have a real-time paper cup data stream to monitor the stock and react to changes? — Absolutely not. If we need to only make that decision once a day we can accomplish that with a simple cronjob that runs every day.

Keep it simple and only do the things your solution needs; just because we could do something more fancy does not mean we should.

Practices to Develop a Product

Spoiler: Best-practices for Data Engineers and Software Engineers are virtually the same. Especially DevOps practices have improved how we build — and while there is also DataOps, the latter is merely a term relating DevOps principles to the data domain.

This does not mean Data Engineering and Software Engineering are the same, but the truth is that Software Engineers have had a head start when it comes to quickly build value-generating digital products, and we should learn from that.

Let us break that down in more detail and link it to the metrics above.

Usage and generated value. This is what matters, we want to deliver a product that has an impact. We know from software development that this can effectively be done by knowing and working closely with our consumers, preferably in the same team. Know the needs, understand the pain points, and address them iteratively. Practically, this means we will have to experiment and evolve our product. Technically, this means we need to be able to keep track of that and constantly be able to develop new features (or rollback changes).

Speed-to-market. Automate processes and tooling, we need to be able to ship and develop a data product fast, at production-grade. The only way to do that efficiently, especially in an analytics context, is having decoupled data assets. If every time we first have to update a gazillion pipelines to add a new dimension to our trusty star schema (and thereby break some undocumented downstream dependency) we have no chance to gain any sustainable velocity.

Quality and Reliability. The first thing we need to have is a healthy continuous improvement mindset when it comes to development and operations. We will never build a perfect product from the start. Technically, this means that we should always have a good CI/CD setup (DevOps). Mentally, this means we have to adopt a postmortem culture. In other words, we try our very best to catch problems utilizing automatic testing/self-healing pods/data quality checks. And when things inevitably go wrong we make sure to learn from it and improve the product!

Availability and interoperability. Availability is not just the presence of a data product, but how easy it is to consume. Not for us, but for our consumers. Our product has to be used in order to maximize value generation so we need to make it easily available. We need to use standard interfaces to facilitate the usage of our product by diverse teams, i.e. we do not make a technology choice for our consumers or make them jump through technology hoops. To stay agile and fast we need to separate the concern of our internal development from the interface we share with our consumers.

All of the above points are derived from lessons learned by software engineers when building digital products. In my opinion, they all apply equally well to data engineering — though their implementation might vary in tech.

One more thing: **Immutability.** This is maybe not as important/widespread in software engineering best-practices and comes more from the functional programming folks, but it is extremely impactful for data engineering. Especially in the analytics space. Try to practice functional data engineering, which means: we do not change history, instead, we snapshot all the data. Not doing this is a path to insanity — something that is worth writing a full post about. For now, I will leave you with a pointer to some of the work of Maxime Beauchemin, who has some great material on functional data engineering ❤️.

Data Engineering Practices

If you made it so far, great! Let's try to boil things down further to some simple tangible practices that we as Data Engineers can apply in our day-to-day.

Version Control — All our work needs to be version controlled, our code, our data, and the documentation of our data. Evolve, experiment, and if necessary roll back!

Automated testing and deployment — Automated tests provide high and sustainable quality. Automated deployment makes us efficient and mitigates errors. We need this (check Python for DevOps to get started)!

Make and keep data available — Make your data available in the easiest way possible. And make sure it stays available — use monitoring and alerting to stay proactive in fixing issues.

Test and communicate data quality — Continuously assess your product's data quality and communicate it, preferably via automated data documentation. Use monitoring and alerting to stay proactive in fixing issues.

Separate Concerns — Ensure that you are not tightly coupling things together. Don't put everything in a star schema right away, keep data products separated as much as possible.

Share best-practices with colleagues — Communicate with our Data Engineering colleagues, learn about the best implementations, and share them. Ultimately we put the best ideas into a framework and abstract the implementation details -> profit!

Build immutable datasets — Data is history and history is immutable. Updating observations (rows) leads to issues when analyzing the history of data. This is immensely impactful for debugging and also machine learning uses cases. We strive to make it simple and enable downstream use of our data products by preserving the history -> data is immutable.

These are my top practices, the ones that (in my humble opinion) give us the most bang for the buck and relate closely to the metrics described earlier. That does not mean there is not more to learn out there!

Notes on Positioning in Big Companies

If we follow the premise of Data as a Product we have to intimately understand our consumers, we have to experiment with them to drive the best possible outcome. We have to jointly create value and constantly change to figure out what works and we have to constantly care for our product to improve it further.

Essentially that means working in a product team rather than a centralized data team (and I can say from experience that it is awesome 😉). Of course, the concept of decentralizing Data Engineers is not new and you could even argue it is just taking the cross-functional team seriously — and I would agree. One of the persons who brought this together and promotes it is Zhamak Dehghani who wrote a great article about this with a descriptive bird-eye view on how this could work at scale (she is also convincing on Youtube 👍). I believe what I am describing in this post is the inside-view and motivation of a Data Engineer and therefore wholeheartedly agree with her perspective.

Conclusion

I used this post to clear my own head and get some of my thoughts in order.

By doing so, I hope I have been able to give some people stuff to think about. Maybe some inspiration on how to improve their day-to-day data engineering work or how to become more productive. If you disagree with my views, I would also love to hear and learn from you.

I believe that the outcomes we can drive through data products are impactful and that we have to make them first-class citizens.
We have all the tools and tech to make that happen, so let's do it! 💪

Data Data Engineering Software Development Organizational Culture

Making Sense Of Big Data

Sign up for The Variable

By Towards Data Science

Every Thursday, the Variable delivers the very best of Towards Data Science: from hands-on tutorials and cutting-edge research to original features you don't want to miss. Take a look.

By signing up, you will create a Medium account if you don't already have one. Review our Privacy Policy for more information about our privacy practices.

Get this newsletter

Get the Medium app

Download on the App Store

GET IT ON Google Play