A Search Divided Cannot Stand

Data-Eng partnership

Doug Turnbull & Chen Karako | MICES, April 26, 2022

shopify

About us

Doug Turnbull

Sr. Staff Relevance Engineer, Shopify
http://softwaredoug.com
http://linkedin.com/in/softwaredoug

Chen Karako

Data Science Lead, Shopify
https://www.linkedin.com/in/chenkarako

shopify

At a Glance

Shopify is a leading multi-channel commerce platform.

2006
Platform Released

$4.6B
Revenue (2021)

10,000+
Employees

Millions
Merchants
— in —
~175 Countries

~$450B
Total Sales on Shopify
by Merchants
(Cumulative)

allbirds

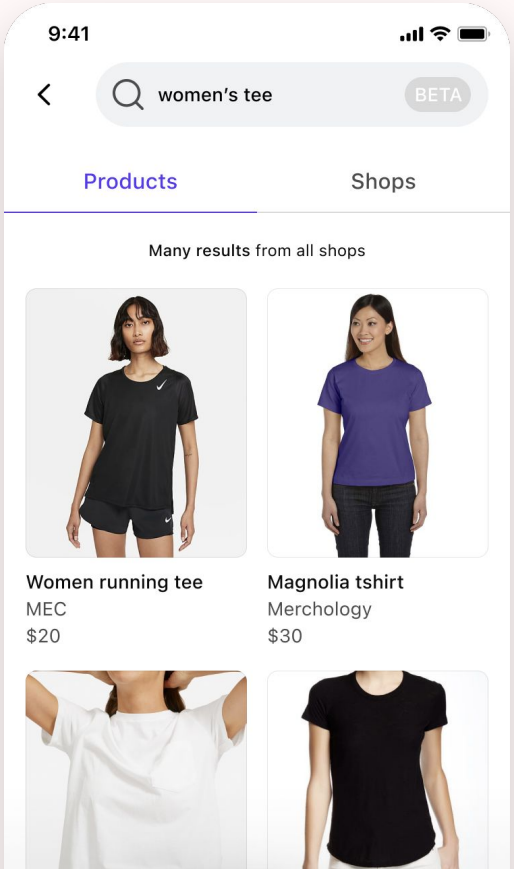
Heinz

FIGS

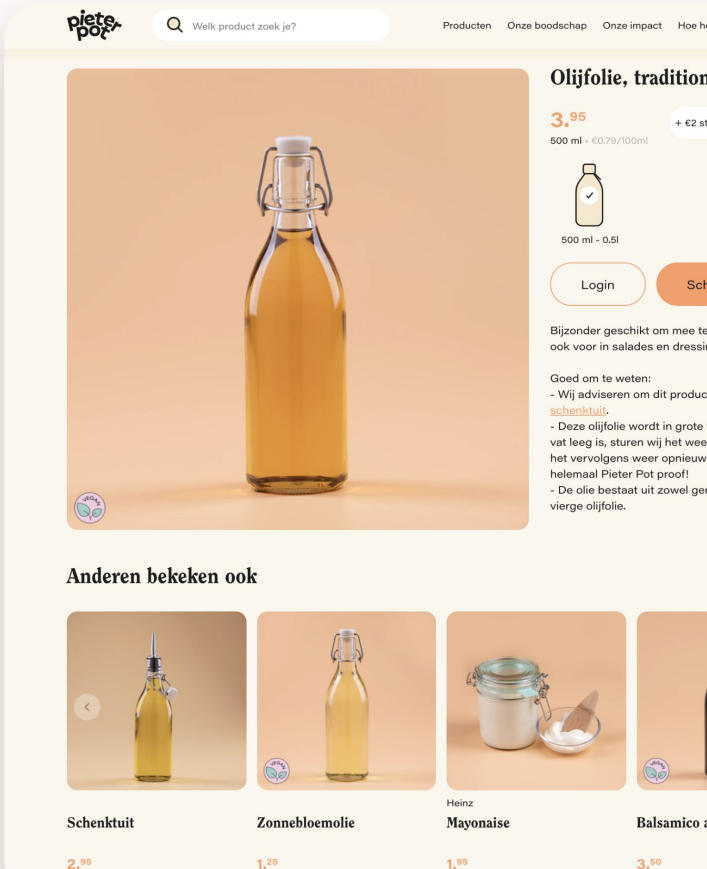
GYMSHARK

Tupperware®

About Discovery Experience at Shopify



- Search and recommendations for millions of merchants
- Empowering merchants by giving them state of the art search and discovery tooling
- Deepening relationships between merchants and their buyers



Collaboration:
Avoiding Data + Eng Dysfunction

(Data)

Chen Karako 4:22 PM

Hey Doug, I was just checking on the A/B test and noticed that very few people are allocated to the treatment group.

Are you sure you set things up correctly?

(Eng)

Doug Turnbull 4:27 PM

I don't want to bore you with boring Ruby / Rails details, but the code seems fine, and it should allocate 50% of users to each group 🤔

By the way, how are you accounting for the impact of caching in your analysis?

Chen Karako 4:31 PM

Caching? 😕

Doug Turnbull 4:40 PM

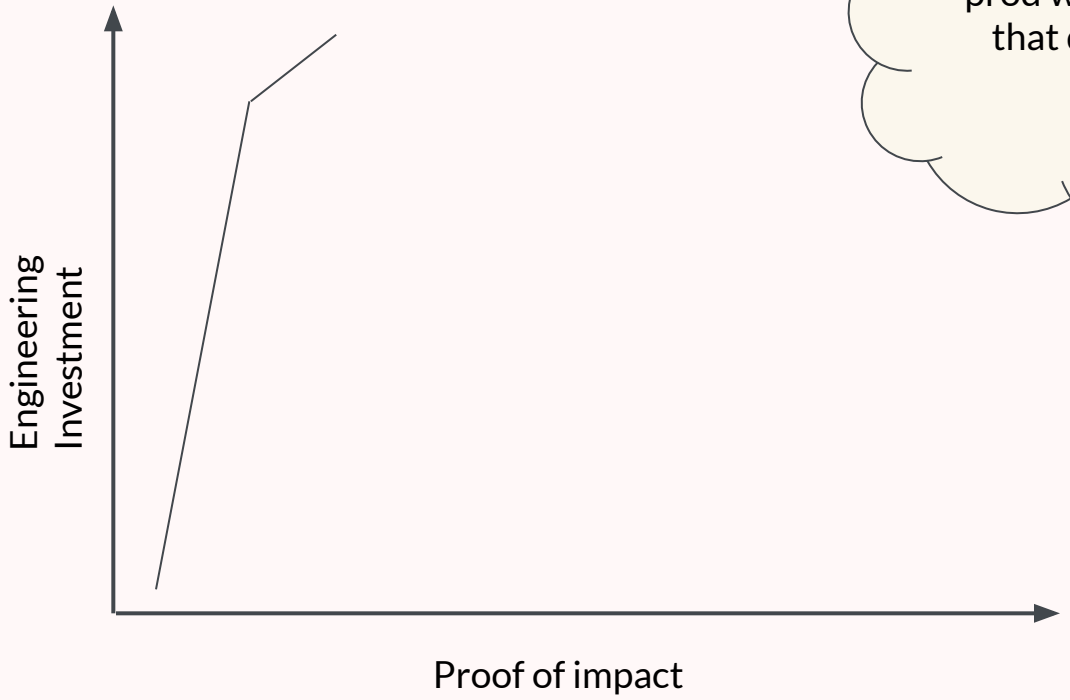
Check this out! I doubled the title boost and I shipped it!!! 🚀

When do you think you'd be able to let me know how well it's doing?

Chen Karako 4:41 PM

😳

Avoid investing too much too early



Overbuilding idea in prod without proof that direction is sound

Chen Karako 4:44 PM

Our new model prototype seems like it'll give us 10% higher conversion - the PM is really excited! We're hoping to release it as soon as possible. Do you think the infrastructure will be ready next week?

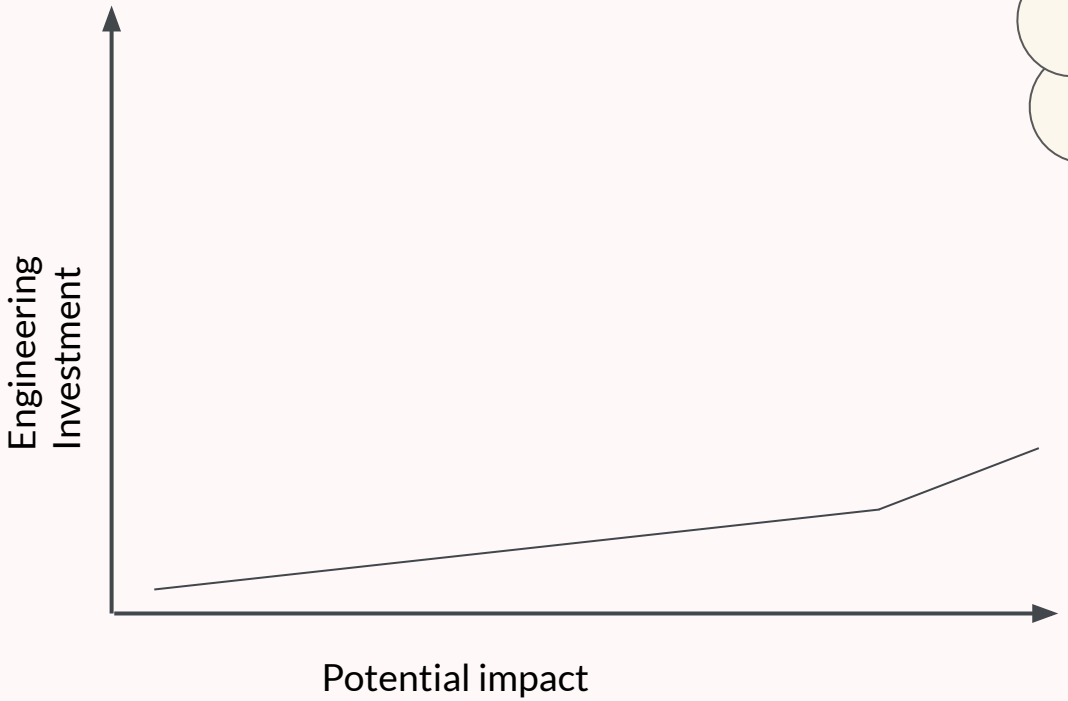
Doug Turnbull 4:44 PM

Netflix Never Used Its $1 Million Algorithm Due To Engineering Costs

Netflix awarded a $1 million prize to a developer team in 2009 for an algorithm that increased the accuracy of the company's recommendation engine by 10 percent. But it doesn't use the million-dollar code, and has no plans to implement it in the future, Netflix announced on its blog Friday. The post goes on to explain why: [...]

https://www.wired.com/2012/04/netflix-prize-costs/

Avoid investing too little too late



Promising direction, but no engineering effort put into building infrastructure

Engineering - Data Collaboration Problems are rife throughout our field

- **Project Delays** - eng / data constraints not taken into account early enough
- **Relevance 'improvements'** that don't have positive impact (or whose impact can't be measured)
- **Slow, unstable data products** - by not incorporating eng, may build complex solns that don't scale
- **Poor eng leads to poor science** - poorly maintained code leads to non-repeatable science

What would the ideal state be?

Creators of data products need both skills

(Software) Engineering

Repeatable, reusable, performant, testable, modular code

Code that we can put in prod and trust

Data

‘Good’ experiments - accuracy, consideration of bias

ML - building robust, repeatable models

With these powers combined…

- Satisfy software constraints
- Go to WebScale 🚀 ™
- Solve user problems
- Accurately measure impact
- Iterate reliably on new ideas

Collaboration Beyond Data – Eng

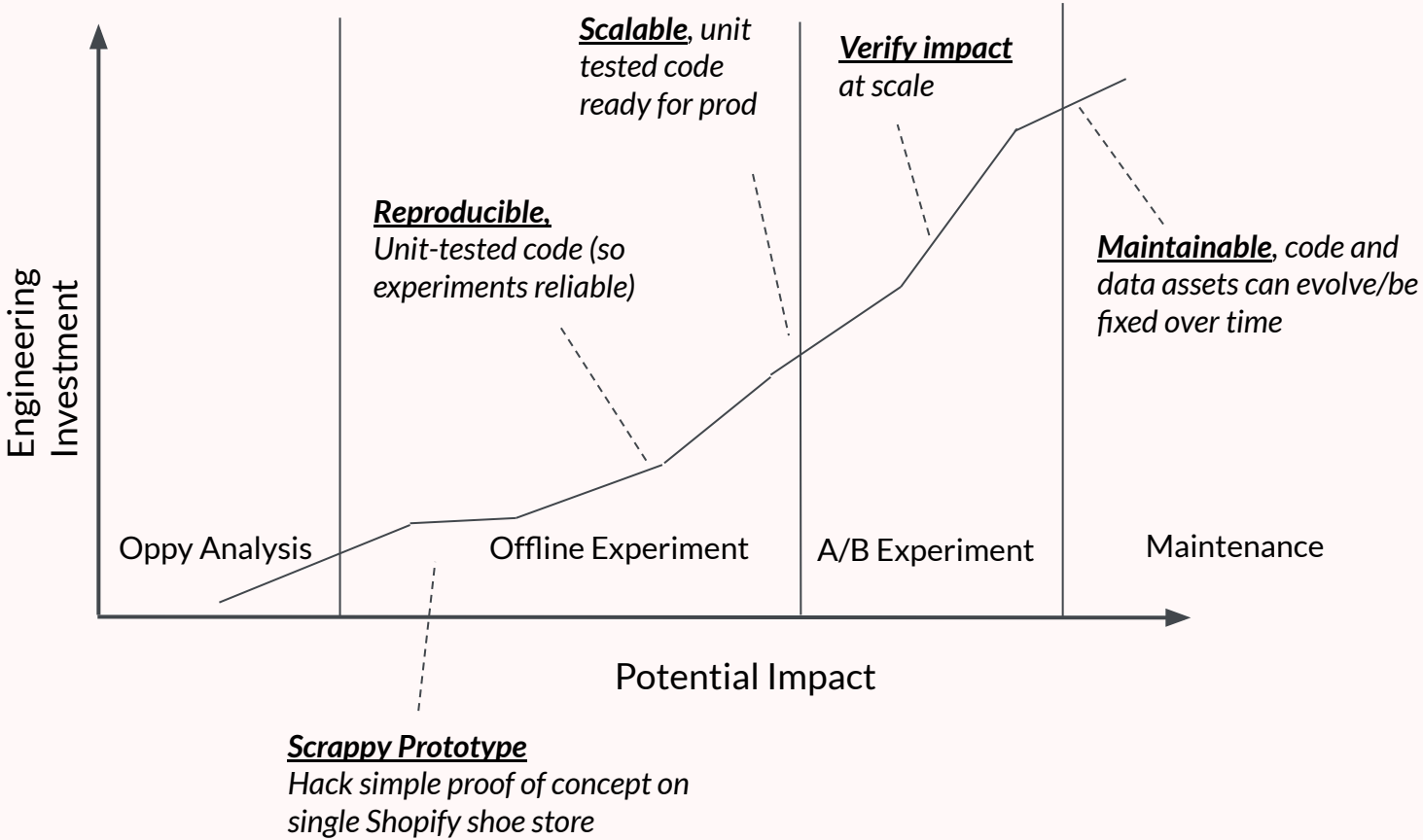
While eng vs data is very salient in search, other disciplines matter too!

Eng & Data can build the ***wrong*** 'relevant' search experience, if don't consult with:

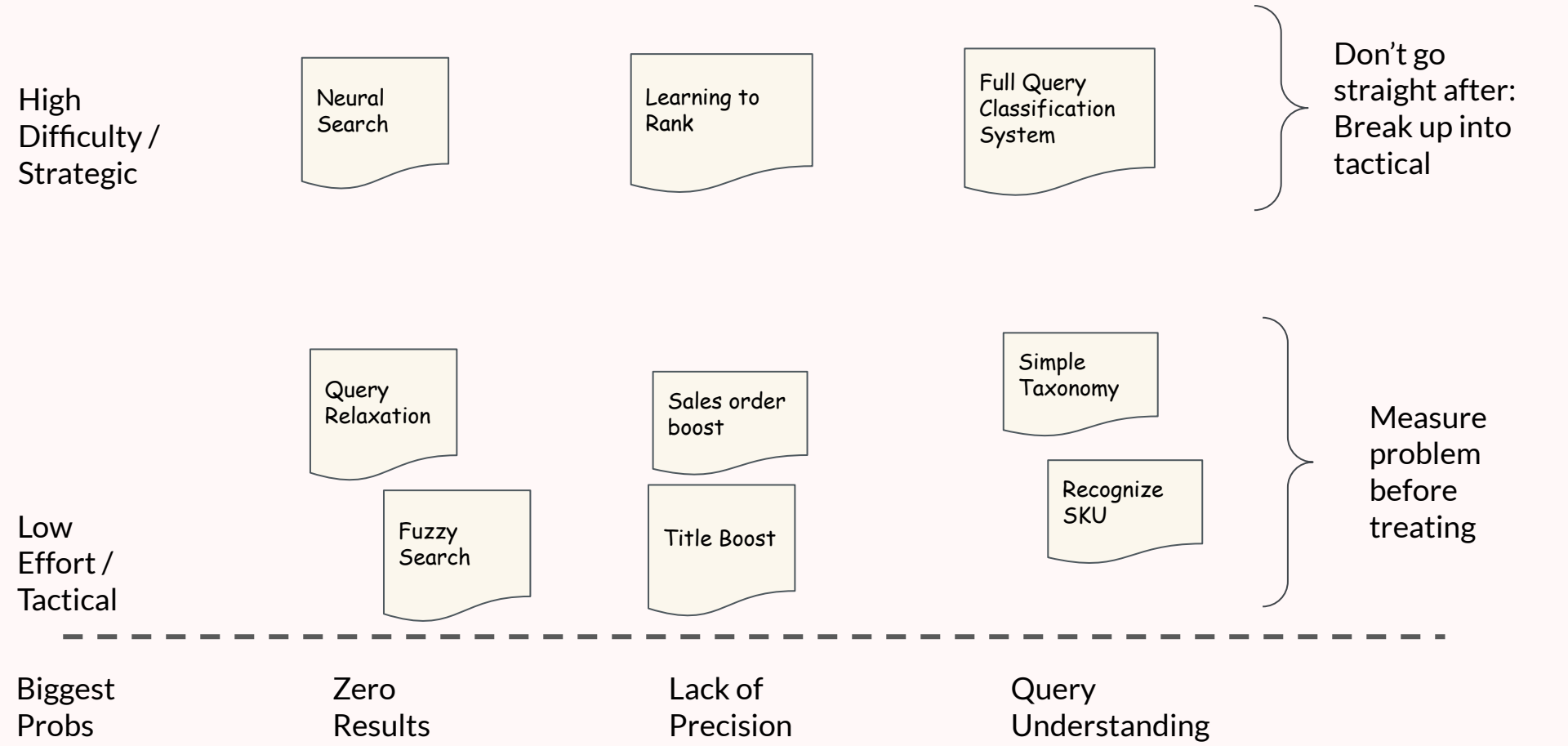
- UX
- Strategy
- Product
- Partners
- …

In practice: how we build great search

Ideal experiment scales investment to evidence



Analysis & Experiment Brainstorming



Before experiment - both groups maintain tooling

- *Methodology* (data)
- *Repeatability* (software eng & data)
- *Extensibility* (software eng & data)

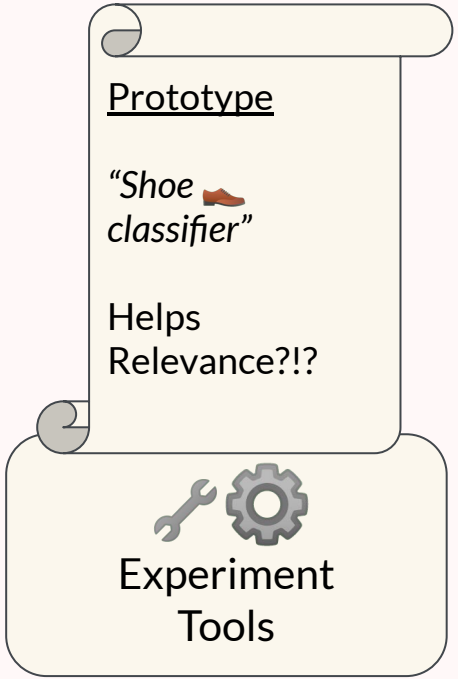
Eng

Data

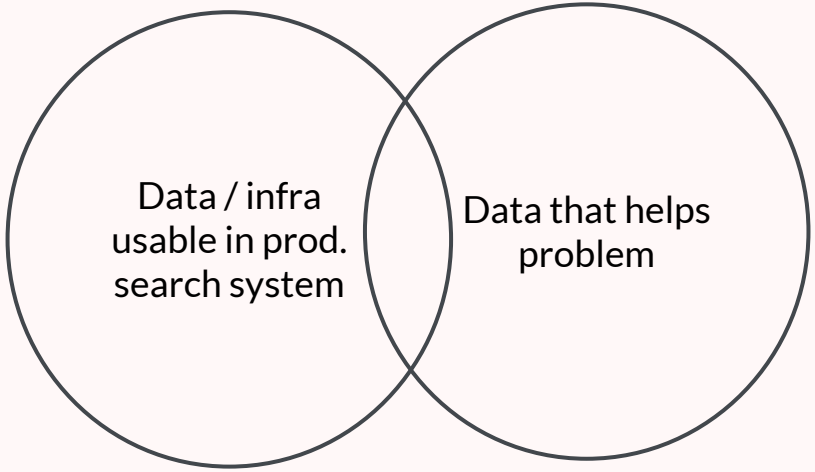
Experiment Tools

Easily try ideas with scrappy prototype

Should be able to easily create experiments and answer questions



Scrappy Prototype -> Mature data assets



Team can discuss what realistic assets would look like in production

Prototype

“Shoe 👞 experiment”

Experiment Tools

Scrappy Prototype -> Mature data assets

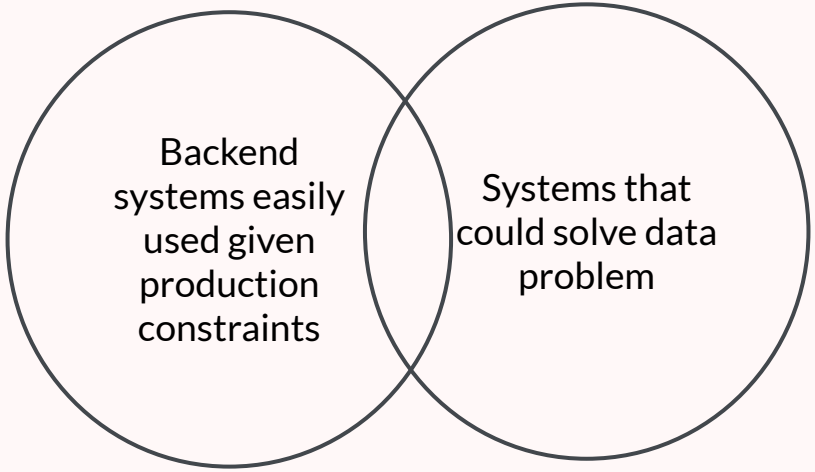
Prototype

“Shoe 👞 experiment”

Experiment Tools

An easy to build data asset using existing, well-trod text datasets

Scrappy Prototype -> Mature production systems



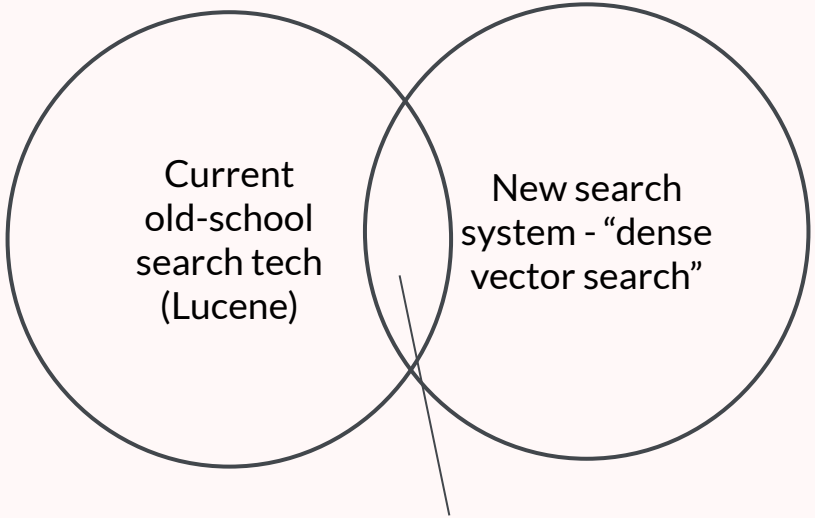
Team can discuss what realistic infra to solve problem looks like in production

Prototype

"Shoe 👞 experiment"

Experiment Tools

Scrappy Prototype -> Mature production systems



Prototype

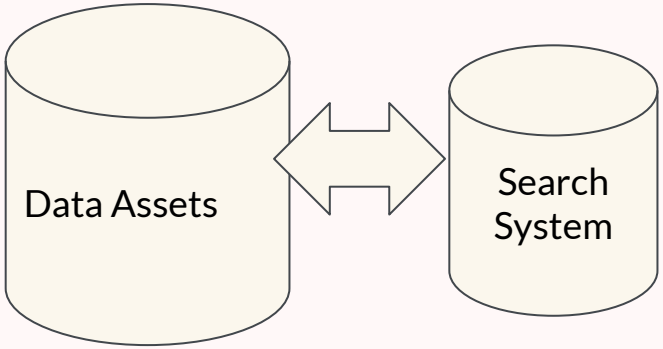
“Shoe 👞 experiment”

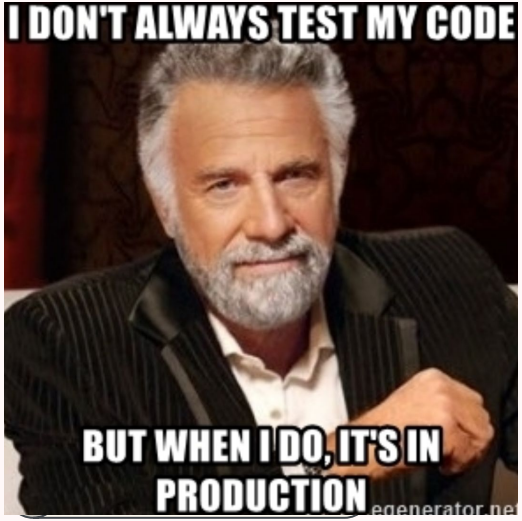
Experiment Tools

Tech available after version update to existing search engine?

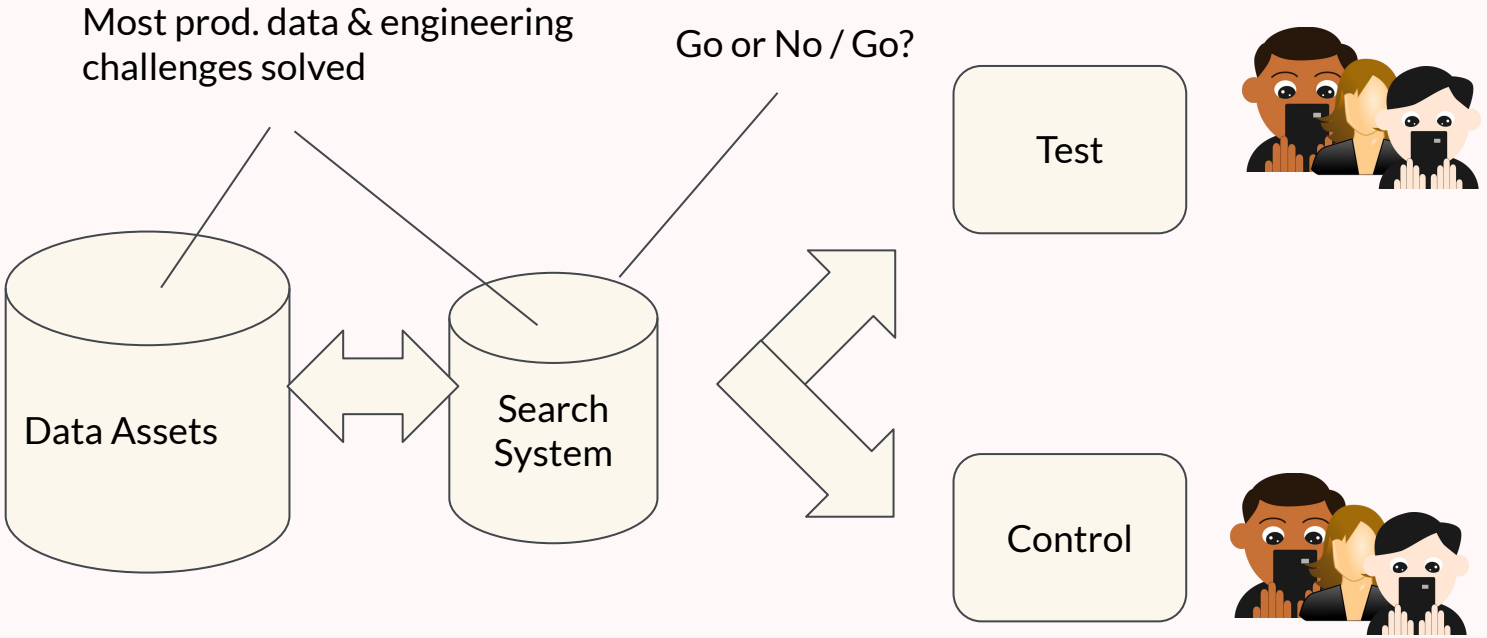
Scrappy reranking in search API rather than perfect solution?

Every step, we ship to prod, re-evaluate with prod infra & data





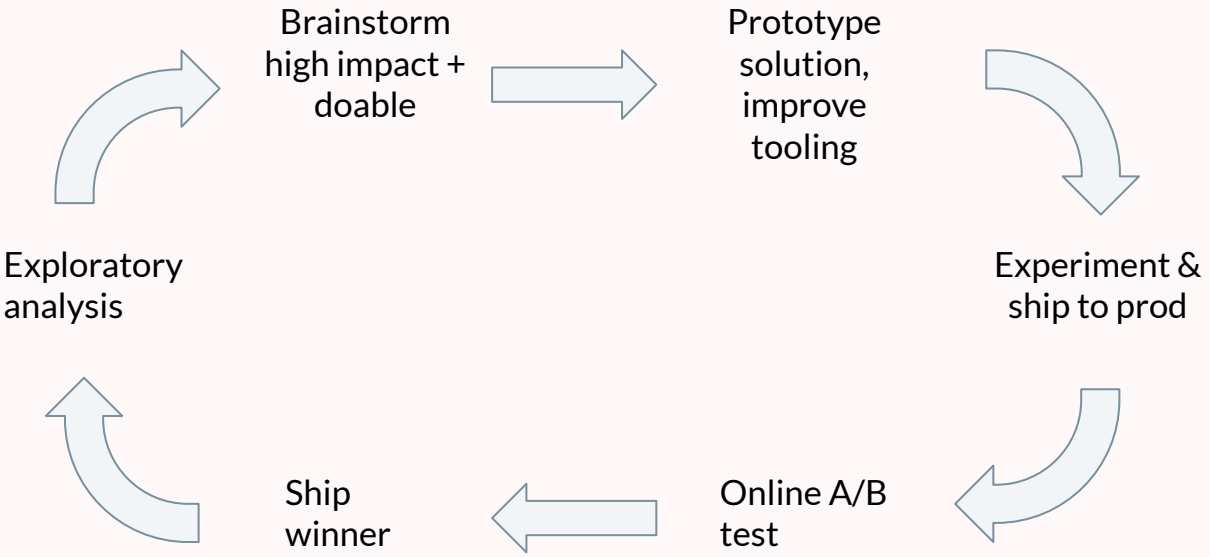
Going from offline to online experiment



We turn on the new solution for some buyers and measure impact

(ie simply by changing the query, or a final deployment step)

Experiment workflow



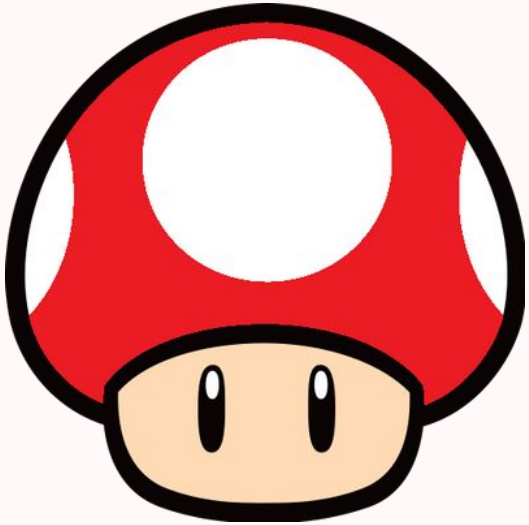
How we built a team that could do this

Team practices

- **Single** team identity across disciplines
- Team planning together
- Pair programming

Growth mindset

- We hire for people who want to push their skills
- Team members constantly stretch and help others grow across eng-data spectrum
- Proactive, can work in ambiguity

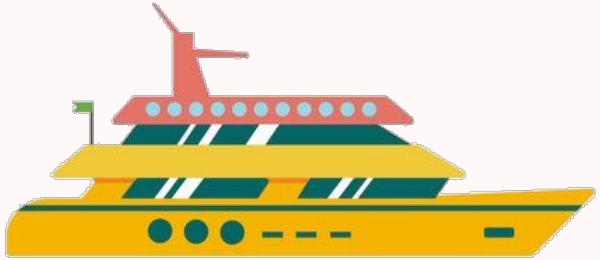


Emotional safety and feedback

- Data and eng leads set example by establishing trust, encouraging diverse opinions
- Whole team owns code, not the person that touched it last
- Hold each other accountable
  - Eng gives feedback on tests and code design, data gives feedback on experimental design
- Everyone shares in the credit
  - it's not (just) about one wizard that built an amazing ML model
  - It's not (just) about the brilliant eng that got that to scale in prod
  - Leads celebrate and amplify team's achievements!

Build a culture of shipping

- Shipping regularly helps the team build confidence in their skills
  - Data scientists write production-ready code
  - Engineers run search experiments
- Gather feedback and iterate



Should we even draw a line between engineers and data scientists?

Pipelines! And unit testing! And search infra!

Datengascientisteer

Experiments! And science! Reproducibility and machine learning!

Sciengadatatist

This is the future, will you join us?

shopify.com/careers

Thank you!