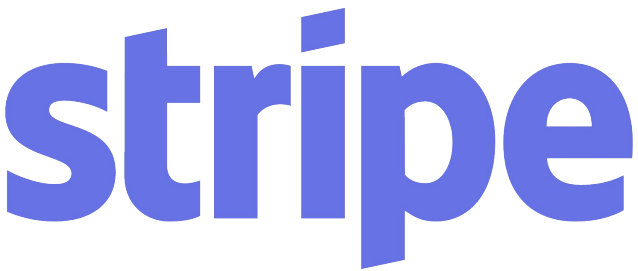
Paying Technical Debt at Scale: Migrations @ Stripe

Will Larson
@lethain

QCon SF
11/6/2018

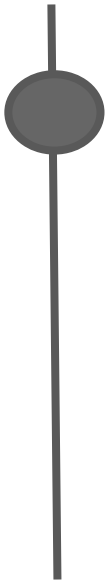

stripe

Uber

Tech debt:
core constraint on your velocity.

Migration:
fully replacing a tool, system or library.

Approach:
treat every migration like a product.



What is a migration?

Do migrations matter?

Effective migrations!

CC image courtesy of harshilshah on Flickr

Migration:
fully replacing a tool, system or library.

Blog › Engineering

Share this post on Twitter

Learning to operate Kubernetes reliably

Julia Evans on December 20, 2017 in Engineering

We recently built a distributed cron job scheduling system on top of Kubernetes, an exciting new platform for container orchestration. Kubernetes is very popular right now and makes a lot of exciting promises: one of the most exciting is that engineers don't need to know or care what machines their applications run on.

AWS re:Invent

Background: Netflix on AWS

- Netflix:
  world's largest
  subscription Internet
  TV Business

- 1,500 Employees
  1,000 Tech & Dev

- 53M members
- 10k starts/sec
- 7B hours streamed Q3
  = 10,000 PB delivered
- >33% of Internet traffic
  (US, last mile, downstream)
- 60,000 instances
- 40 PB S3

Architecture Open Source

Why Uber Engineering Switched from Postgres to MySQL

July 26, 2016

138 22K 12 2K G+

By Evan Klitzke

Introduction

The early architecture of Uber consisted of a monolithic backend application written in Python that used Postgres for data persistence. Since that time, the architecture of Uber has changed significantly, to a model of microservices and new data platforms. Specifically, in many of the cases where we previously used Postgres, we now use Schemaless, a novel database sharding layer built on top of MySQL. In this article, we'll explore some of the drawbacks we found with Postgres and explain the decision to build Schemaless and other backend services on top of MySQL.

Search

Sign up for Uber Engineering updates:

Your email address

Subscribe

Popular Articles

Uber's Big Data Platform: 100+ Petabytes with Minute Latency

October 17, 2018

Forecasting at Uber: An

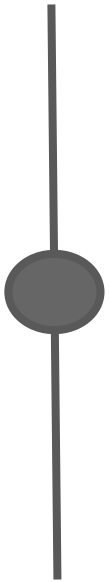
How we rolled out one of the largest Python 3 migrations ever

Max Bélanger and Damien DeVille | September 25, 2018

540 0

Dropbox is one of the most popular desktop applications in the world: You can install it today on Windows, macOS, and some flavors of Linux. What you may not know is that much of the application is written using Python. In fact, Drew's very first lines of code for Dropbox were written in Python for Windows using venerable libraries such as `pywin32`.

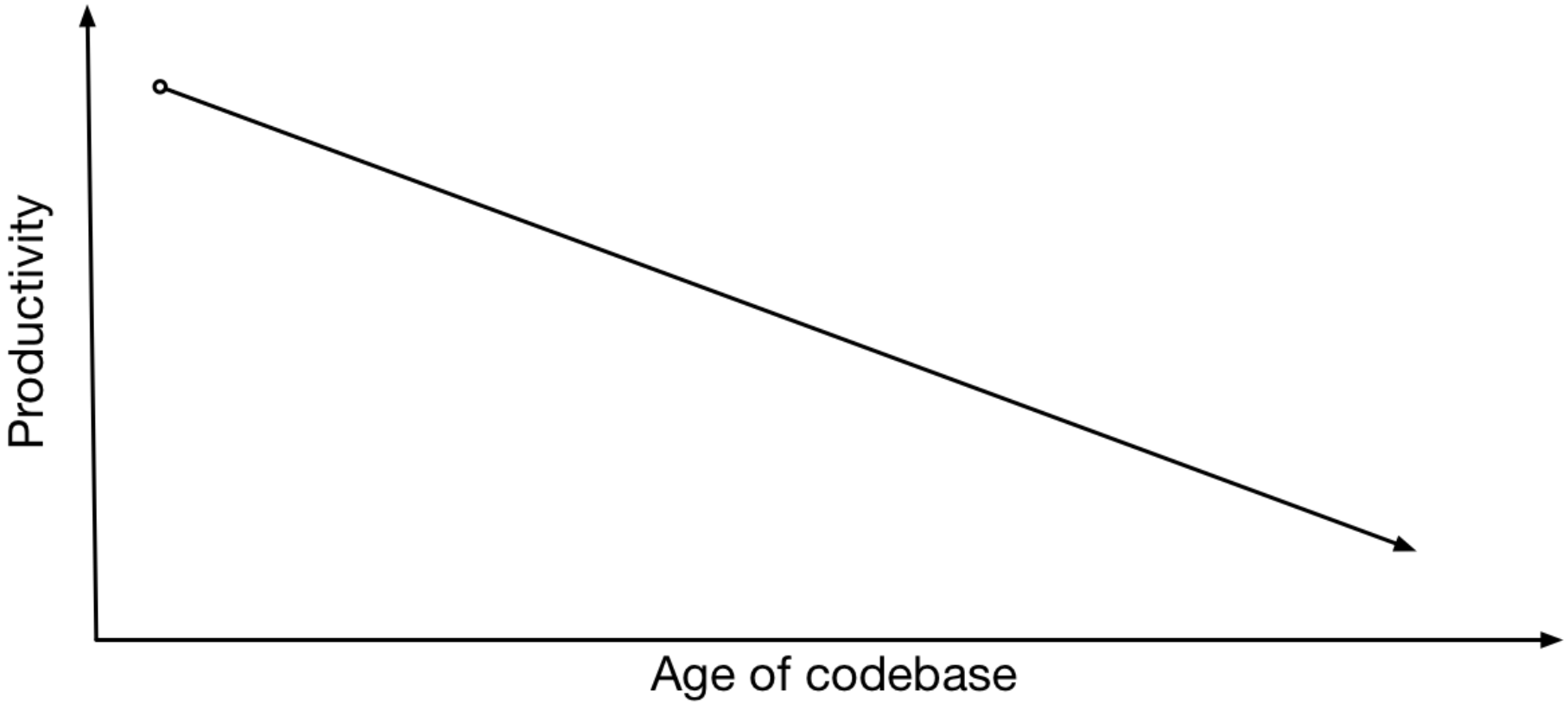
Migration:
fully replacing a tool, system or library.

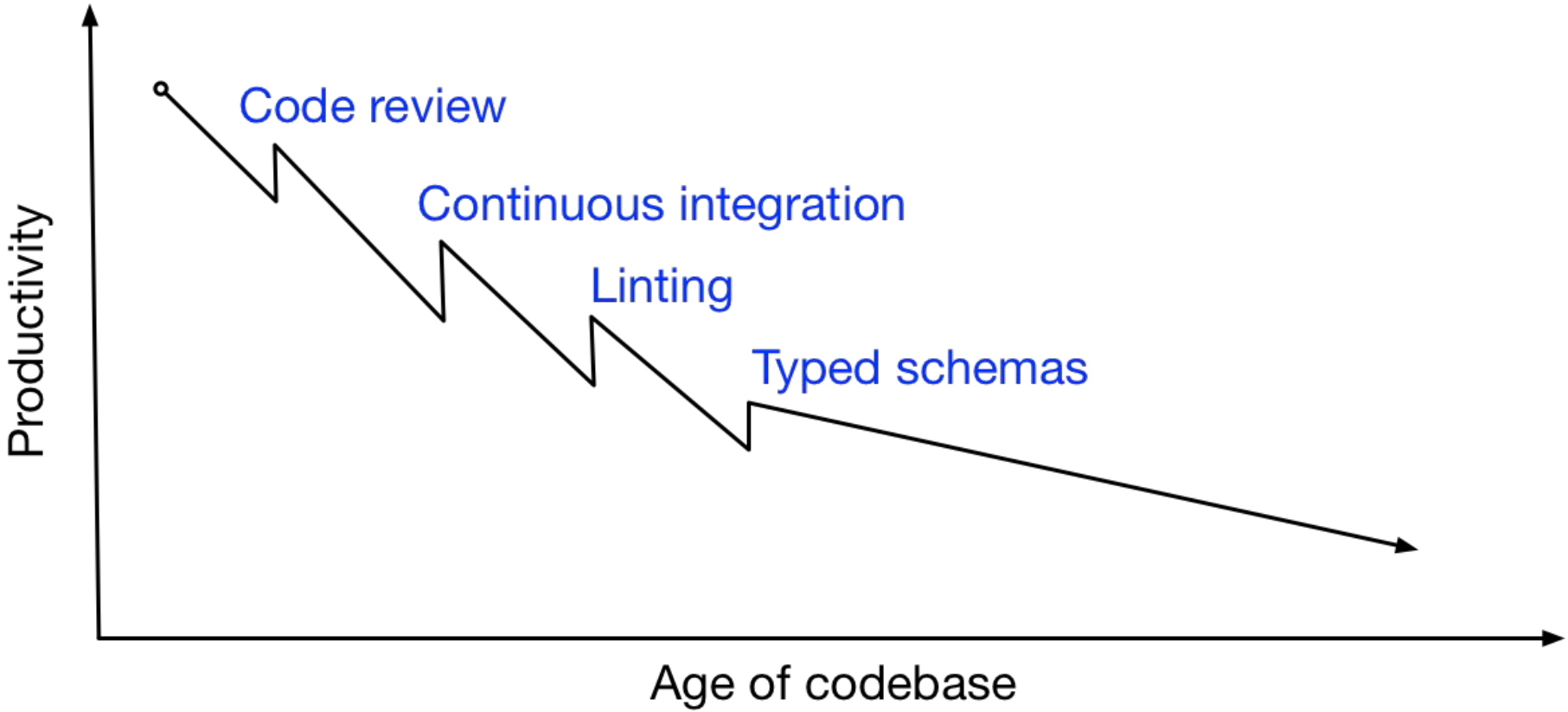


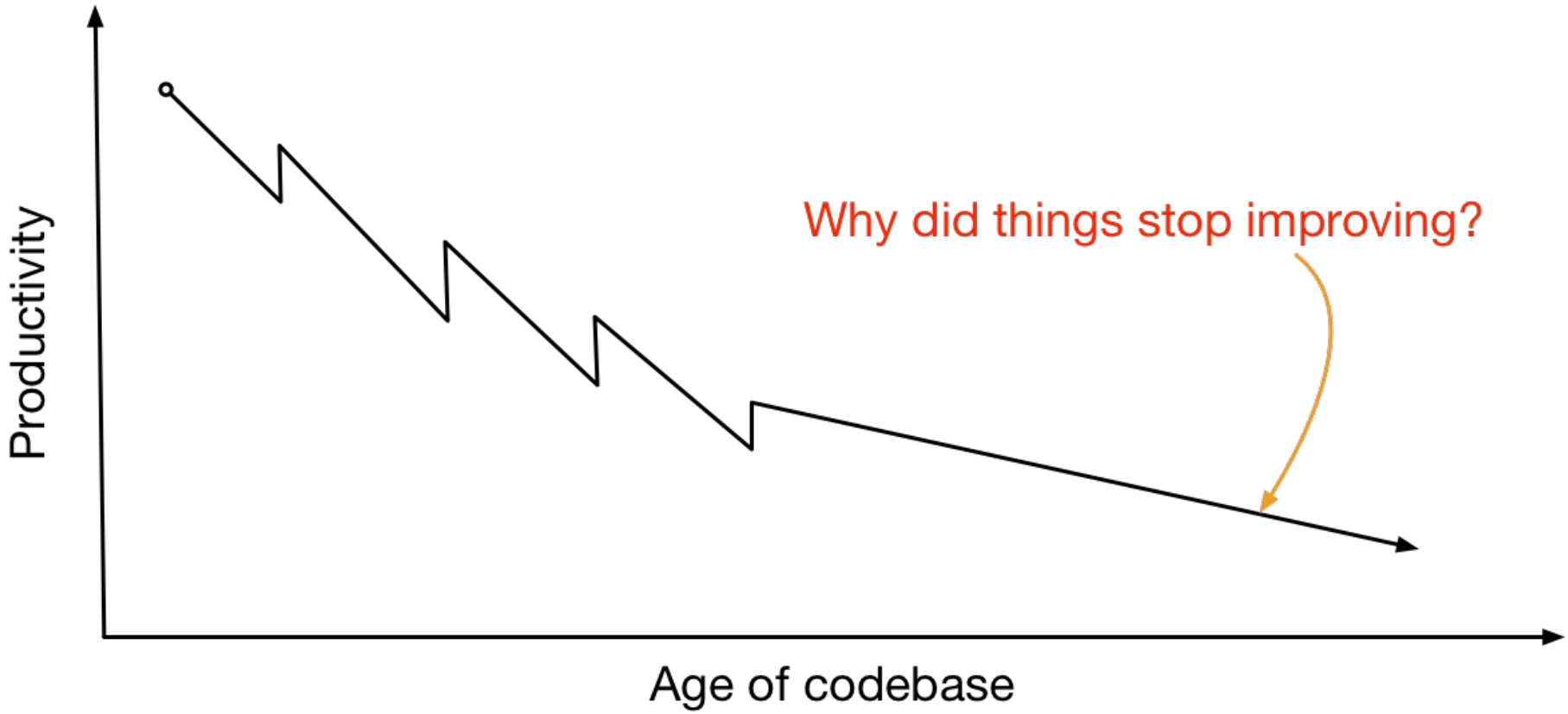
What is a migration?

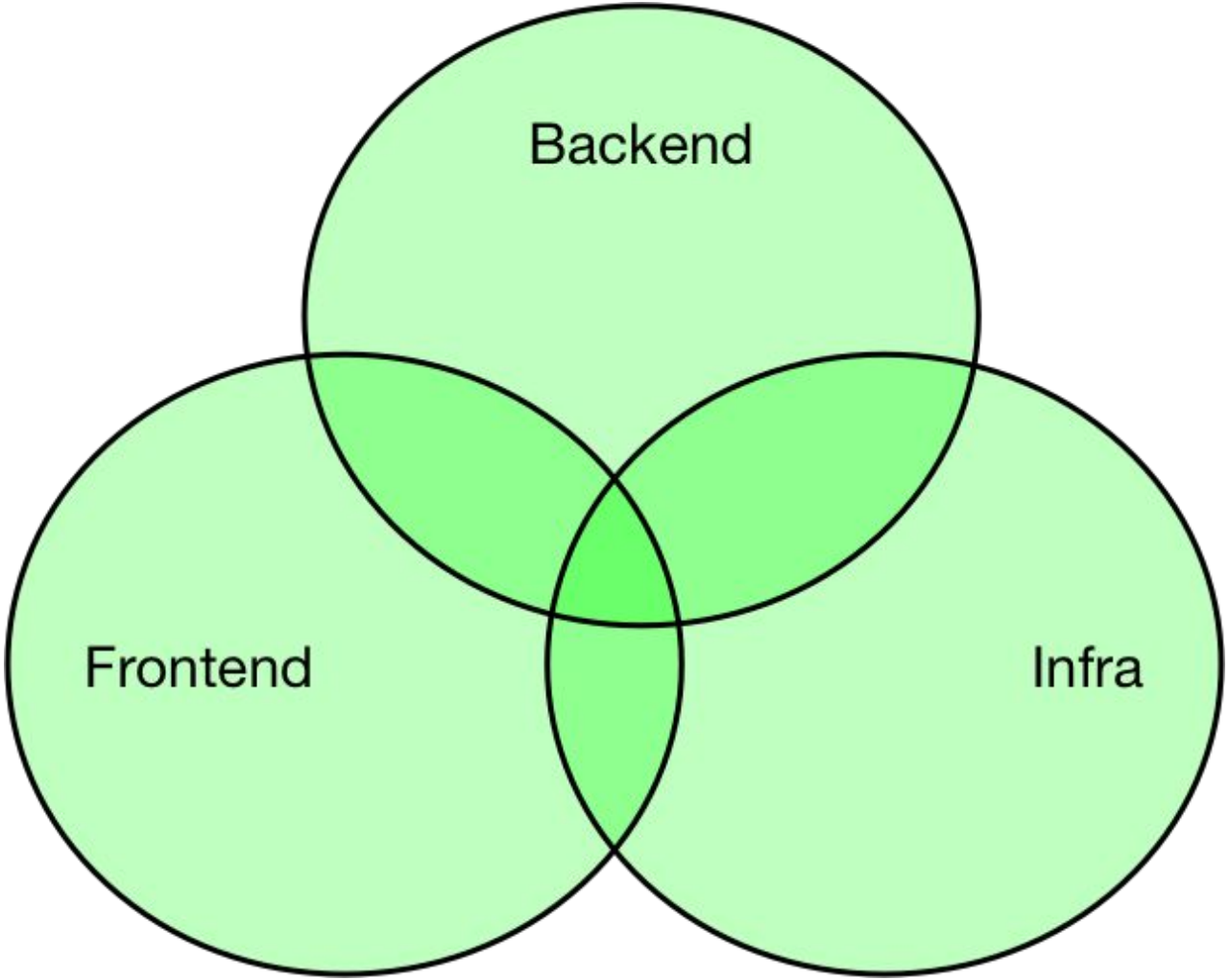
Do migrations matter?

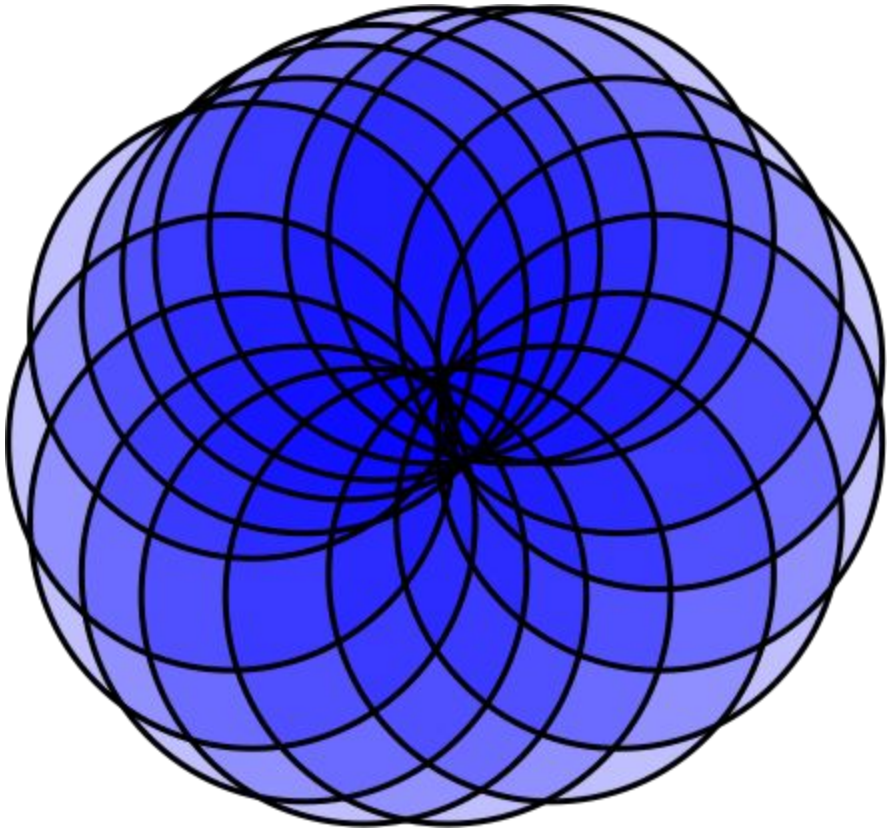
Effective migrations!

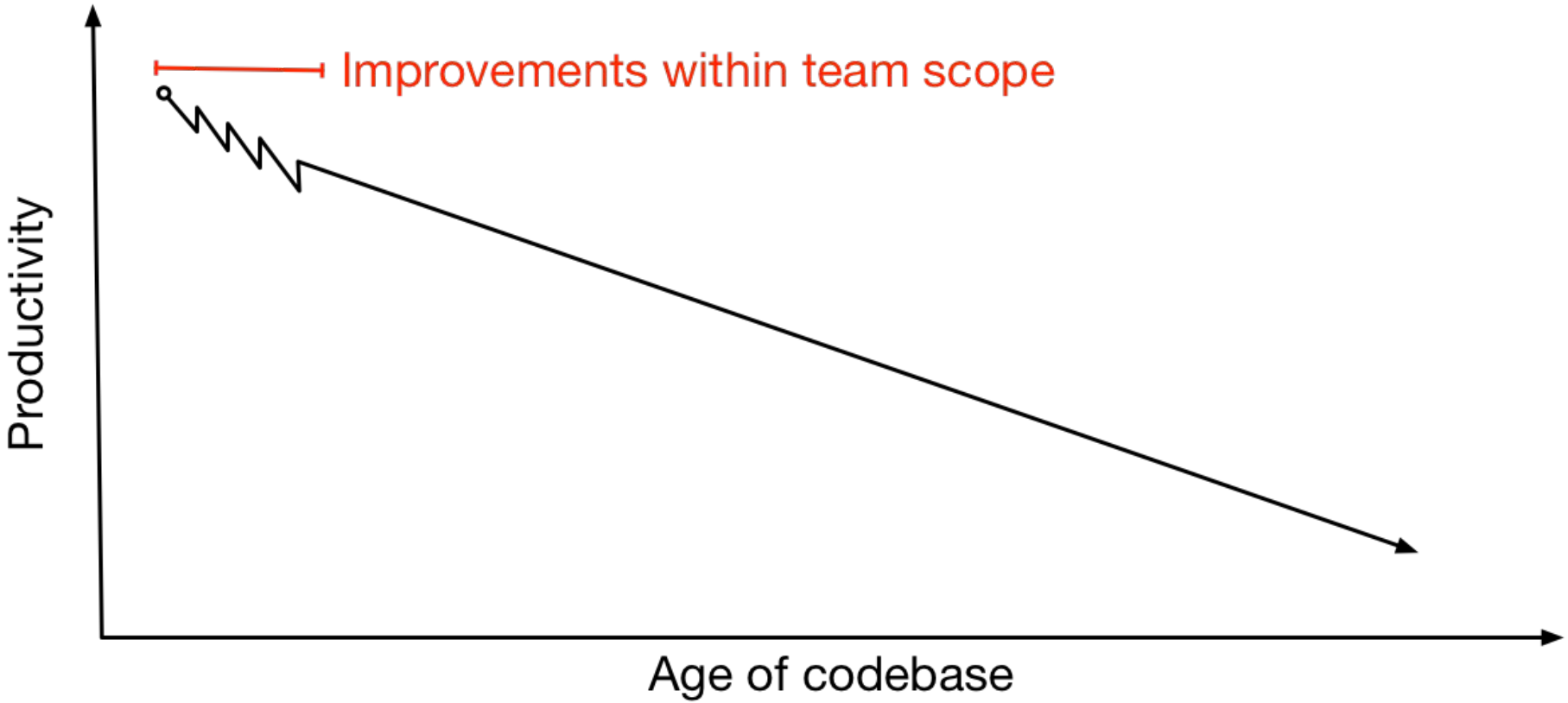

Productivity
Age of codebase

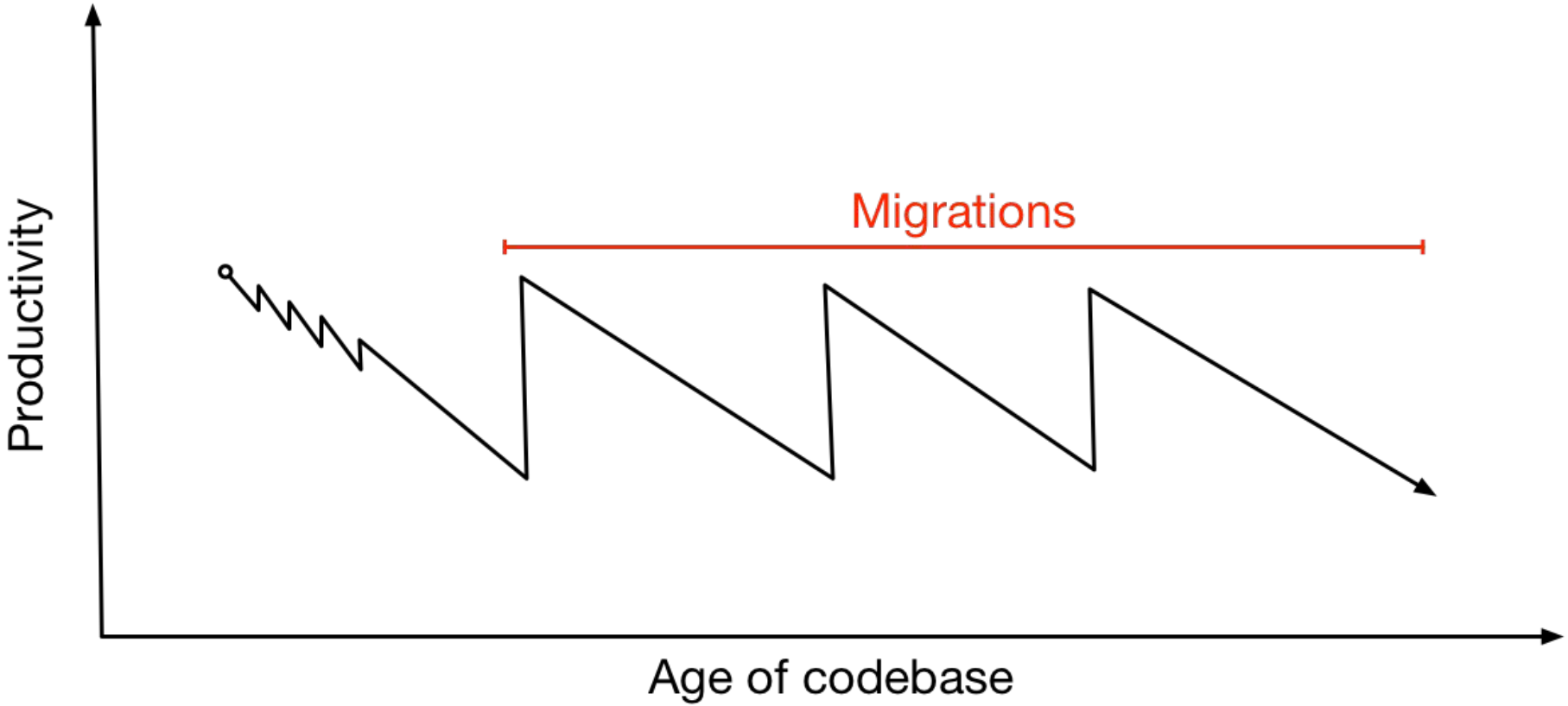

Code review
Continuous integration
Linting
Typed schemas
Productivity
Age of codebase

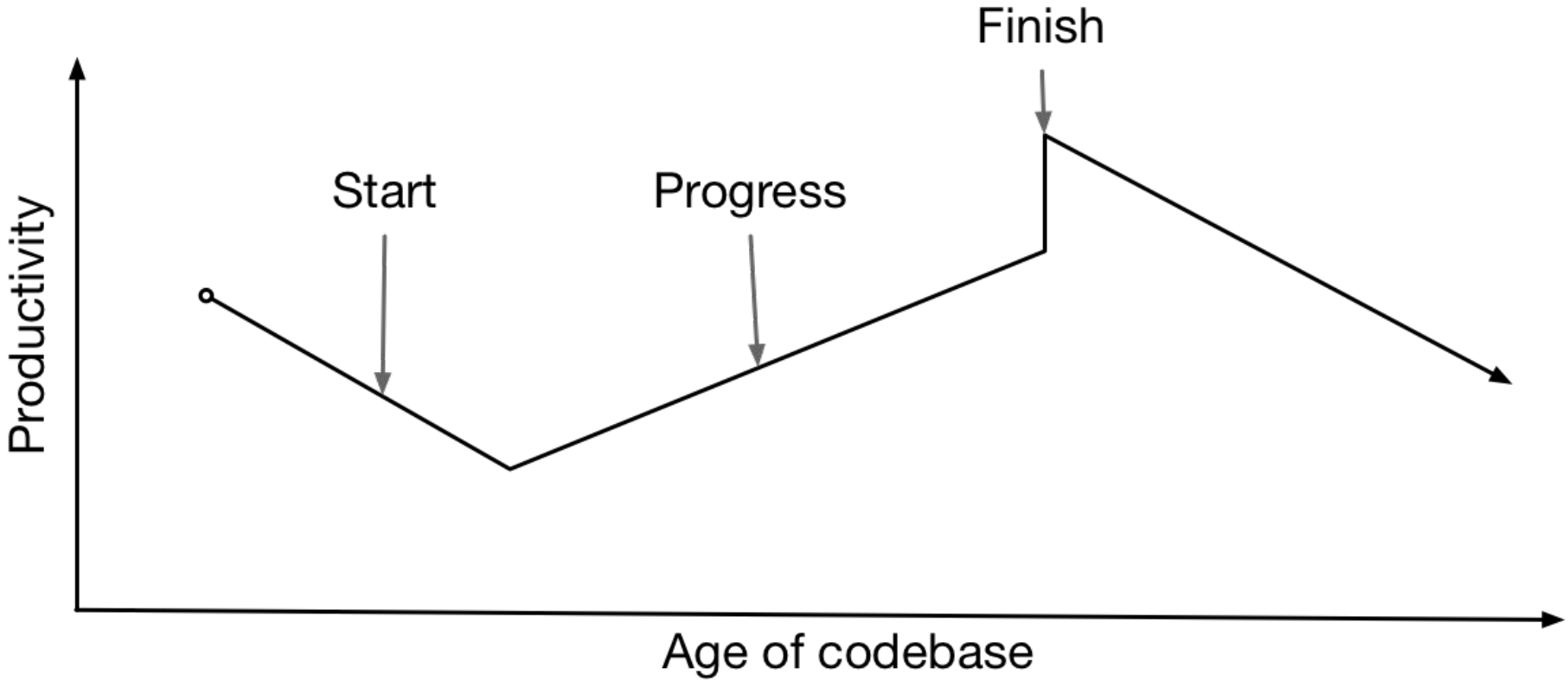

Productivity
Why did things stop improving?
Age of codebase

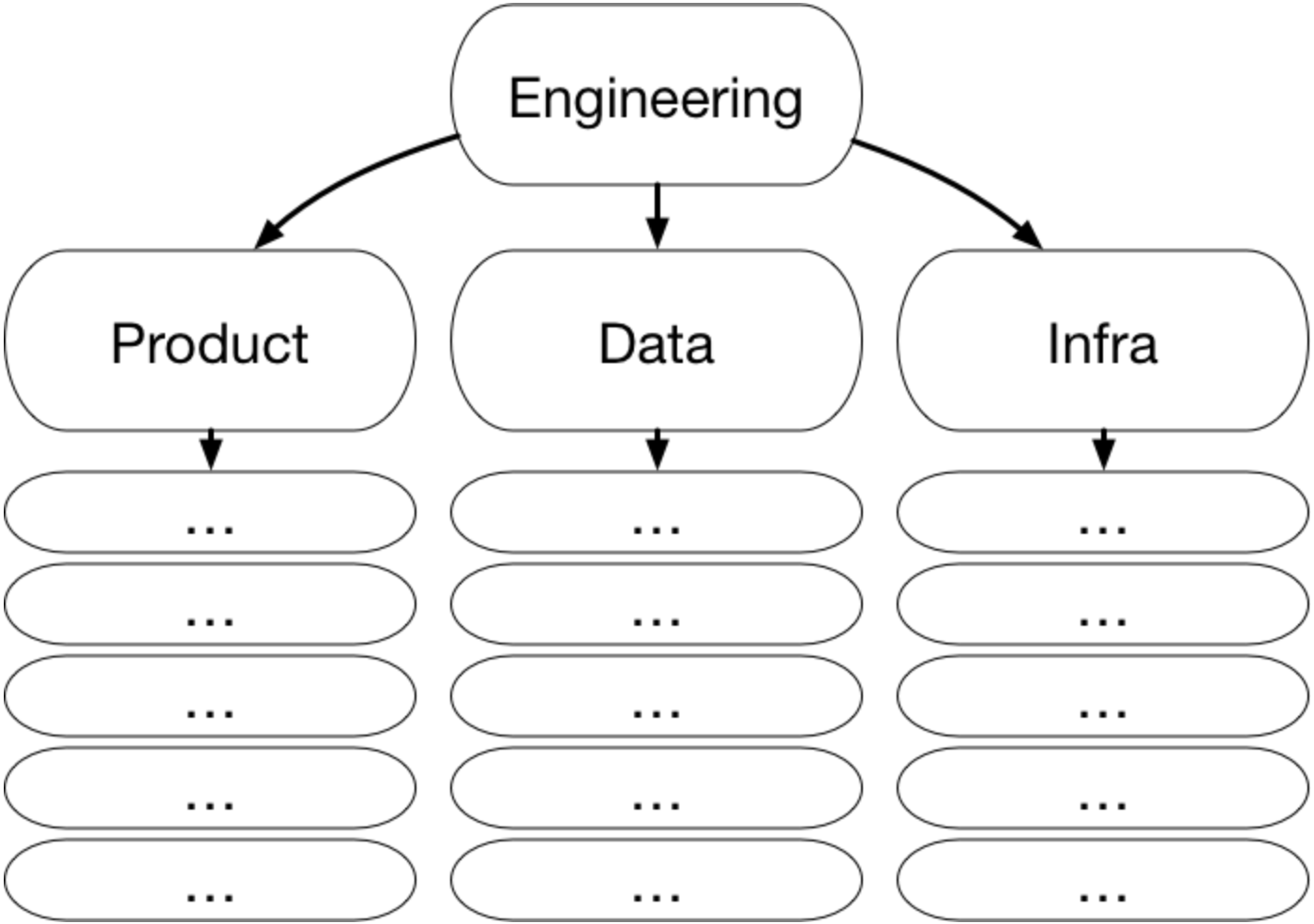

Backend
Frontend
Infra

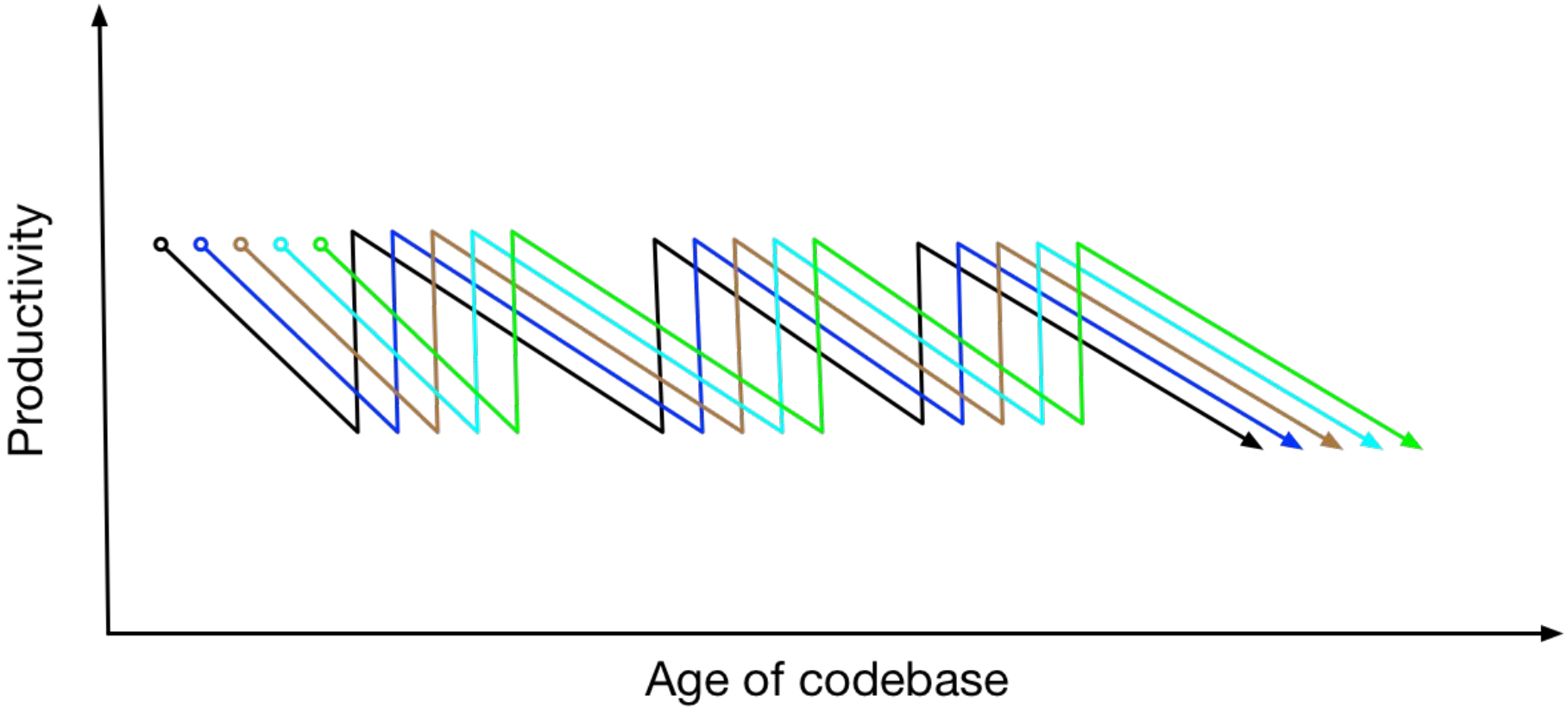


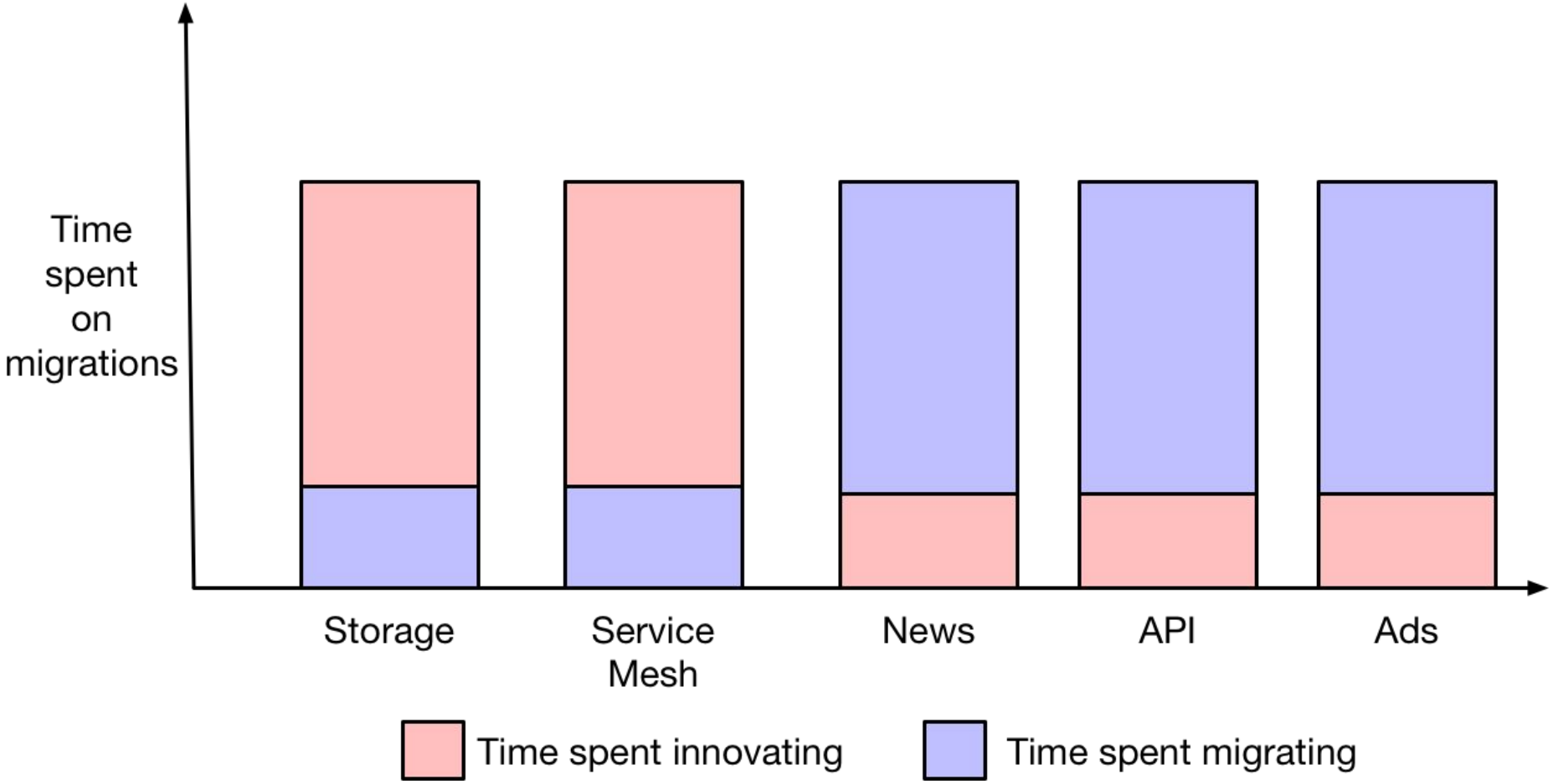

Improvements within team scope
Productivity
Age of codebase


Migrations
Productivity
Age of codebase


Finish
Start
Progress
Productivity
Age of codebase

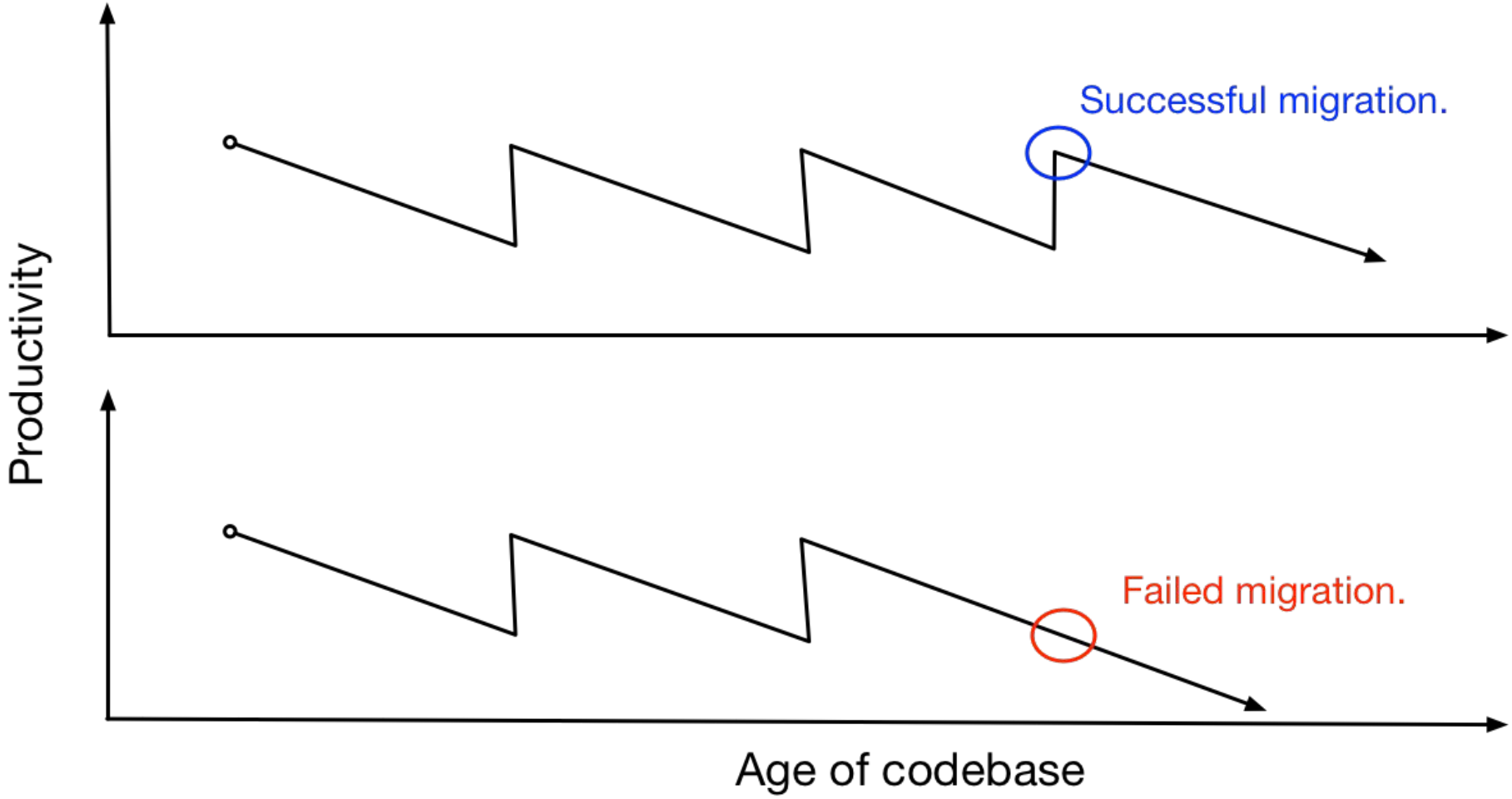

Engineering
Product
Data
Infra
...
...
...
...
...
...
...
...
...
...
...
...
...
...
...


Productivity
Age of codebase


Time spent on migrations
Storage
Service Mesh
News
API
Ads
Time spent innovating
Time spent migrating

So now you’re busy...

...and it gets worse!


Successful migration.
Productivity
Failed migration.
Age of codebase

Hyperbahn.

uber-archive / **hyperbahn**

Watch 1,882 | Star 380 | Fork 50

Code | Issues 25 | Pull requests 15 | Projects 0 | Wiki | Insights

Service discovery and routing for large scale microservice operations

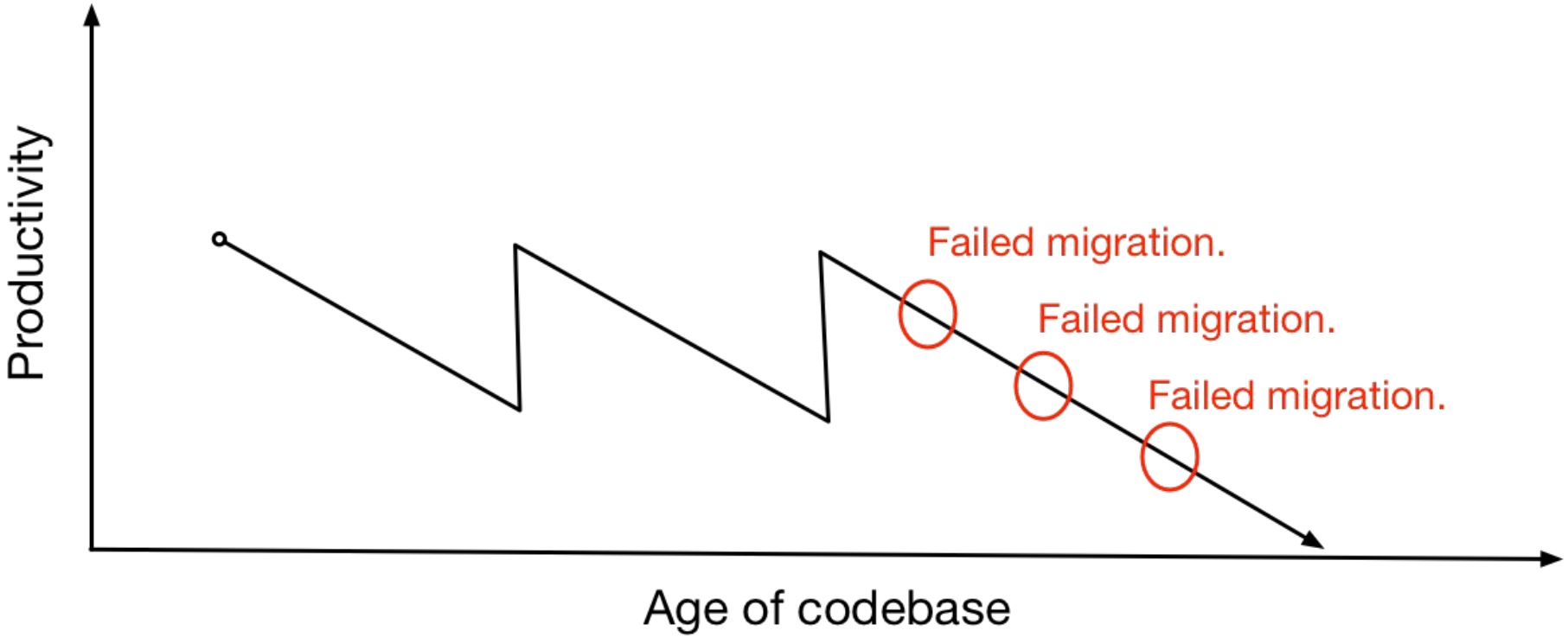
1,444 commits | **41** branches | **140** releases | **15** contributors | MIT

Branch: **master** | New pull request | Create new file | Upload files | Find file | Clone or download

kriskowal 2.20.1 — Latest commit 3b5ebee on Oct 24, 2016

benchmarks	enable cheat configurations	3 years ago
bin	remove obsolete debug log	3 years ago
clients	Ease the connection stale period to 10s (#314)	2 years ago
config	In case of emergecy, raise ping req timeout (#313)	2 years ago

Teams bet on migrations.


Productivity
Failed migration.
Failed migration.
Failed migration.
Age of codebase

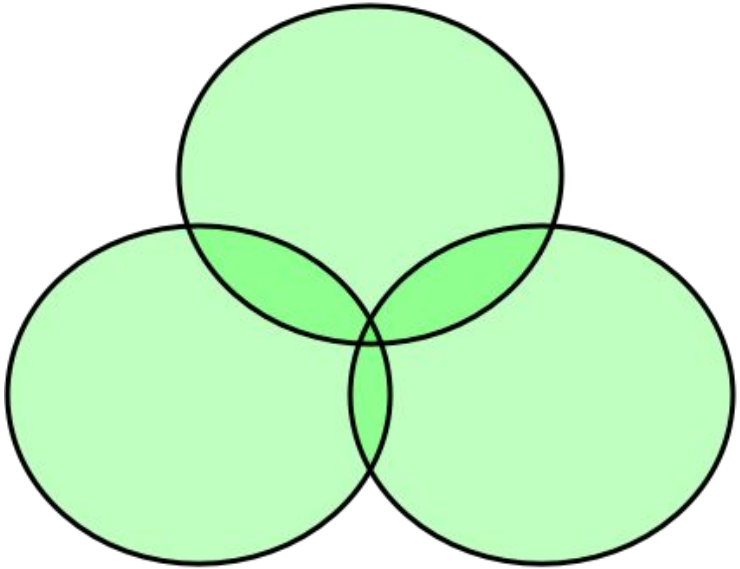
Digg v4.

So... do migrations matter?

Yes.

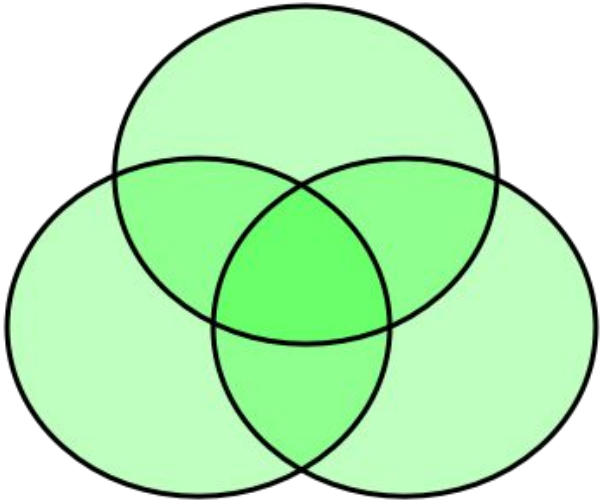
Quick aside: interfaces!

Strong interfaces

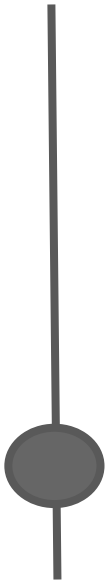


Most implementation
within team scope

Weak interfaces



Most implementation
within shared scope



What is a migration?

Do migrations matter?

Effective migrations!

Every migration is a product.

Effective migrations!

Derisk.

Enable.

Finish.

Place good bets.

1) Is this worth doing?

Find sponsor.

Opportunity cost.

Not invented here.

Not here.

Hammer looking for a nail.

Hammer
for a n

2) Will it work?

Design document.

Prototype.

Embed with early adopters.

Blog › Engineering

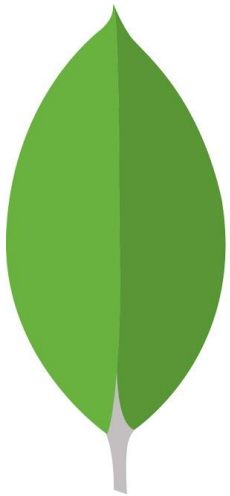
Share this post on Twitter

Learning to operate Kubernetes reliably

Julia Evans on December 20, 2017 in Engineering

We recently built a distributed cron job scheduling system on top of Kubernetes, an exciting new platform for container orchestration. Kubernetes is very popular right now and makes a lot of exciting promises: one of the most exciting is that engineers don't need to know or care what machines their applications run on.

One easy, then hard.



mongoDB

Effective migrations!

Derisk.

Enable.

Finish.

User testing.

User testing: interfaces.

User testing: documentation.

User testing: operations.

User testing: debugging.

Slow down to go fast.

Self-service.

Automate the migration.

Sorbet:
A Typechecker for Ruby

Dmitry @darkdimius Petrashko
Nelson @nelhage Elhage
Paul @ptarjan Tarjan

Incremental and reversible tools.

Dark launch.

Interfaces.
Interfaces.
Interfaces.

Effective migrations!

Derisk.

Enable.

Finish.

100%

Stop the bleeding.

μContainer.

Tracking.

Tracking: tickets.

Tracking: reports.

Tracking: nudges.

Finish it yourself.

How to Break Up With Your Vendor

Amy Nguyen (@amyngyn)
Cory Watson (@gphat)

Velocity NYC
October 2, 2018

Celebrate when it’s over.

What is a migration?

Do migrations matter?

Effective migrations!

Tech debt:
core constraint on your velocity.

Migration:
only way to pay tech debt at scale.

Approach:
treat every migration like a product.

Q&A

@lethain

https://lethain.com