Platform

The History of Platform Insights

How Platform became data informed

@yarin

November 20, 2024

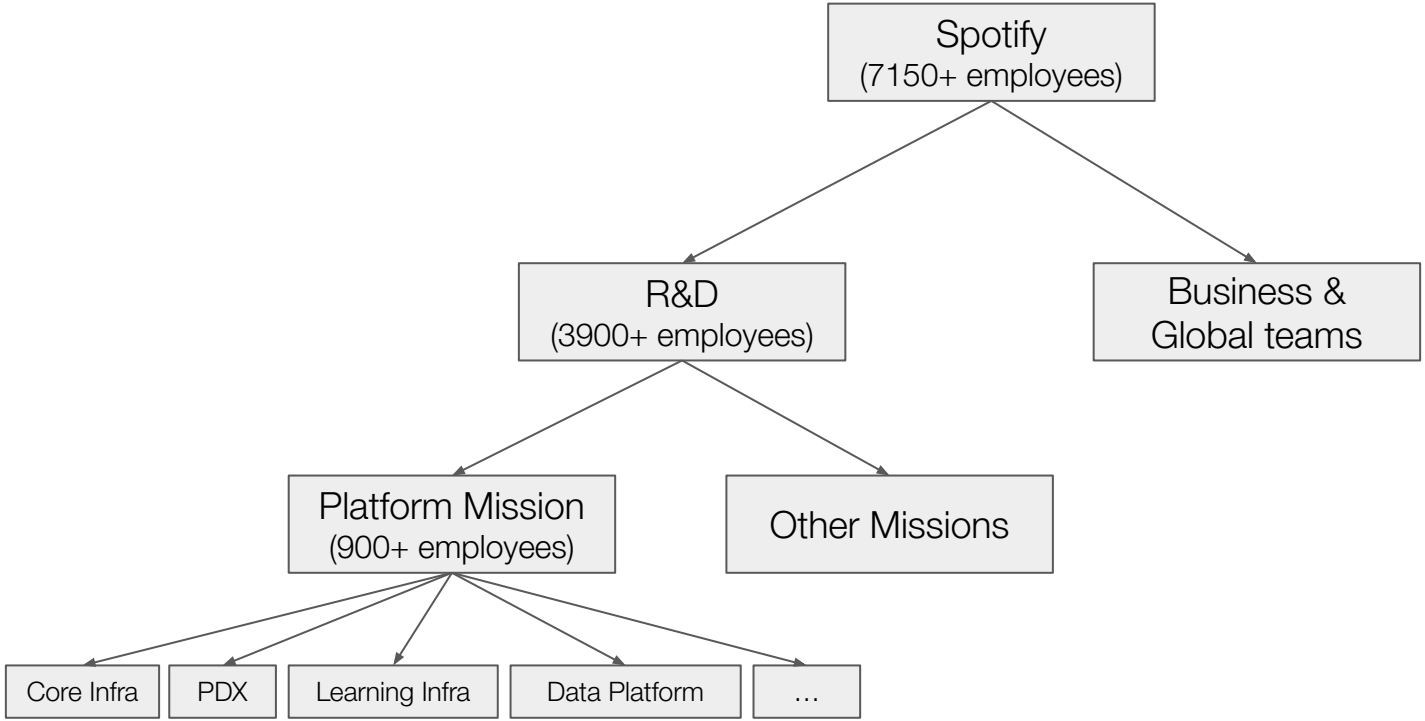
About me

- Jimmy Mårdell (@yarin)
- Staff Engineer
- 14 years at Spotify
- 6 years in Platform Insights

yarin@spotify.com
https://www.linkedin.com/in/jimmymardell/

Spotify & Platform

Spotify & Platform

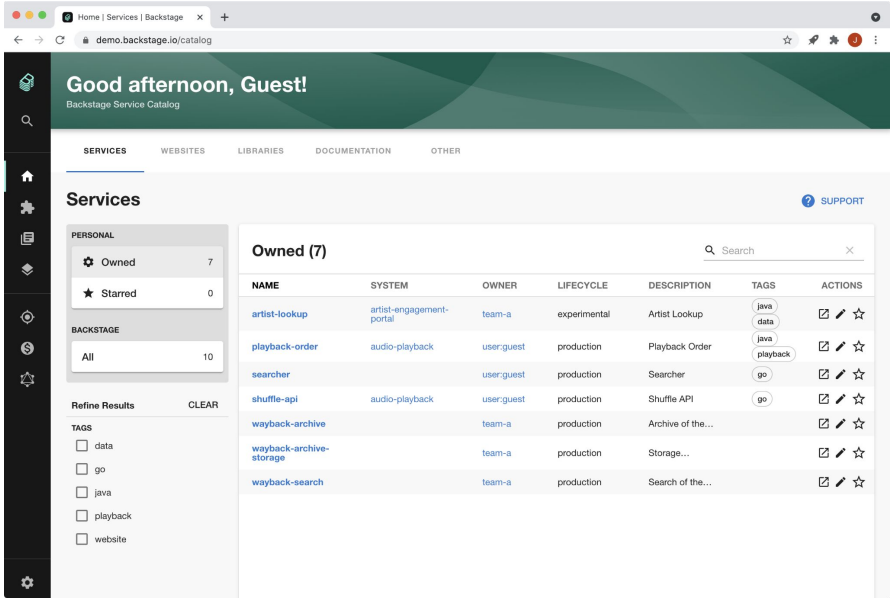


Platform Mission scope

- Developer Experience
  - CI/CD, version control
  - Backstage
  - IDE's / GenAI
  - Streamlined developer user journeys (Golden Path's)
- Infrastructure
  - Service control plane, monitoring
  - Fleet management / Kubernetes
  - Data ecosystem
  - Core libraries
  - End-user authentication
- Security
- …and a lot more

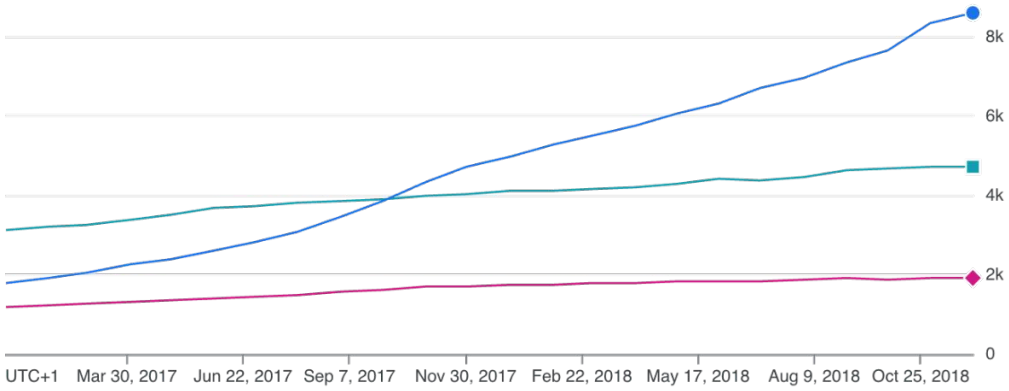
Backstage

- Developer portal
- Tying together the Spotify Tech Ecosystem since 2015
- Open Source since 2020
  - Also an Enterprise product



R&D in 2018

- 1800 employees in R&D
- Manually managed org chart (json!)
- Getting harder for any one person to "know" the whole picture
- Continuous hypergrowth



Between 2017-2019, number of employees in R&D (🔴) increase by 60%, number of components (🔵) increased by 370%

Are we getting slower??

Hard to answer questions such as

- How many Android developers do we have?
- What are they working on?
- Which disciplines are struggling the most? Where should we invest?
- Does it take longer time to ship features today than a year ago?

Ad-hoc investigation started by curious engineers

Problem statement

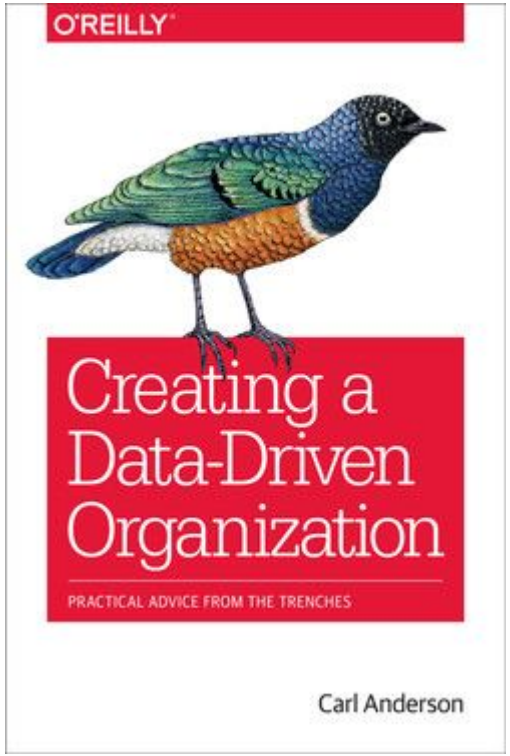
"[Platform] is not operating with a data informed approach. Data is not collected to the extent needed and the collected data is not accessible for analysis in a simple manner. We rely to a large extent on qualitative data, experience and external requirements in decision making.

We are not able to measure our impact on customers and the ROI to the extent desired.

As a result our processes and practises are not relying on data and we are lacking high quality success metrics to guide and validate our decision making."

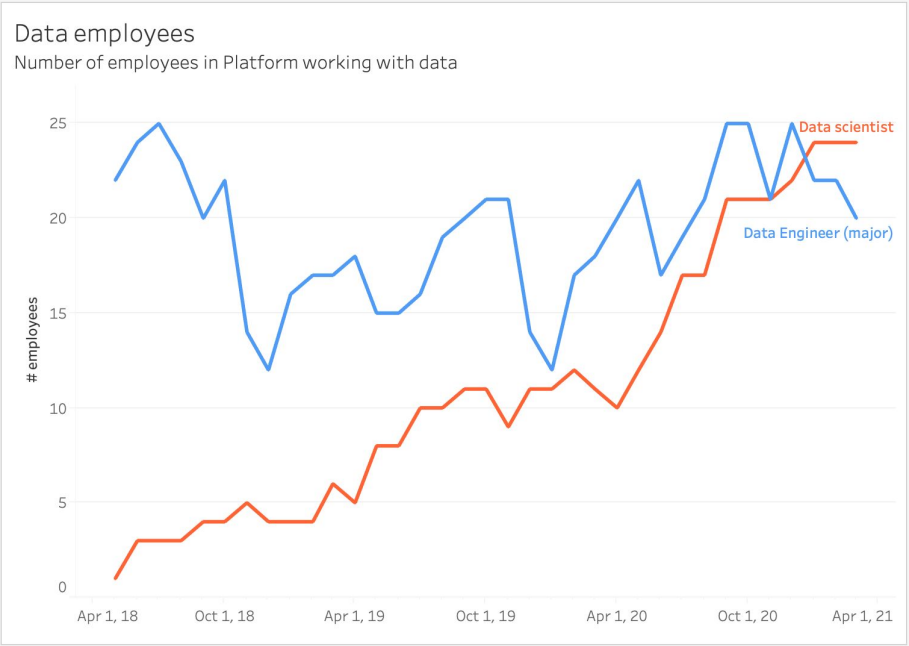
Requirements for a Data-~~Driven~~Informed Organization

- Data collection
  - Relevant, timely, accurate, clean, unbiased, trustworthy
- Data accessibility
  - Joinable, queryable, shareable
- Culture
  - Open trusting
  - Broad data literacy
  - Goal-first
  - Inquisitive, questioning
  - Iterative, learning
  - Anti-HiPPO



Platform goes data

- Platform Insights created in 2018
- More specialised insights teams created in Platform the upcoming years



Data learnings

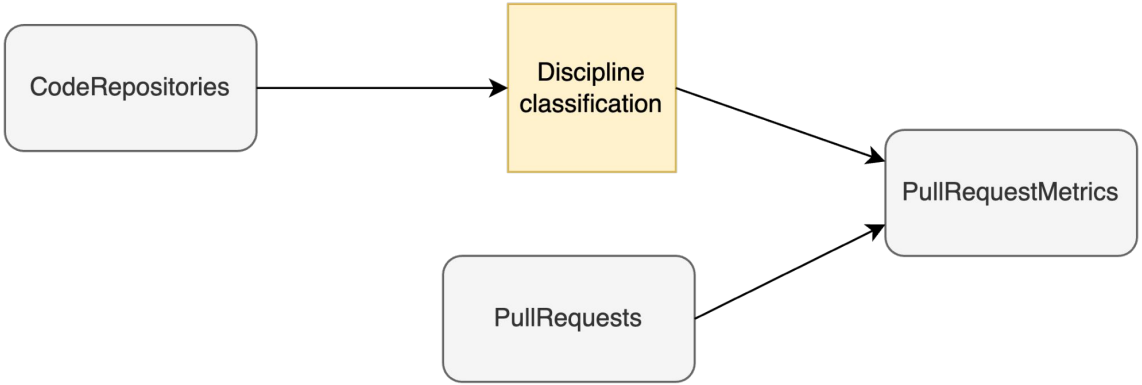
- Engineering disciplines
- Onboarding
- Tech Architecture
- etc

Disciplines & T-shapedness

- Engineers have different skill sets, or *disciplines*
  - Backend, Data, Web, Android, iOS, ML etc
- Level of expertise in a particular discipline is either *deep* or *shallow*
- T-shaped engineers are deep in one discipline and shallow in (at least) one other

How can we measure this?

Disciplines & T-shapedness



Disciplines & T-shapedness

	% of Pull Requests last 90 days	Major	Minors
Alex	Android 20%, Backend 65%, Other 15%	Backend	Android
Brandon	Android 30%, iOS 40%, Other 30%		Android, iOS
Claire	Data 90%	Data	
David	iOS 30%, Backend 20%, Data 25%, ML 25%		iOS, Backend, Data, ML

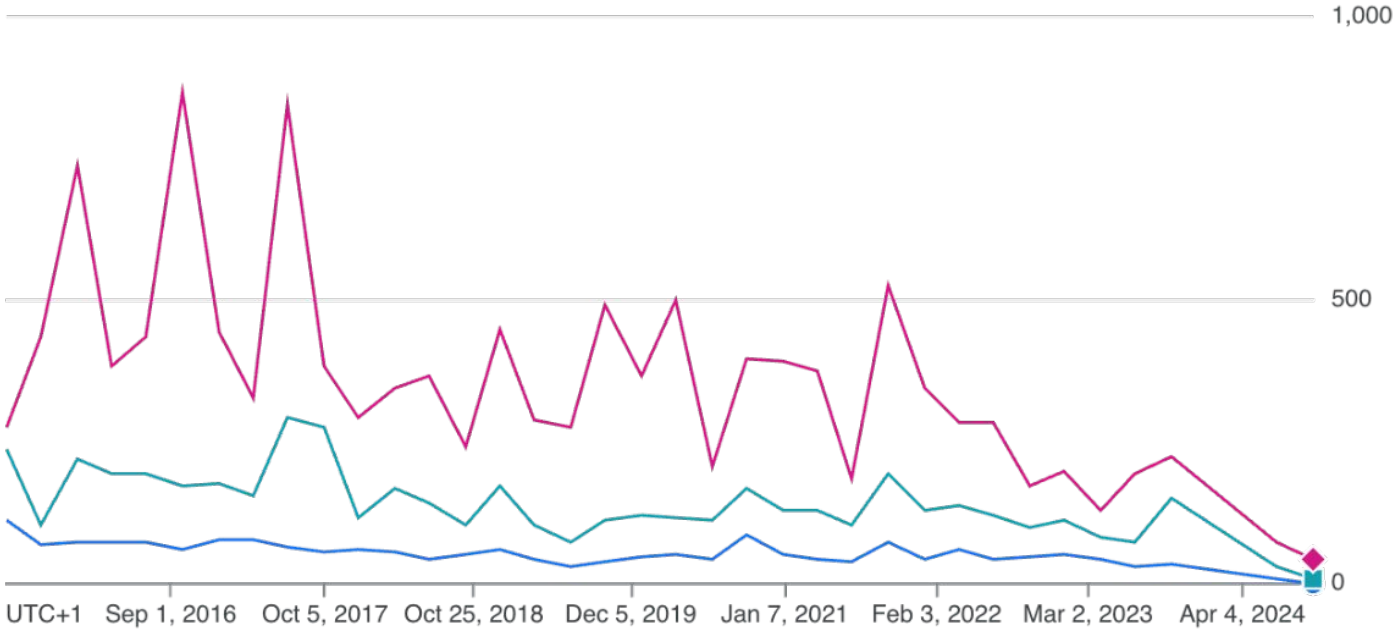
Onboarding

- Indications that it took long time for new hires to get up-to-speed
- How effective were our bootcamps?
- How effective were our Golden Paths?

How to measure this?

Onboarding

days 10th PR (p50, p75, p90)

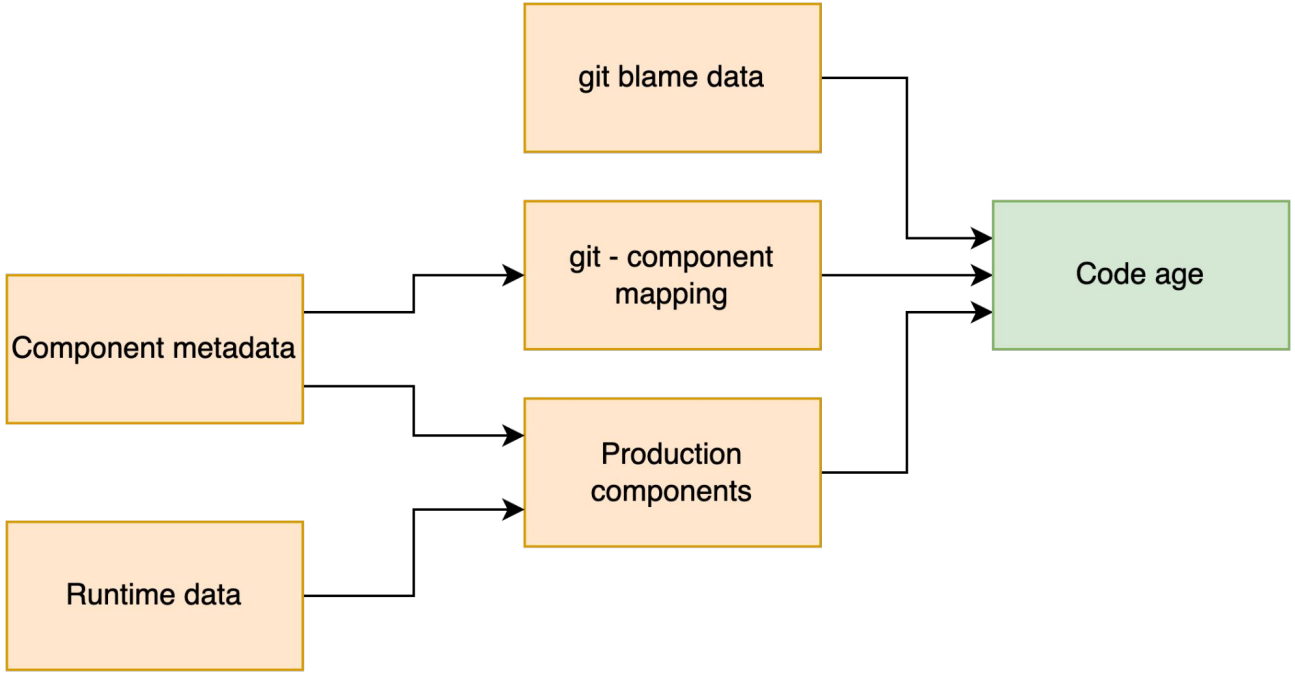


Tech Architecture

- To what extent is the code in production still being maintained?
- Do we even understand all the code running in production?
- How interdependent is our code base?
- Can our teams work in isolation?
- How will the Spotify service be affected if we disband team X?

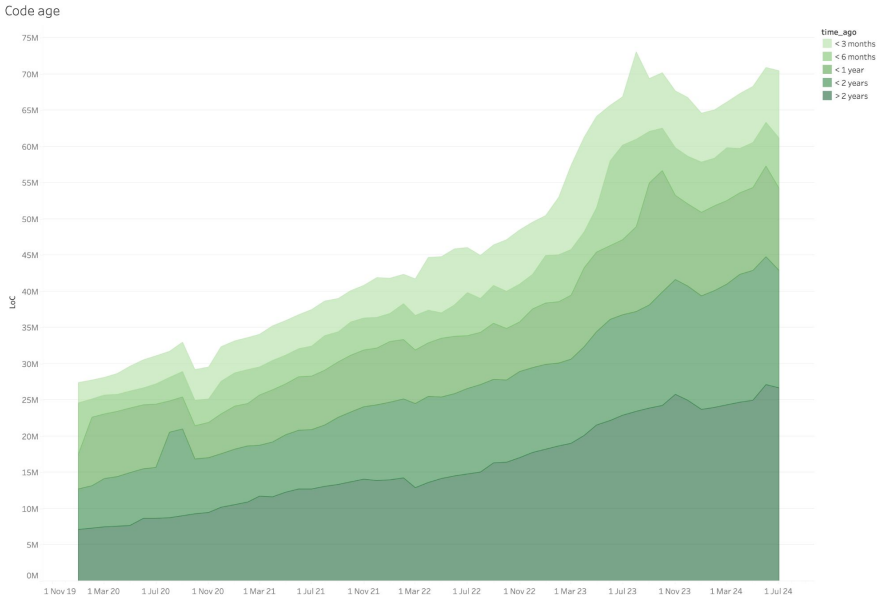
How can we measure or visualize this?

Tech Architecture



Tech Architecture

Code age

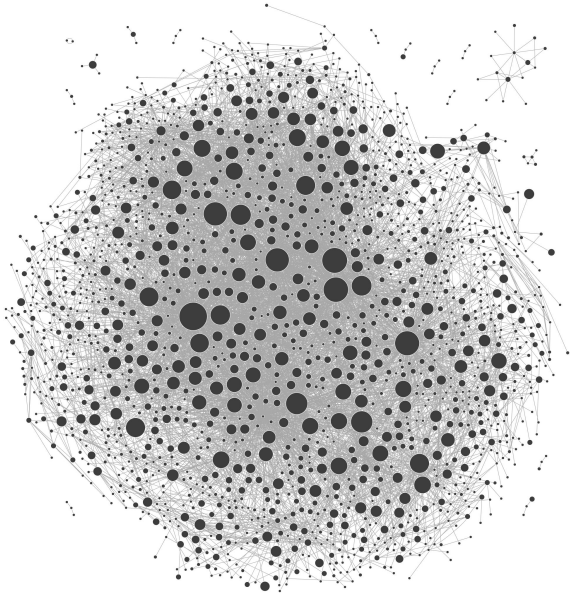


The plot of sum of num_lines for date. Color shows details about time_ago. The view is filtered on date, which ranges from 2020-01-01 to 2024-07-23.

```
WITH ranges AS (
  SELECT fa.date, TIMESTAMP_DIFF(TIMESTAMP(fa.date),
r.commit.author.date, day)
    AS delta_days, r.num_lines
  FROM `ghe_file_annotations.ghe_file_annotations_*` AS fa
  INNER JOIN `production_components.production_components_*` AS pc
    ON fa.main_component_id=pc.component AND
       fa._table_suffix=pc._table_suffix
  CROSS JOIN UNNEST(ranges) AS r
)
SELECT
  date,
  CASE
    WHEN delta_days < 91 THEN '< 3 months'
    WHEN delta_days < 182 THEN '< 6 months'
    WHEN delta_days < 365 THEN '< 1 year'
    WHEN delta_days < 365*2 THEN '< 2 years'
    ELSE '> 2 years'
  END AS time_ago,
  SUM(num_lines) AS num_lines
FROM ranges
GROUP BY date, time_ago
```

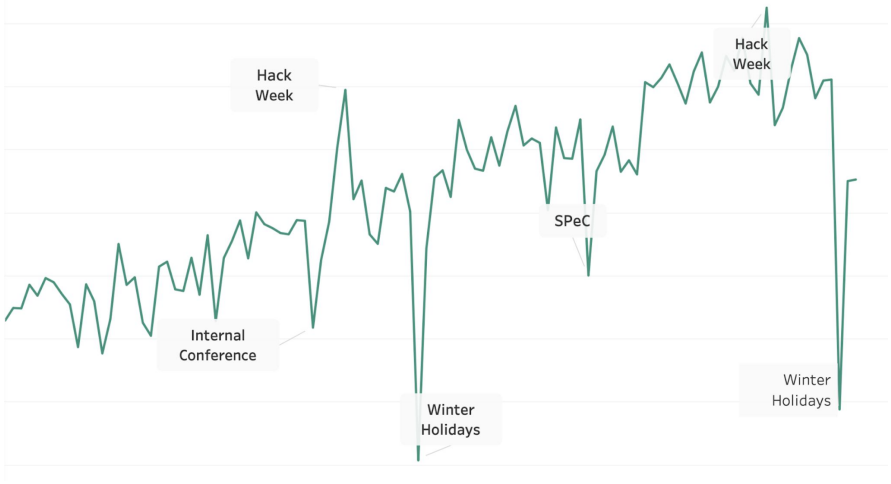
Tech Architecture

Backend Service dependency graph



Other small random data tidbits

- Understanding the impact of Log4shell
- Impact of Working from Anywhere
- Productivity spikes during Hack Week



Data to collect

Starting points

- People and organization chart
- Source code
  - Pull Requests and Code reviews
  - Actual contents of all source code files
- Software metadata
  - Backstage components
  - Dependencies (service mesh, API and lib usage, data lineage etc)
- Server logs and dumps of data from internal systems
  - CI/CD systems
  - Alerting and monitoring
  - JIRA

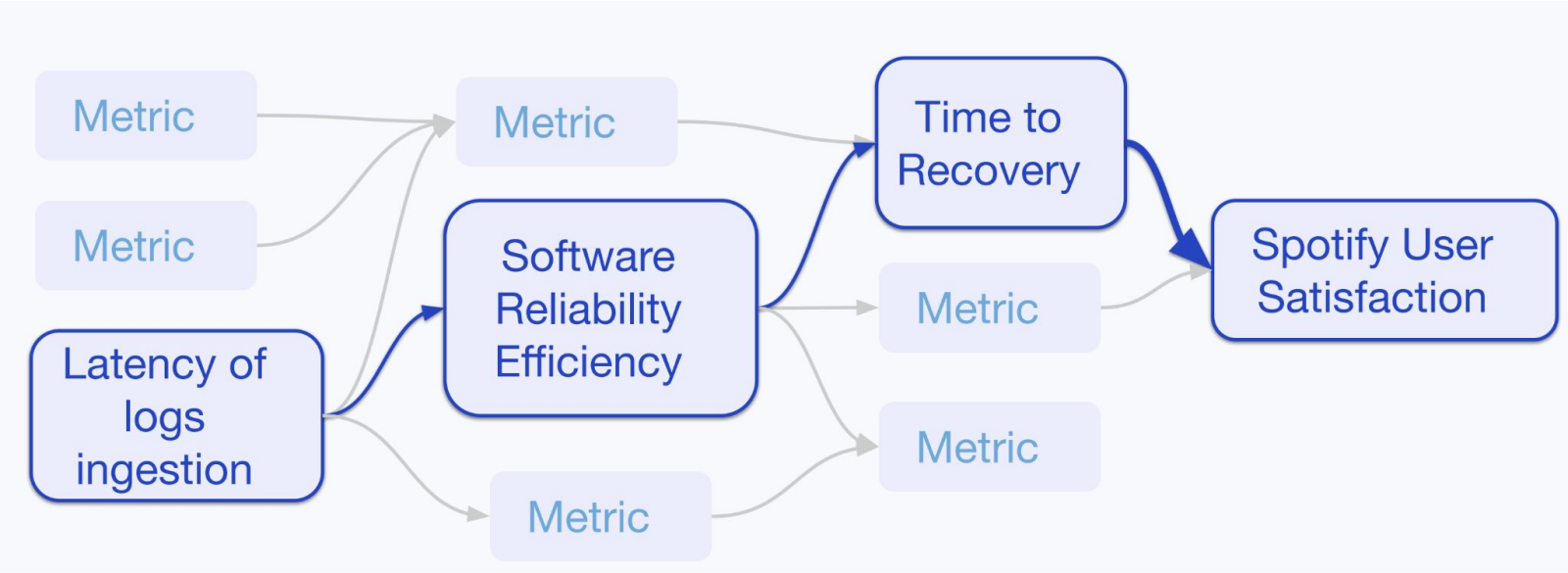
Metrics

- Mental model
- Best practices
- Some metrics we use

A mental model for metrics

Leading	Lagging
Action-focused	Value / Impact-focused
Noisy	Low noise
Actionable	Move slowly
Tactical	Strategic
Gameable	Trickier to measure
Vanity Metrics	

Metrics influence each other



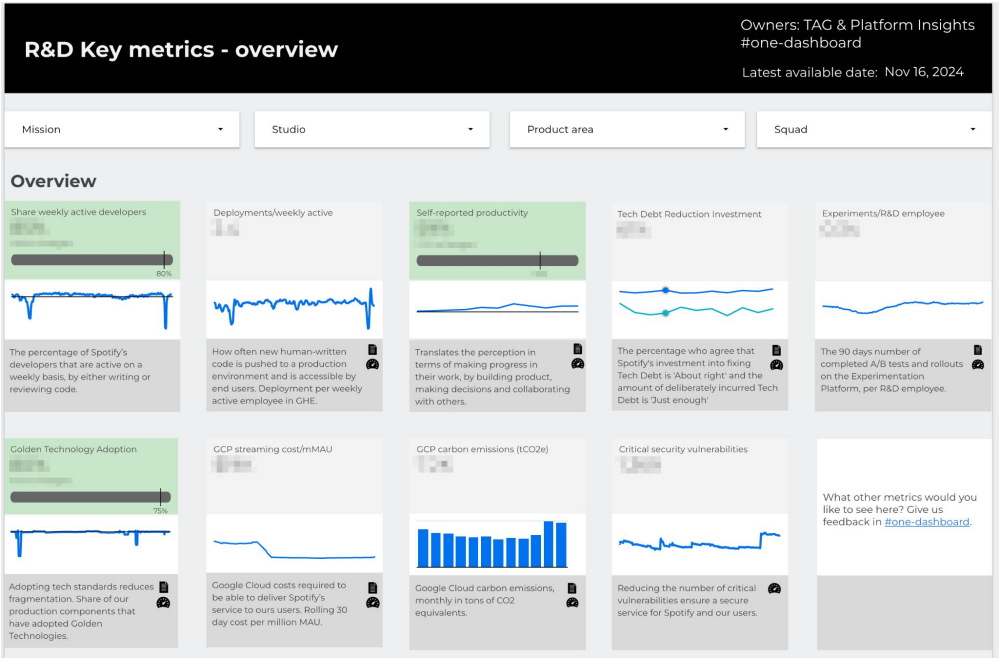
Metrics best practices

- There's **no one single metric** to capture developer productivity
- Survey data are also metrics
- Target setting requires a base line
- Metrics are products
- Metrics need to be drill-downable

R&D Key metrics

High value metrics

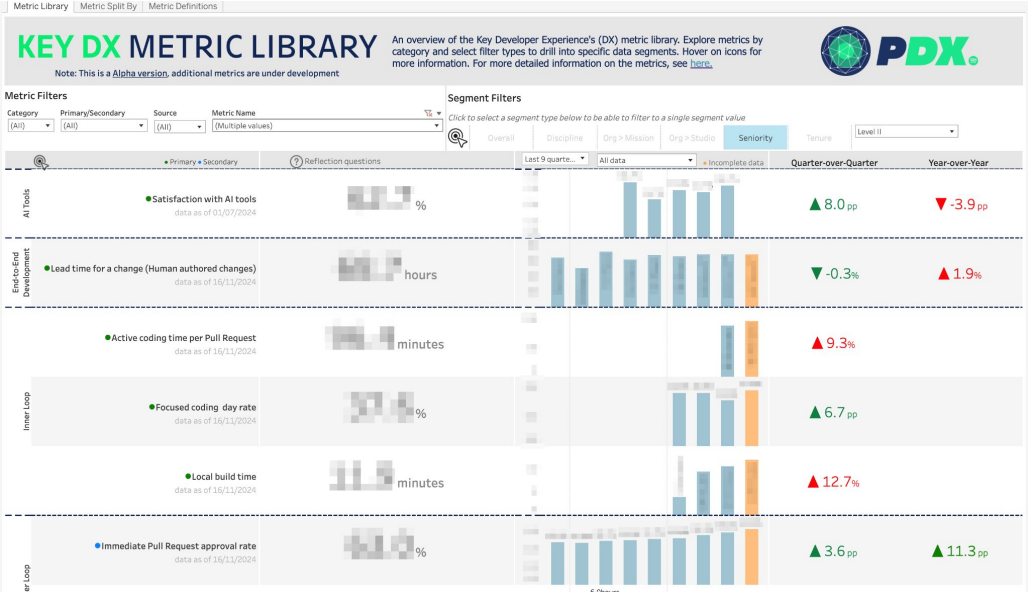
- Deployment Frequency
- Self-reported productivity
- # completed A/B tests
- # security vulnerabilities
- Golden Technology adoption
- Streaming costs



Key DX metrics library

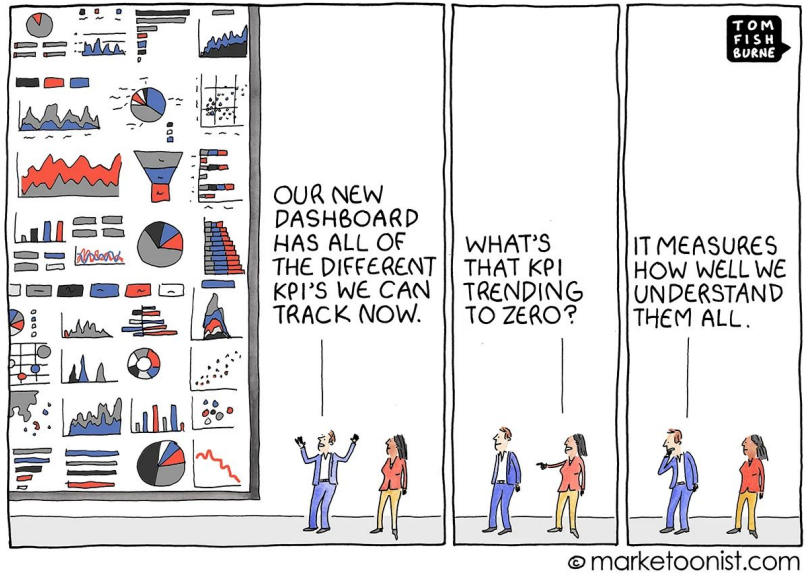
Selection of DX metrics

- Satisfaction with AI tools
- Lead time for Change
- Active Coding time per PR
- Focused Coding (day rate)
- Local build time
- Immediate PR approval rate
- Time waiting for Code Review



A word of caution…

- With great metrics comes great responsibilities…
- Be **data informed**, not **data driven!**



Key takeaways

Platform data and metrics are different!

Define a set of KPI's that you want to track over time

Get the data - one dataset at a time

Data must be easily available and trustworthy all the time

Invest in a longitudinal developer survey

Be *data informed*, not *data driven*

Data and metrics creates a common language

Thank you