Embracing DORA Metrics at Airbnb
Measuring Developer Productivity
Measuring Developer Productivity
Measuring Developer Productivity
Measuring Developer Productivity
Measuring Developer Productivity
Measuring Developer Productivity
Measuring Developer Productivity
Measuring Developer Productivity
Measuring Developer Productivity

Everything pizza
How to measure developer productivity
How to measure developer productivity

- PR frequency
- Time to 10th PR
How to measure developer productivity

- PR frequency
- Time to 10th PR
- LoC
How to measure developer productivity

- PR frequency
- Time to 10th PR
- LoC
- LoC
How to measure developer productivity

- PR frequency
- Time to 10th PR
- LoC
- LoC
- Saw a blog
How to measure developer productivity

- PR frequency
- Time to 10th PR
- LoC
- LoC
- Single metric
- Saw a blog
How to measure developer productivity

- Time to 10th PR
- PR frequency
- Tell me what to fix
- Single metric
- LoC
- LoC
- Saw a blog
How to measure developer productivity

- Time to 10th PR
- LoC
- PR frequency
- Tell me what to fix
- Single metric
- Saw a blog

So what?
What changed?
DORA to the rescue

Industry standard metrics

Research backed

Get past the initial debate
Implementation
In theory, a metric like lead time is super easy. But in reality, there’s a ton of nuance. Do we look at when code gets committed to a local branch, a feature branch, or to the trunk? When is code in production – is it when we roll it out to a beta user, to 5% of our users, or to 100% of our users?

On the surface, these metrics are simple and easy to understand. But once you dig in, there can be a ton of nuance to them. My recommendation is to agree to something, and be very clear and transparent in what you’re measuring.

Nathen Harvey, Google
Monorepo release pipeline

A developer makes changes to services A & B and submits a PR

1

Service A

Service B

After merge, snapshots are created for each service change

2

Snapshot A

Snapshot B

Snapshots are deployed independently by service owners

3

Prod

Prod

Lead Time for Service A

Lead Time for Service B
A developer makes changes to services A & B and submits a PR

After merge, snapshots are created for each service change

Snapshots are deployed independently by service owners

Monorepo release pipeline
Calculating Lead Time

1. PR Lead Time
2. Lag Time to Deploy
3. Deploy Time
Calculating Lead Time

1. PR Lead Time
   - Time to 1st review
   - Time in review
   - Merge lag time

2. Lag Time to Deploy

3. Deploy Time
What did we learn
Lag Time to Deploy was a major bottleneck

1. PR Lead Time
2. Lag Time to Deploy
3. Deploy Time

>85% of lead time
DORA Takeaways

Standardize on definitions
Use alongside other signals
Have a plan for driving change
Can set goals but be careful
Don’t boil the ocean
Where DORA worked less well

Measuring impact of specific projects
Not directly actionable
Focused on DevOps pipeline
What’s Next
Beyond DORA

SPACE framework & feedback loops
Identifying and measuring key drivers
Thanks!