Trunk Based Development

How to improve software delivery cycle

<Placeholder> for another eye catching subtitle

23.04.2023 Maksym Prokopov, Devsmeetup Freiburg

Agenda

- Introduction
- Team Performance
  - How to **measure**
- How to **improve**
- Trunk Based Development
- Feature Flags

Introduction

Hello, W0r1d!

The problem

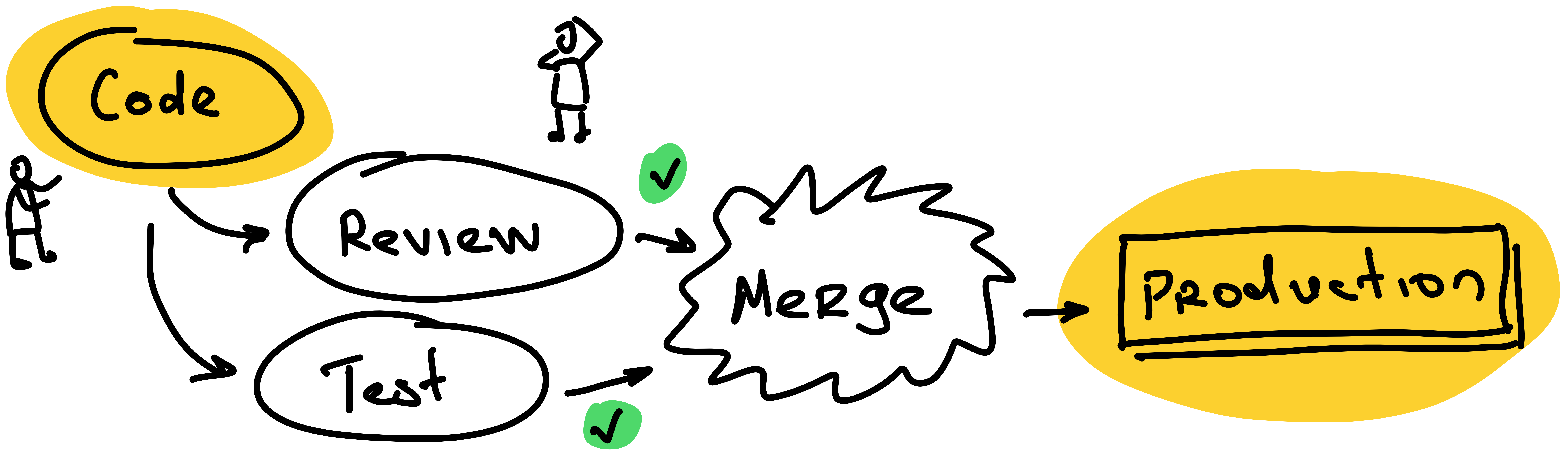
Houston, is there even a problem with performance?

- “If you can't **measure** it, you can't **improve** it”
  Peter Drucker
- What does it even mean, the term **team performance**?

What do we do?
The development cycle

- Code **Changes**
- We ship changes to the **customers**, i.e. to **production**
- Everything else is a **WIP**, sorry, but it doesn't count.

The pipeline



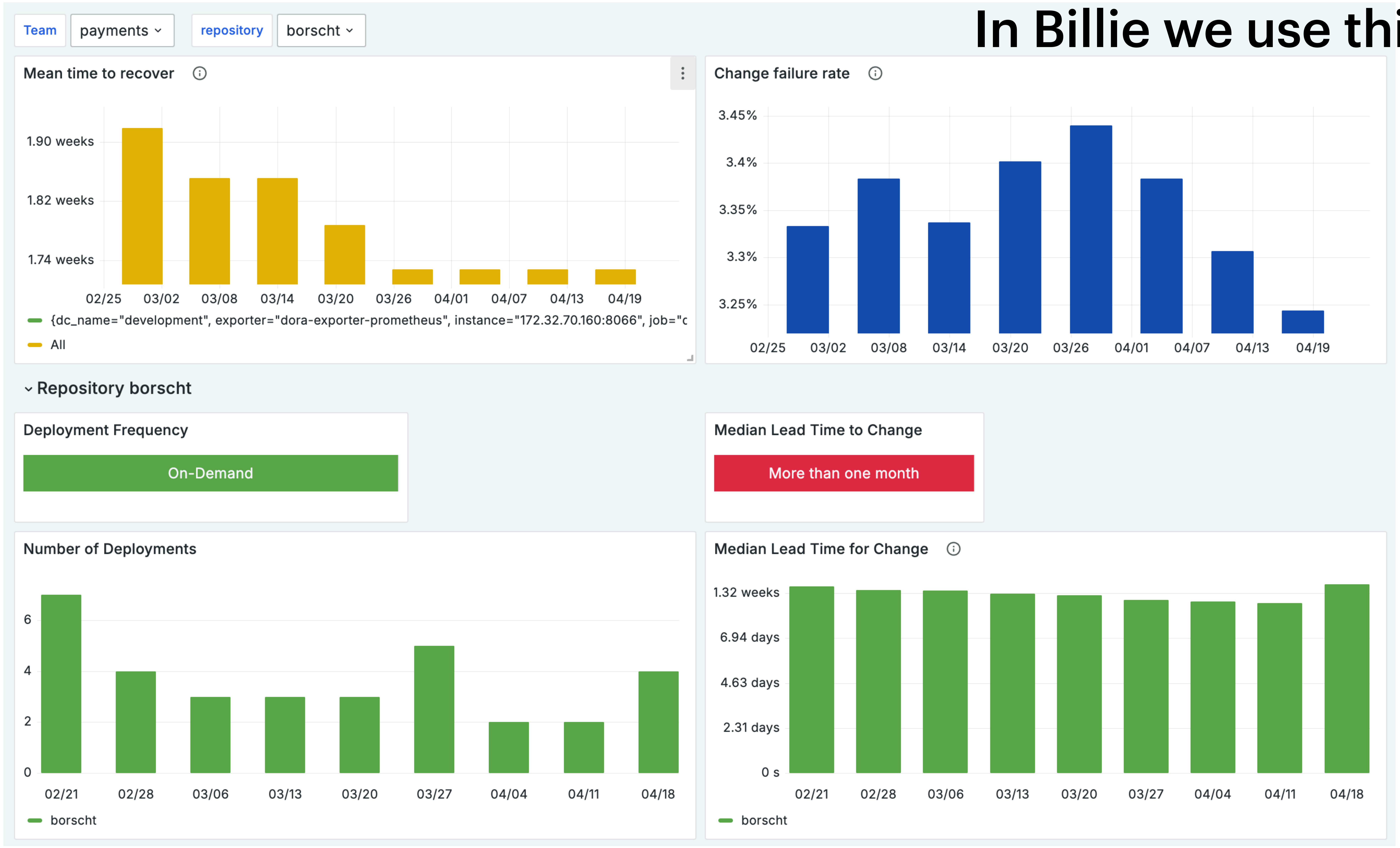
The definition of performance

According to DevOps Research and Assessment

- **Throughput**
  - Deployment frequency
  - Mean Time for Change
- **Stability**
  - Change failure rate
  - Mean time to recover

Metrics Example

In Billie we use this dashboard



Throughput

Deployment Frequency

- How often your changes land production
- The **more** the **better**
- **Confidence** gained by the cadence
- Great teams **deploy multiple times per day**
- Stripe (source: https://newsletter.pragmaticengineer.com/p/stripe-part-2)
  - on average **16,4** deployments to prod every day
  - **5978** deployments to Core API in 2022 🤯

Throughput

Mean Time to Change

- How long the **change** reaches **production** from the beginning
- Or how long is the **PR open in average**
  - Open PR = WIP
- sum(Time of deployment - Time of first commit) / sum(deployments)

Improve Throughput
Source code

- Make the change **smaller**, split one big into many smaller
- Simplify **branch** management
  - Probably, **main** and PR against it is enough
- Make CI/CD **pipelines faster**
  - Find a good balance between **safety** and **speed**

Improve Throughput

Code Review

- Make a change **smaller**, easier to review
- **Don't stick** to strict **review** process if
  - The change is an obvious **one-liner**. Merge it.
  - The change is an output of a **pair programming**. Merge it.
  - The change doesn't require inputs, just **for information**. **Notify** a reviewer **after** you merged it.

Stability

Change Failure Rate

- How often a **change leads** to an **incident**
- The **less** incidents the **better**
- sum(deployments) / sum(incidents)

Stability

Mean Time To Recover

- How **fast** can you **recover from incident** in average
- The **faster** the **better**
- sum(incident duration)/sum(incidents)

Improve Stability

Reduce incident number and impact

- Have a **balanced test** suite
  - Mind testing **pyramid**
  - Use **quality** gates
- Reduce a PR size. The **smaller** the **better**
- Fast or even automated **Rollback**
- **Feature Flags** for even better control
- Advanced **deployment techniques** blue-green, canary

Improve Stability

Making the change smaller

- Split **one big** PR into several **smaller** PRs
- Maybe start with a PR that contains only **tests**
- **Hide** the code that isn't ready yet under the **feature flag**
- Be **creative** and implement **own** solutions

Improve Stability
Feature Flags

- **Simplest** example
  - If feature:
    - # do this
- Typical sources
  - **Environment** variable
  - **Database** or KV store entry
  - **External** Service
- Watch out
  - Feature flags require **maintenance** and **lifecycle** management

Trunk Based Development

- Fancy name for working against **main** branch
- Versus **GitFlow**
- It’s all about keeping changes **small** to **improve** release cycle
- **Stop** creating a branch dedicated to a **feature**
- Do branch “by **abstraction**” instead
- Important to **keep** builds **green**

How to start
Tracking performance vitals

- **Deployment**
  - one deployment = one change
- **Incident + Incident Duration**
- **Change duration**
  - duration from initial commit to merging to **main**
- Metric Dimensions
  - team
  - repository

Want to track DORA metrics?

Solutions

- **Developed in Billie tailored, simplified dora-exporter**
  - Grafana based, no storage required
  - Integrated with GitHub, Jira and Backstage
  - https://github.com/mprokopov/dora-exporter
- **Apache DevLake community driven, feature rich solution**
  - Grafana based, storage MySQL
  - Integrated with GitHub, Jira
  - https://devlake.apache.org/

Summary

- **Reduce** shipping cycle
- Make code changes **small**
- **Simplify** and **improve** code review process
- Improve **tests** and CI/CD pipelines
- Use **feature flags**
- Establish and use **DORA** metrics

Feature Flag Management Tools

- BYO feature flag, custom implementation
- Commercial/Cloud https://launchdarkly.com
- OSS https://www.flipt.io/
- Cloud-provided facilities
  - AWS AppConfig

References

- https://trunkbaseddevelopment.com
- https://martinfowler.com/bliki/DarkLaunching.html
-

Let's keep in touch

https://prokopov.me

Maksym Prokopov

Staff Site Reliability Engineer at Billie