ThoughtWorks® presents:

Trunk-Based Development

Kief Morris

Michael Lihs

Chris Ford

Welcome

ThoughtWorks presents:

Trunk-Based Development

with **Kief Morris, Chris Ford** and **Michael Lihs**

ThoughtWorks®

This event will be recorded

Optional:
Keep your video on
Rename yourself

Please stay
muted

Questions?
Write them
in the chat

ThoughtWorks®

ThoughtWorks Online Code of Conduct

„Be excellent to each other“

While free speech is important -
intolerance of any kind will not be tolerated.

If you’re happy with this - we welcome you.

We will add the link to the whole Code of Conduct to the Zoom chat soon.

ThoughtWorks®

ThoughtWorks®

GLOBAL SOFTWARE CONSULTANCY

Kief Morris
@kief

Chris Ford
@ctford

Michael Lihs
@kaktusmimi

Agenda

- What is trunk-based development (TBD)
- Support TBD in your development workflow
- Tricky Questions
- Discussion
- Summary

The history behind this meetup

Trunk-based code review - tools and techniques

Subscribe

98 views

Kornelis Sietsma Dec 13, 2020, 12:11:08 PM

to Software Development

Hi folks - this has been on the back of my mind for a bit.

One of the arguments people have for branches and PRs is effectively about the mechanics PRs offer for code reviews. There is lots of good tooling available for reviewing a bunch of changes, and the tools are familiar to lots of devs.

Infrastructure as Code

Exploring better ways to build and manage cloud infrastructure

Home Book Author Speaking Archive Feed

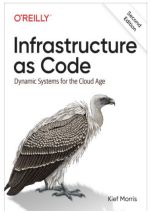
Why your team doesn't need to use pull requests

Jan 2, 2021

Github introduced the pull request practice, and features to support it, to make it easier for people who run open-source projects to accept contributions from outside their group of trusted committers.

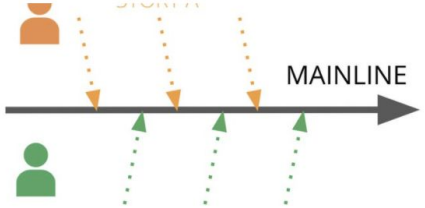
Committers are trusted to make changes to the codebase routinely. But a change from a random outsider needs to be assessed to make sure it works, doesn't take the project in an unwanted direction, and meets the standards for style and

Second edition is out now!



Michael Lihs @kaktusmimi Jan 4

at @thoughtworks we strongly advocate for #trunkBasedDevelopment - this blog post by my dear colleague @kief explains the backgrounds and why you should probably prefer it over a #pullRequest based approach infrastructure-as-code.com/book/2021/01/0...



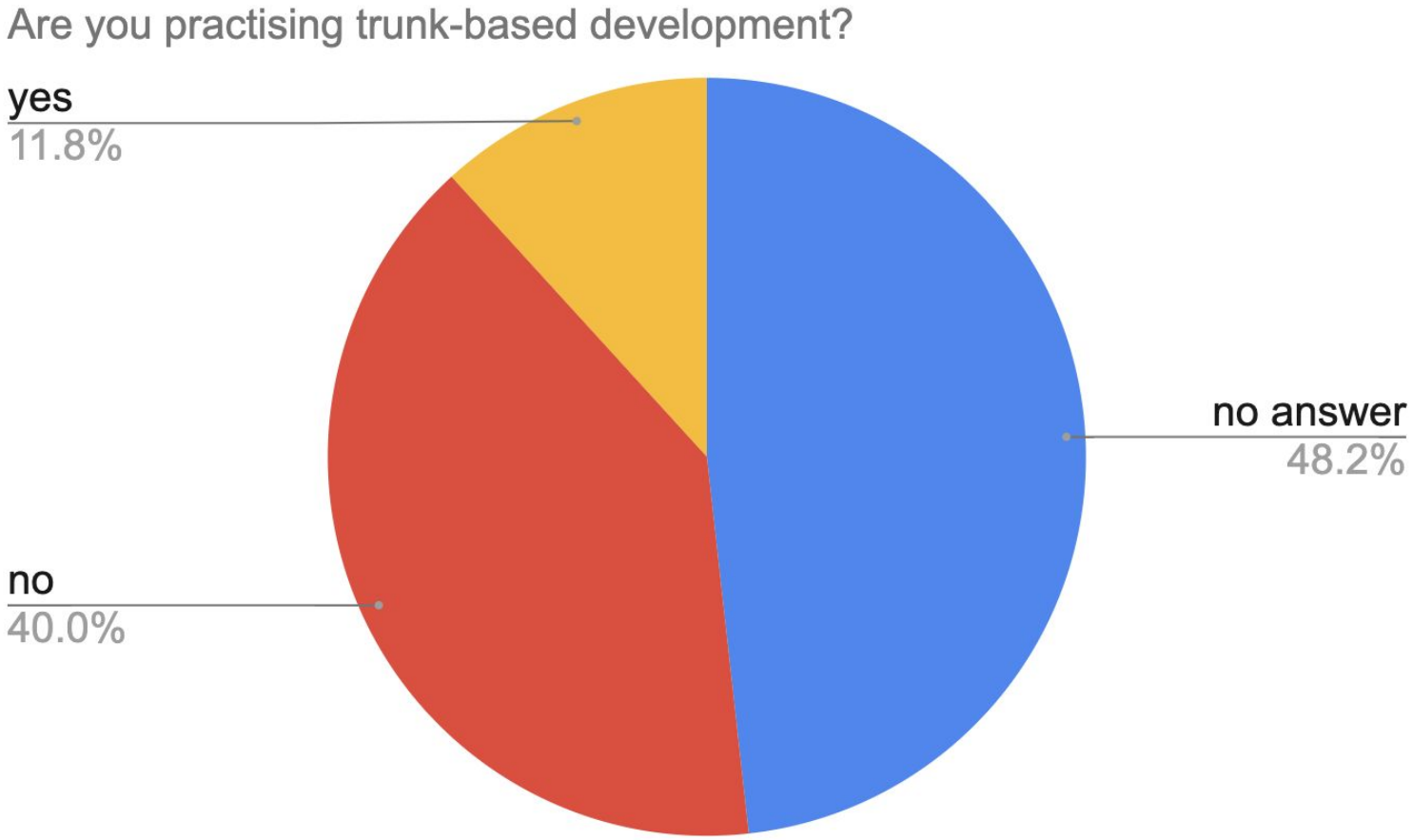
3 2 17

Christian Kühn @CYxChris Jan 5

Replying to @CYxChris @kaktusmimi and 2 others

2/2 in the spirit of discussing stuff like this after a devops meetup with you, which i miss, i propose we have an open discussion on whatever video platform in the next couple of weeks where we discuss this! i'm organizing if you're in!

3 2 5

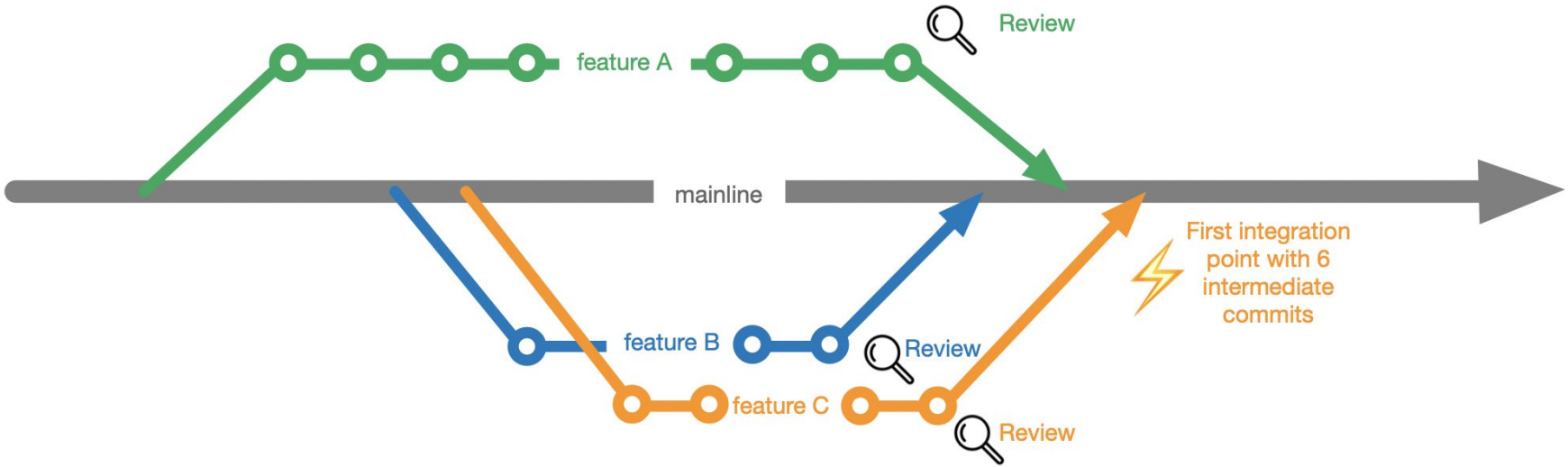


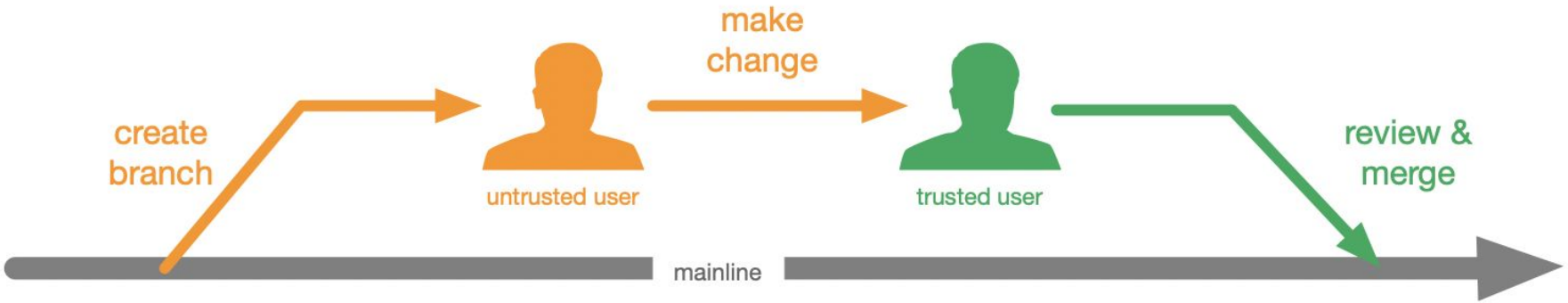
Introducing Trunk-based Development

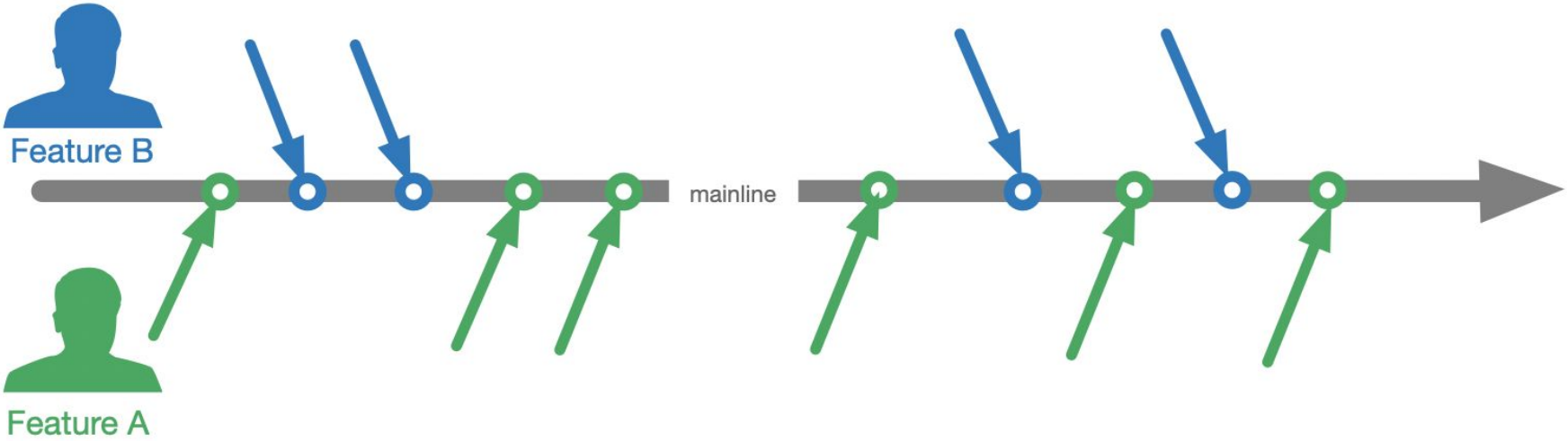
Continuous Integration

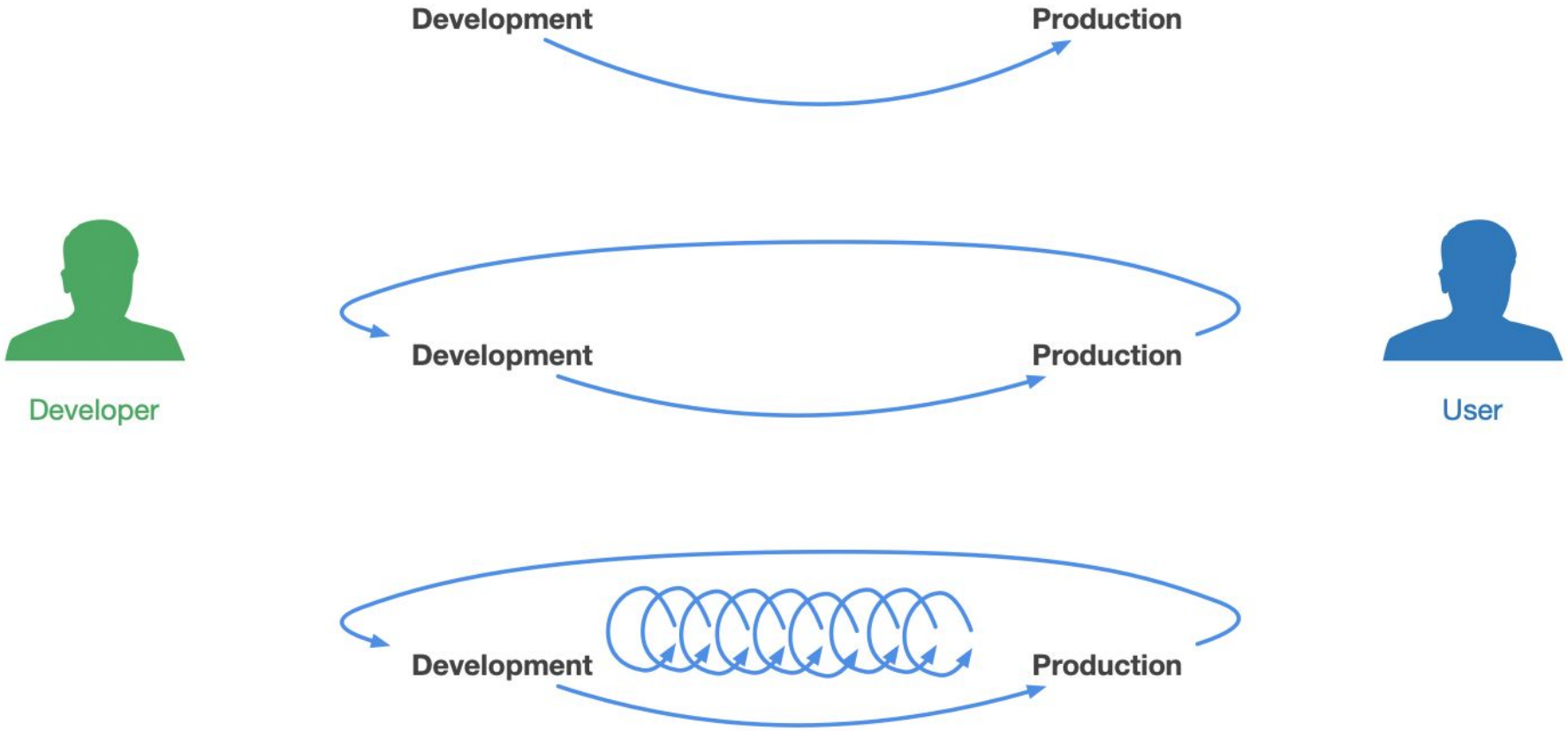
Continuous Integration (CI) is a development practice that requires developers to **integrate** code into a shared repository **at least once a day**. Each check-in is then verified by an automated build, allowing teams to **detect problems early**.

Image source https://www.thoughtworks.com/de/continuous-integration





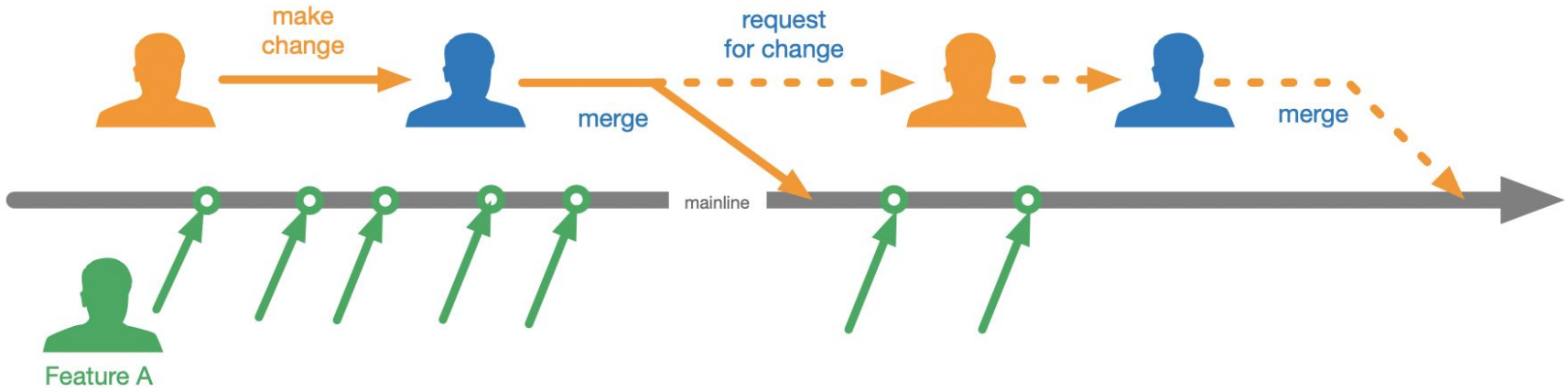




Lean Management

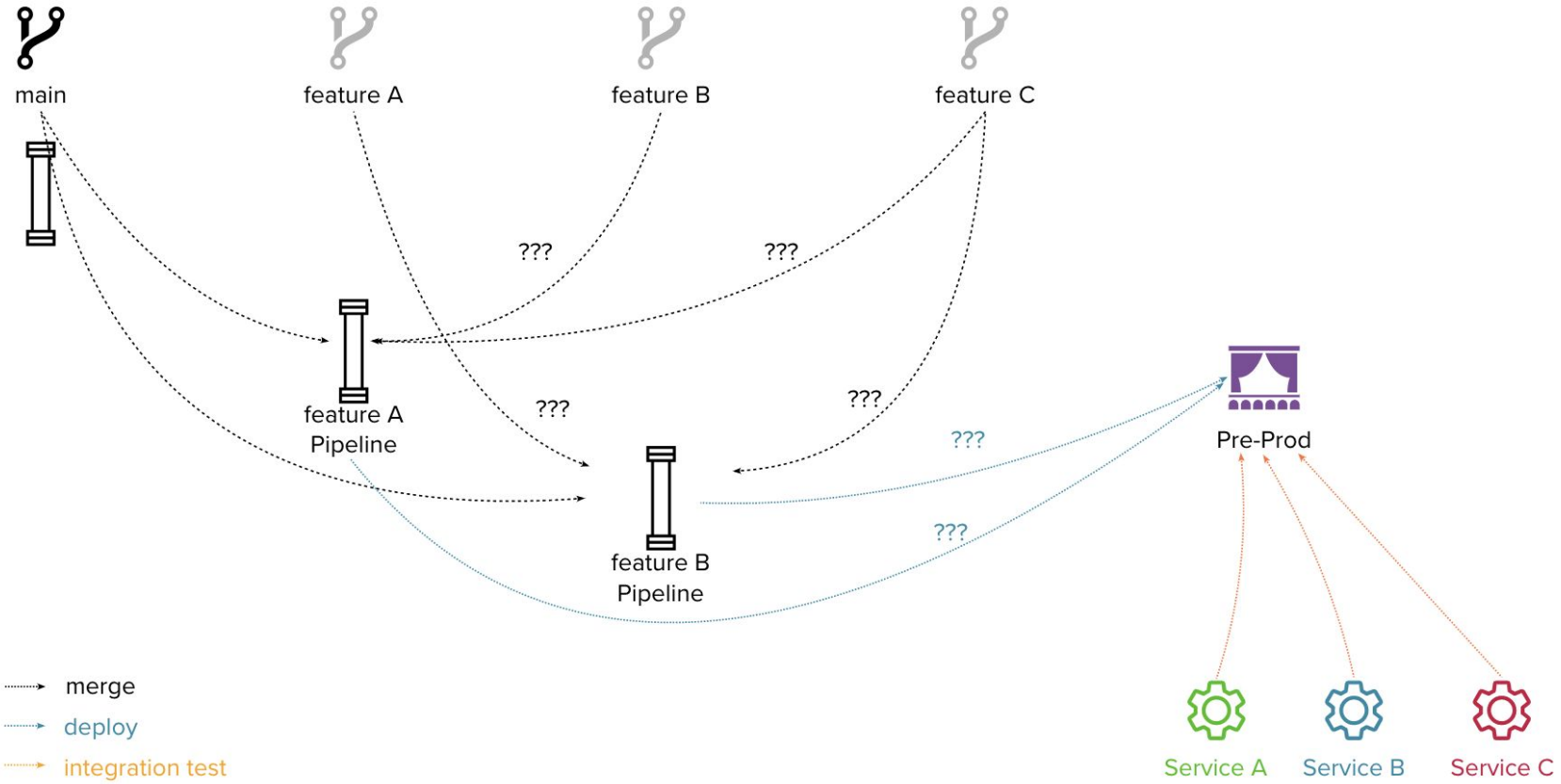
Reduce Waste

PULL REQUEST:
Wait for review and merge



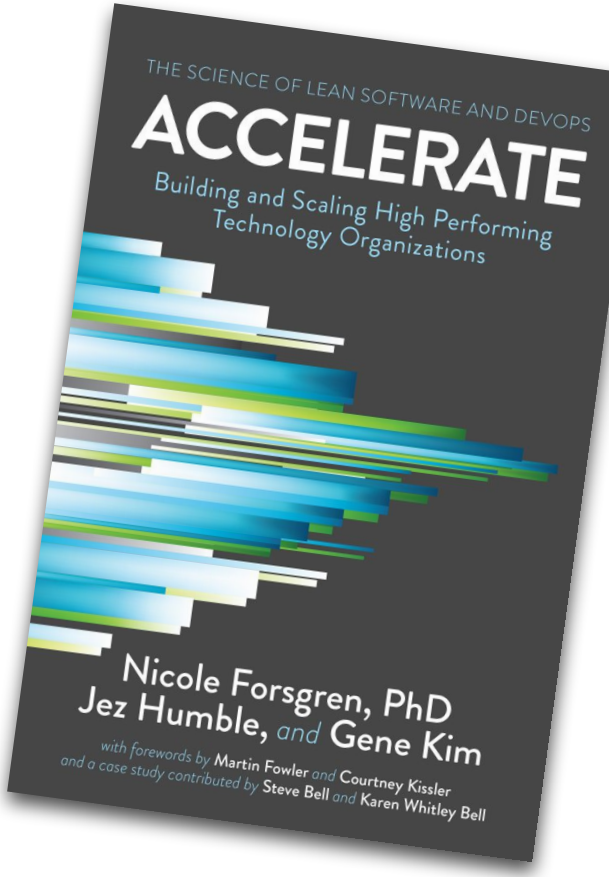
CONTINUOUS INTEGRATION:
Immediate and continuous feedback

Problems with CI-ing feature branches



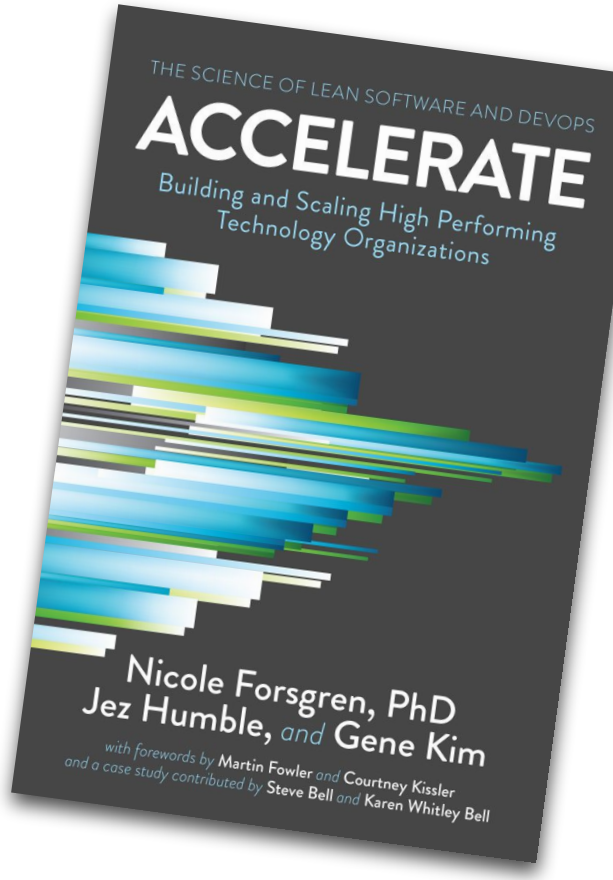
TBD and Team Performance

“ Our research also found that developing [trunk-based] rather than on long-lived feature branches was **correlated with higher delivery performance**. Teams that did well had fewer than three active branches at any time, their **branches had very short lifetimes (less than a day)** before being merged into trunk and never had “code freeze” or stabilization periods. It’s worth re-emphasizing that these results are independent of team size, organization size, or industry.



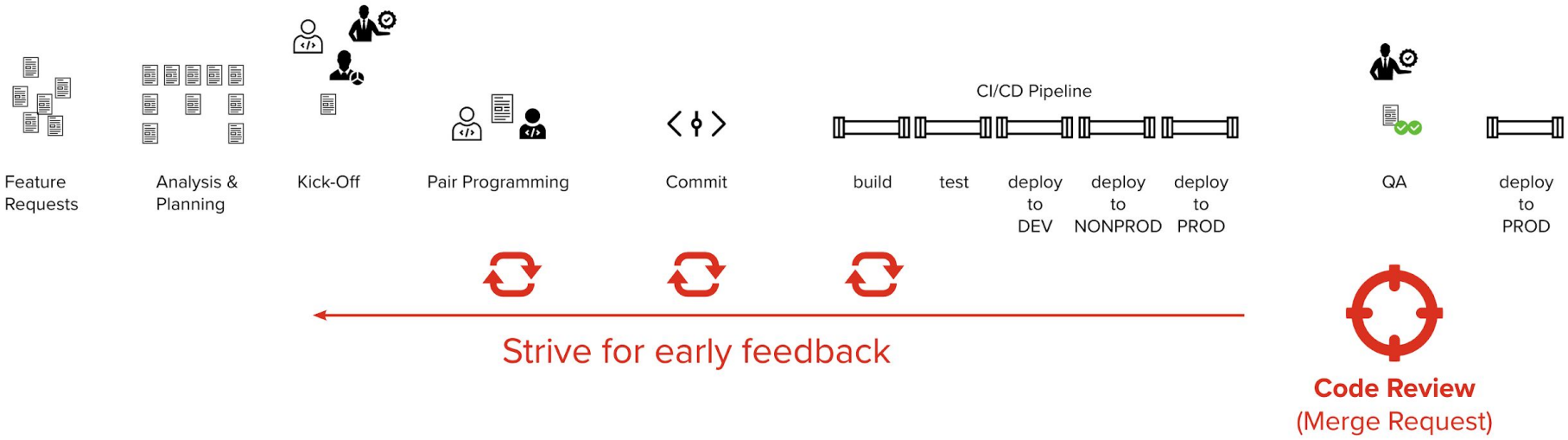
TBD and Team Performance

“We conducted additional research and found that teams using branches that live a short amount of time (integration times less than a day) combined with short merging and integration periods (less than a day) do better in terms of software delivery performance than teams using longer-lived branches. Anecdotally, and based on our own experience, we hypothesize that this is because having multiple long-lived branches discourages both refactoring and intrateam communication.




YOU DON'T SIMPLY DO
TRUNK-BASED DEVELOPMET
imgflip.com

Support TBD with your workflow

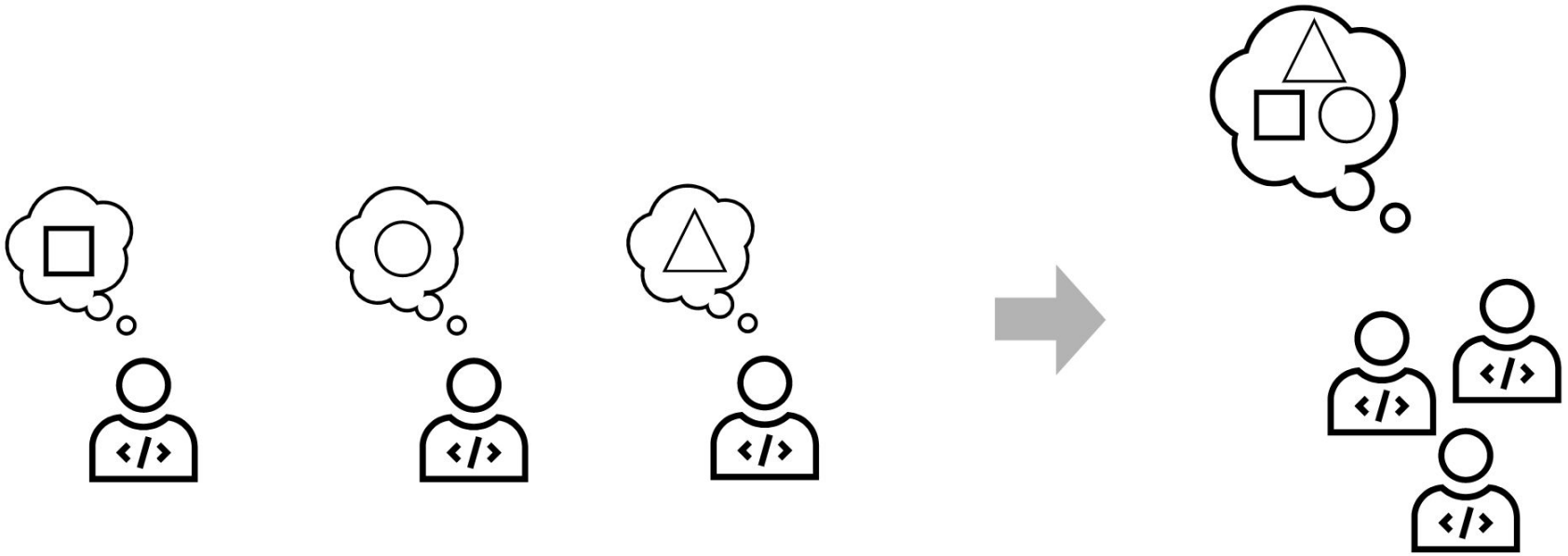


Trunk-based development works best supported by other practices that provide early review

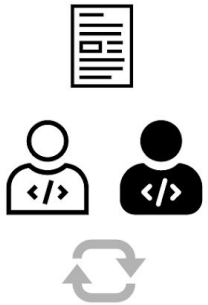
Story-Analysis as a shared responsibility

- Include Client (Business), QA, XD and Devs in the story analysis
- Analyse iteratively
- Assess whether a story is well understood in your estimation sessions
- Have a driver for a story that “anchors” the topic

Tech Huddles

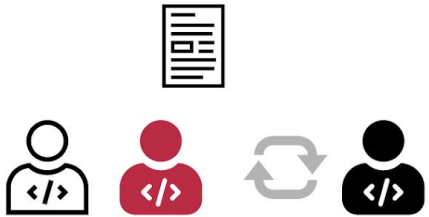


Pair Programming



Rotate Roles

Alternate
Pairing Methods

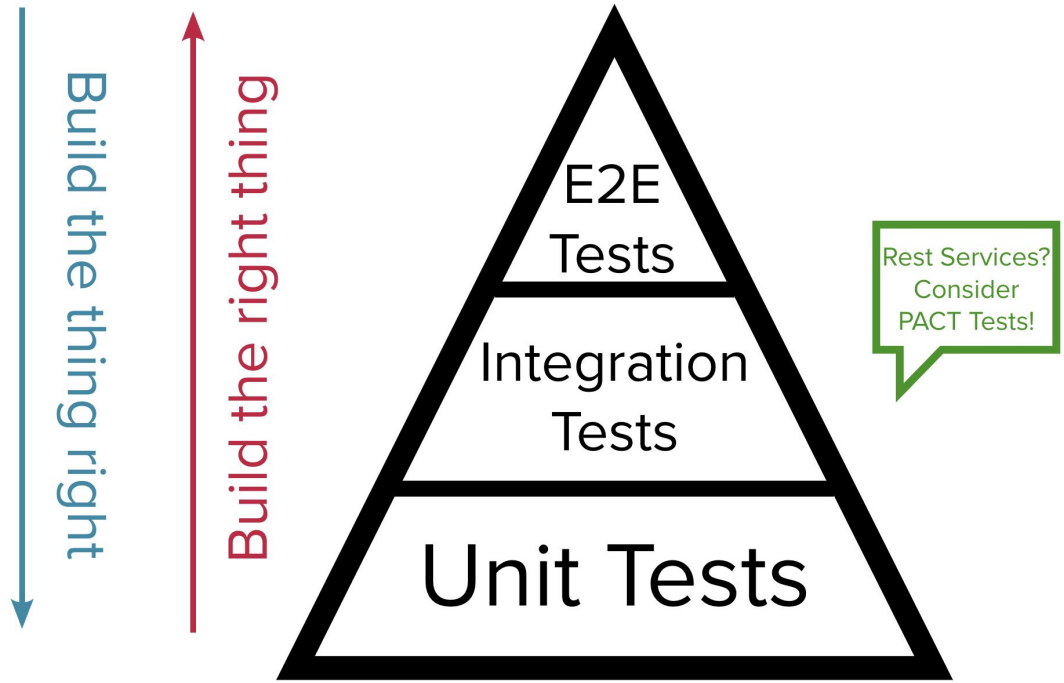


Rotate People

Reduce Bus Factor

Trunk-based development requires the adoption of some technical practices to be safe

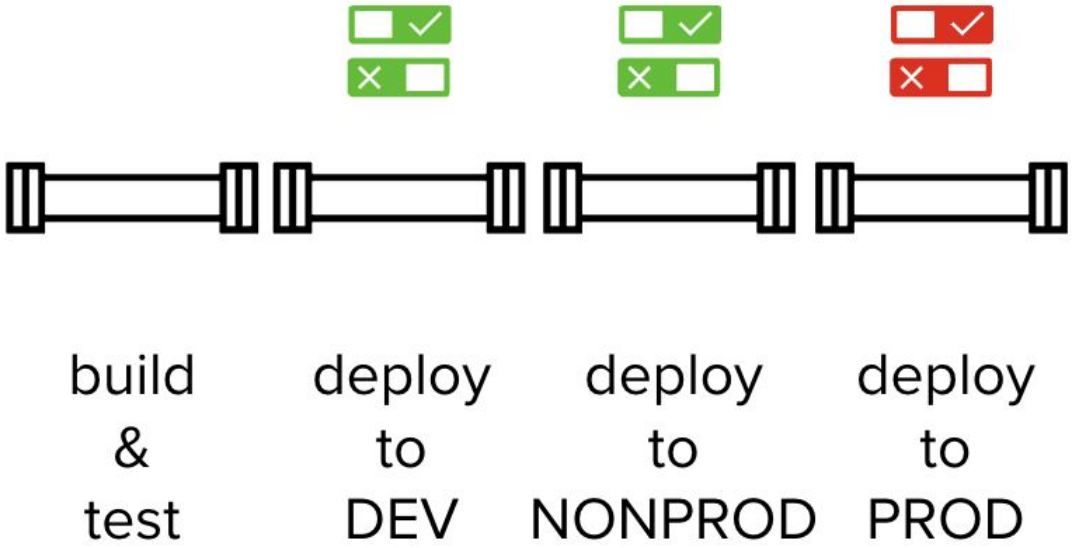
Balanced Test Suite



Continuous Integration - Best Practices

- Check in frequently (at least once a day)
- Every commit should build on an integration machine
- Keep the build fast (<< 10 min)
- Test in a clone of the production environment
- Don’t check in broken code
- Don’t check in untested code
- Don’t check in when the build is broken
- Don’t go home after checking in until the system builds

Feature Toggles



Summary

Summary

- **Integrate** your code changes **at least once a day** to enable Continuous Integration
- Optimise for **fast** and **specific feedback**
- Aim for **non-blocking** (automated) **feedback**
- Go for a "**Prevent instead of Detect**" approach
- "Distribute" (code) reviews across the development lifecycle
- Invest in a **proper test suite** rather than in code reviews
- Avoid long-living branches
- Continuously evaluate and re-adjust
- Improving **intra-team communication** improves code quality and delivery speed

Tricky Questions

- Why do so many teams use pull requests?
- When would you recommend pull requests?
- Should people deploy on a Friday?
- Does a stable staging environment makes sense?
- Does TBD work if I really really need code reviews?
- ...

Resources

References

- https://infrastructure-as-code.com/book/2021/01/02/pull-requests.html
- https://trunkbaseddevelopment.com/
- https://www.thoughtworks.com/de/continuous-integration
- https://www.martinfowler.com/articles/continuousIntegration.html
- https://martinfowler.com/articles/branching-patterns.html
- https://itrevolution.com/the-devops-handbook/
- https://itrevolution.com/book/accelerate/

THANKYOU

Kief Morris @kief

Chris Ford @ctford

Michael Lihs @kaktusmimi

Kief.morris | chris.ford | michael.lihs@thoughtworks.com

ThoughtWorks®