



Paul Börding

Developer



Board games

@PaulB5g



Michael Lihs

Infrastructure Consultant

Tübingen (Germany)

Cycling
Guitar

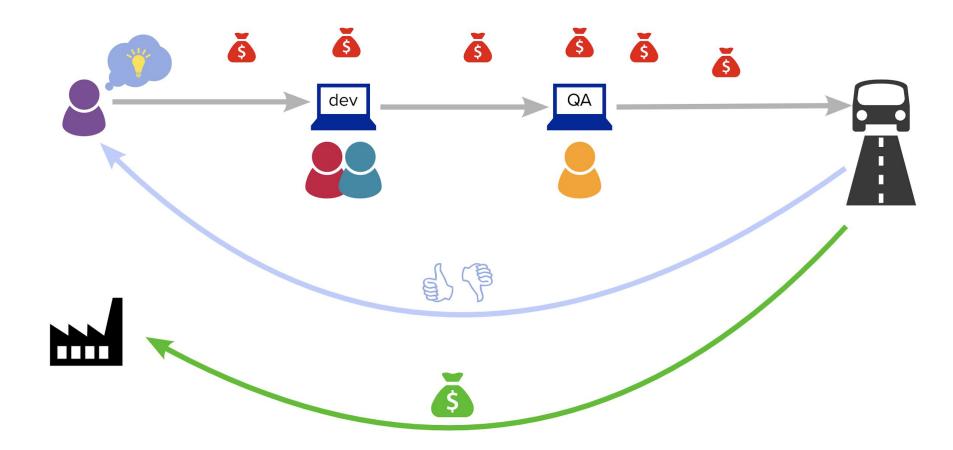


@kaktusmimi

THOUGHTWORKS: An international Software Consultancy







Continuous Delivery



Continuous delivery (CD) is a software engineering approach in which teams produce software in **short cycles**, ensuring that the software can be **reliably released** at any time [...] It aims at **building**, **testing**, and **releasing** software with greater speed and frequency.

Wikipedia

Continuous Integration

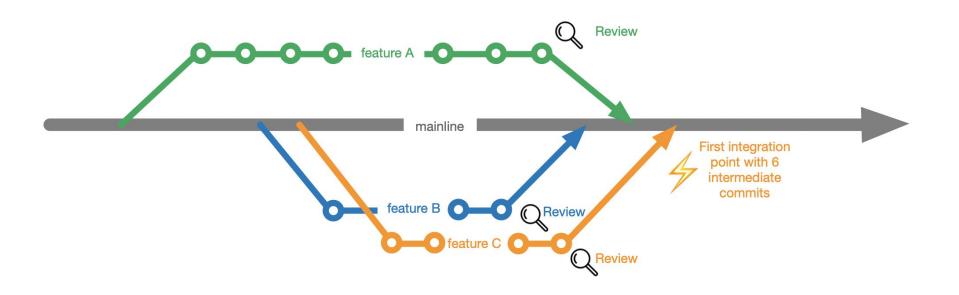


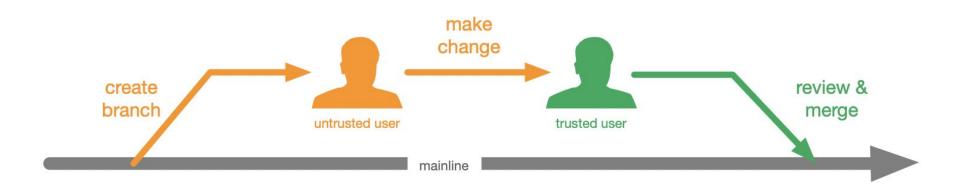
Continuous Integration (CI) is a development practice that requires developers to integrate code into a shared repository at least once a day. Each check-in is then verified by an automated build, allowing teams to detect problems early.

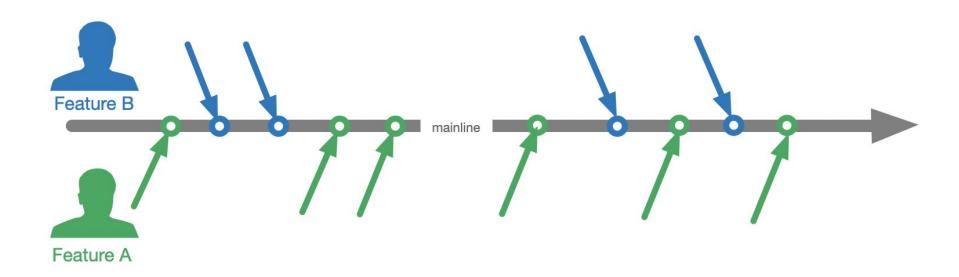
Continuous Integration Certification

- 1. Every developer commits to a shared mainline (trunk) at least daily
- 2. Every commit to mainline **triggers an automated test and build**
- 3. If the build or test fails, it is **fixed in 10 minutes**

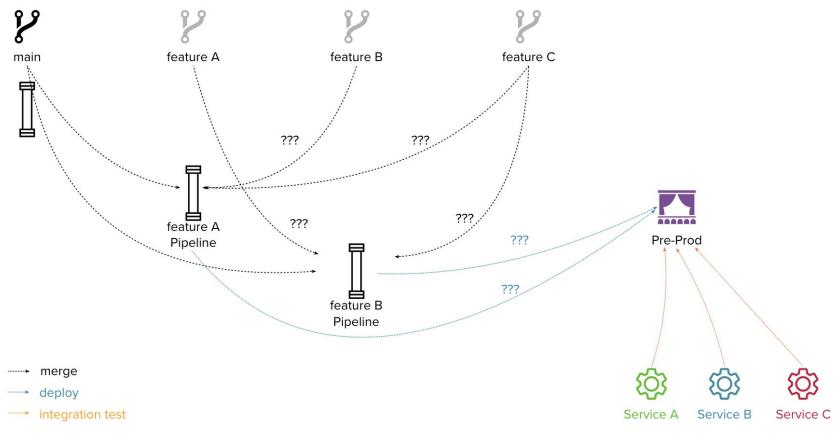
Steve Upton - fellow ThoughtWorker







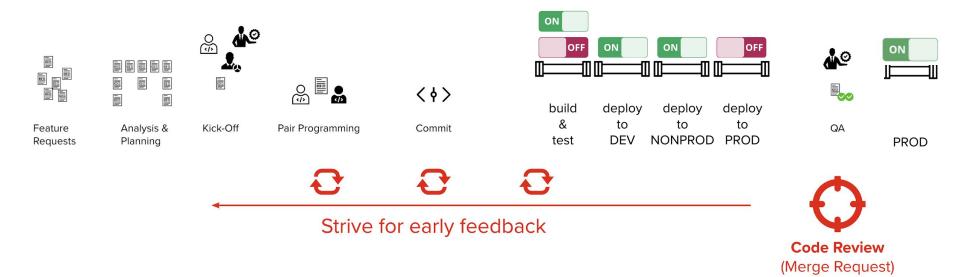
Problems with CI-ing feature branches



Trunk-based development is a mandatory prerequisite for Continuous Integration



Support TBD within 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



Pair Programming









Alternate
Pairing Methods

Rotate People

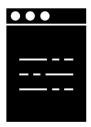
Reduce Bus Factor

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

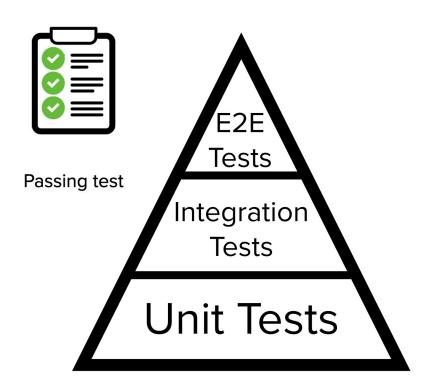
Test-Driven Development



Write a failing test



Write only as much code to make it pass

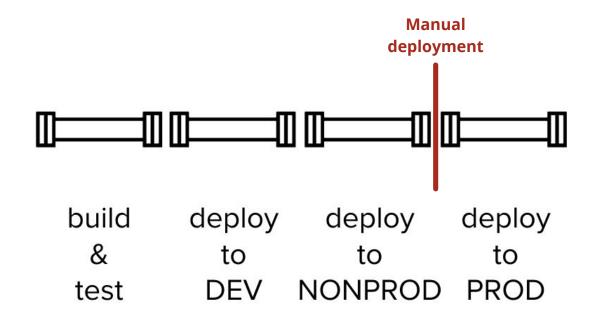


Continuous Integration - Best Practices

- 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



Enable Continuous Delivery with Feature Toggles



Continuous Delivery makes releasing a business decision not a technical decision

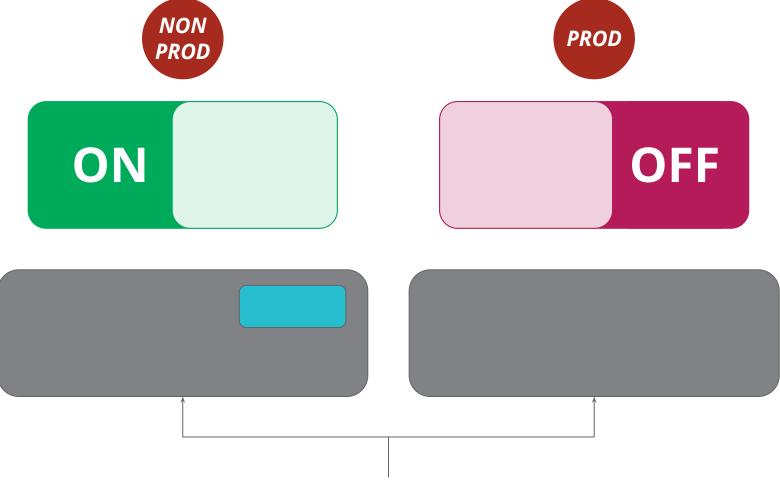


MY WEBSITE



MY WEBSITE

COOL NEW FEATURE



Both are valid states!

ThoughtWorks®

GLOBAL SOFTWARE CONSULTANCY

A Feature Toggles Life Cycle





#60 New Feature

#62 Bug Fix

#63 Feature Improvement





#60 New Feature

#62 Bug Fix

#63 Feature Improvement Create a toggle first!





#62 Bug Fix

#63 Feature Improvement #60 New Feature

Create a toggle first!

- Usually takes less than 1 hour
- Enable toggle on all non-prod environments
- Disable toggle on prod environment





#62 Bug Fix

#63 Feature Improvement Create a toggle first!







#60 New Feature

 $\left[\begin{array}{c|c} T1\end{array}\right]\left[\begin{array}{c} T2\end{array}\right]\left[\begin{array}{c} T3\end{array}\right]\left[\begin{array}{c} T4\end{array}\right]\left[\begin{array}{c} T5\end{array}\right]$

Unaffected

T1 T2 T3 T4 T5 ON

Unaffected

T1 T2 T3 T4 T5 ON

Unaffected

T1

T3

T4

ON

Affected

T2

T5

Unaffected

T1

T3

T4

ON

Affected (**legacy**)

Copies of Affected

T2'

T2

T5

T5'



T1

T3

T4

ON

Affected (**legacy**)

T2

T5

OFF

Copies of Affected

T2'

T5'



T1

T3

T4

ON

Affected (**legacy**)

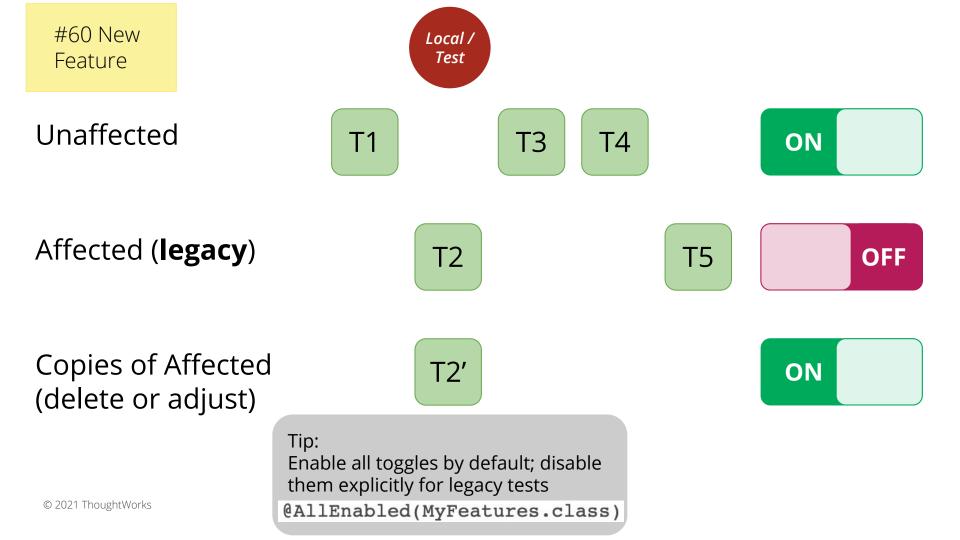
T2

T5

OFF

Copies of Affected (delete or adjust)

T2'





Affected (**legacy**)

T1

T2'

T2

T3

T4 T5

ON

OFF ON

(delete or adjust)

© 2021 ThoughtWorks

New Tests

Copies of Affected

N1

N2

N3



Affected (**legacy**)

Copies of Affected (delete or adjust)

New Tests

N1

T1

N2

T2

T2'

N3

T3

T4

T5

ON

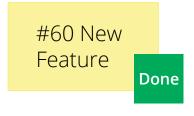


OFF









T1

T3

T4

ON

Copies of Affected (delete or adjust)

T2'

ON

New Tests

© 2021 ThoughtWorks

N2 N1

N3



T1

T3

T4

ON

Copies of Affected (delete or adjust)

N1

T2'

N3

ON

New Tests

© 2021 ThoughtWorks

N2







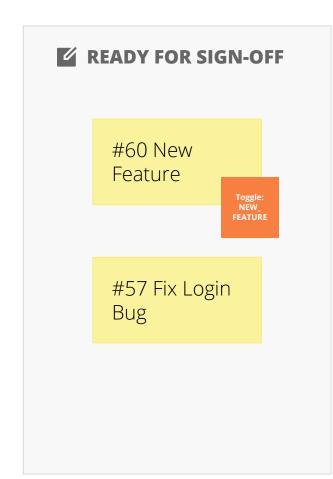
#57 Fix Login Bug

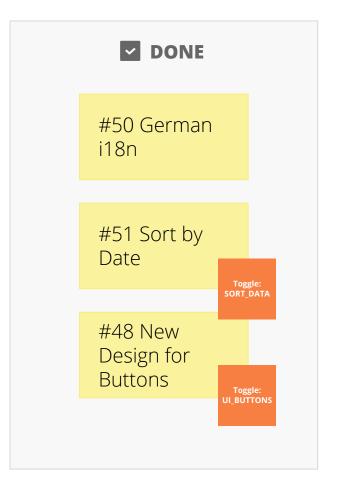


#50 German i18n

#51 Sort by Date

#48 New Design for Buttons









READY FOR SIGN-OFF

#60 New
Feature

Toggle:
NEW_
FEATURE

#57 Fix Login Bug



READY FOR SIGN-OFF

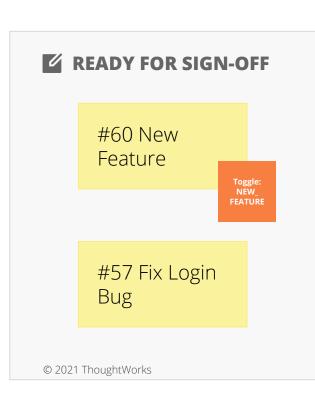
#60 New
Feature

Toggle:
NEW_
FEATURE

#57 Fix Login Bug

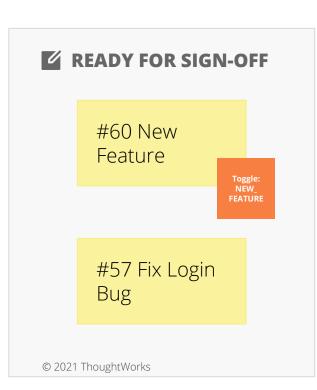
● PROD/TOGGLED ON

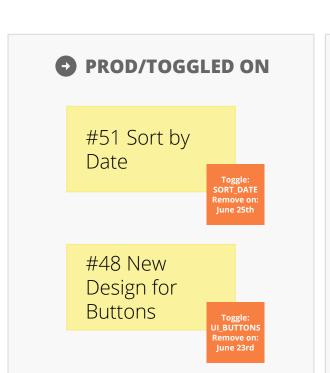






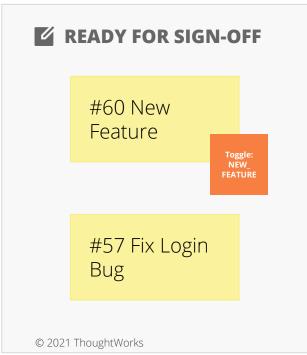


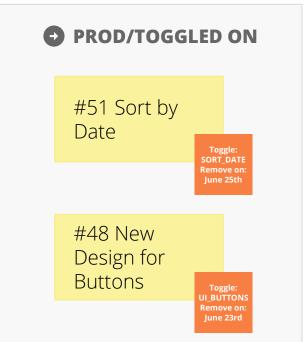








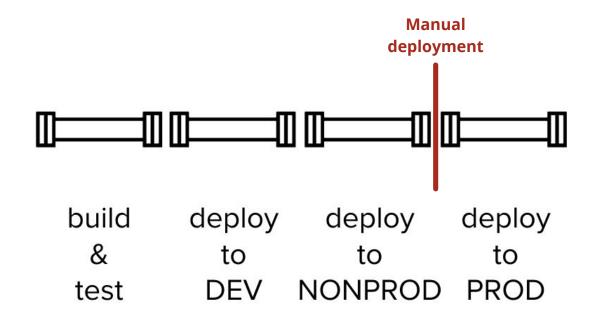




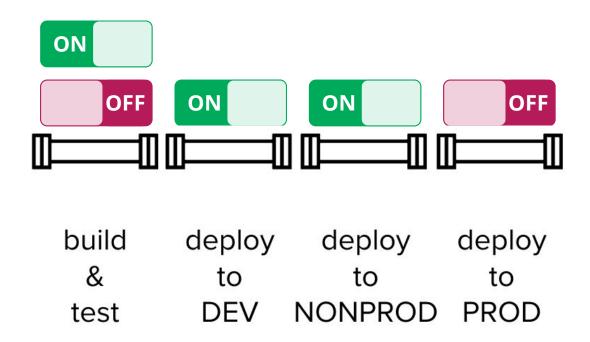




Enable Continuous Delivery with Feature Toggles



Enable Continuous Delivery with Feature Toggles



Continuous Delivery Technology

Managing feature toggles in teams



Paul Börding



Steve Upton

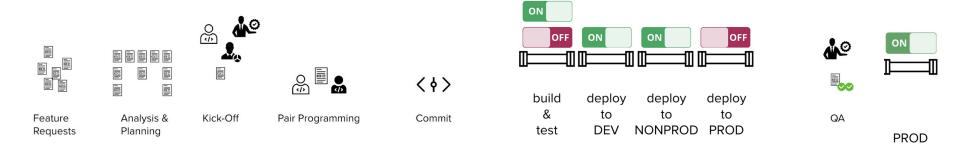
Published: Aug 26, 2020

https://www.thoughtworks.com/insights/blog/managing-feature-toggles-teams

https://youtu.be/6e88toloUiU



Achieve Continuous Delivery







References

- https://www.thoughtworks.com/insights/blog/managing-feature-toggles-teams
- 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/
- https://lihsmi.ch/meetup/2021/01/21/q-and-a-trunk-based-development.html

THANKYOU

Michael Lihs

@kaktusmimi

michael.lihs@thoughtworks.com

Paul Börding

@PaulB5g

paul.boerding@thoughtworks.com

ThoughtWorks[®]