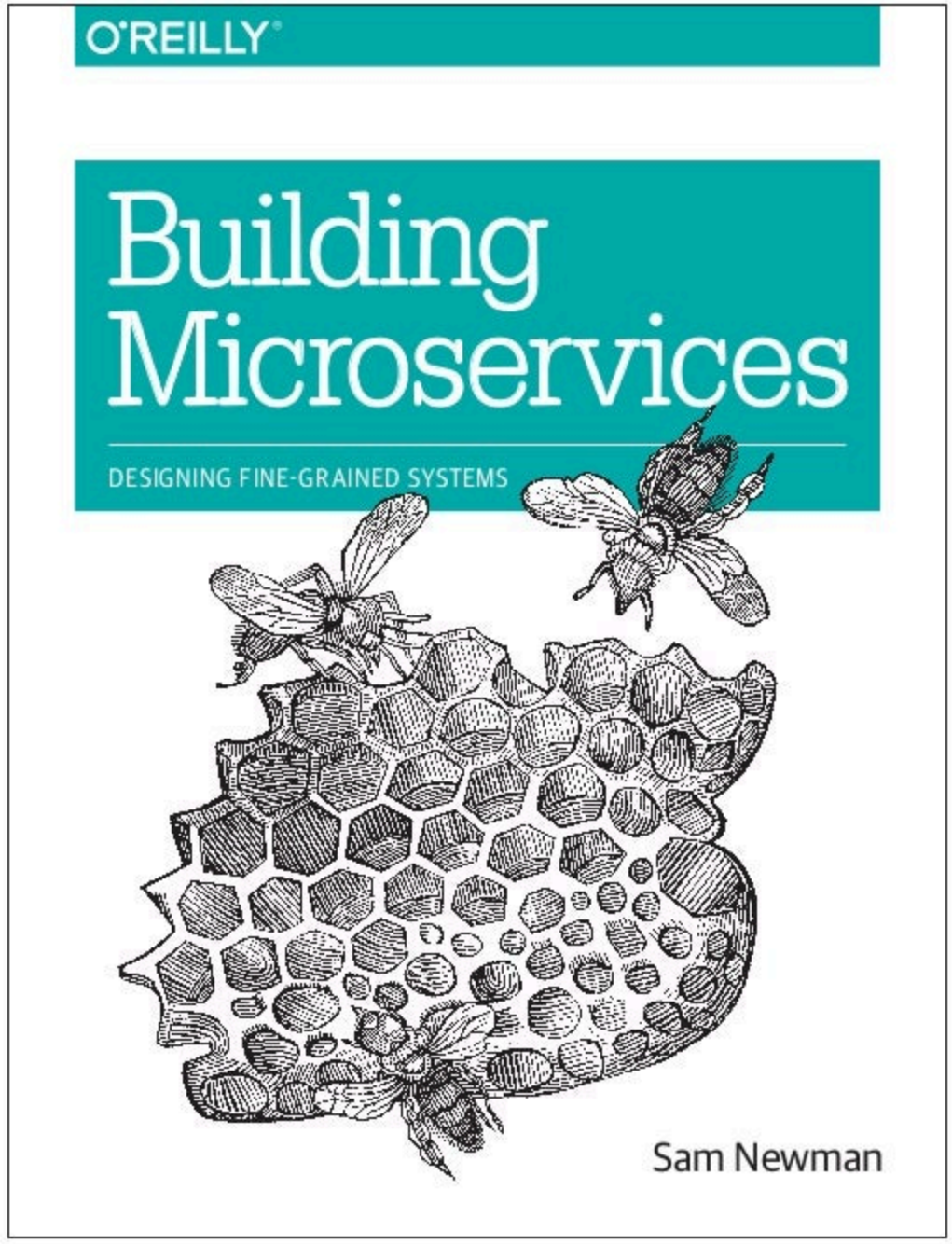
Feature branches and toggles in a post-GitHub world

Devoxx Poland 2017

Sam Newman

Sam Newman & Associates


O'REILLY
Building Microservices
DESIGNING FINE-GRAINED SYSTEMS
Sam Newman

2004


16:40
16:40


na
Coventry
Rugby
Long Buckby
Northampton
Wolverton
Leamington Spa
Milton Keynes Central
Bletchley
Leighton Buzzard
Cheddington
Tring
Berkhamsted
Hemel Hempstead
Apsley
Kings Langley
Watford Junction
Bushey
Harrow & Wealdstone
Wembley Central
London Euston
Bedford
Bedford St Johns
Kempston Hardwick
Stewartby
Millbrook
Lidlington
Ridgmont
Aspley Guise
Woburn Sands
Bow Brickhill
Fenny Stratford
to
Luton
St Albans
London S
St Albans Abbey
Park Street
How Wood
Bricket Wood
Garston
Watford North
to
Banbury
Oxford
Reading
High Wycombe
London Marylebone

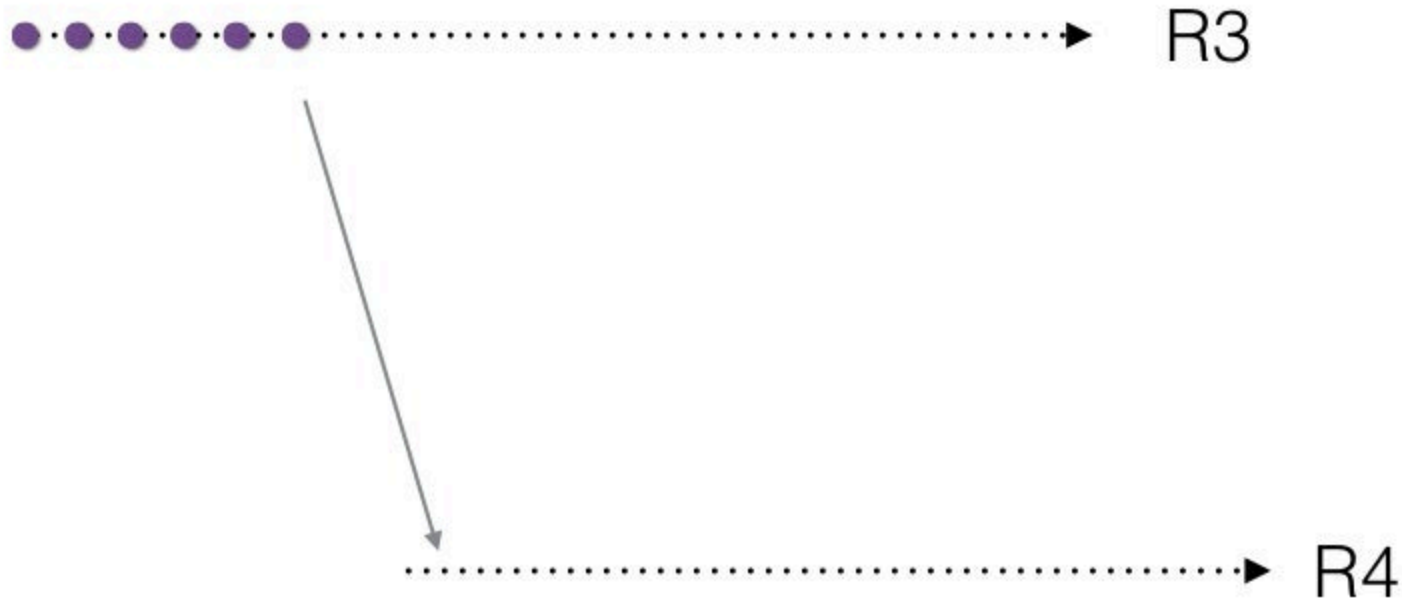

na
Coventry
Rugby
Long Buckby
Northampton
Wolverton
Leamington Spa
Milton Keynes Central
Bedford
Bedford St Johns
Kempston Hardwick
Stewartby
Millbrook
Lidlington
Ridgmont
Aspley Guise
Woburn Sands
Bow Brickhill
Fenny Stratford
to
Luton
St Albans
London S
Bletchley
Leighton Buzzard
Cheddington
Tring
Berkhamsted
Hemel Hempstead
Apsley
Kings Langley
Watford Junction
Bushey
Harrow & Wealdstone
Wembley Central
London Euston
St Albans Abbey
Park Street
How Wood
Bricket Wood
Garston
Watford North
to
Banbury
Oxford
Reading
High Wycombe
London Marylebone

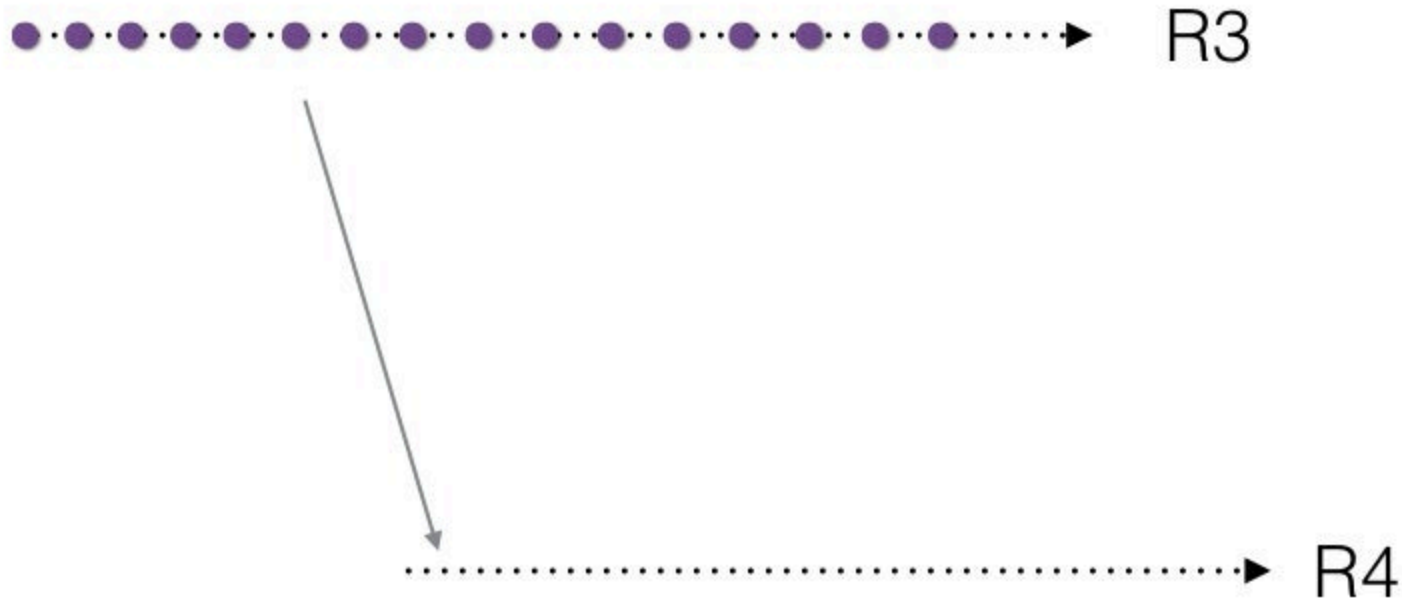

na
Coventry
Rugby
Long Buckby
Northampton
Wolverton
Leamington Spa
Milton Keynes Central
Bletchley
Leighton Buzzard
Cheddington
Tring
Berkhamsted
Hemel Hempstead
Apsley
Kings Langley
Watford Junction
Bushey
Harrow & Wealdstone
Wembley Central
London Euston
Bedford
Bedford St Johns
Kempston Hardwick
Stewartby
Millbrook
Lidlington
Ridgmont
Aspley Guise
Woburn Sands
Bow Brickhill
Fenny Stratford
to
Luton
St Albans
London S
St Albans Abbey
Park Street
How Wood
Bricket Wood
Garston
Watford North
to
Banbury
Oxford
Reading
High Wycombe
London Marylebone

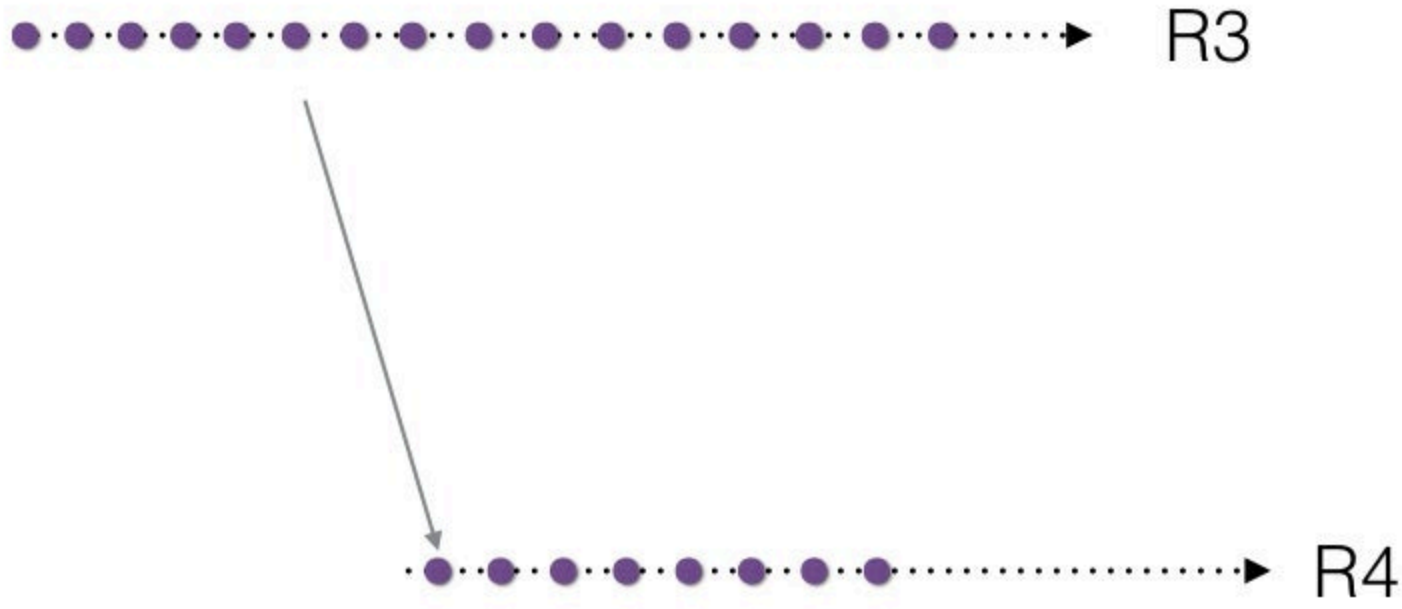

na
Coventry
Rugby
Long Buckby
Northampton
Leamington Spa
Wolverton
Milton Keynes Central
Bletchley
Leighton Buzzard
Cheddington
Tring
Berkhamsted
Hemel Hempstead
Apsley
Kings Langley
Watford Junction
Bushey
Harrow & Wealdstone
Wembley Central
London Euston
Bedford
Bedford St Johns
Kempston Hardwick
Stewartby
Millbrook
Lidlington
Ridgmont
Aspley Guise
Woburn Sands
Bow Brickhill
Fenny Stratford
to
Luton
St Albans
London S
St Albans Abbey
Park Street
How Wood
Bricket Wood
Garston
Watford North
to
Banbury
Oxford
Reading
High Wycombe
London Marylebone

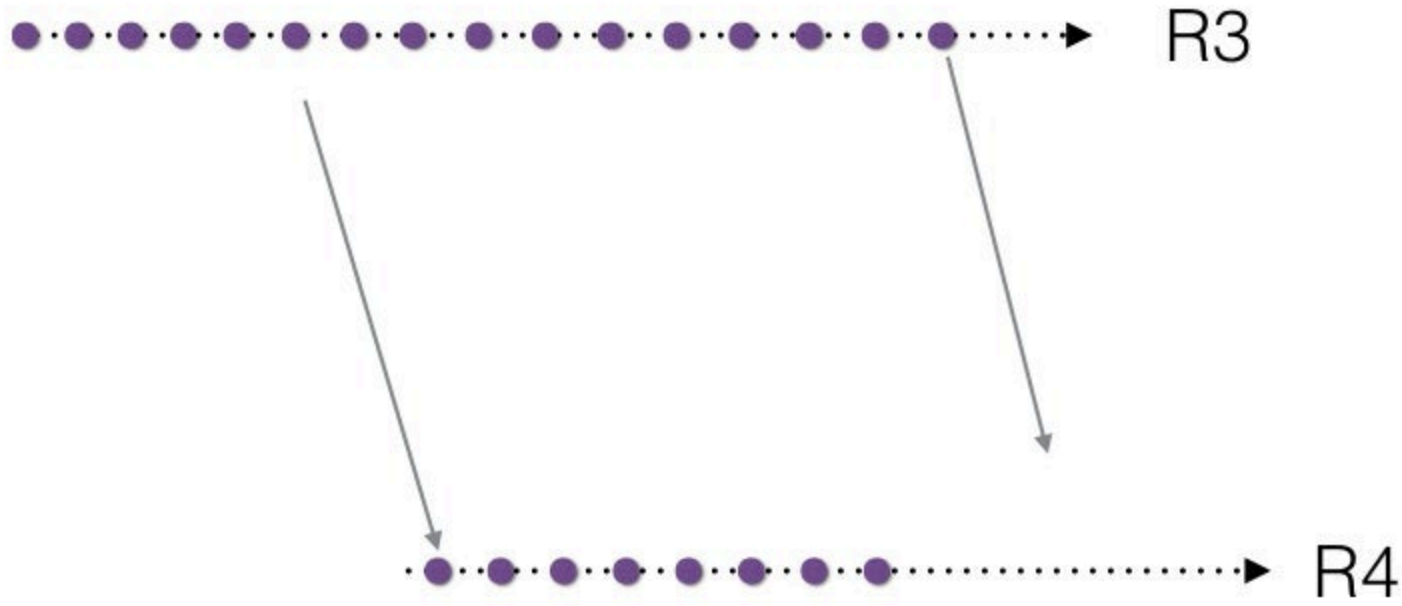

R3

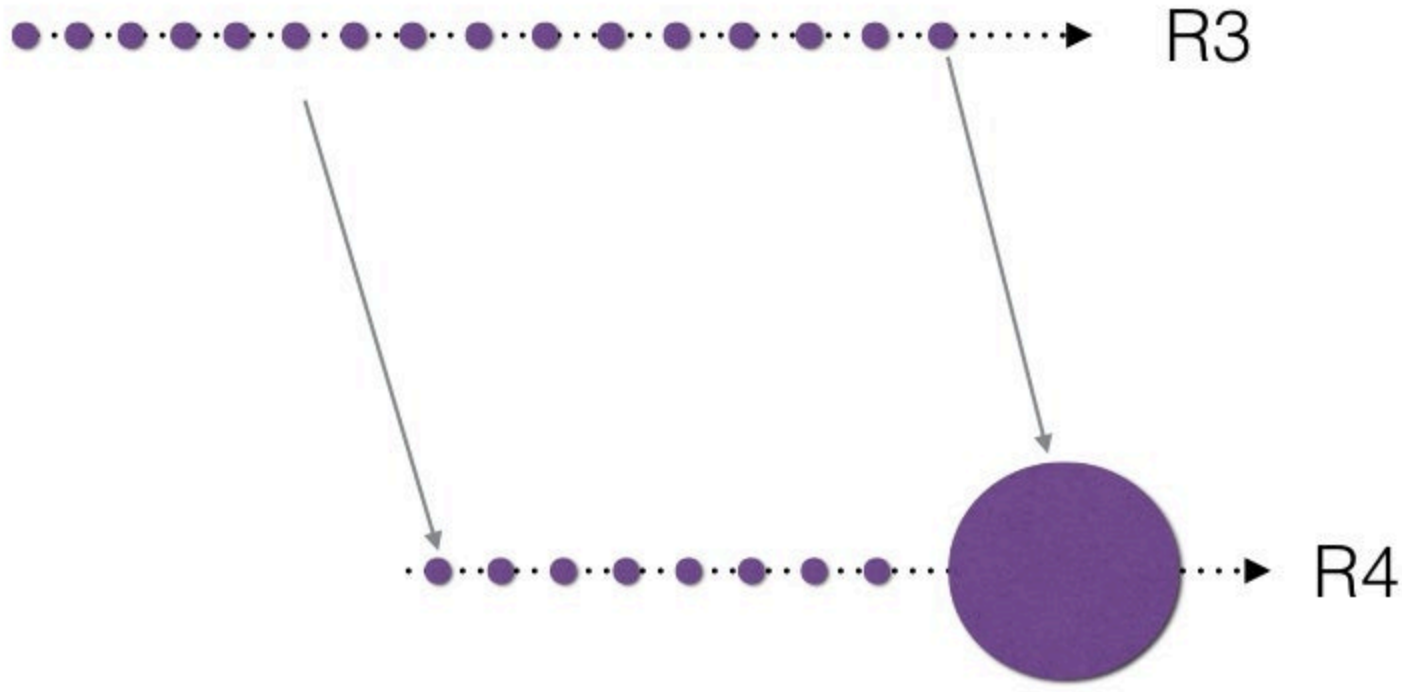

R3

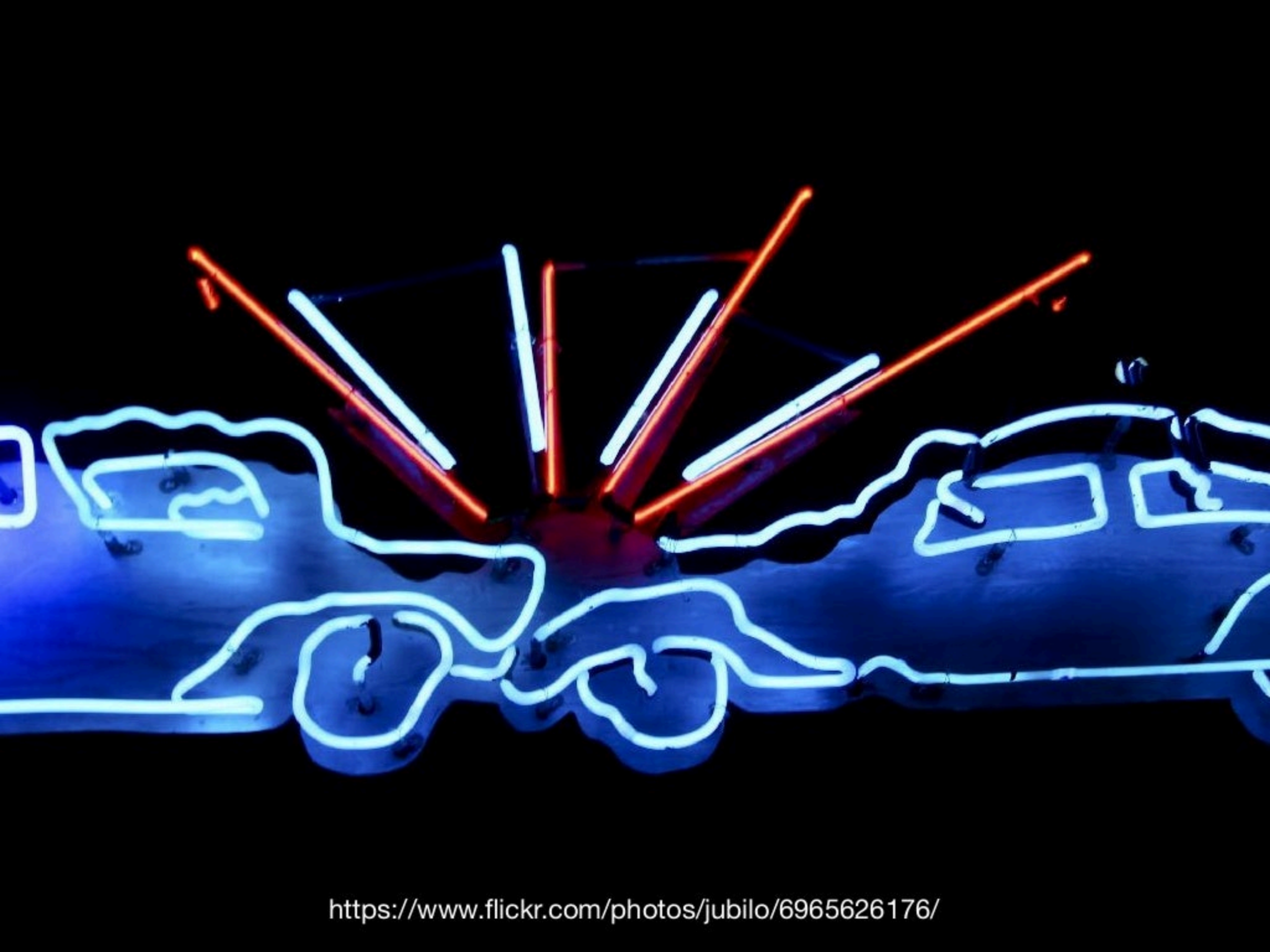

R3
R4


R3
R4


R3
R4


R3
R4


R3
R4



https://www.flickr.com/photos/jubilo/6965626176/

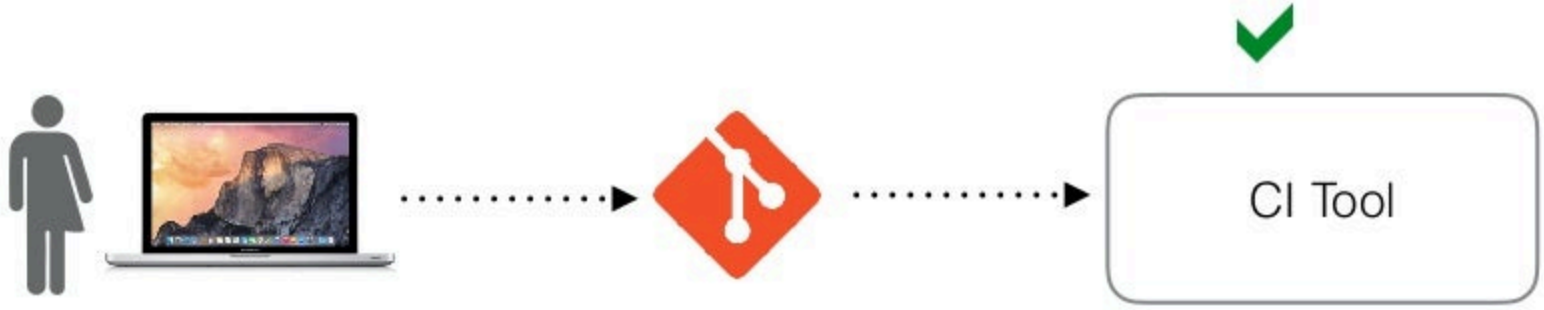
R3-R4 Merge Bug Fix Team

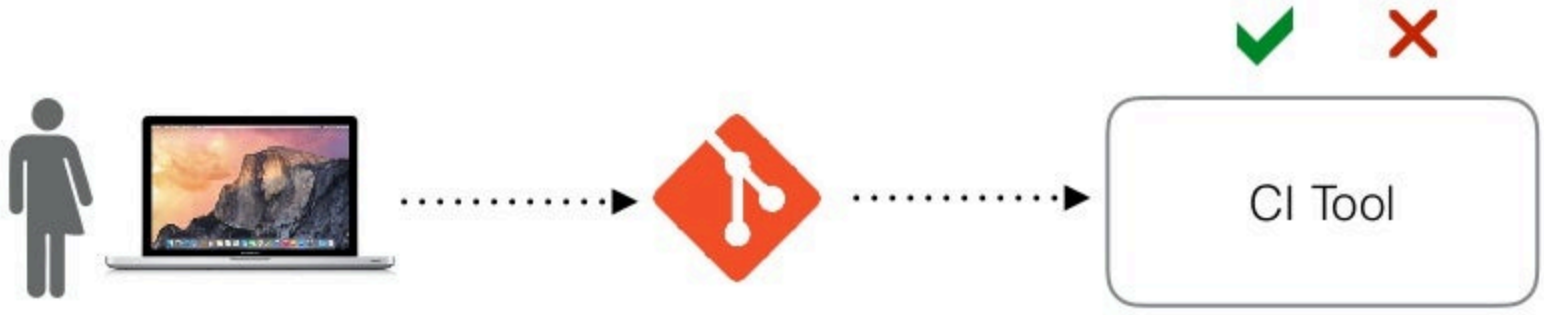
R3-R4 Merge Bug Fix Team

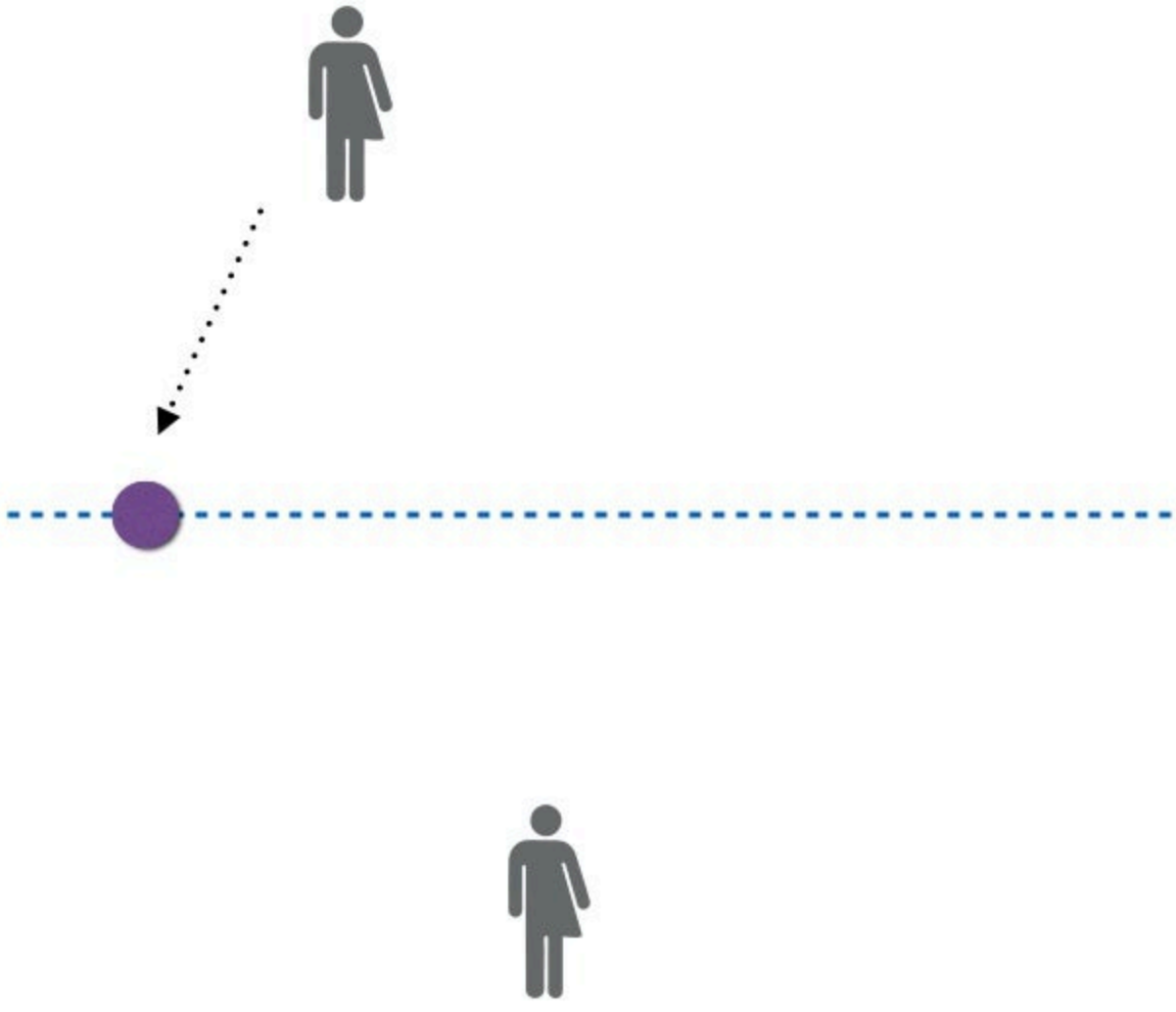
yay

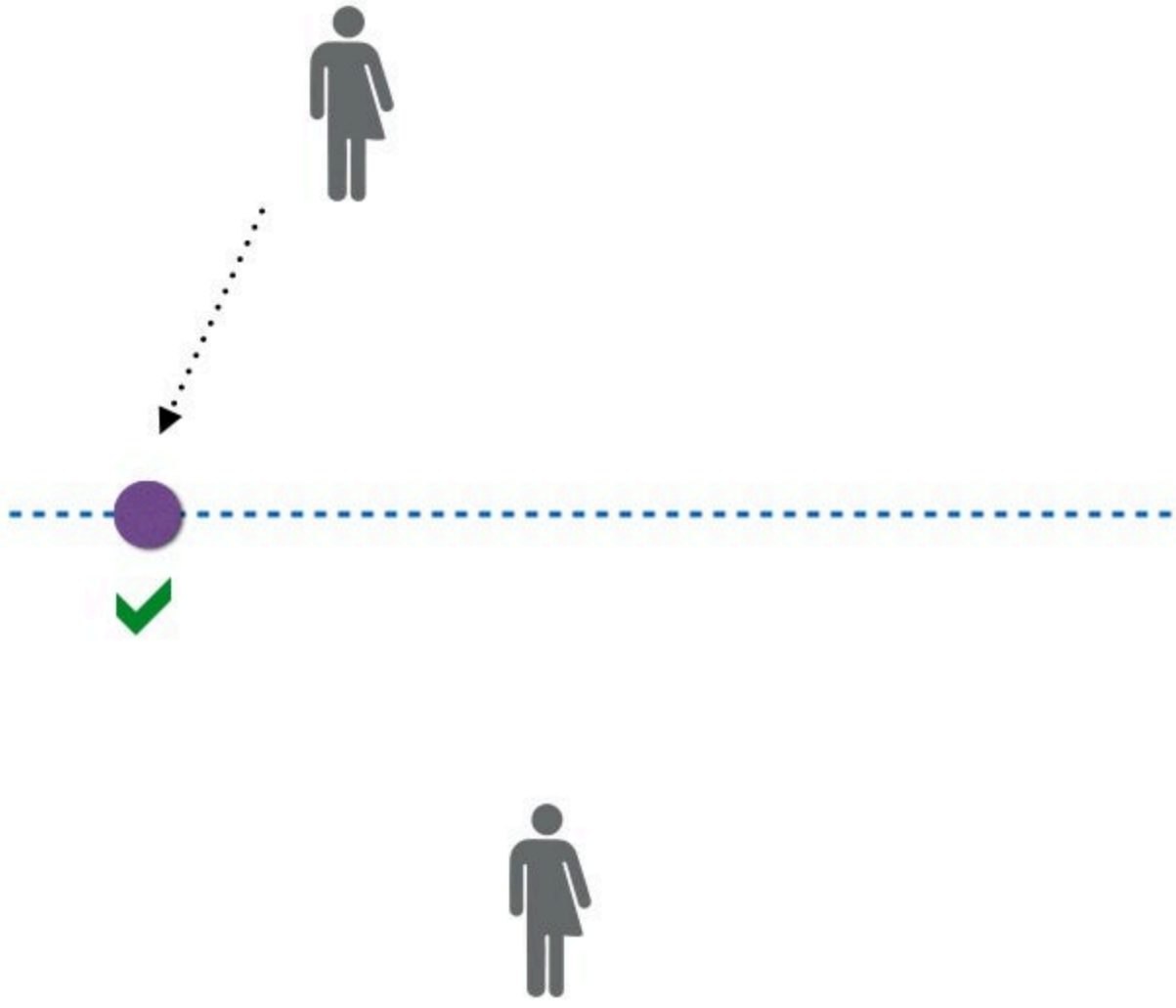
Continuous Integration

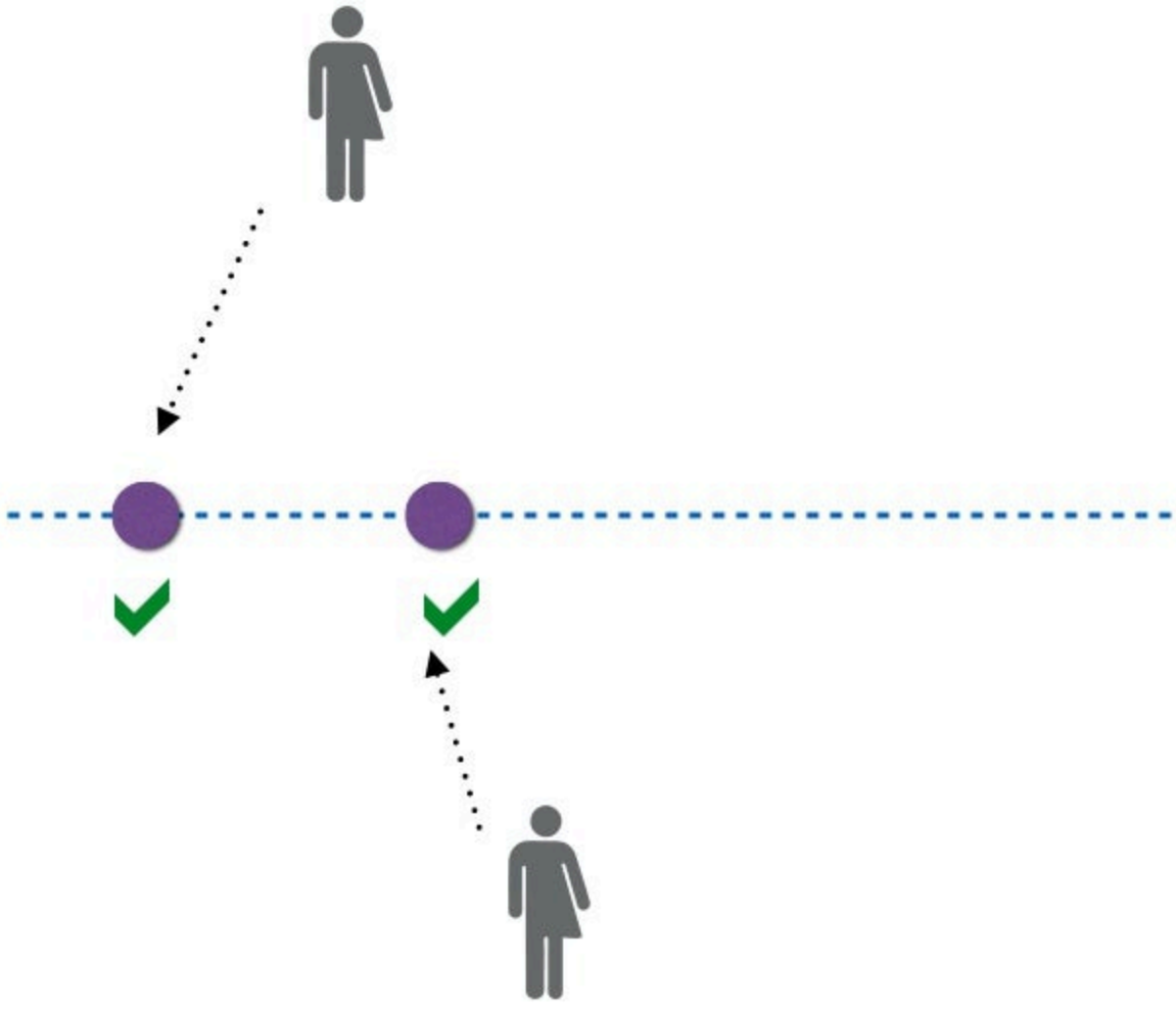

CI Tool


CI Tool

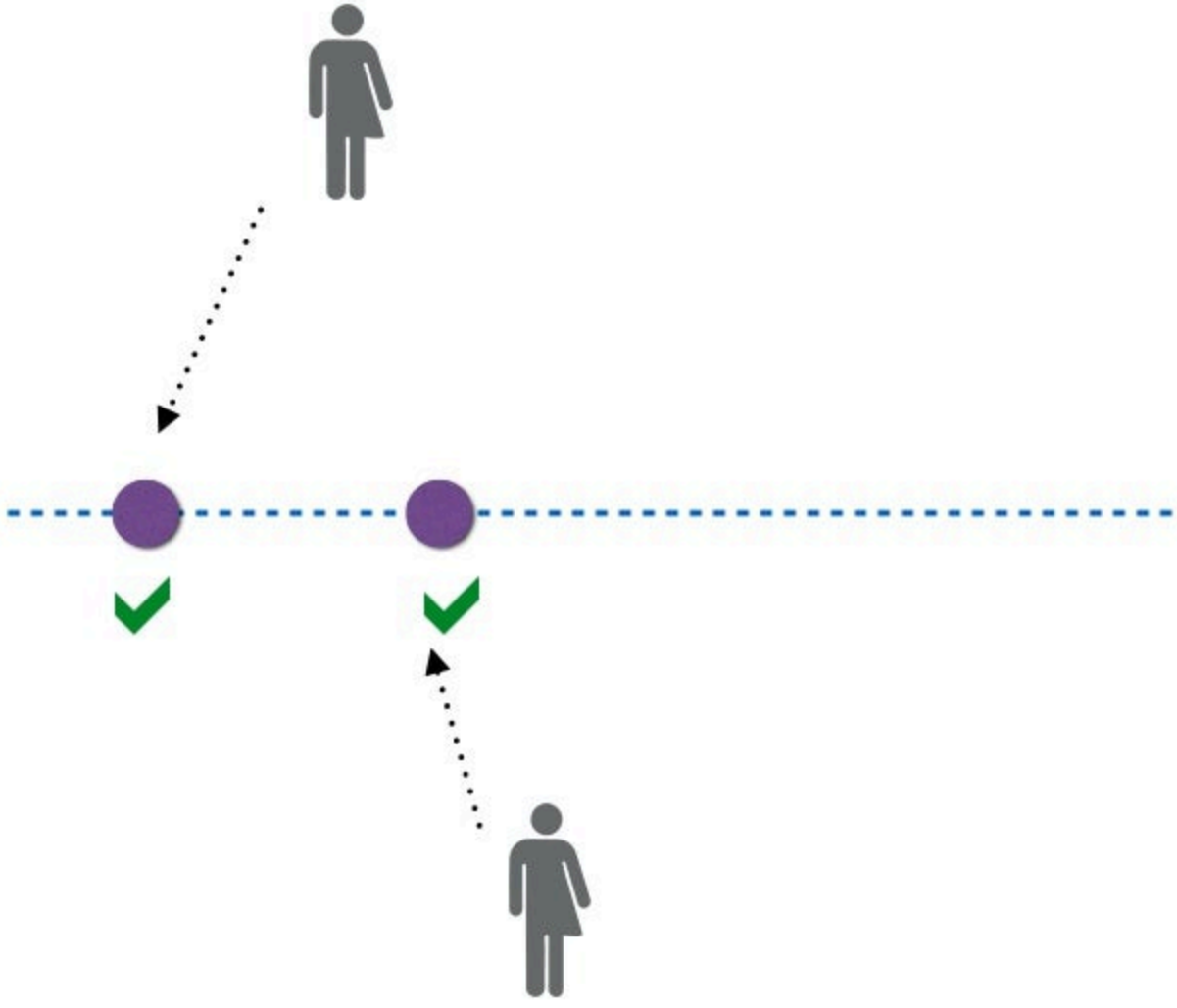

CI Tool



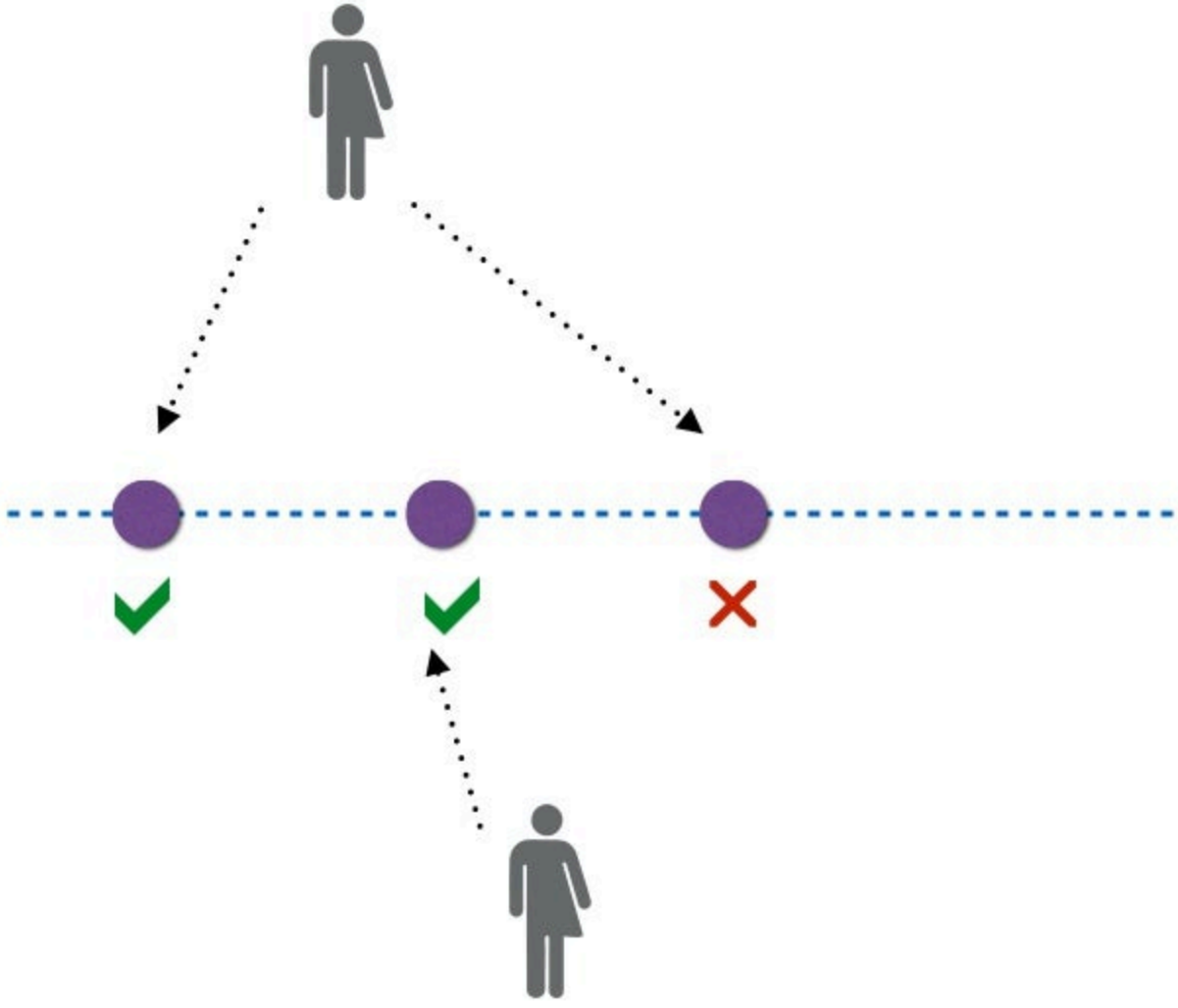




1. Validate the integration

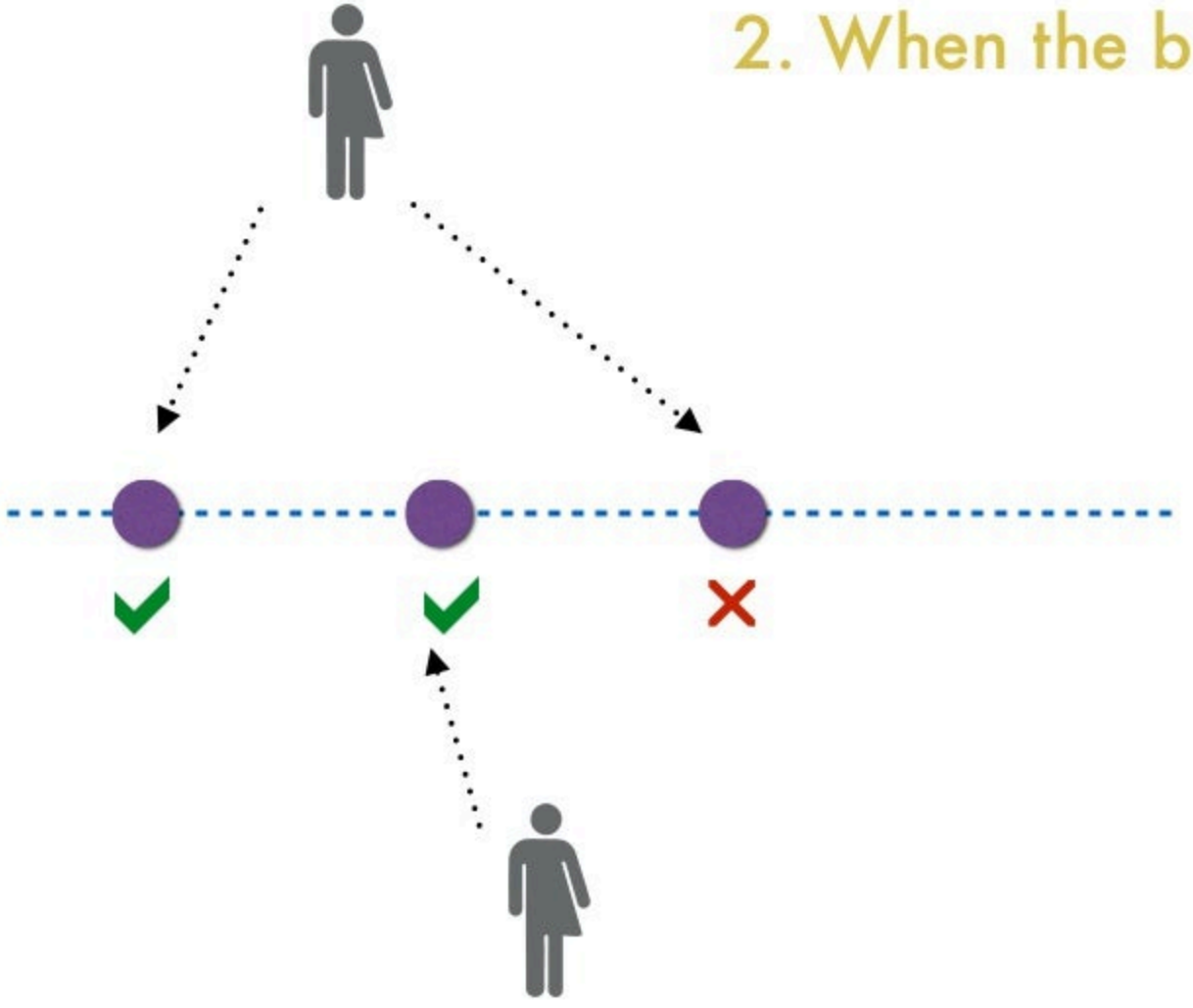


1. Validate the integration



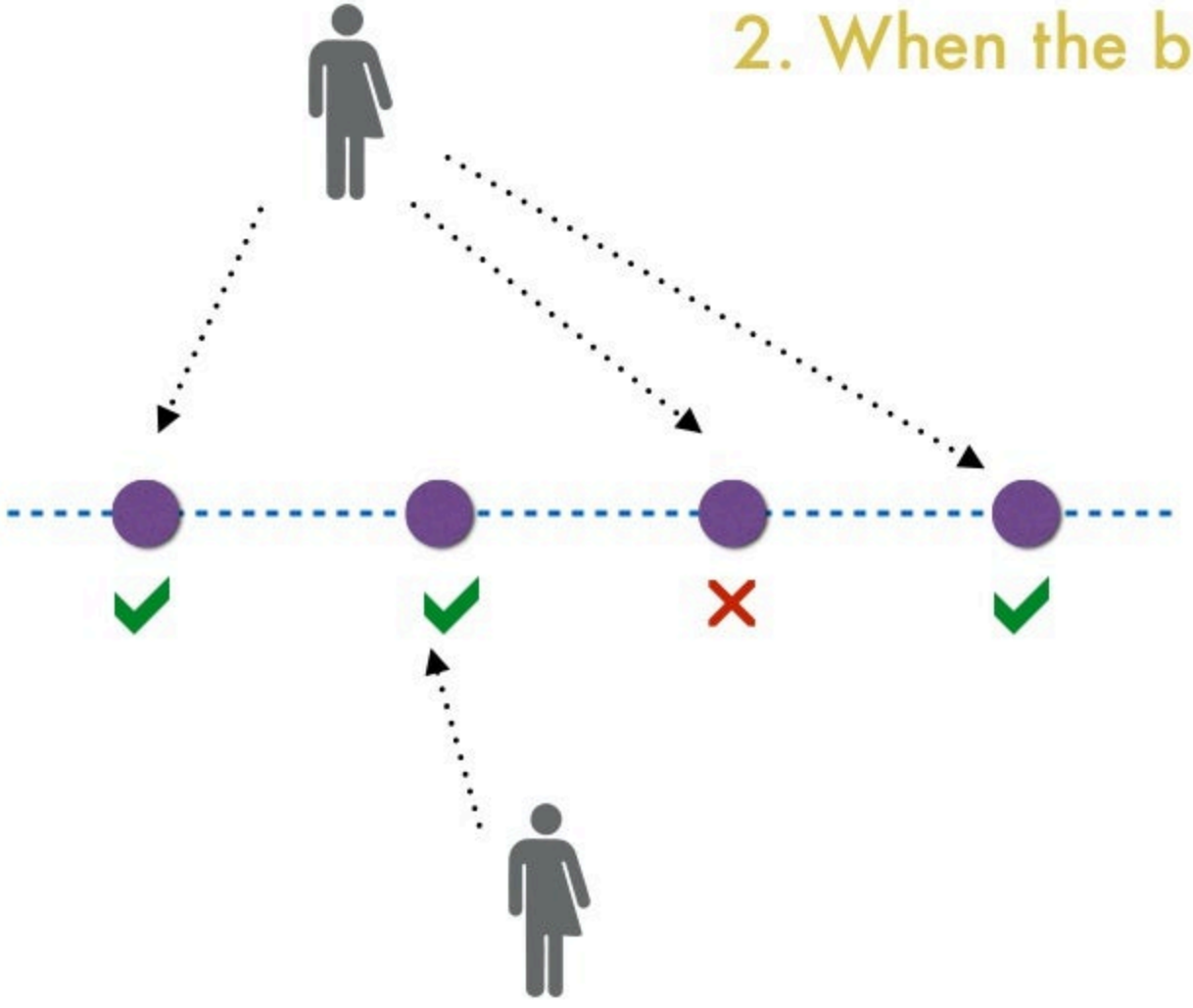
1. Validate the integration

2. When the build breaks, fix it!



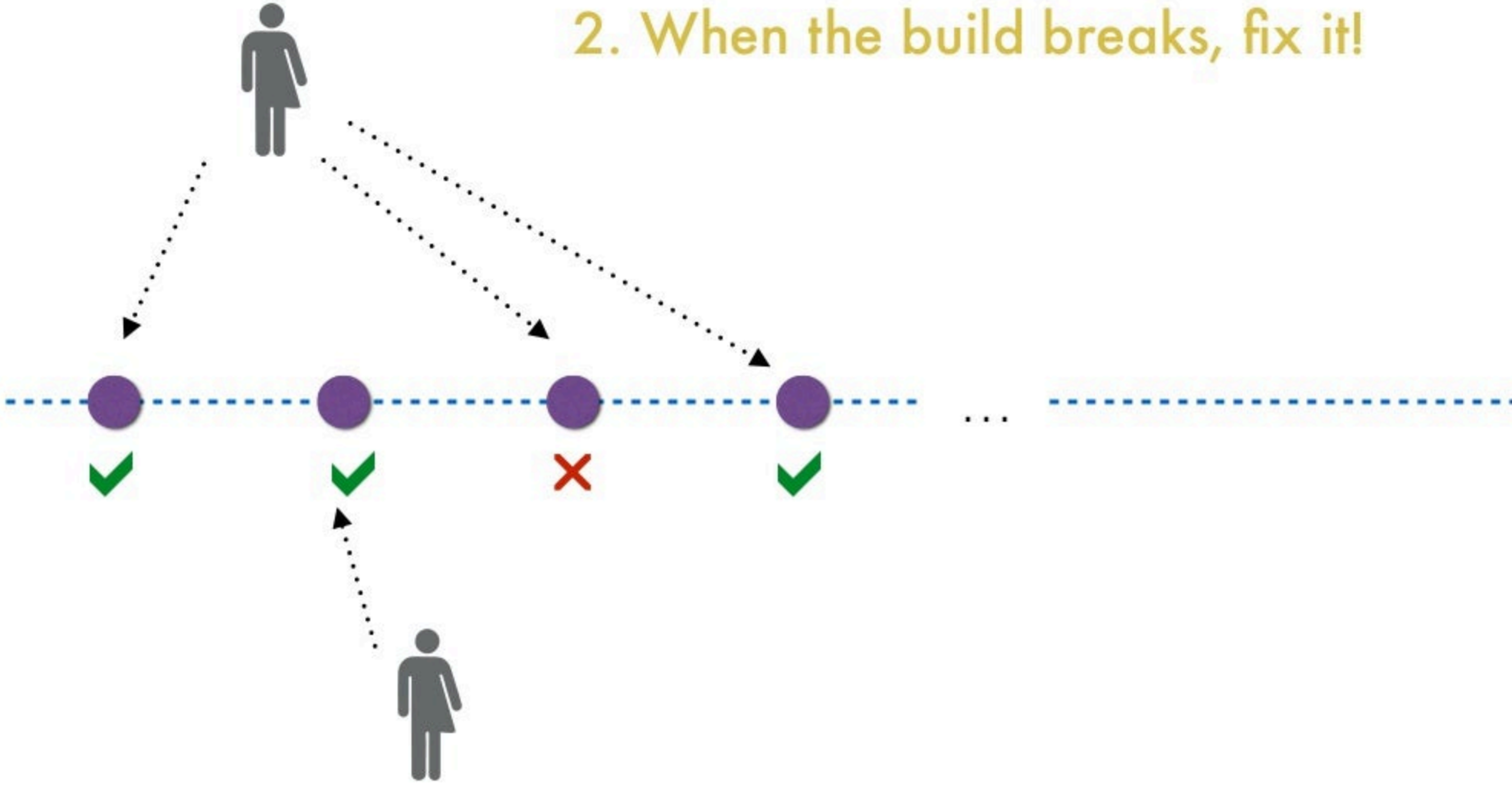
1. Validate the integration

2. When the build breaks, fix it!



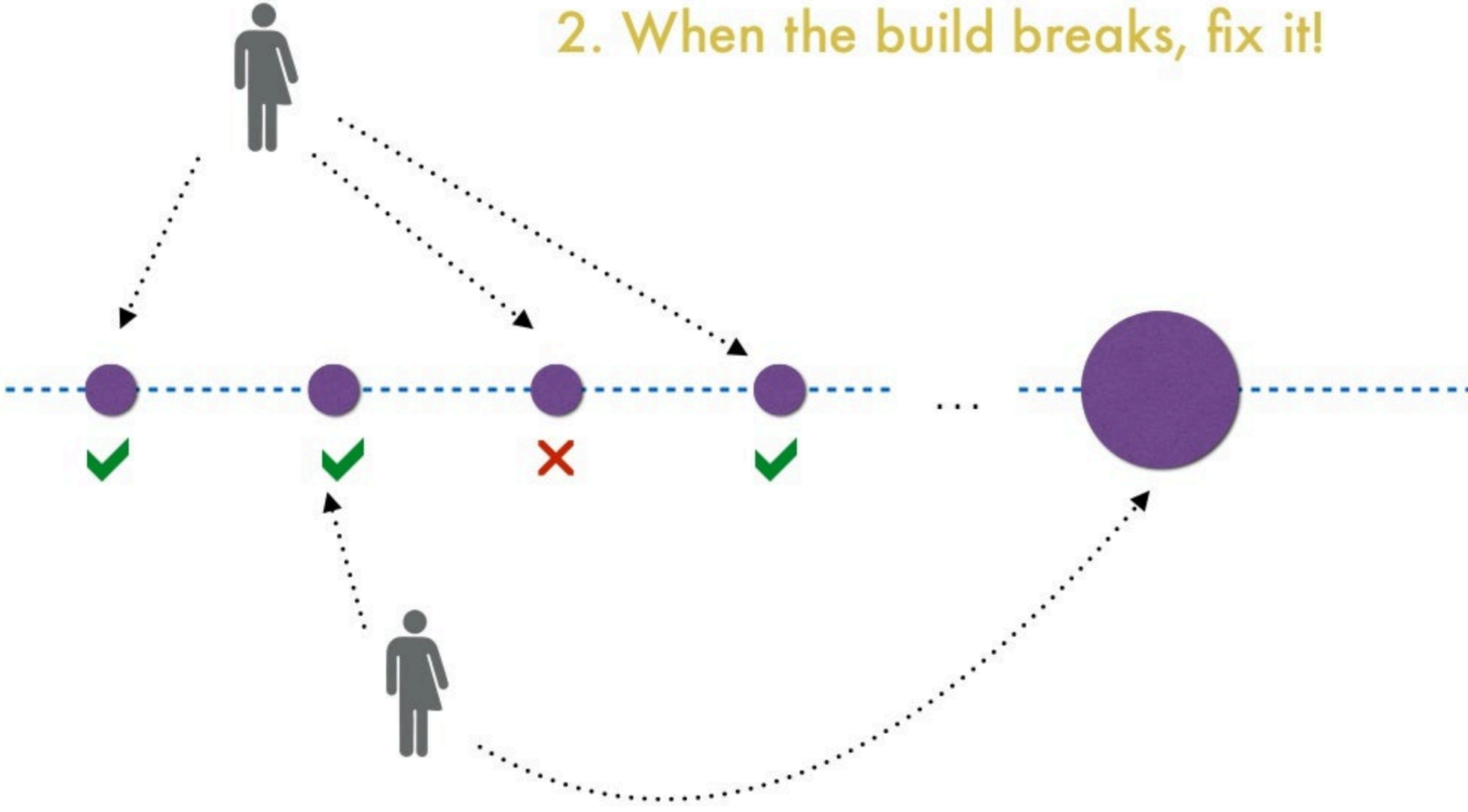
1. Validate the integration

2. When the build breaks, fix it!



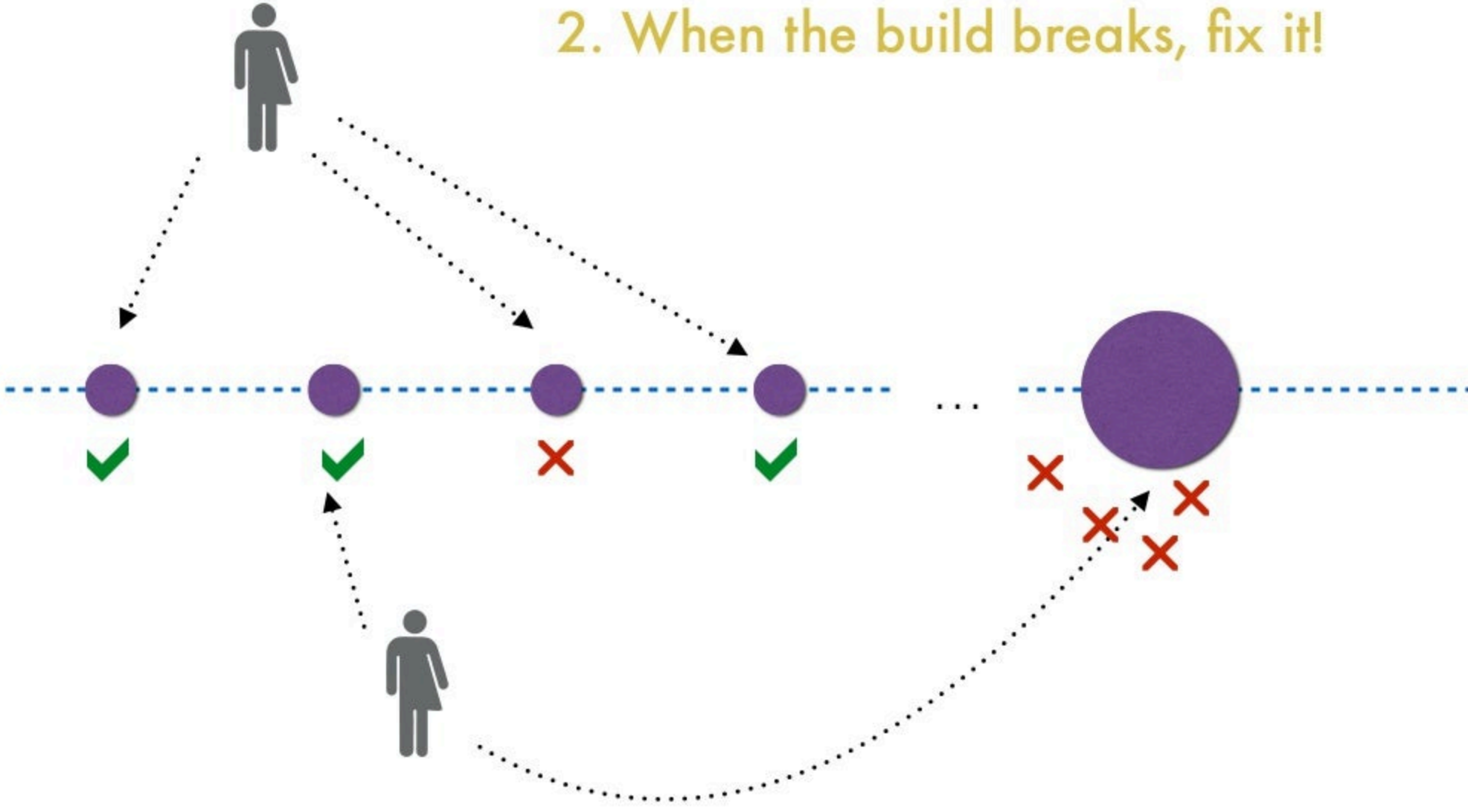
1. Validate the integration

2. When the build breaks, fix it!

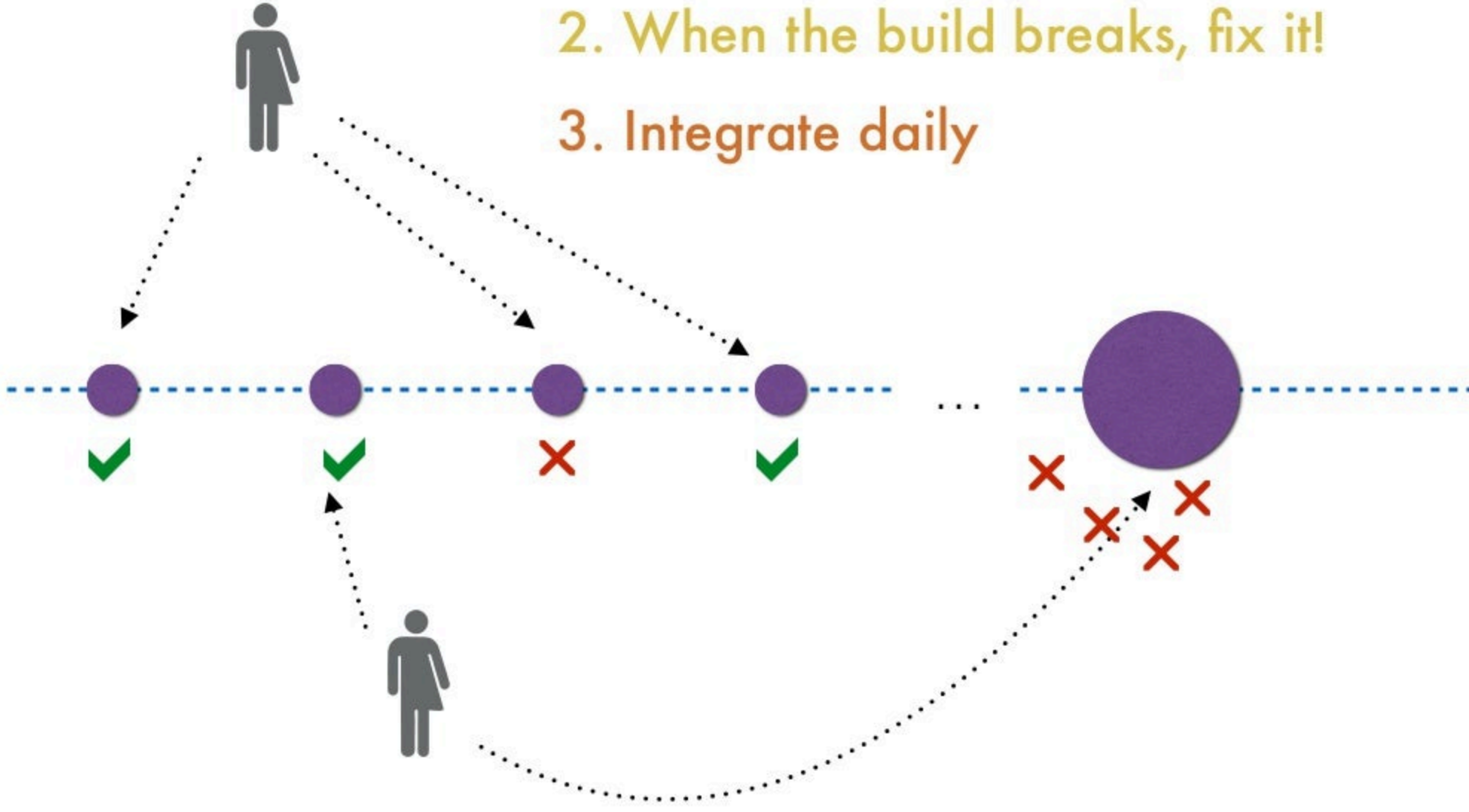


1. Validate the integration

2. When the build breaks, fix it!

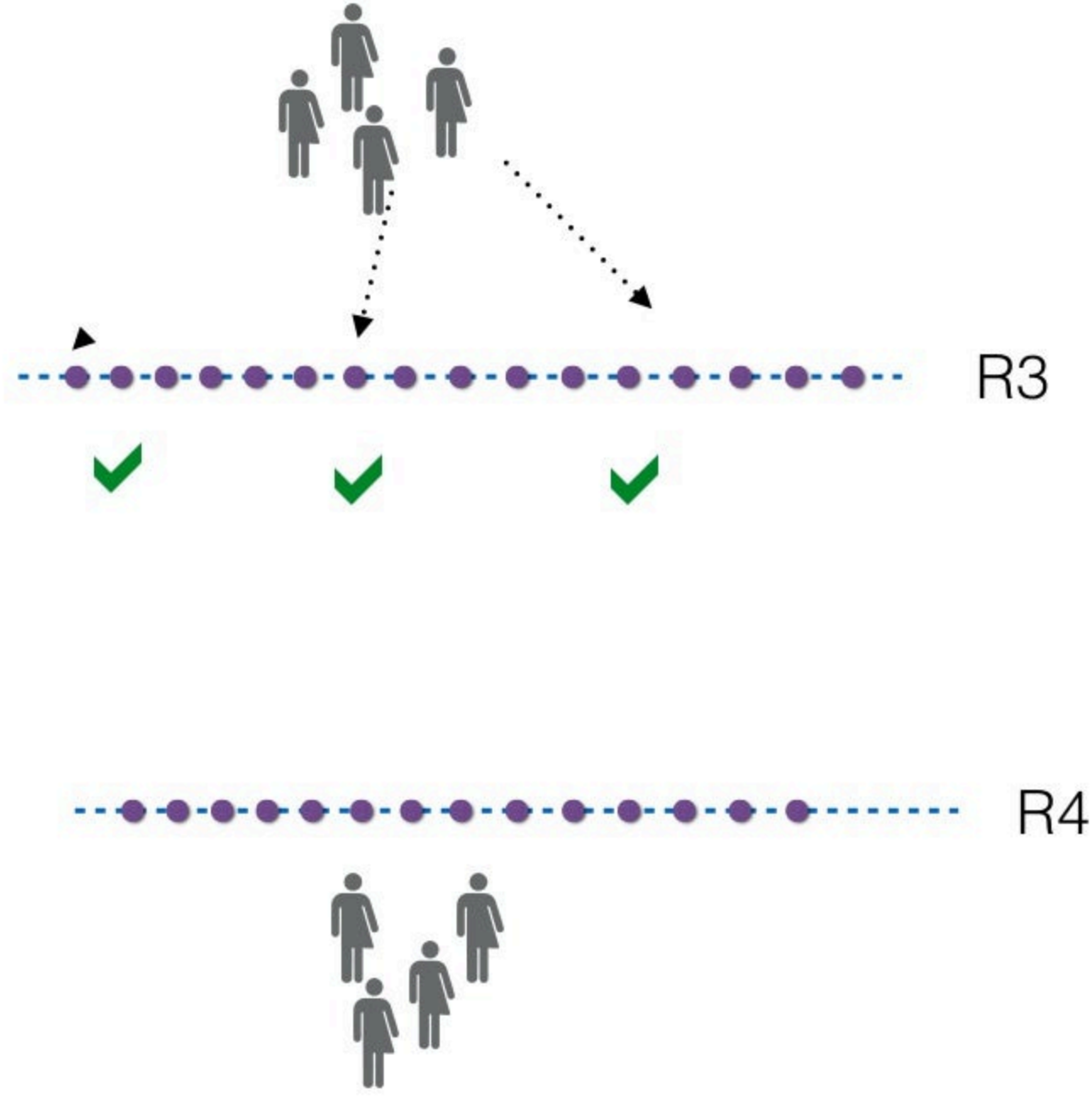


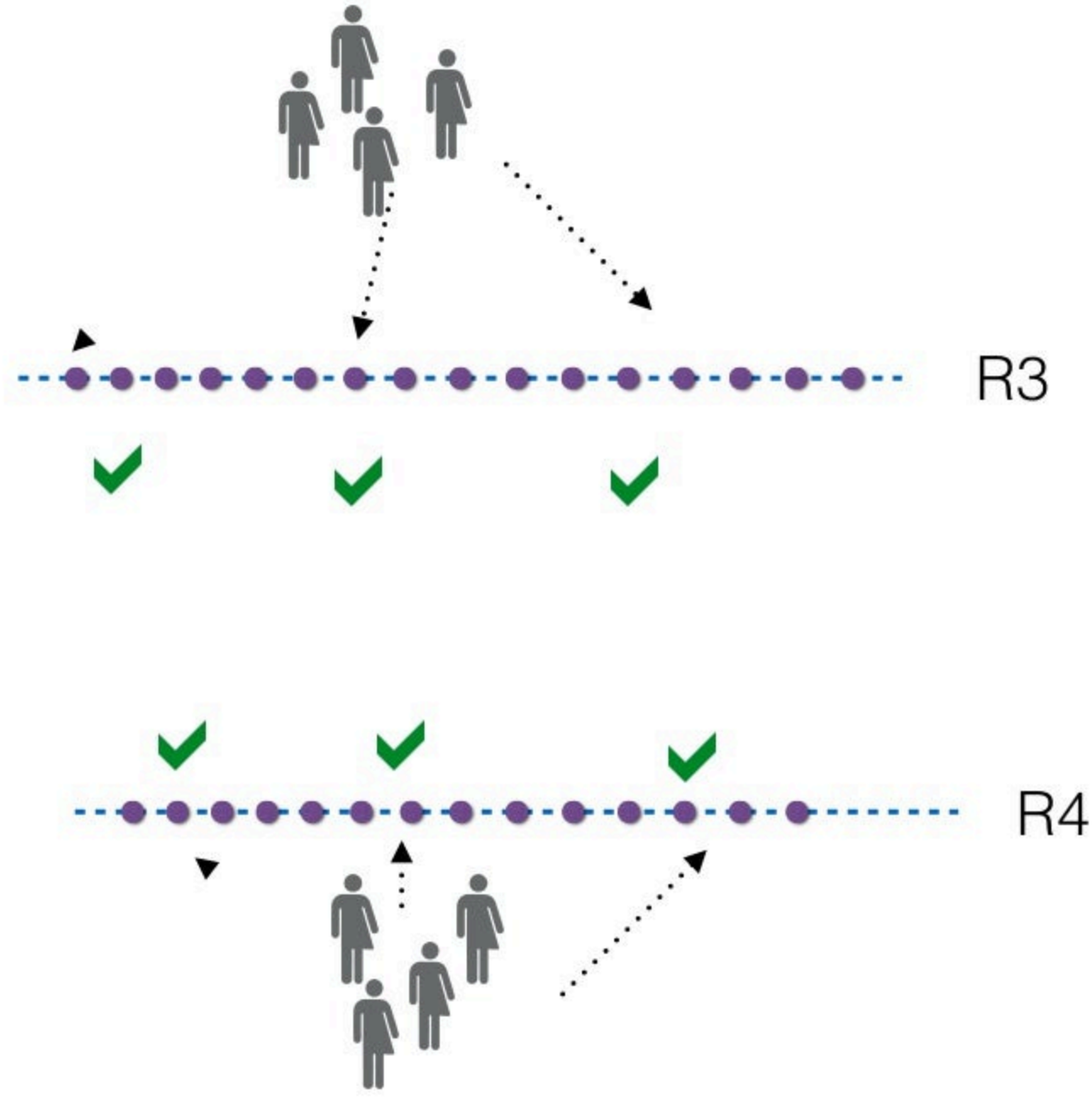
1. Validate the integration
2. When the build breaks, fix it!
3. Integrate daily

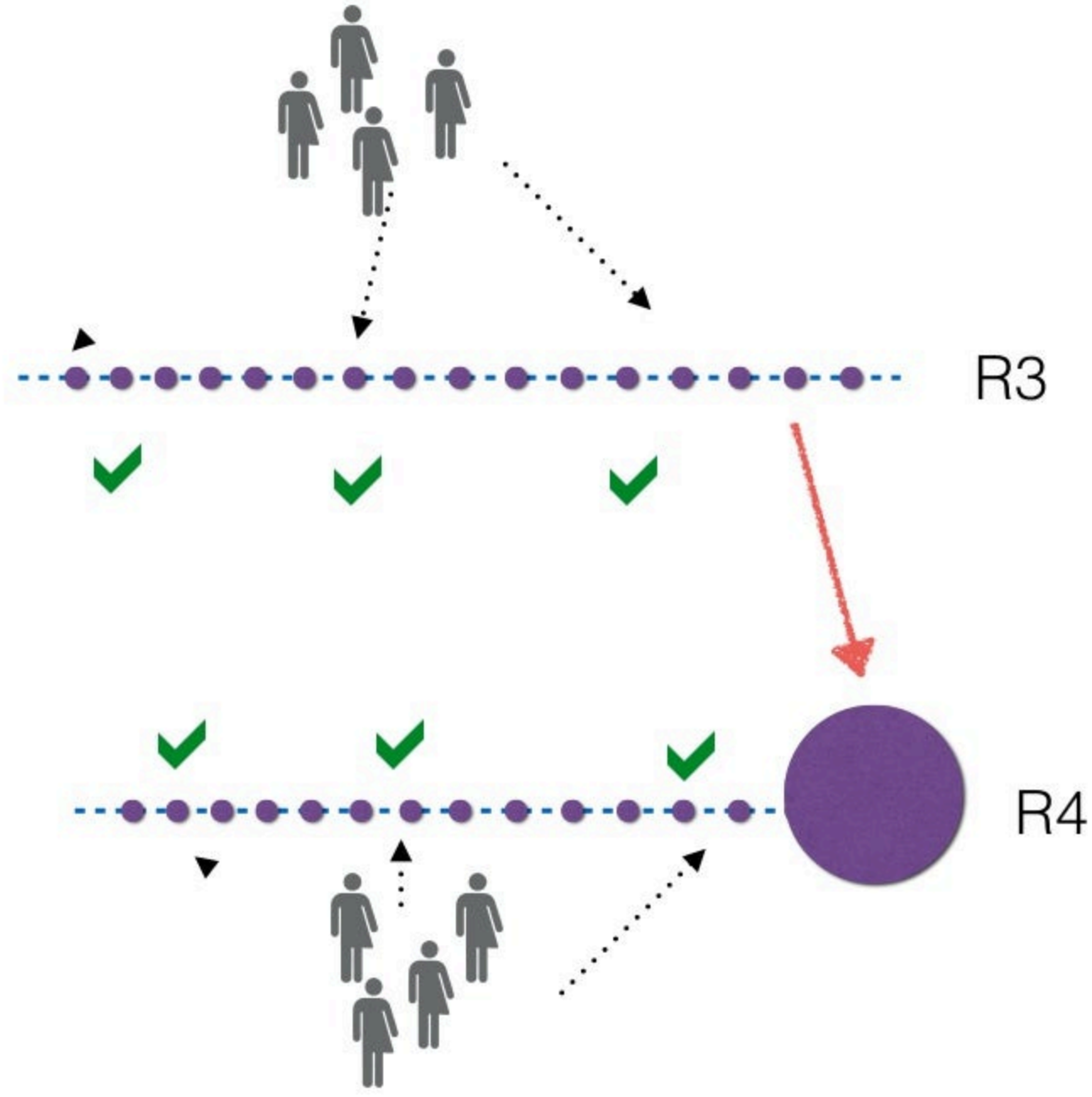



R3


R4


R3
R4


R3
R4


R3
R4

Integrate Once A Day

How can you deal with half-finished work?

Option 1:

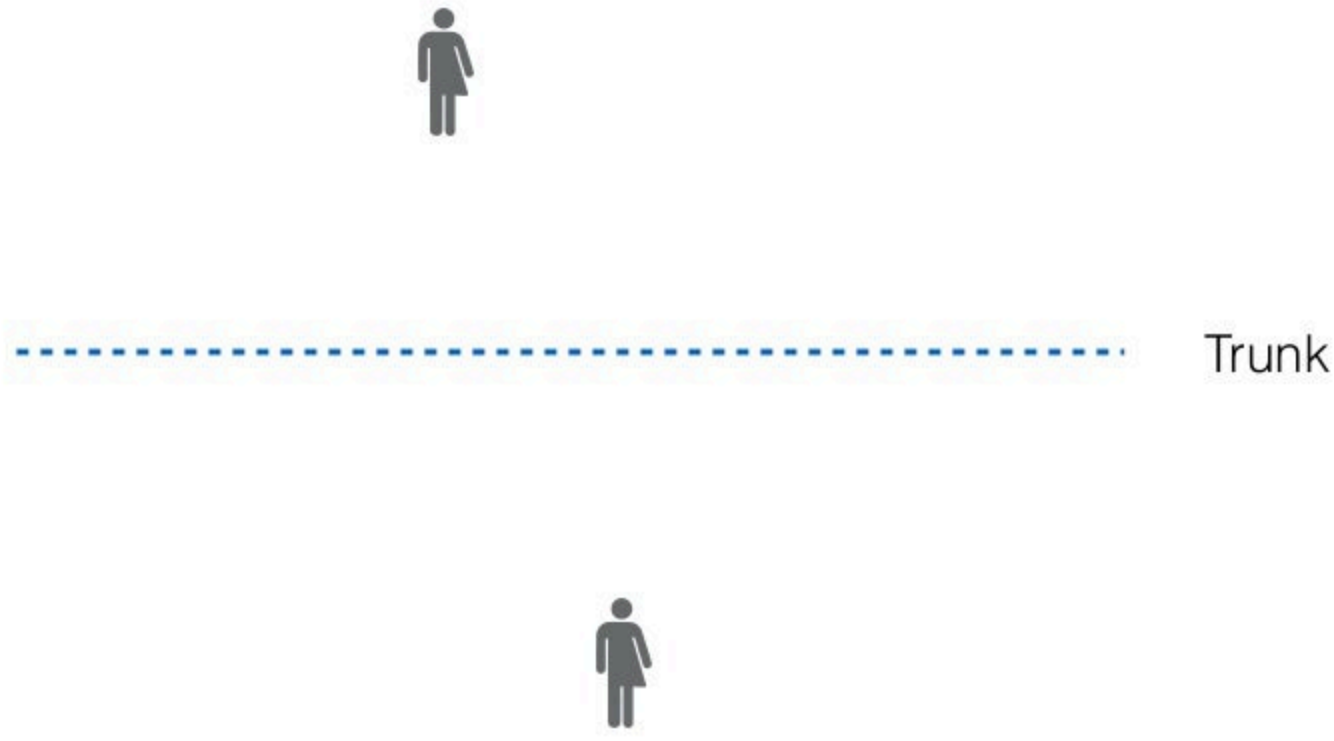
Option 1:

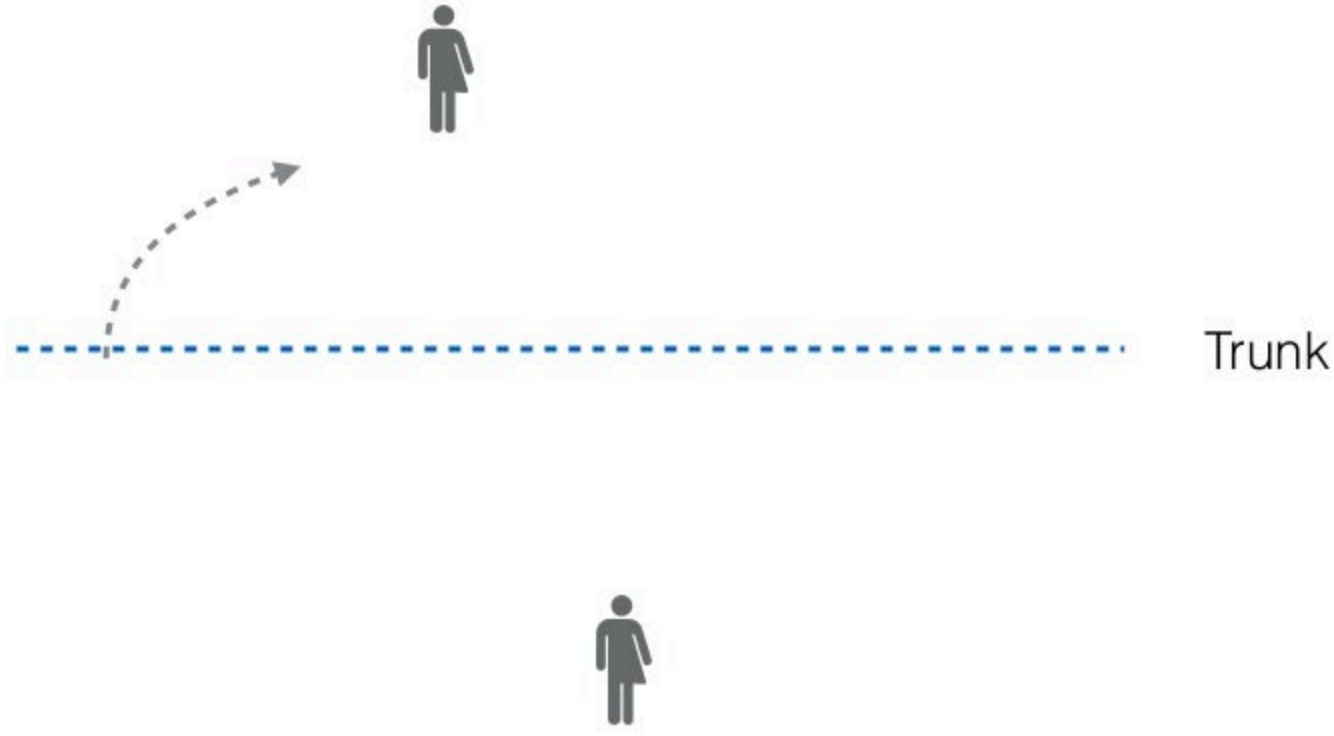
Wait to check in

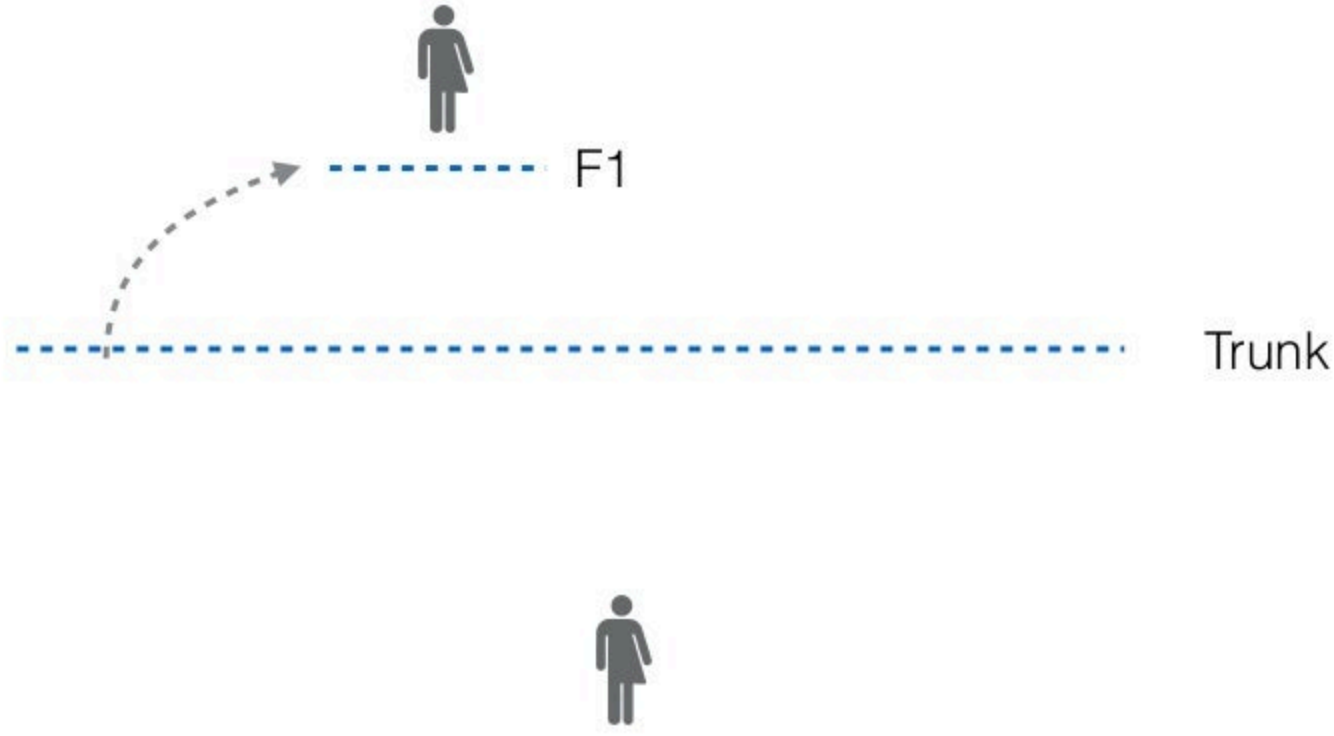
Option 2:

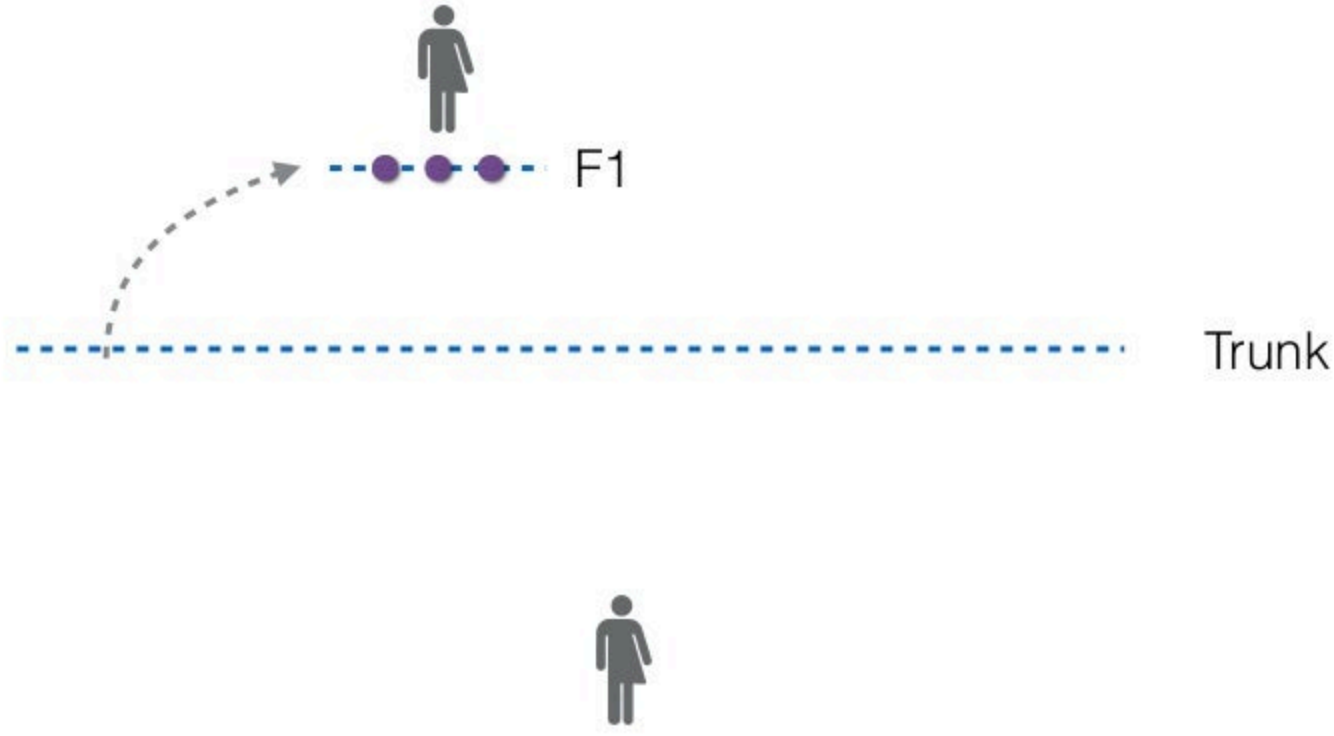
Option 2:

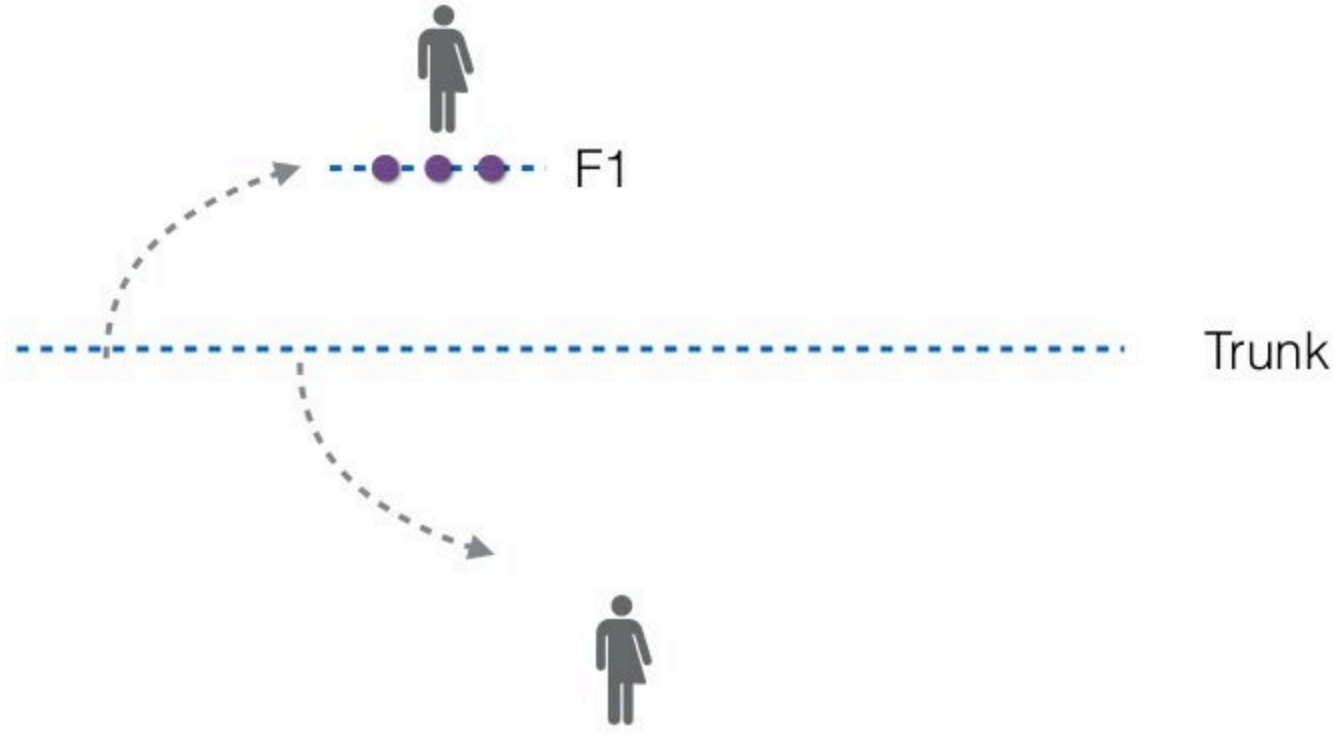
Make a branch!

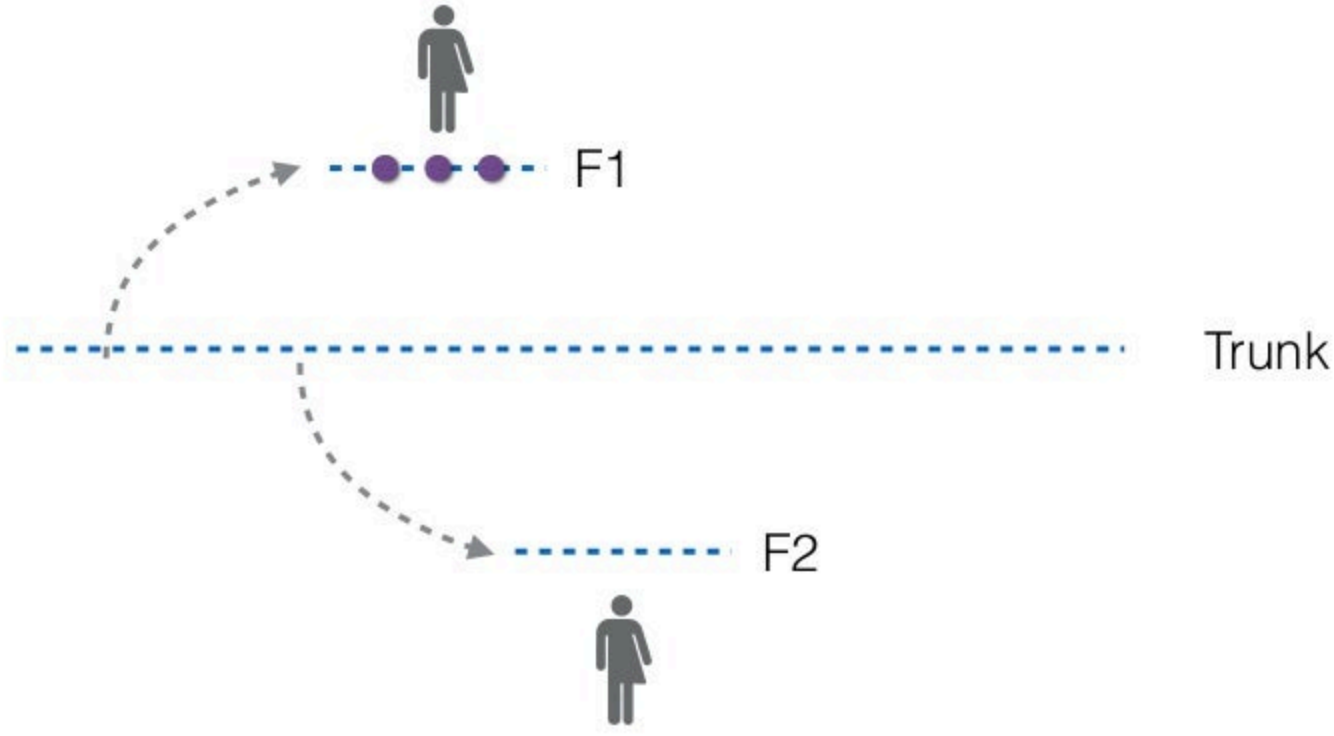

Trunk

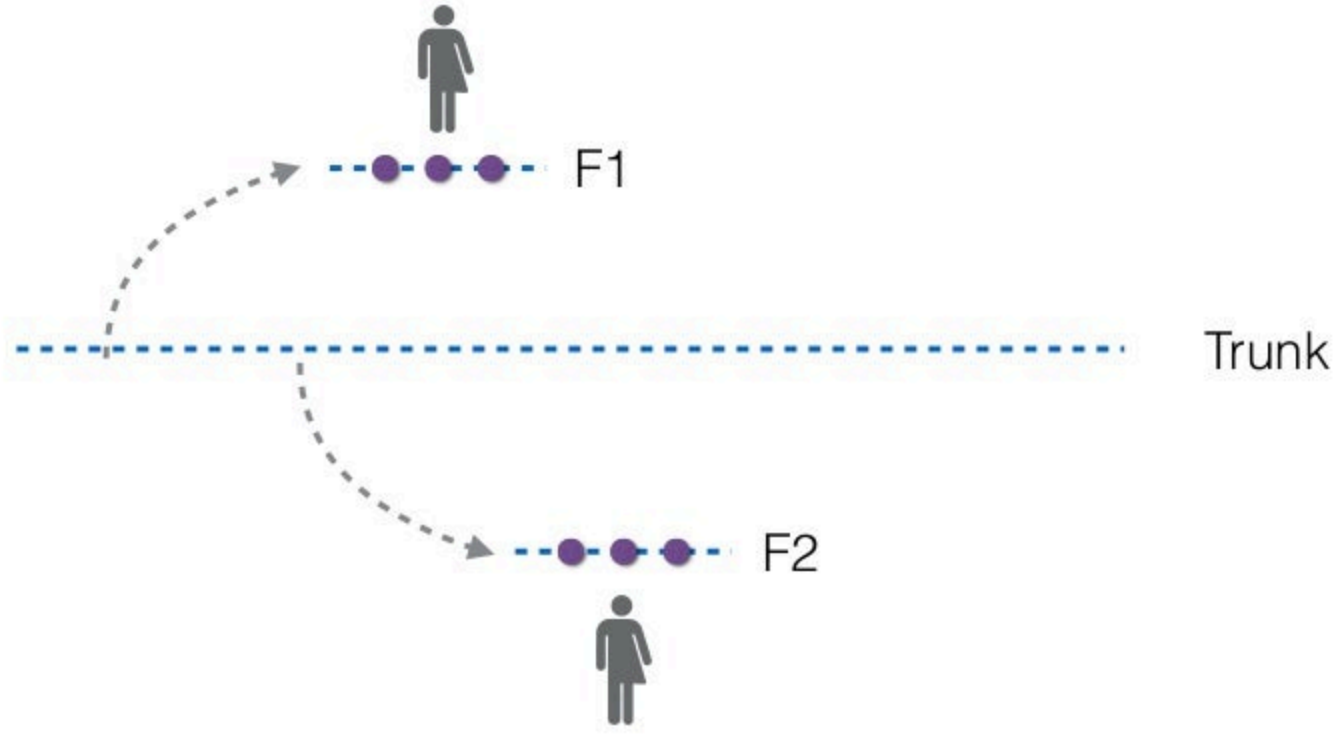

Trunk

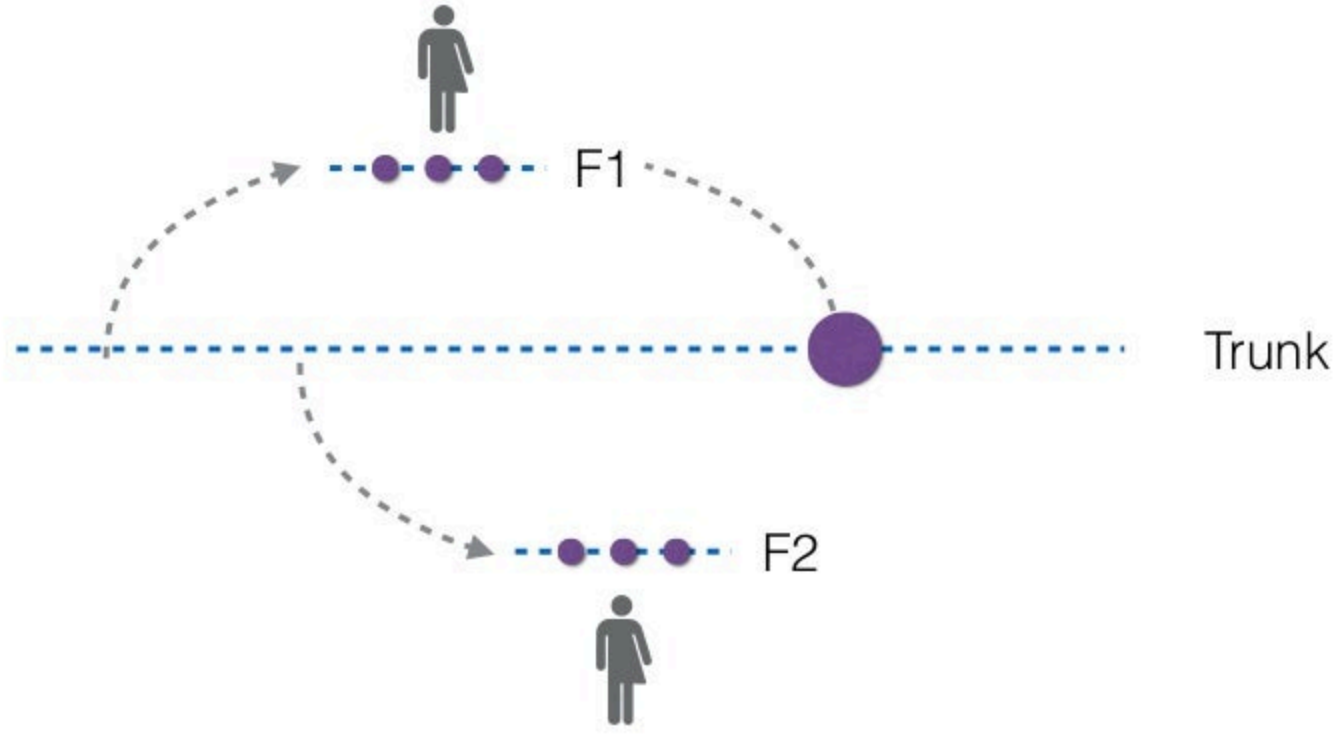

F1
Trunk

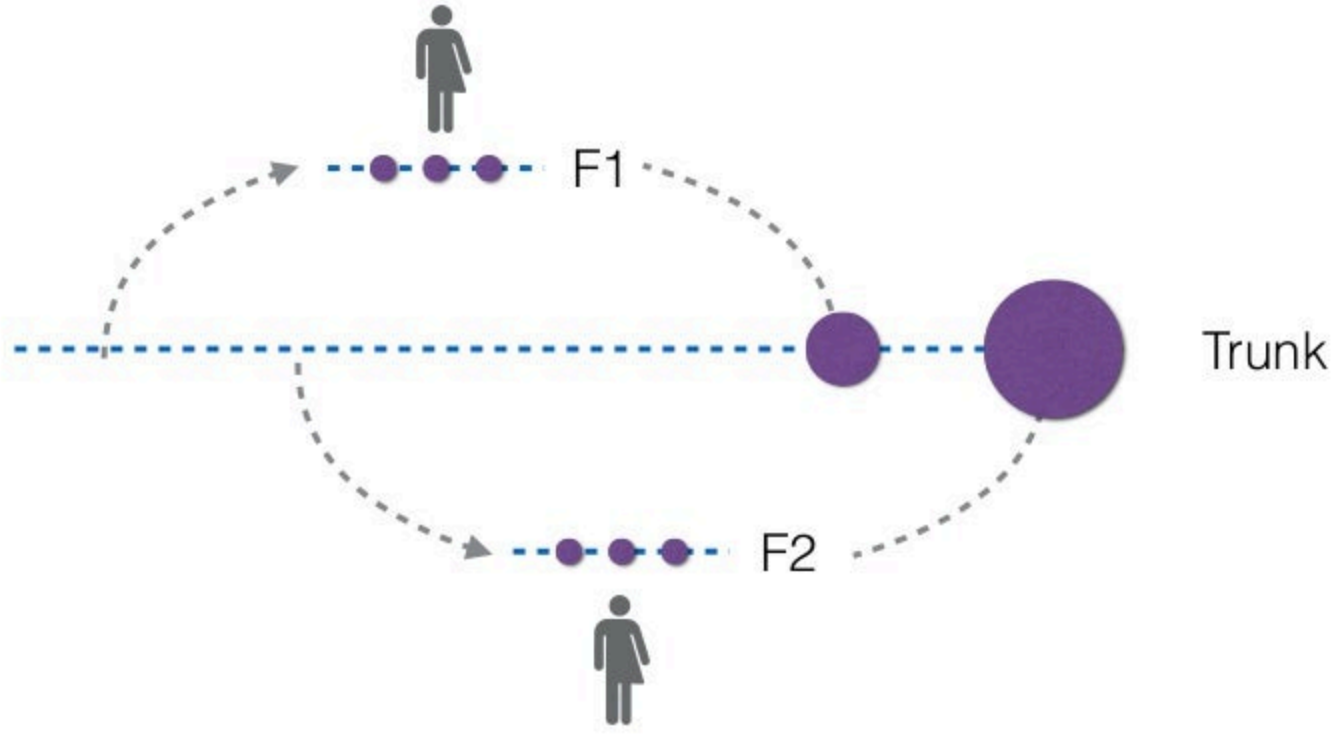

F1
Trunk

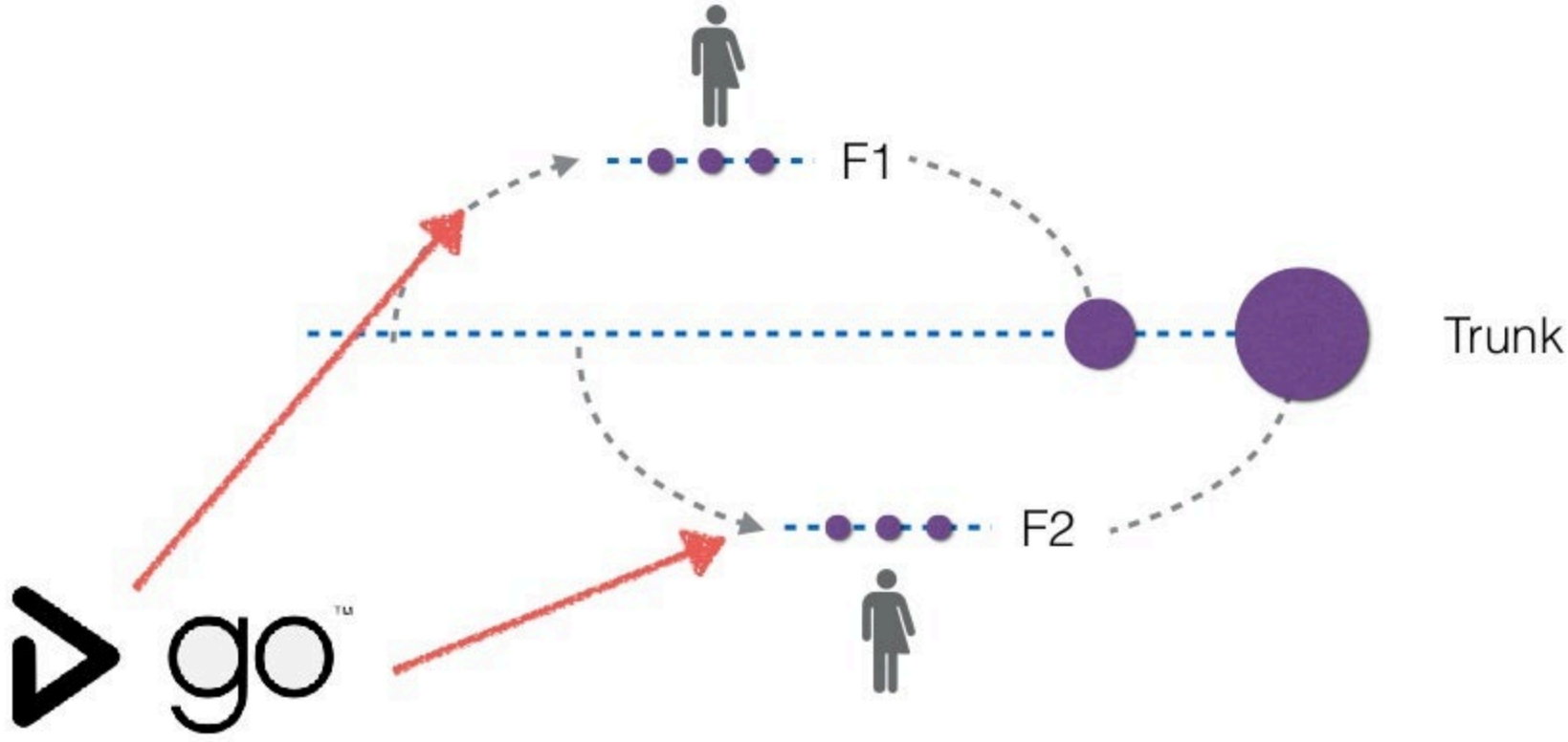

F1
Trunk

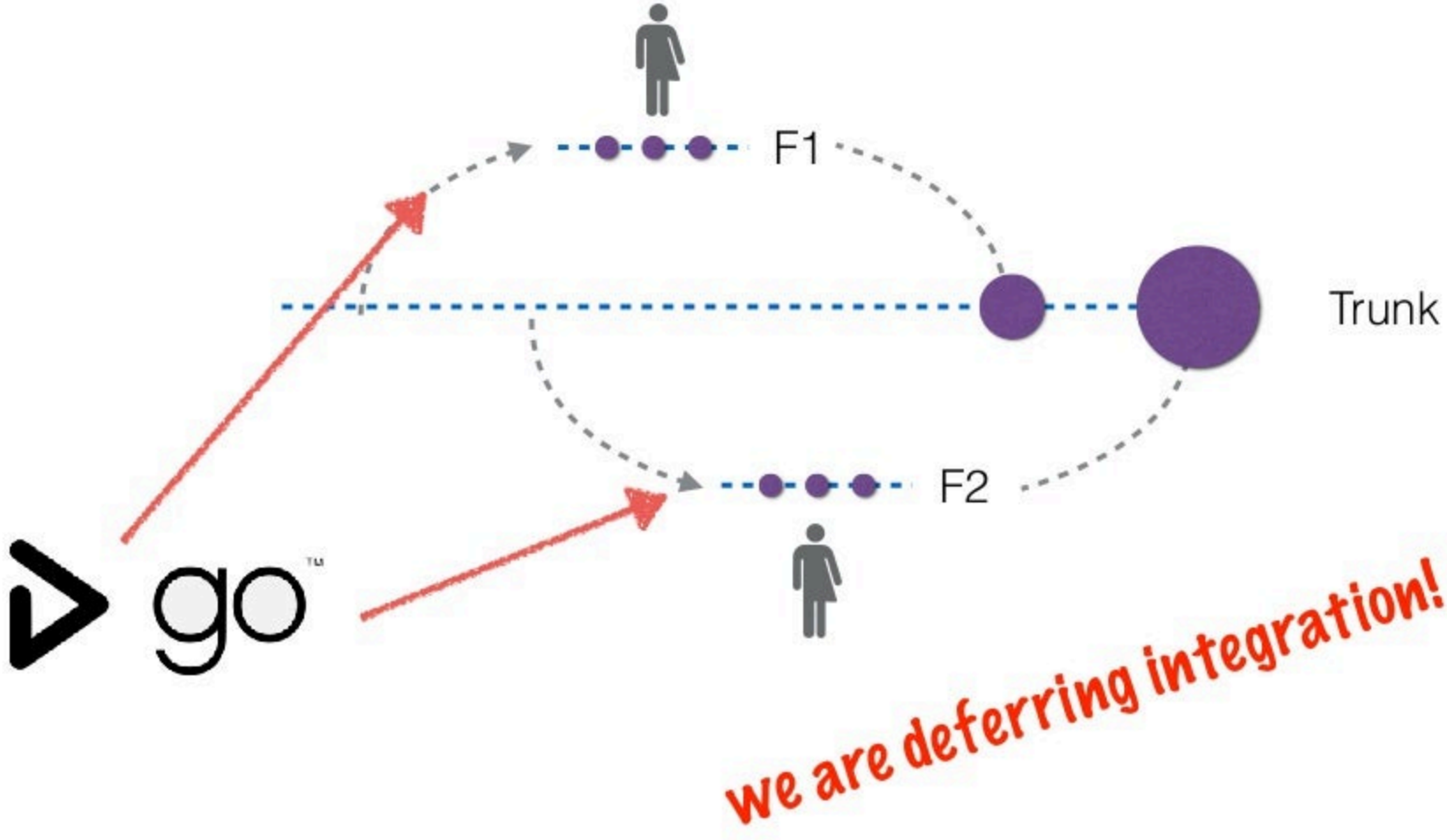

F1
Trunk
F2


F1
Trunk
F2


F1
Trunk
F2


F1
Trunk
F2


F1
Trunk
F2
go



we are deferring integration!

Pain of merge = fn (
size_of_merge,
duration_since_last_merge)

Big merges = commit race!

Merging refactoring is
really hard

Option 3:

Option 3:

Check in anyway

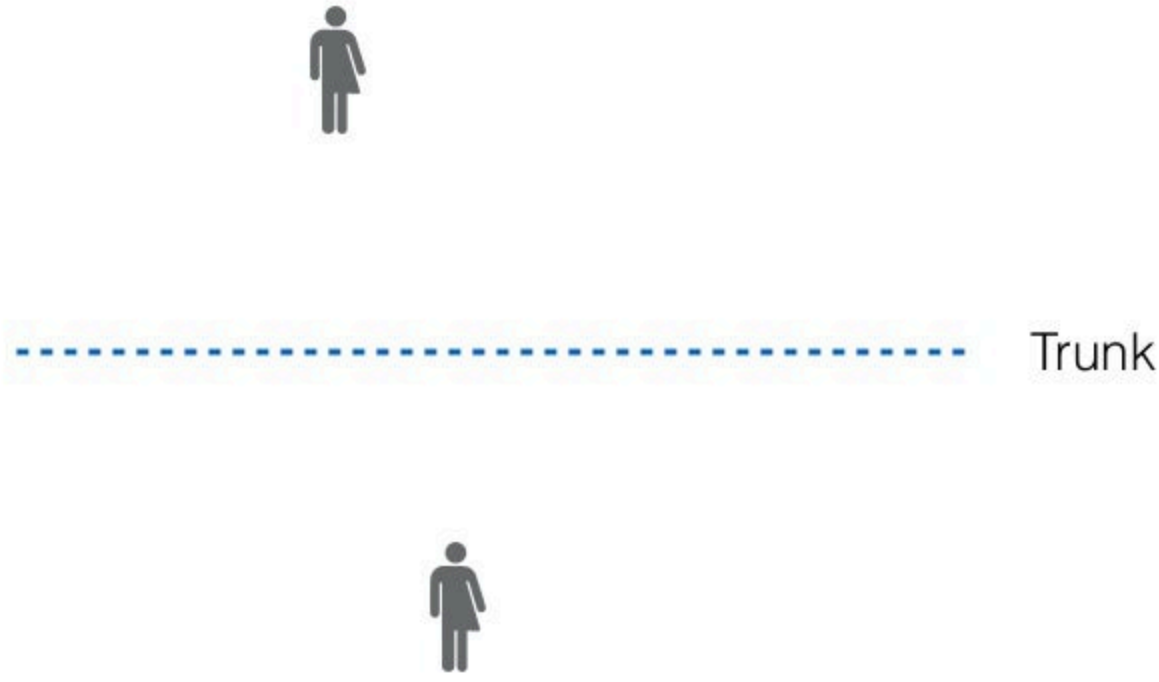
Option 3:

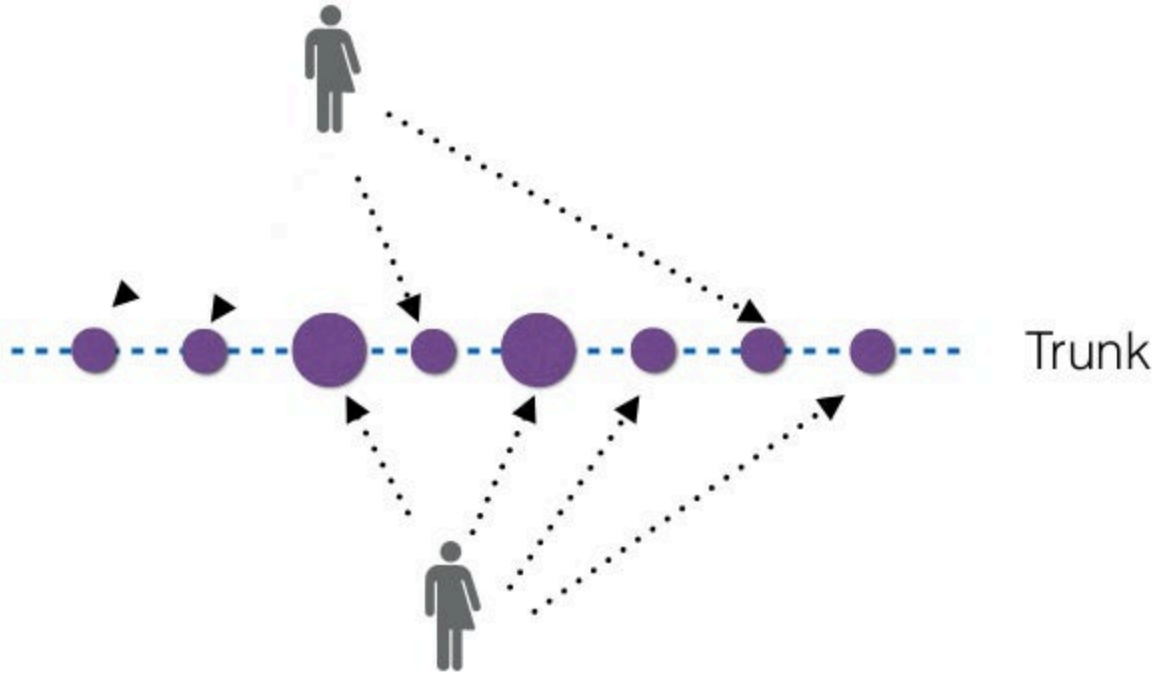
Check in anyway

err...wat?

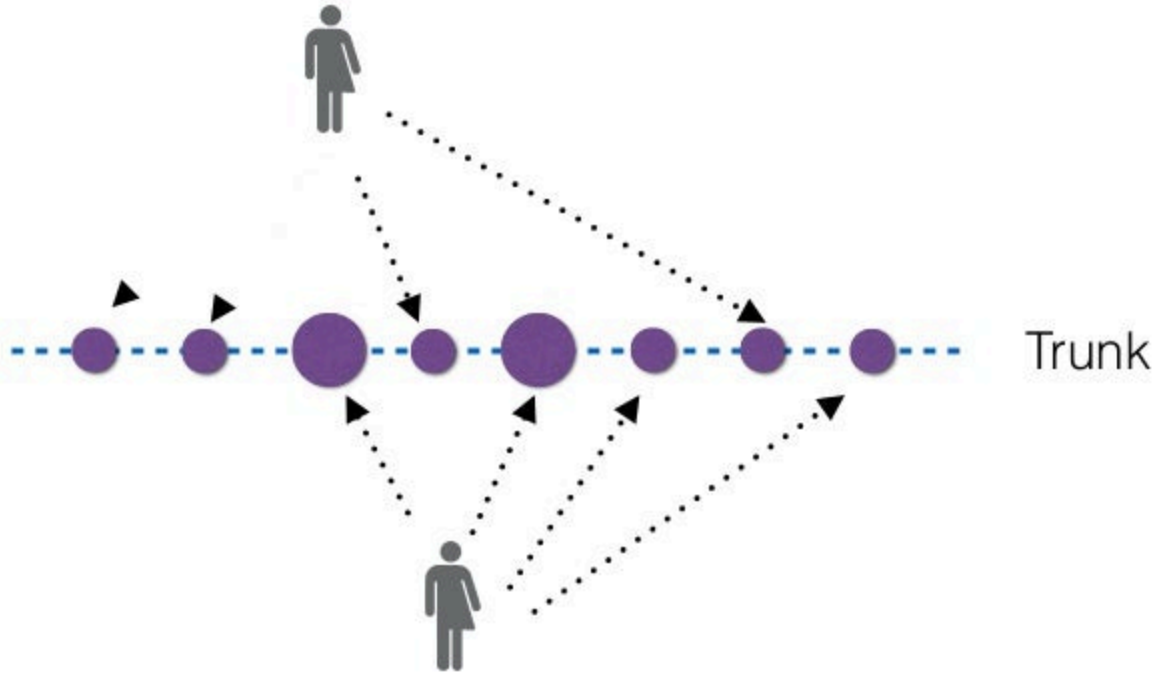
Trunk-based development

Everyone integrates into trunk


Trunk

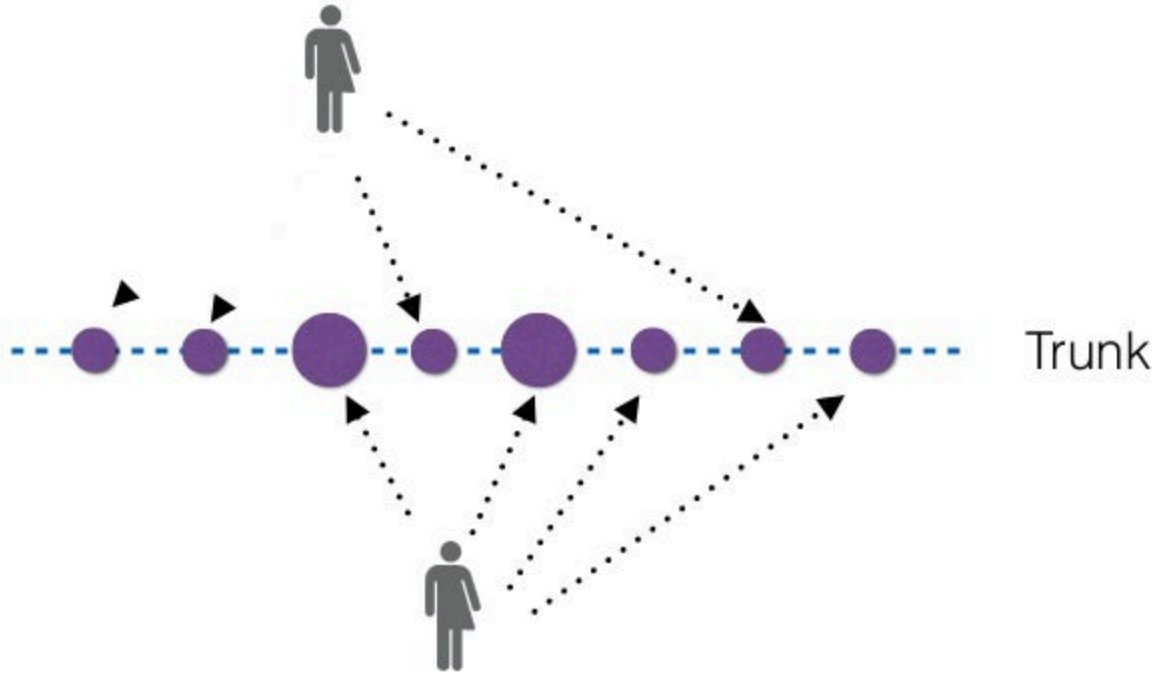

Trunk

integrate often...



integrate often...

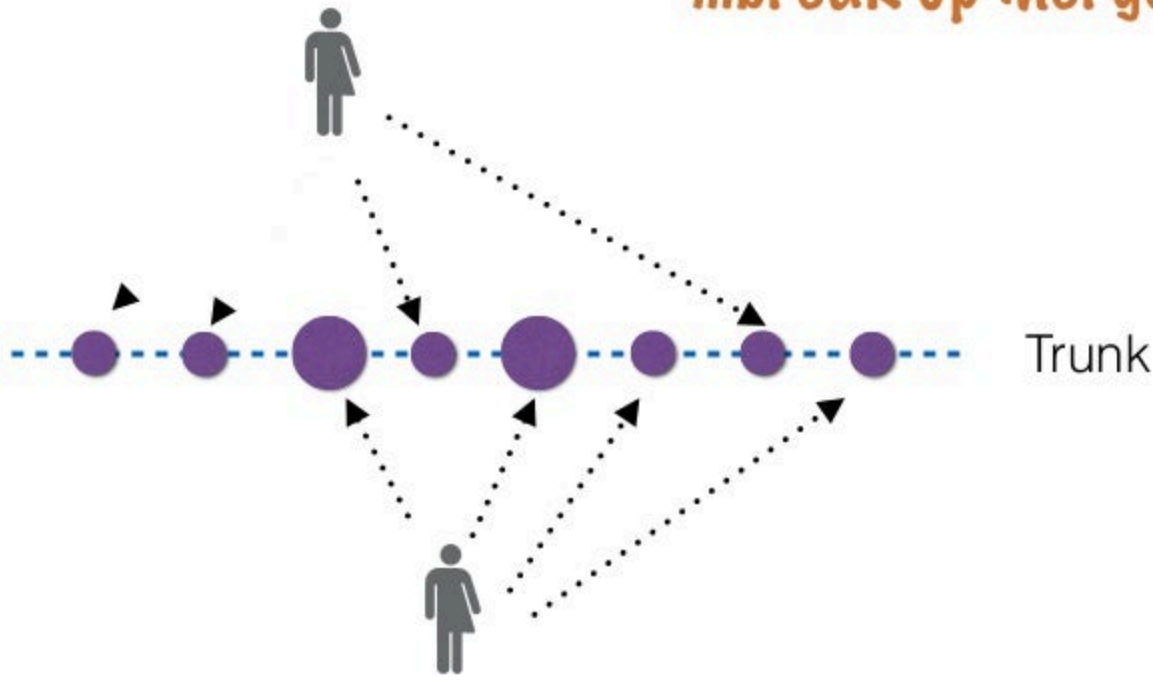
...fast feedback...



integrate often...

...fast feedback...

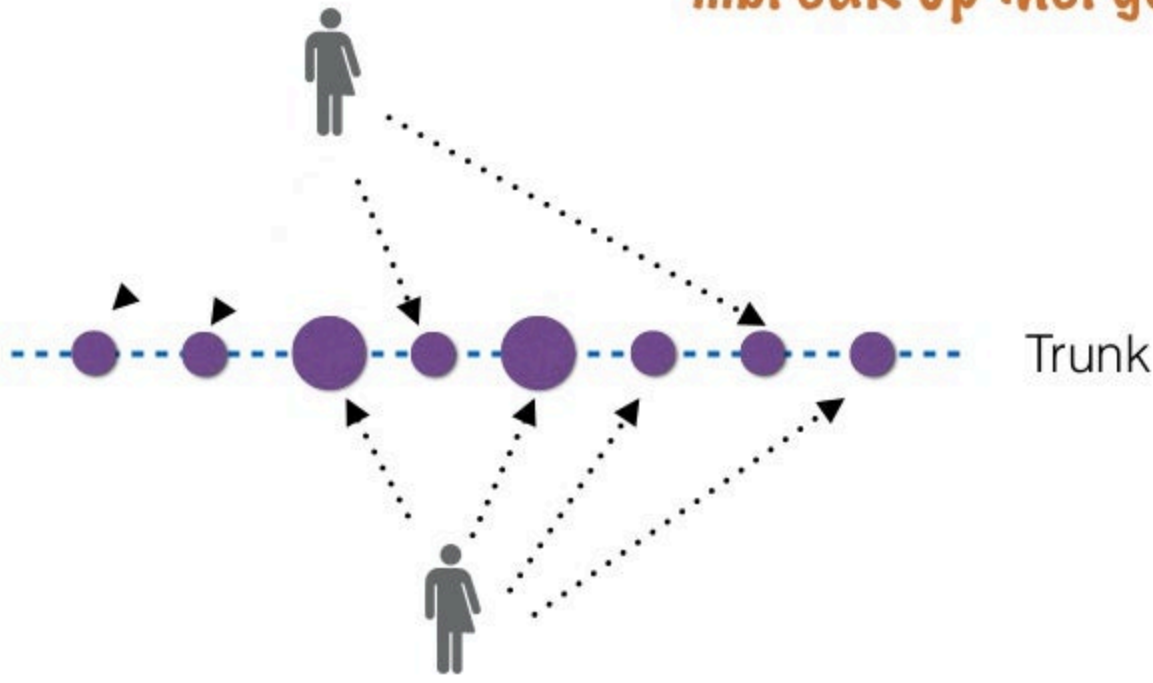
...break up merge pain



integrate often...

...fast feedback...

...break up merge pain

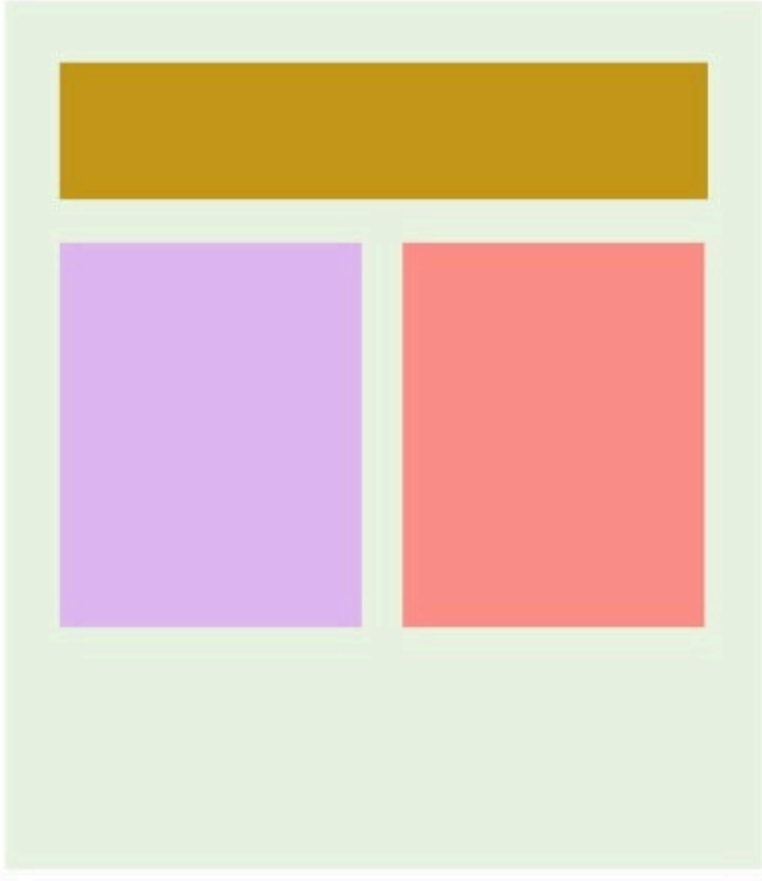


err...but what about half-finished features?

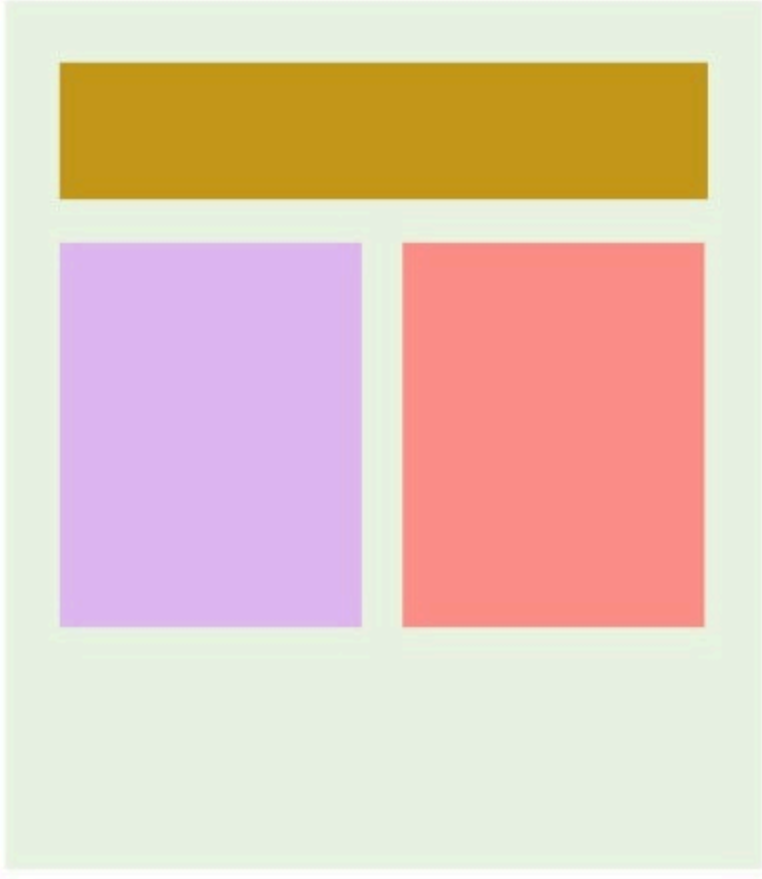
Feature Toggles

Feature Toggles

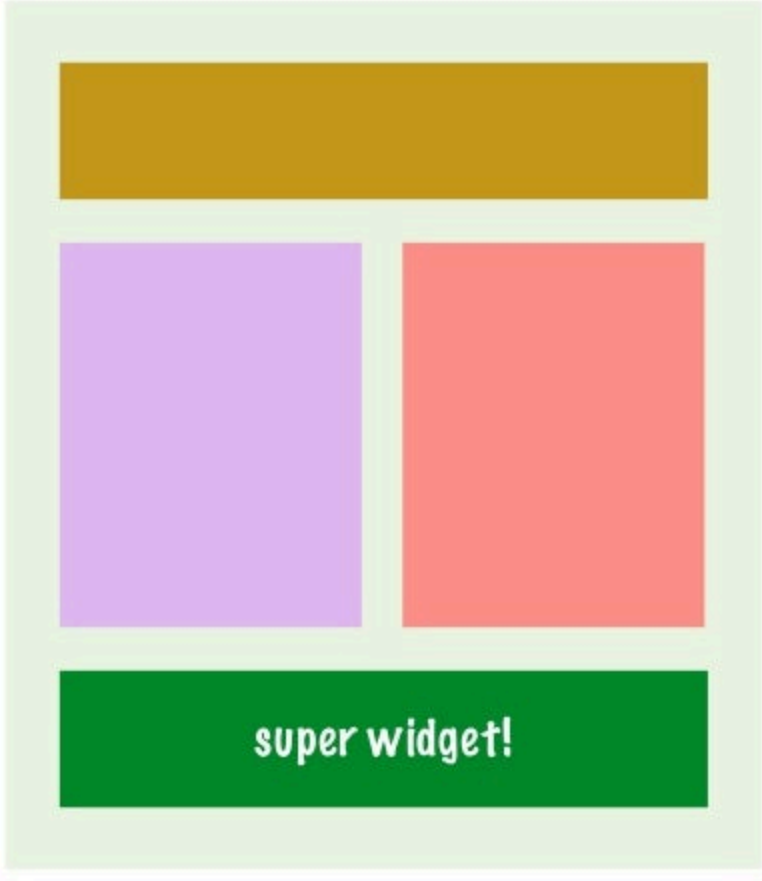
Hide the partially implemented feature in the running system



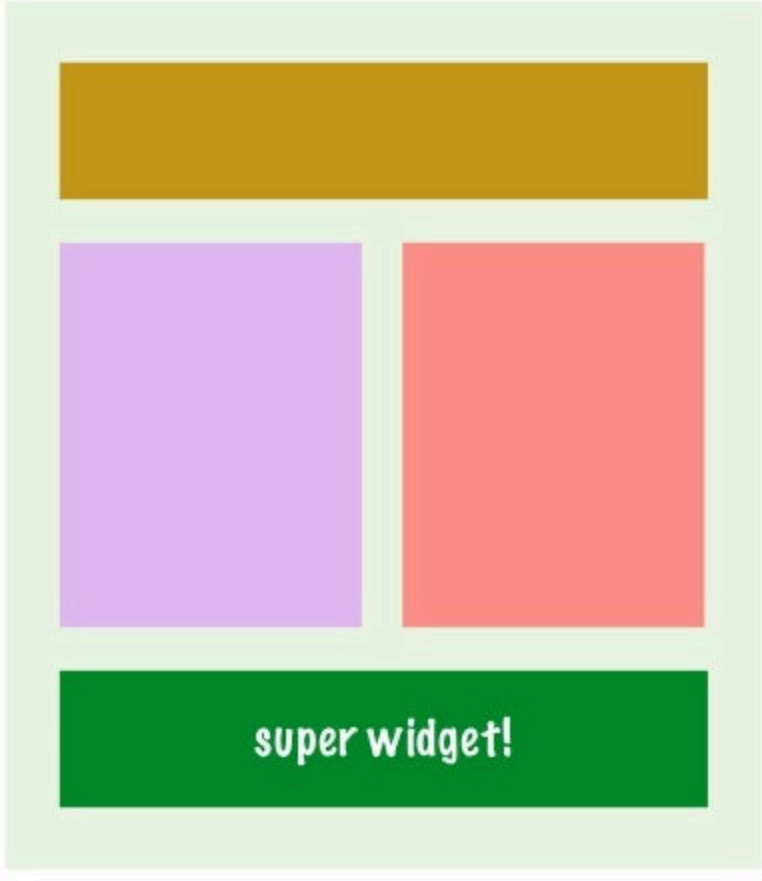
```
…
 super_widget = off
…
```



```
…
  super_widget = on
…
```

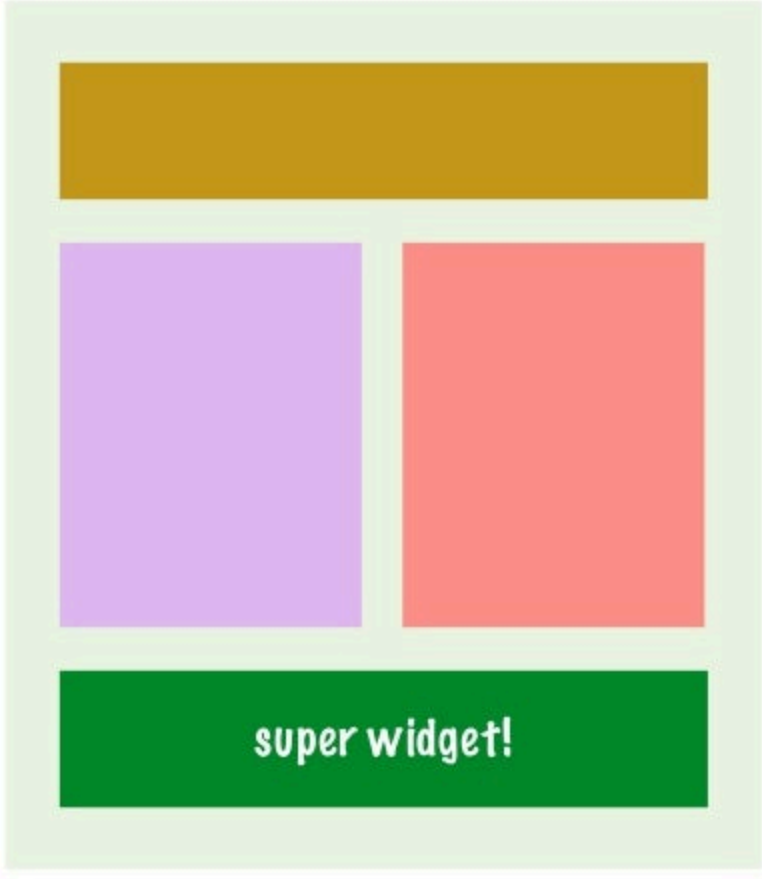



```
…
  super_widget = on
…
```




```
…
 super_widget = on
…
```

```
$. run -Dsuper_widget=on
```

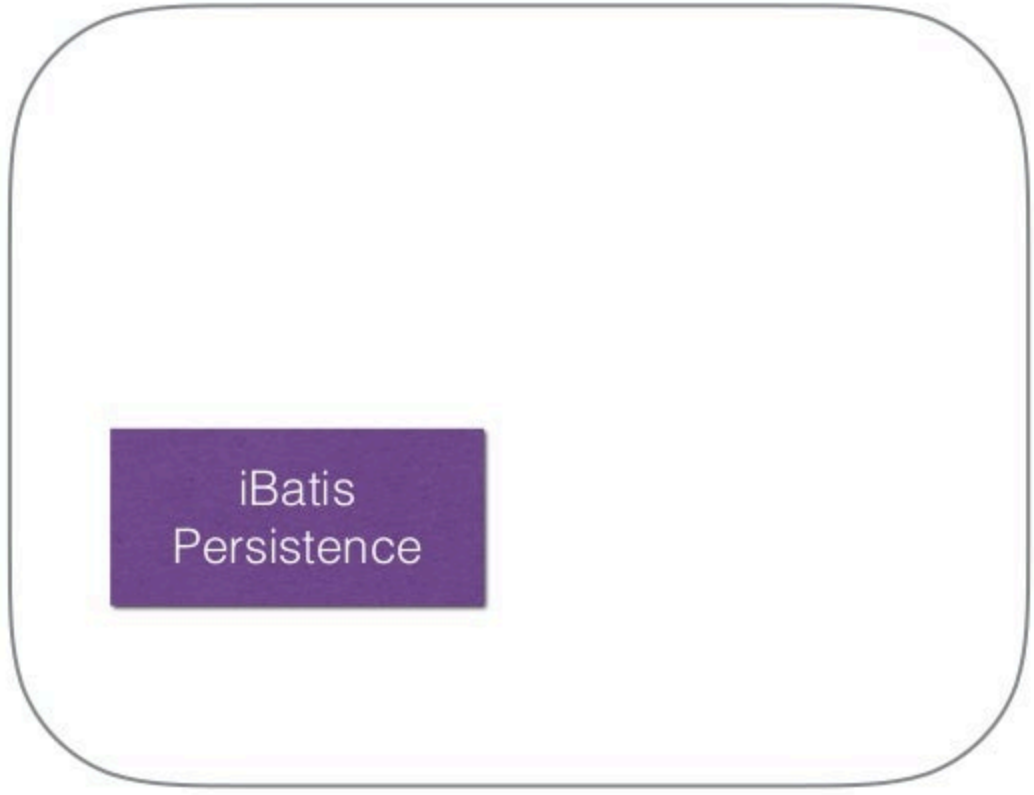


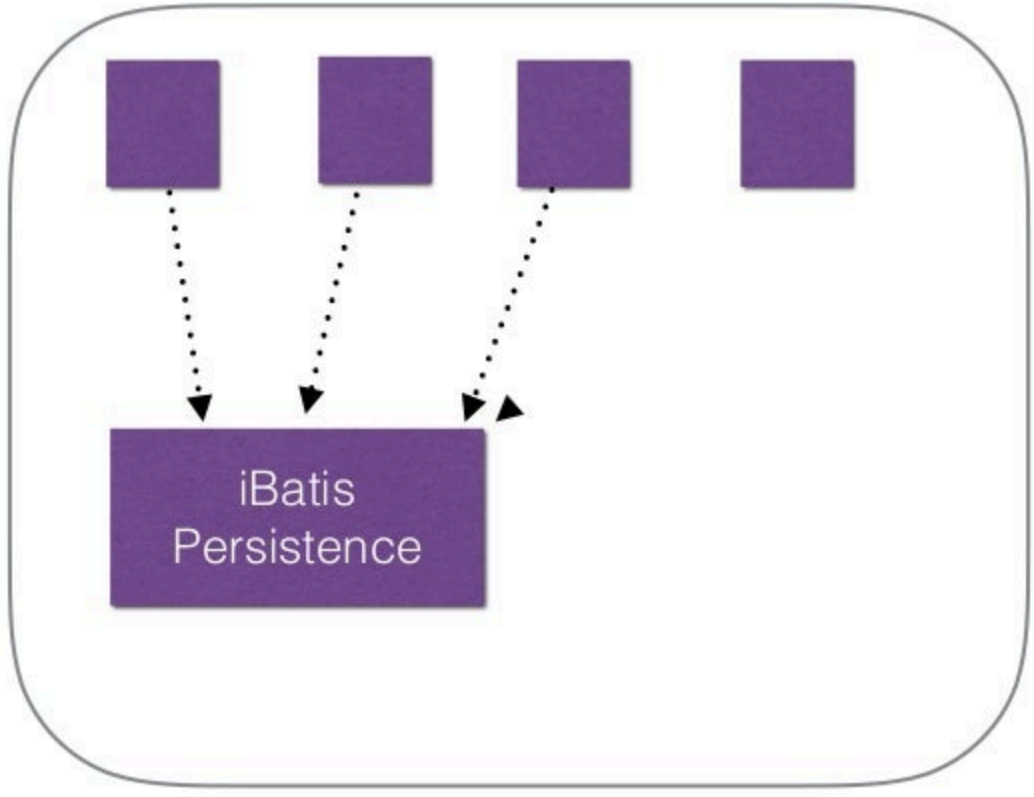

```
…
  super_widget = on
…
```

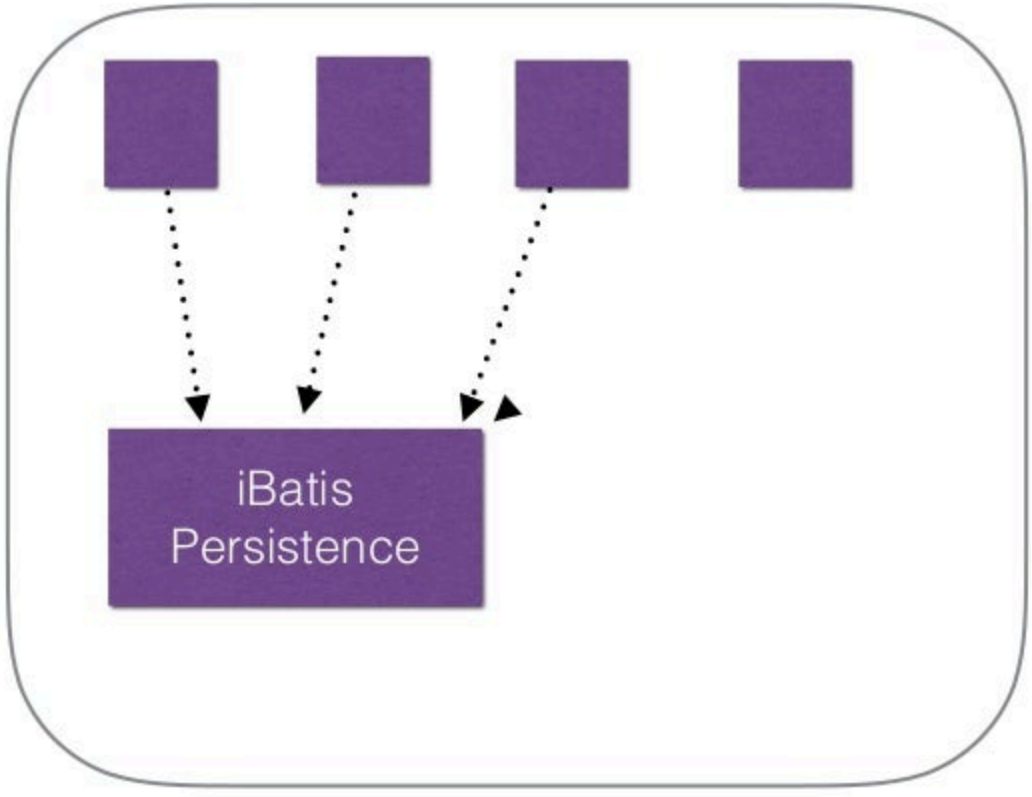
```
$. run -Dsuper_widget=on
```

But what about changes to existing functionality?

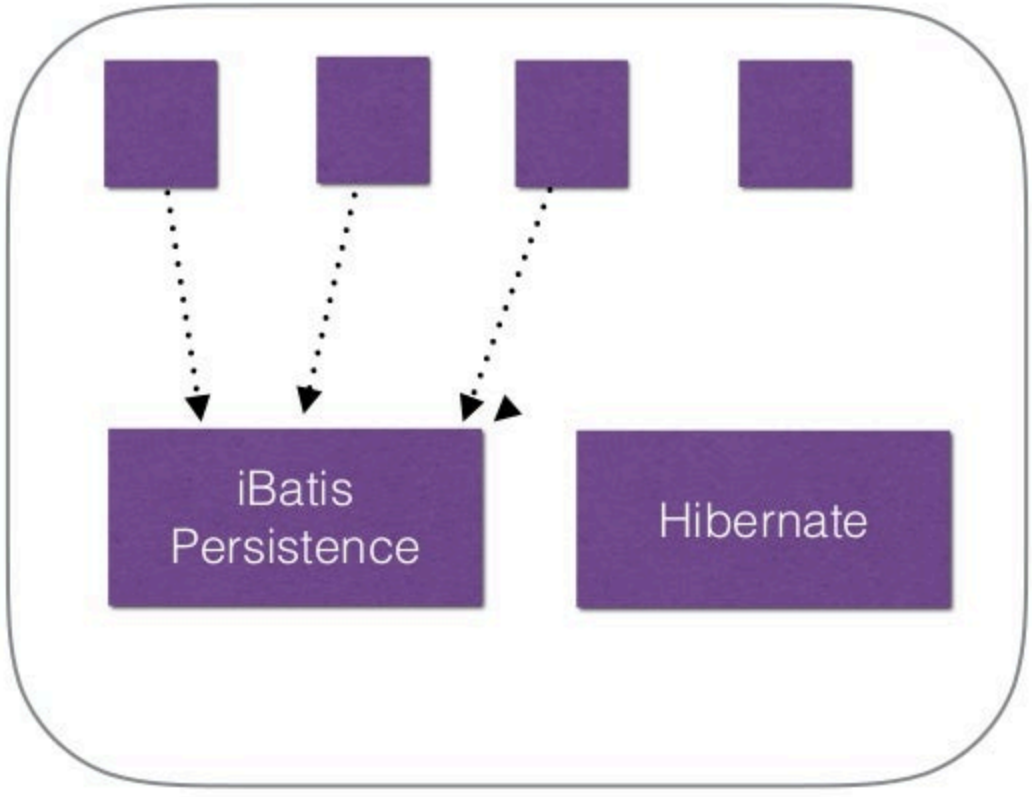

go
TM


iBatis
Persistence

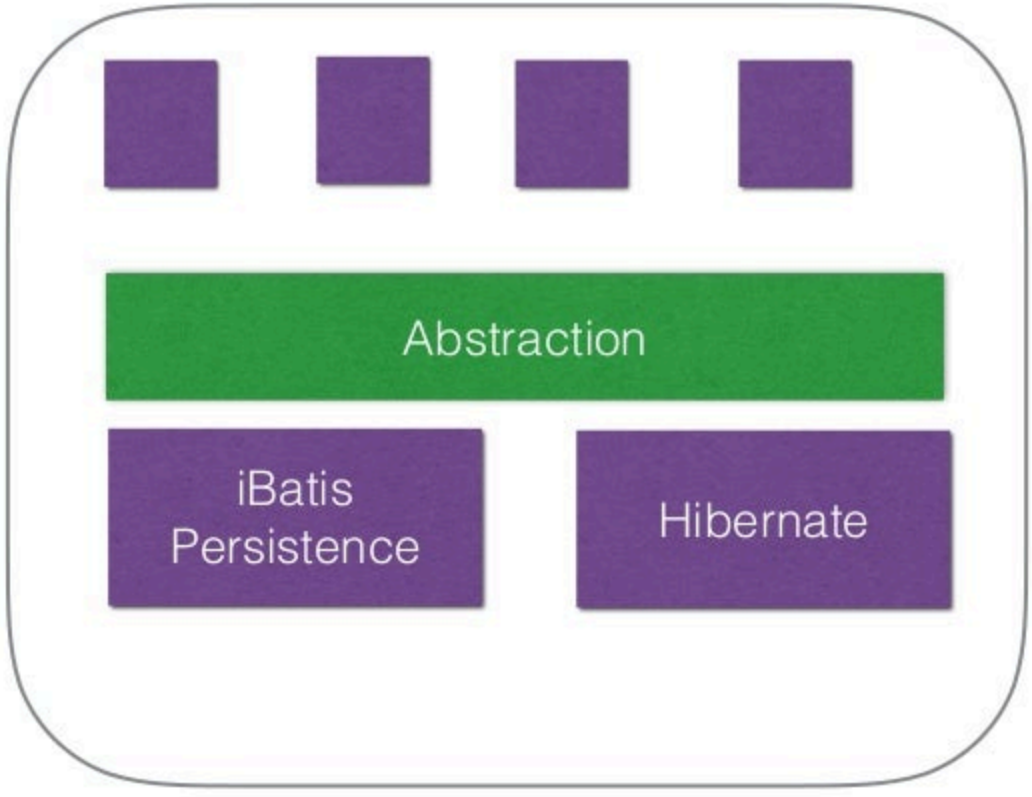

iBatis
Persistence



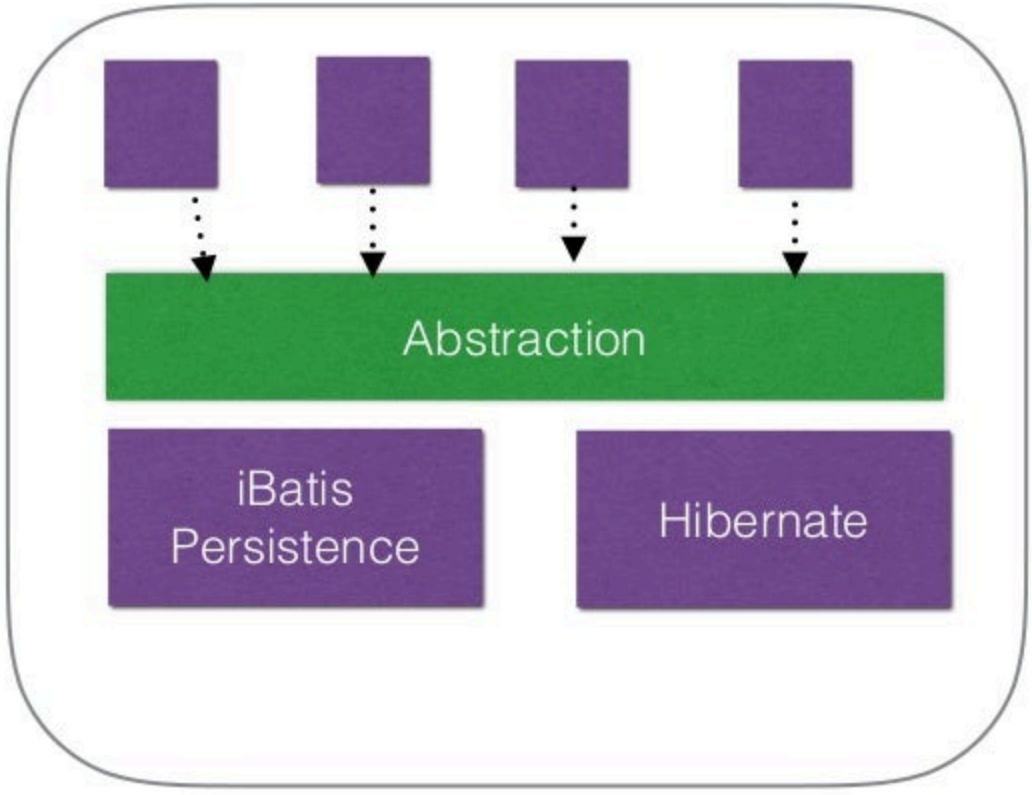
shipping every two weeks...



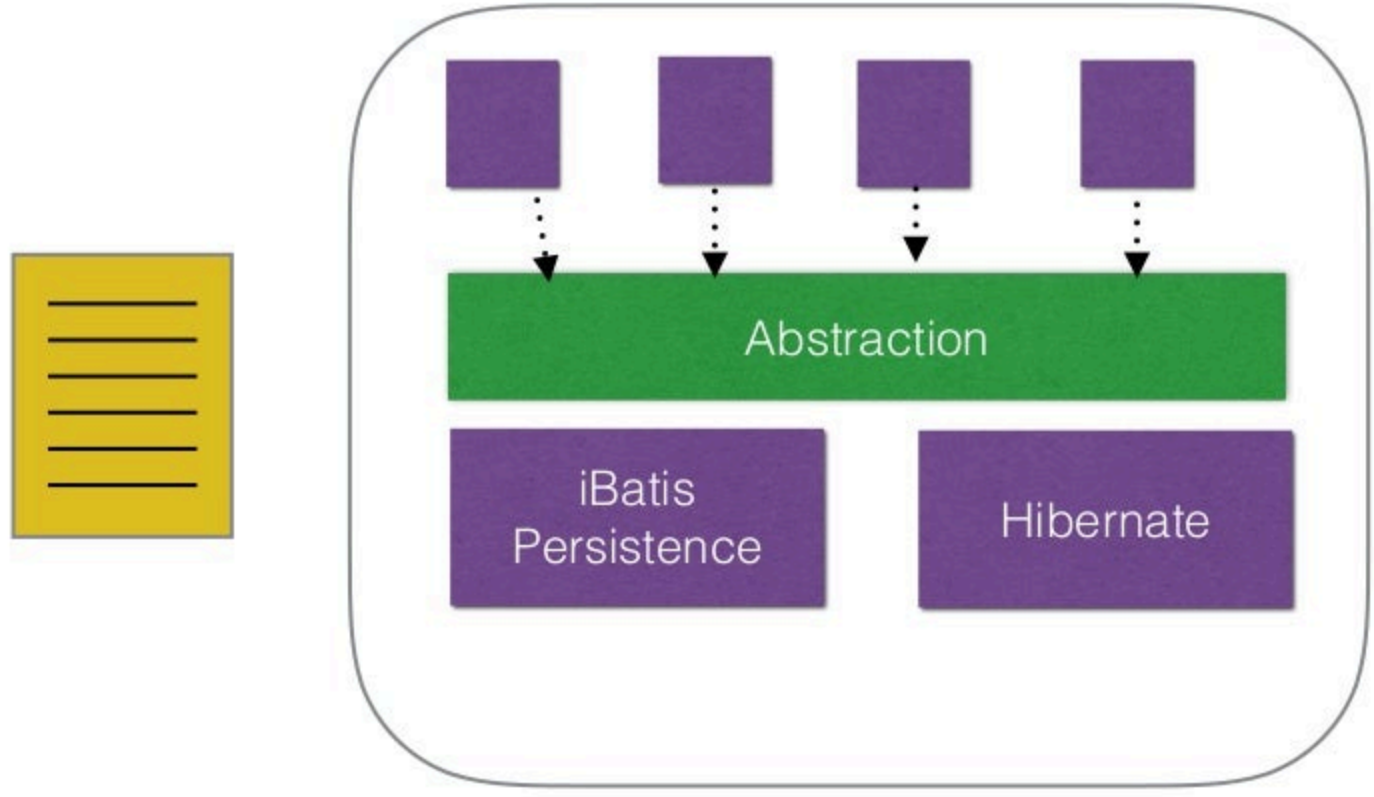
shipping every two weeks...



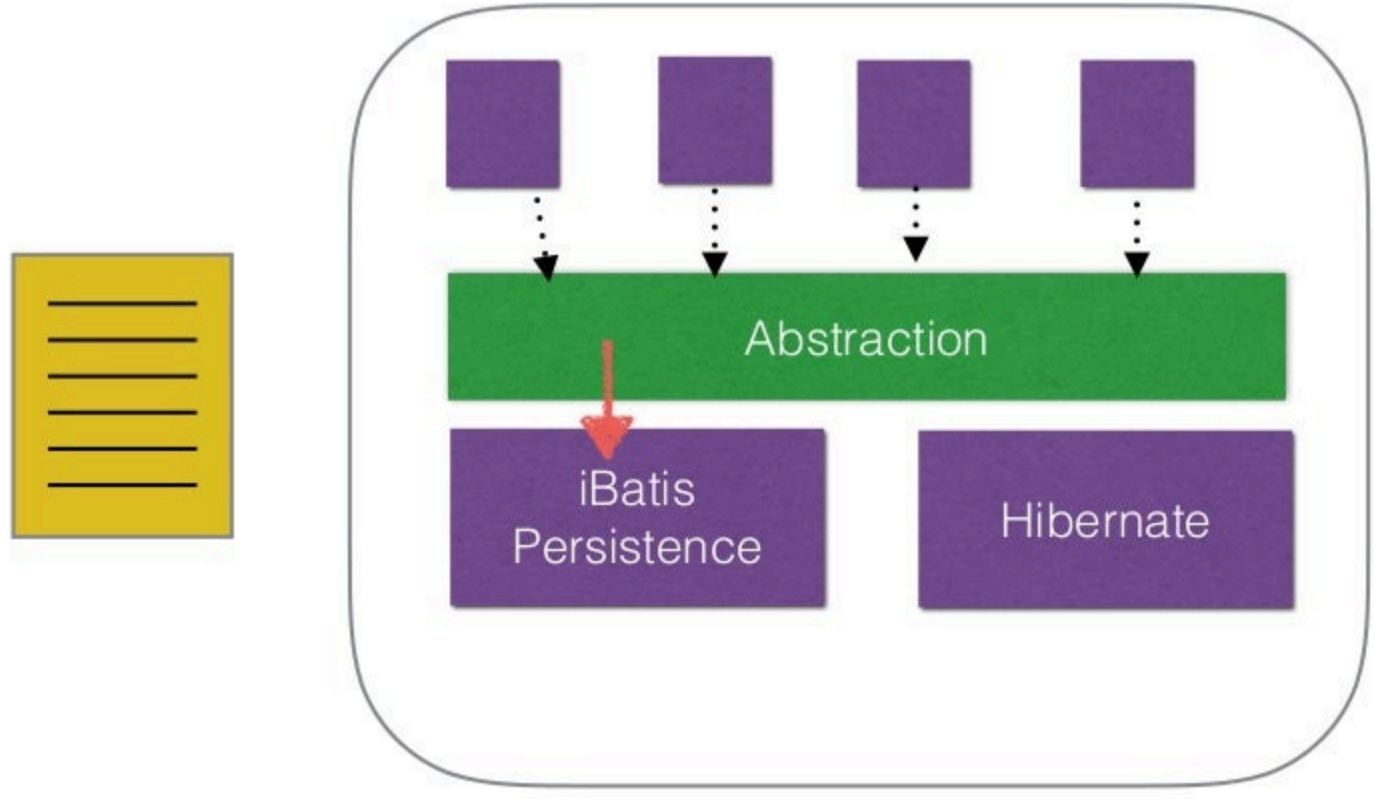
shipping every two weeks...



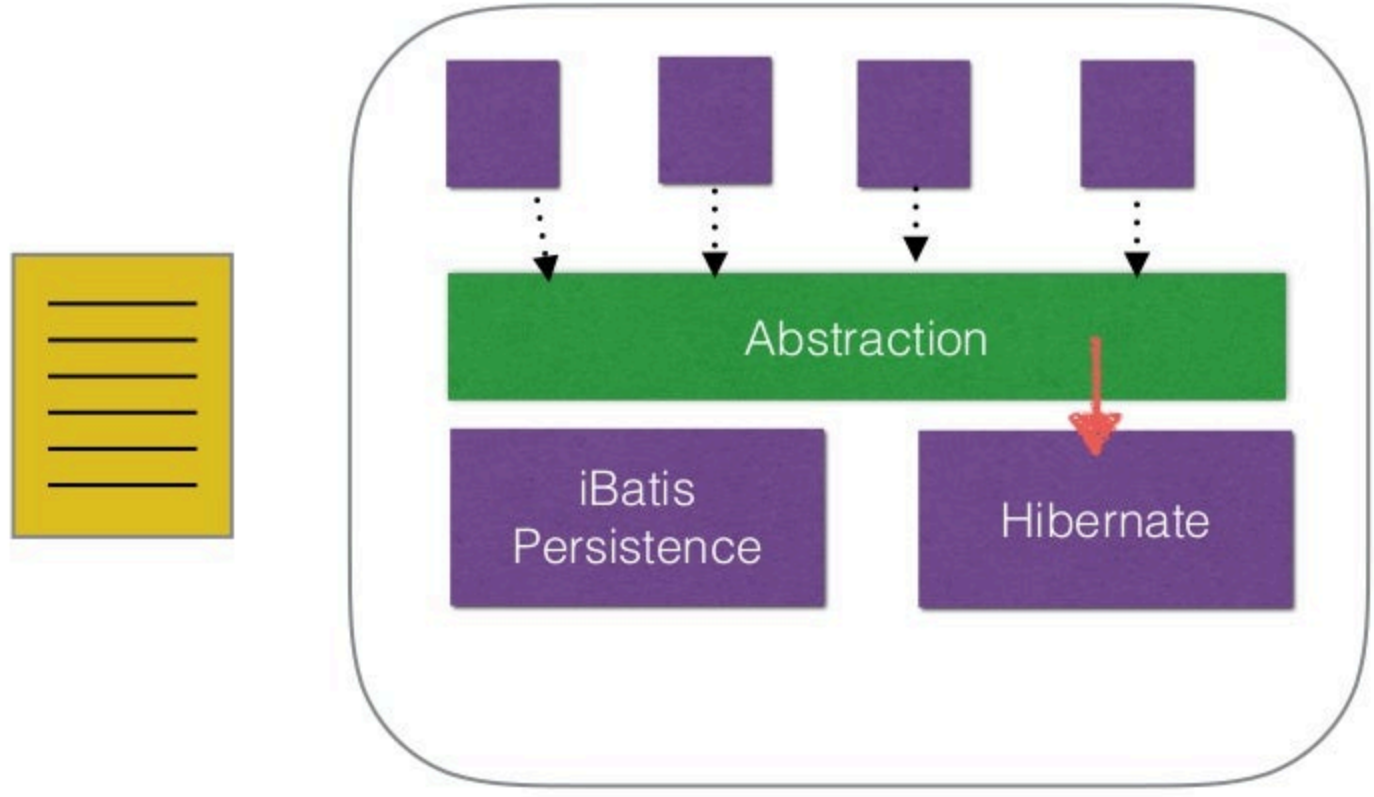
shipping every two weeks...



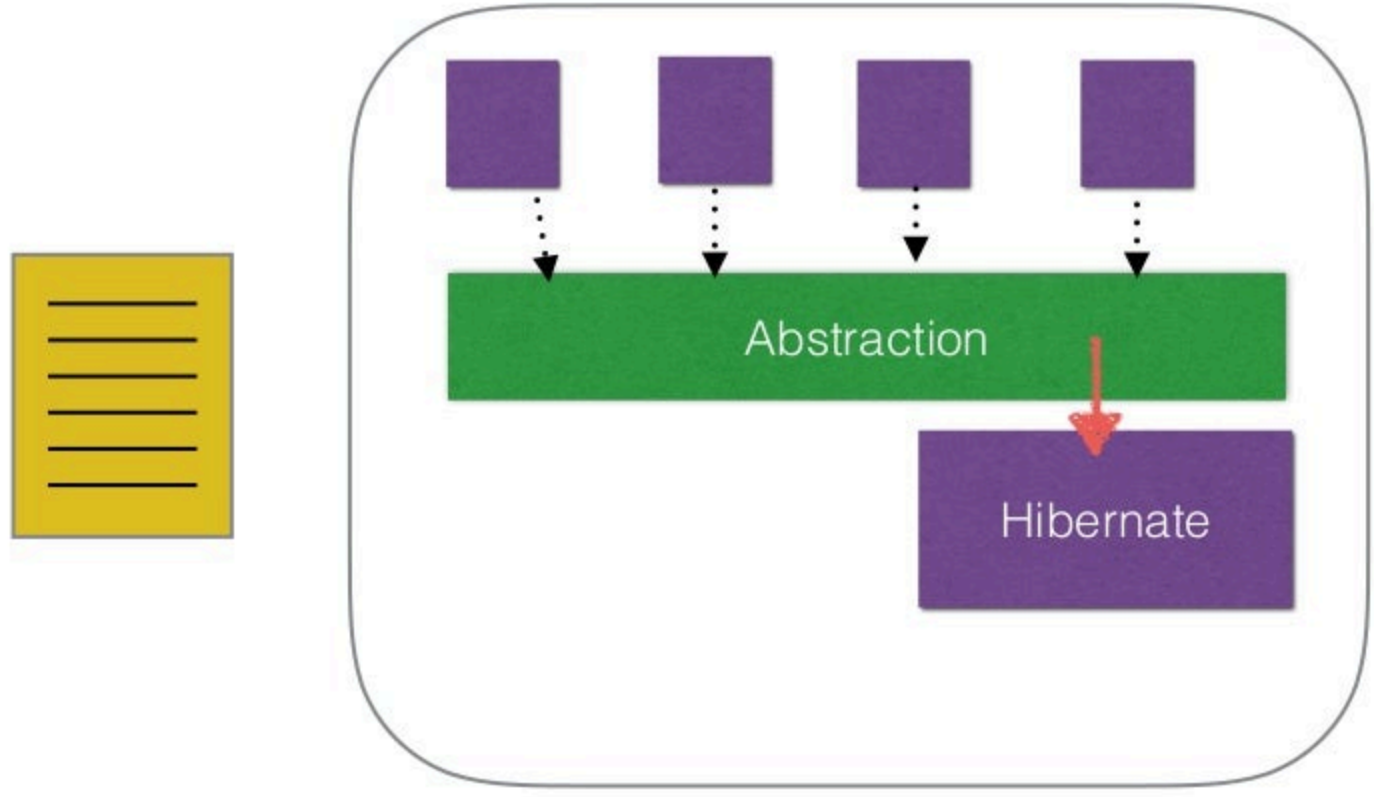
shipping every two weeks...



shipping every two weeks...



shipping every two weeks...



shipping every two weeks...

Branch by abstraction

Important Rules:

Important Rules:

Use a flag in as few places as possible

Important Rules:

Use a flag in as few places as possible

Remove them once you're done

Side Benefits:

Side Benefits:

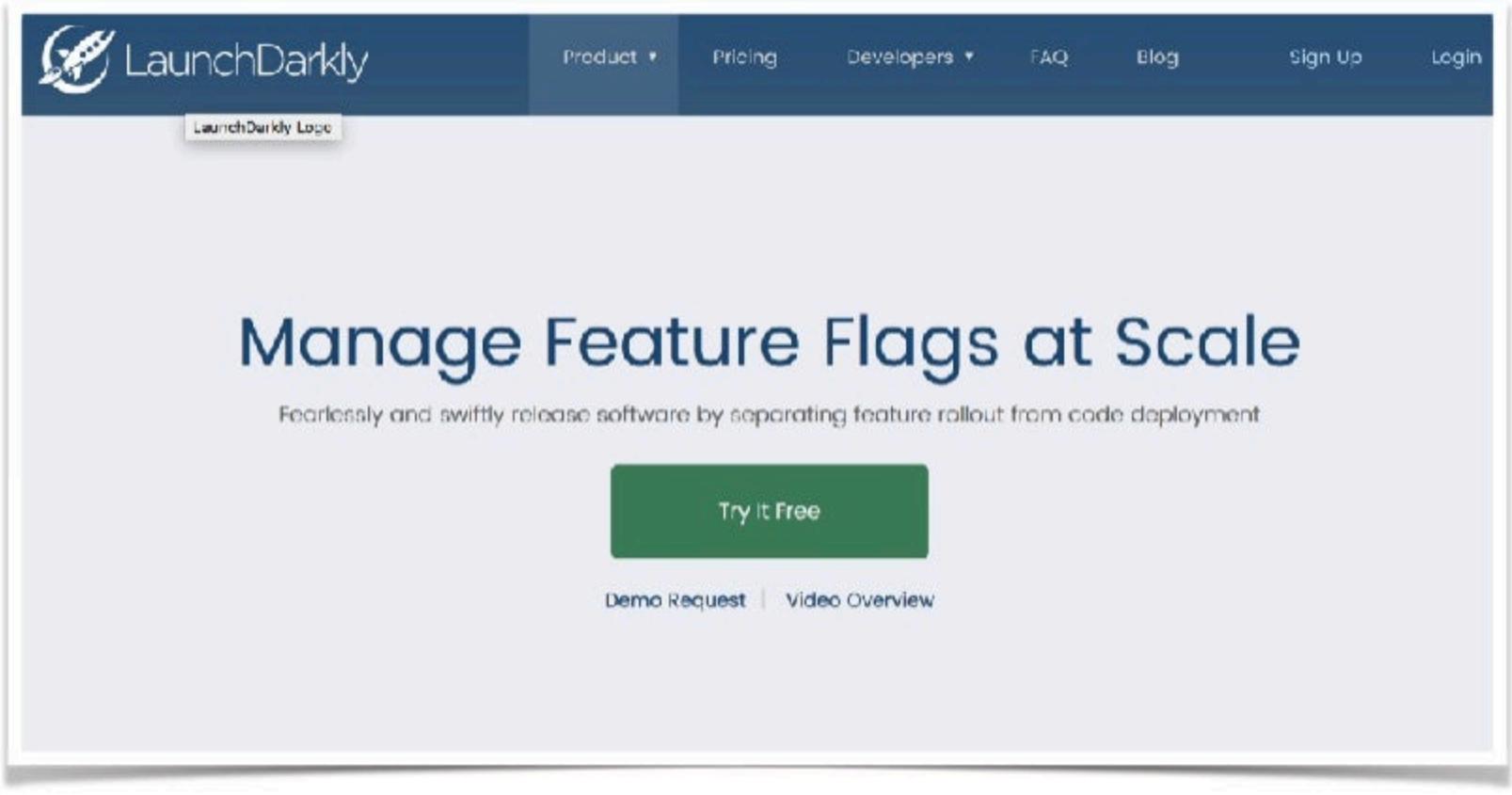
Can be used for A/B testing

Side Benefits:

Can be used for A/B testing

And with some work, canary releasing

split.io, launch darkly


LaunchDarkly
Product
Pricing
Developers
FAQ
Blog
Sign Up
Login
LaunchDarkly Logo
Manage Feature Flags at Scale
Fearlessly and swiftly release software by separating feature rollout from code deployment
Try It Free
Demo Request
Video Overview

LaunchDarkly

Product ▾ Pricing Developers ▾ FAQ Blog Sign Up Login

LaunchDarkly Logo

Manage Feature Flags at Scale

Fearless

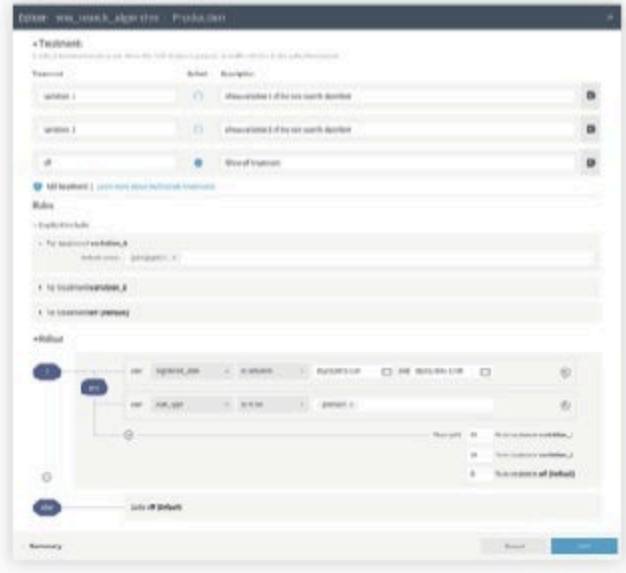
split

Product Resources Company Log In Sign Up

Build better software, faster.

Split is the best way to roll out new features to the right audience—and protect your customers from failure. Try it free for 30 days.

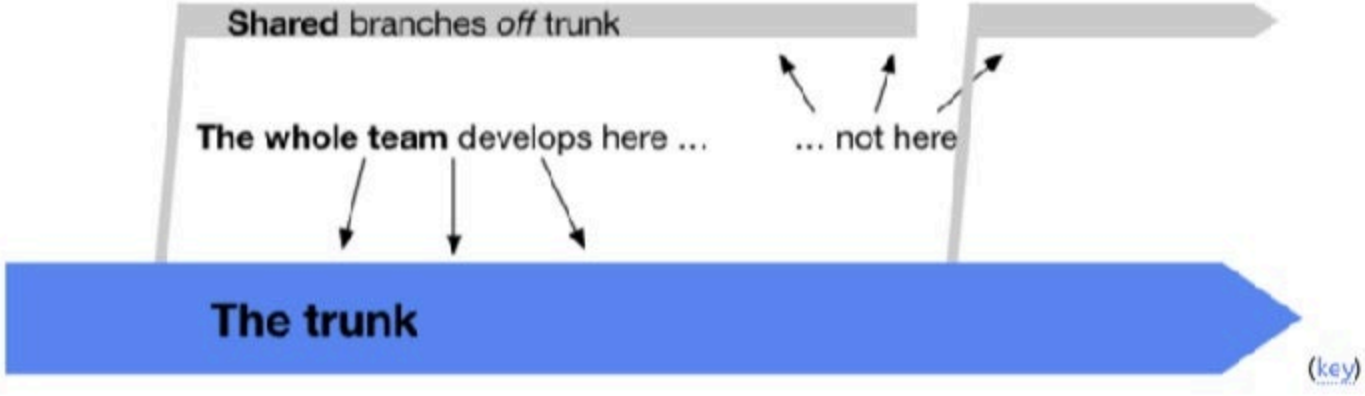
Start your free trial



Introduction

One line summary

A source-control branching model, where developers collaborate on code in a single branch called 'trunk' *, resist any pressure to create other long-lived development branches by employing documented techniques, avoid merge hell, do not break the build, and live happily ever after.

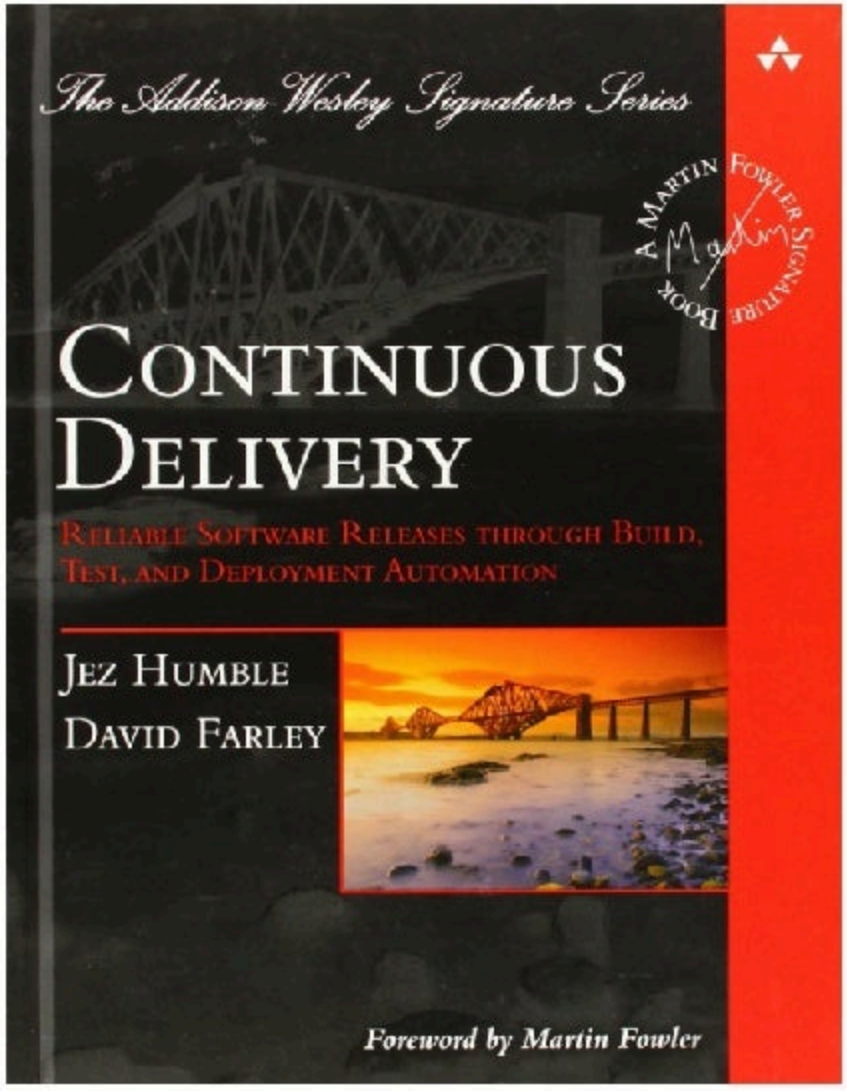


* 'master', in Git nomenclature

Trunk-Based Development is a key enabler of Continuous Integration and by extension Continuous Delivery. When individuals on a team are committing their changes to the trunk multiple times a day it becomes easy to satisfy the core requirement of Continuous Integration that all team members commit to trunk at least once every 24 hours. This ensures the codebase is always releasable on demand and helps to make Continuous Delivery a reality.

https://trunkbaseddevelopment.com

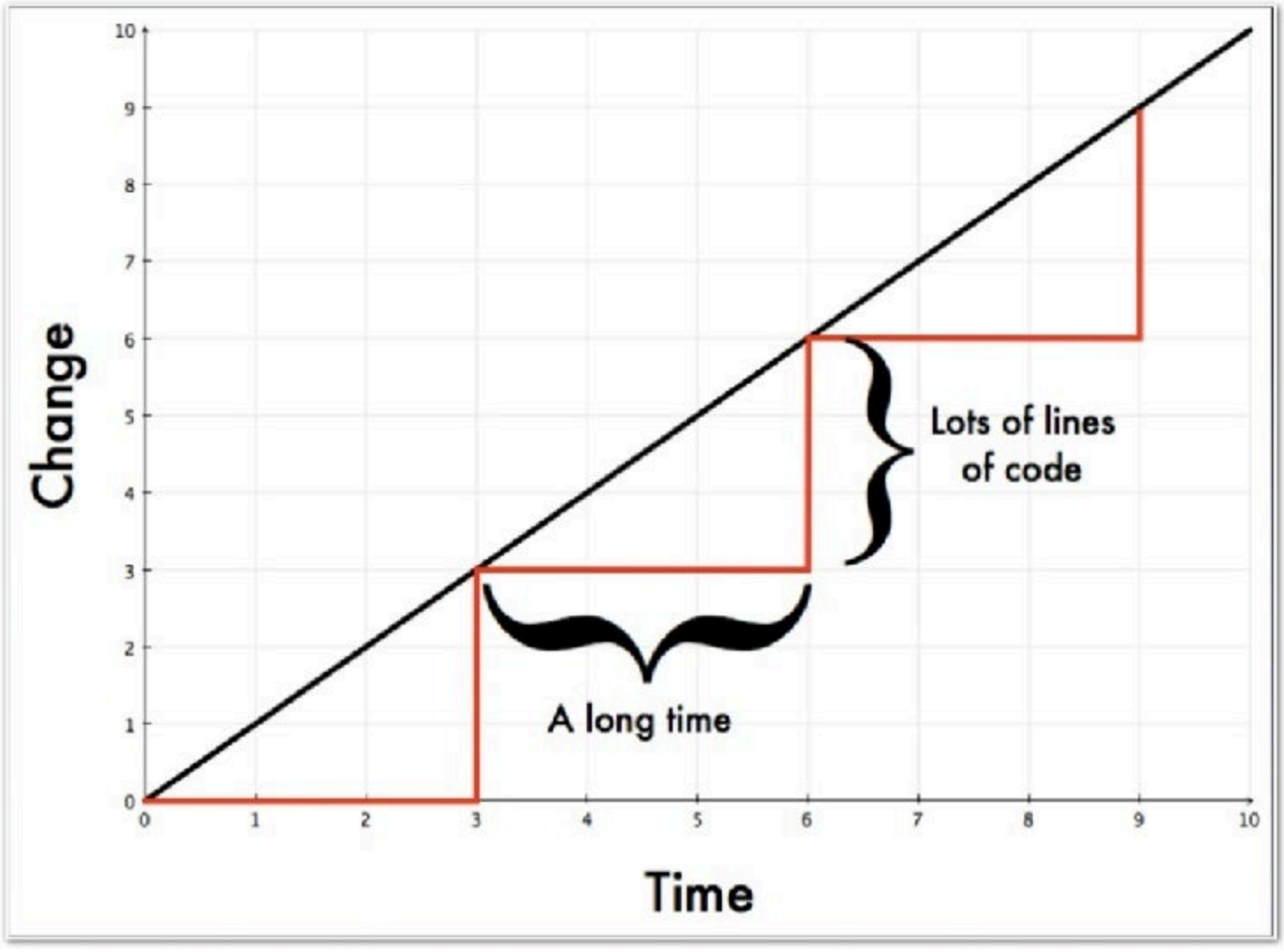
Continuous Delivery Book...



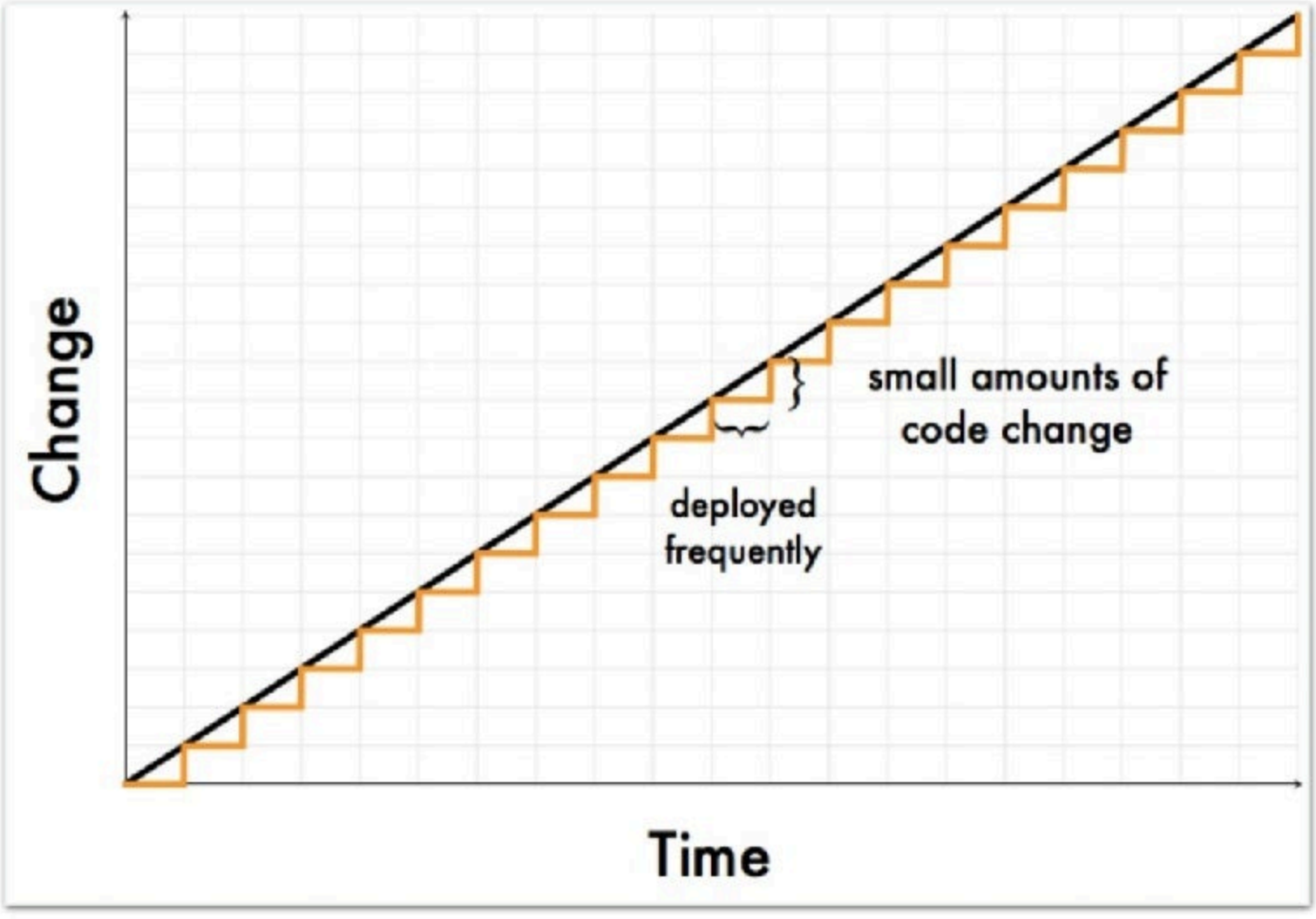
Published in 2011

Treat every check-in as a release candidate

https://www.slideshare.net/jallspaw/ops-metametrics-the-currency-you-pay-for-change-4608108



Ops Meta-Metrics - John Allspaw



Ops Meta-Metrics - John Allspaw

Keep batch size small

Keep batch size small

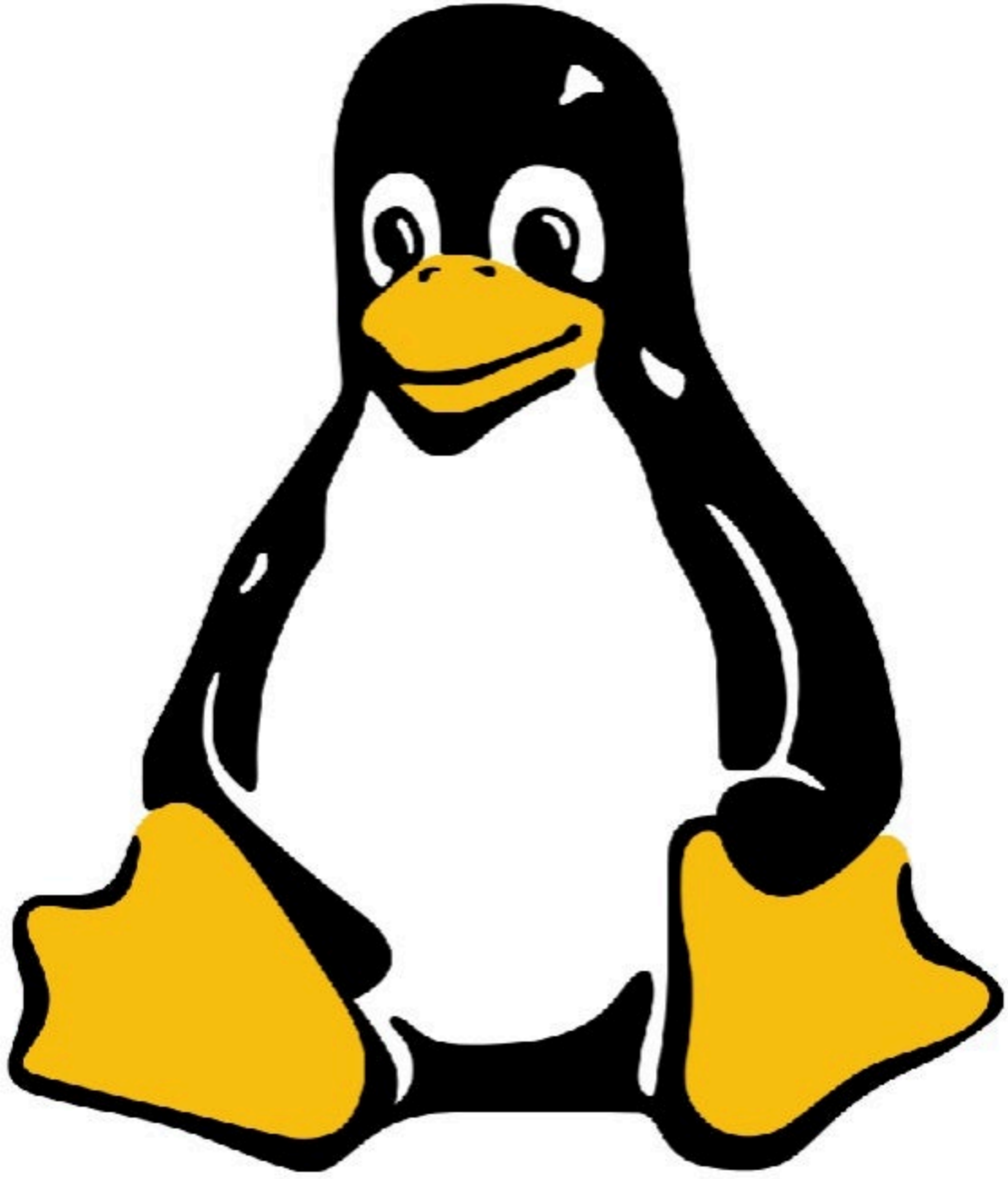
Integrate often

Keep batch size small

Integrate often

Ship often

And there there was Git



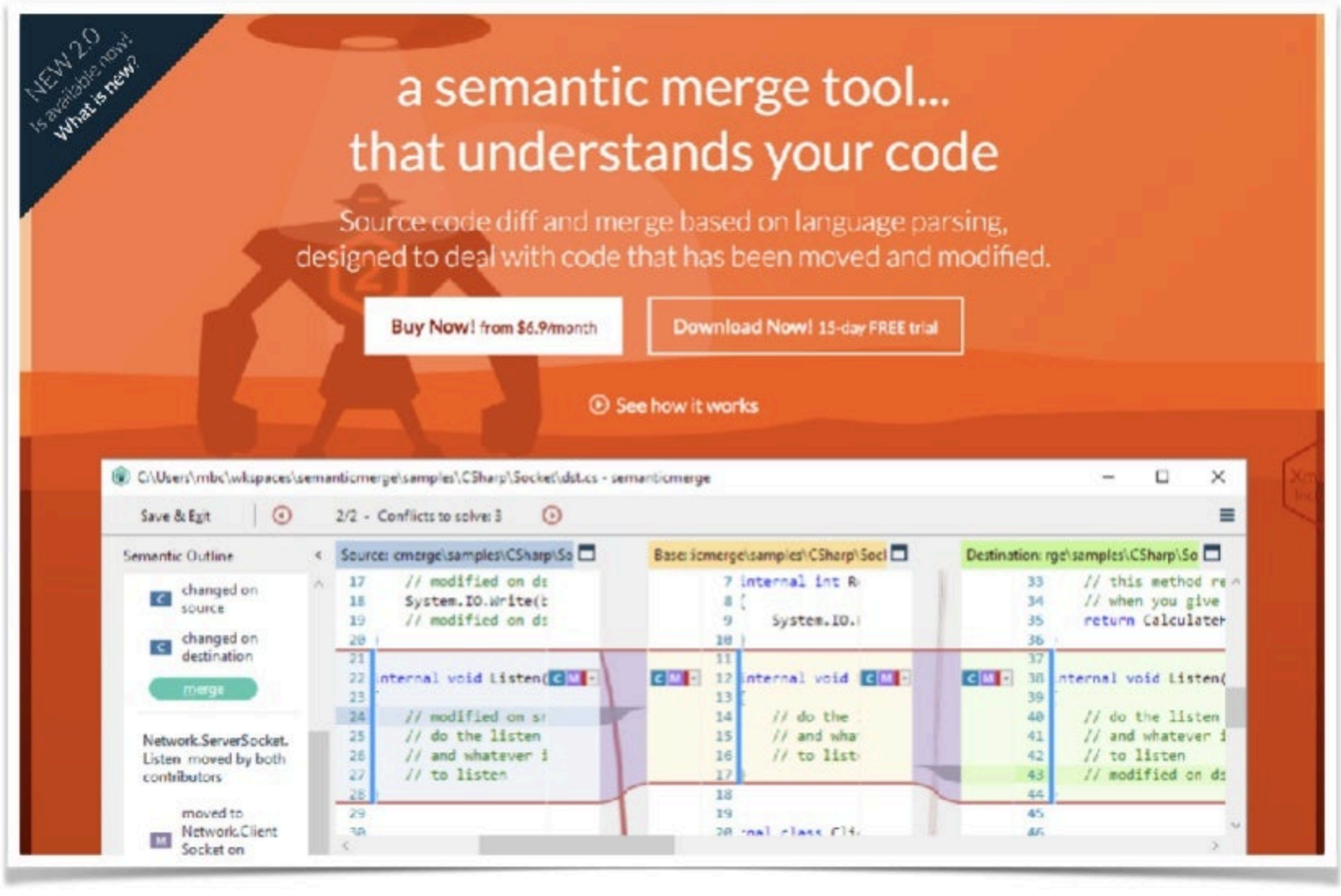
Goal

Goal

Merge a patch in less than 3 seconds

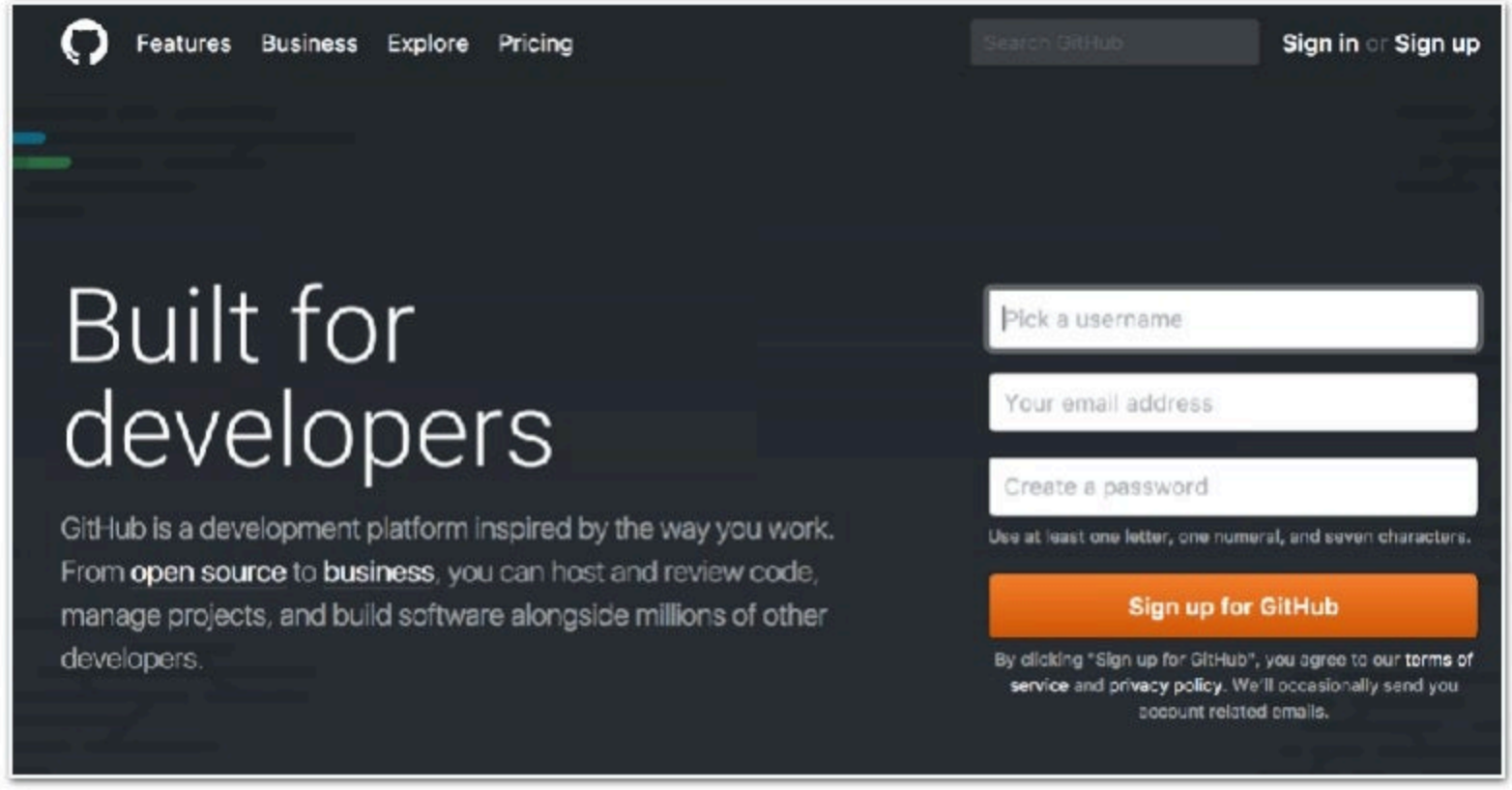
Branches much more lightweight

Merging of text is better but...



https://www.semanticmerge.com

And then there was GitHub...

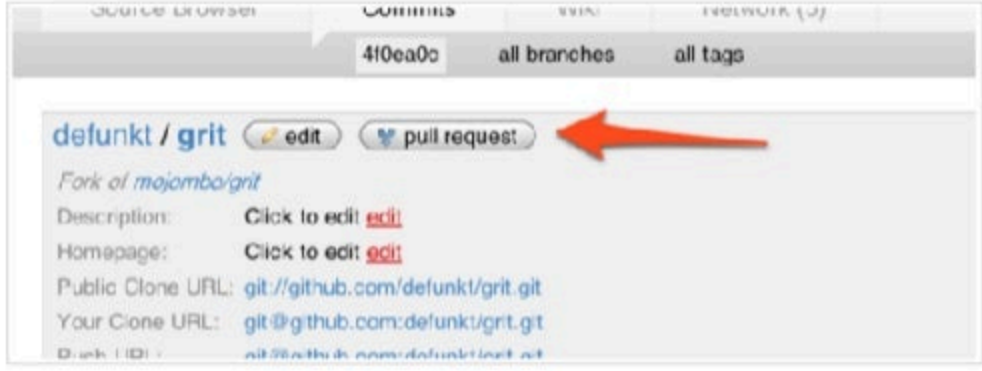

Features
Business
Explore
Pricing
Search GitHub
Sign in or Sign up
Built for developers
GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside millions of other developers.
Pick a username
Your email address
Create a password
Use at least one letter, one numeral, and seven characters.
Sign up for GitHub
By clicking "Sign up for GitHub", you agree to our terms of service and privacy policy. We'll occasionally send you account related emails.

2008

Oh yeah, there's pull requests now

February 23, 2008 defunkt New Features

Last night I pushed out a feature Tom and I have been talking about since day one: pull requests. That's the short walkthrough.

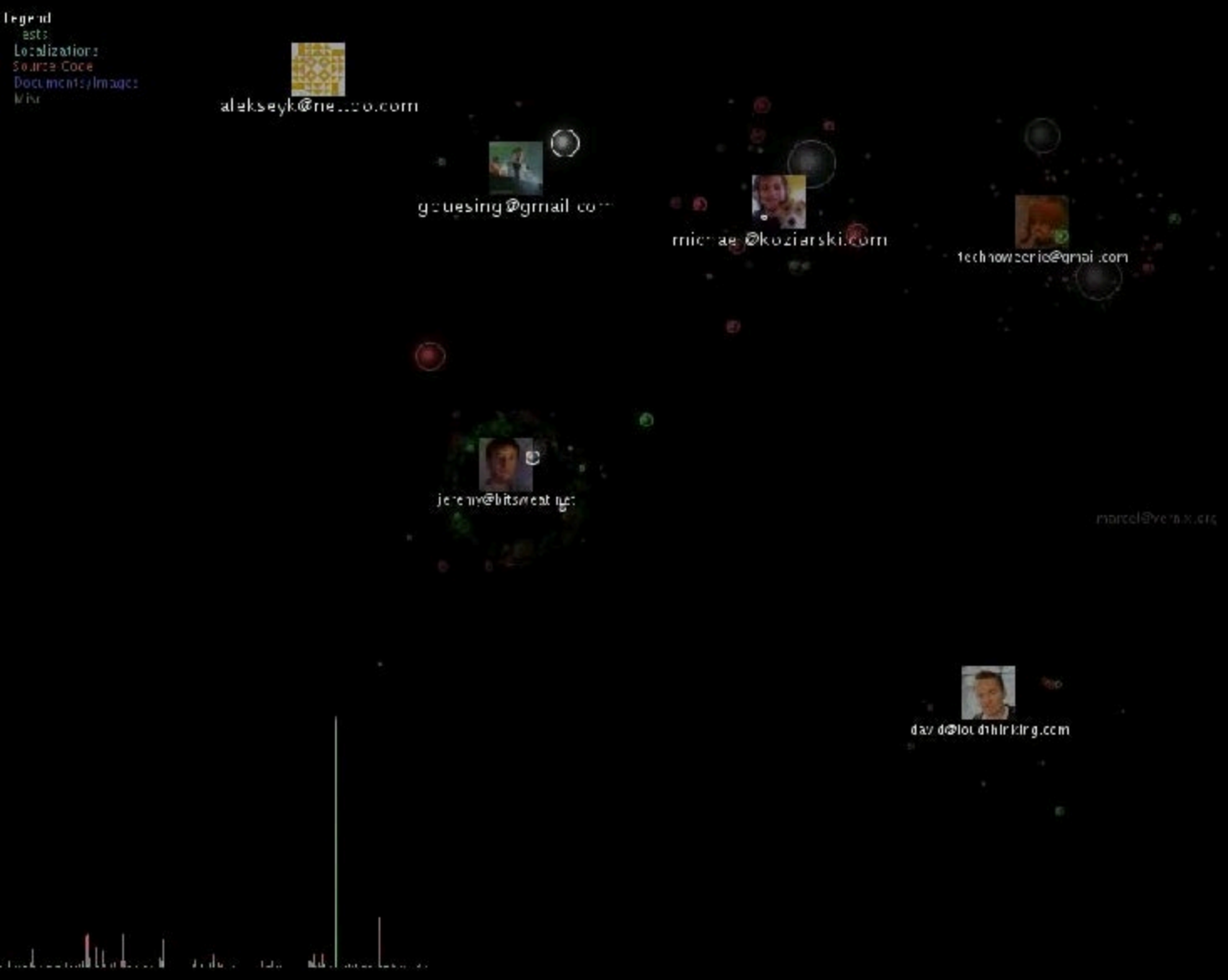


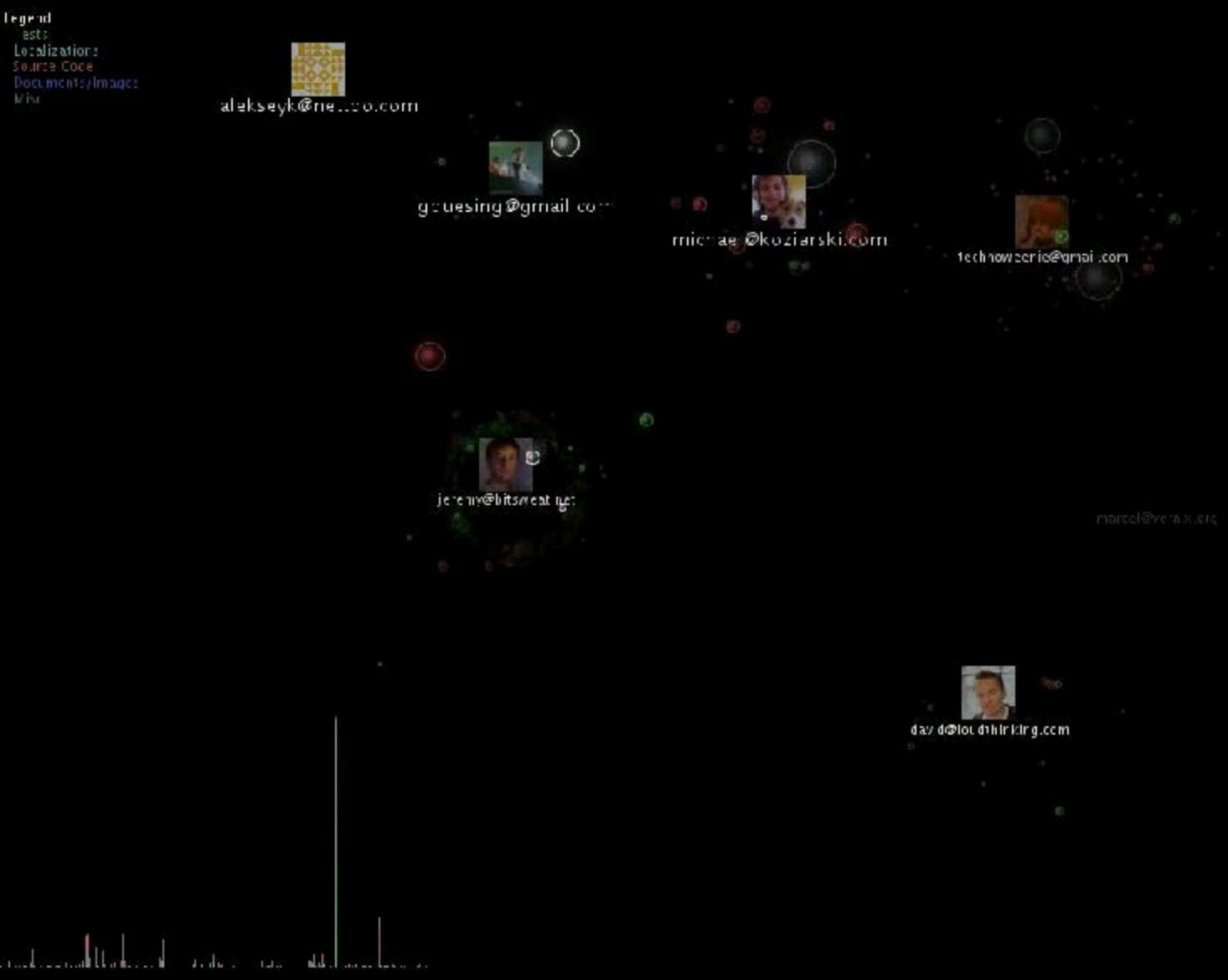
You can use it to tell people who forked from you they need to pull, or they can use it to ask you to pull. It's also great for letting someone else know you have a cool feature pushed to some non-master branch.

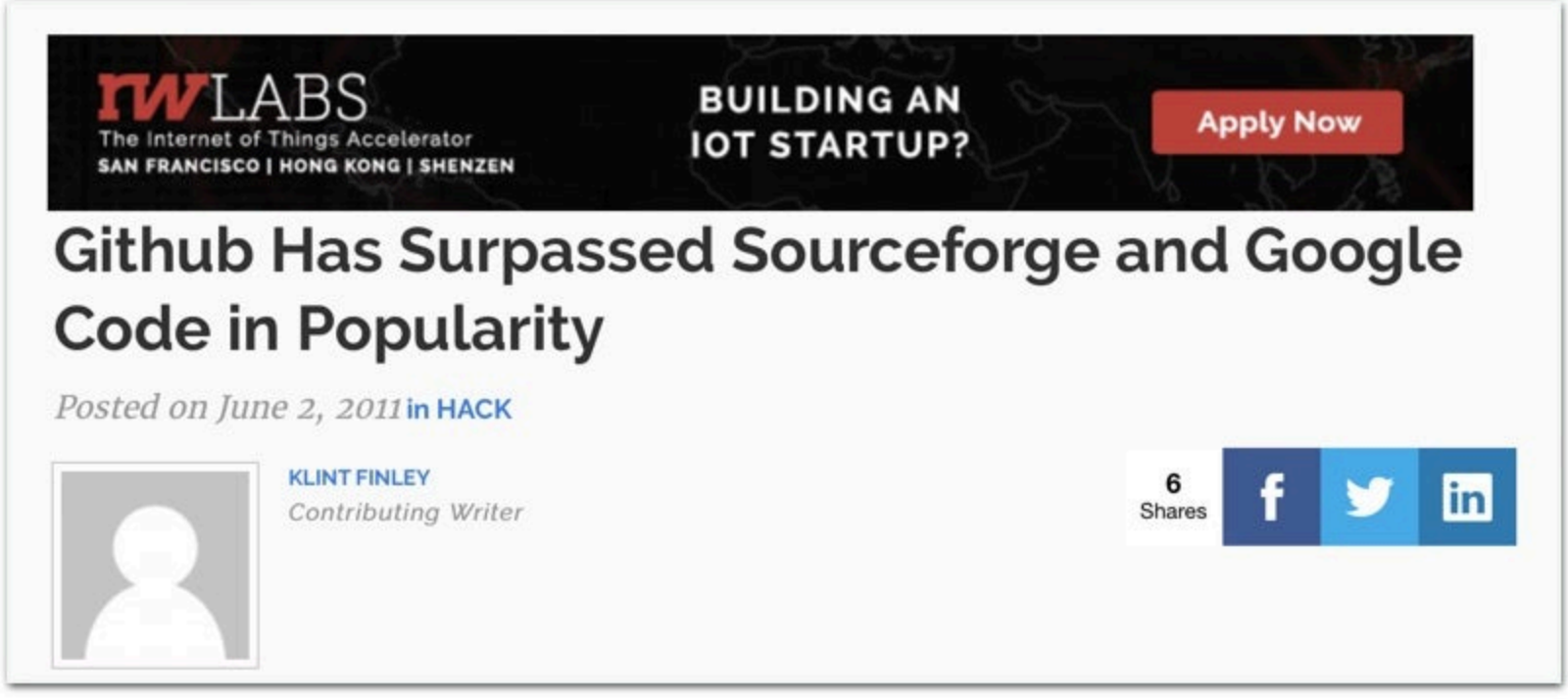
Have feedback on this post? Let @github know on Twitter.

Need help or found a bug? Contact us.

https://github.com/blog/3-oh-yeah-there-s-pull-requests-now


Legend
tests
Localizations
Source Code
Documents/Images
Misc
technowierdo@gmail.com
marcel@vernix.org
Jan 17, 2008


Legend
tests
Localizations
Source Code
Documents/Images
Misc
technowierdo@gmail.com
marcel@vernix.org
Jan 17, 2008



http://readwrite.com/2011/06/02/github-has-passed-sourceforge/

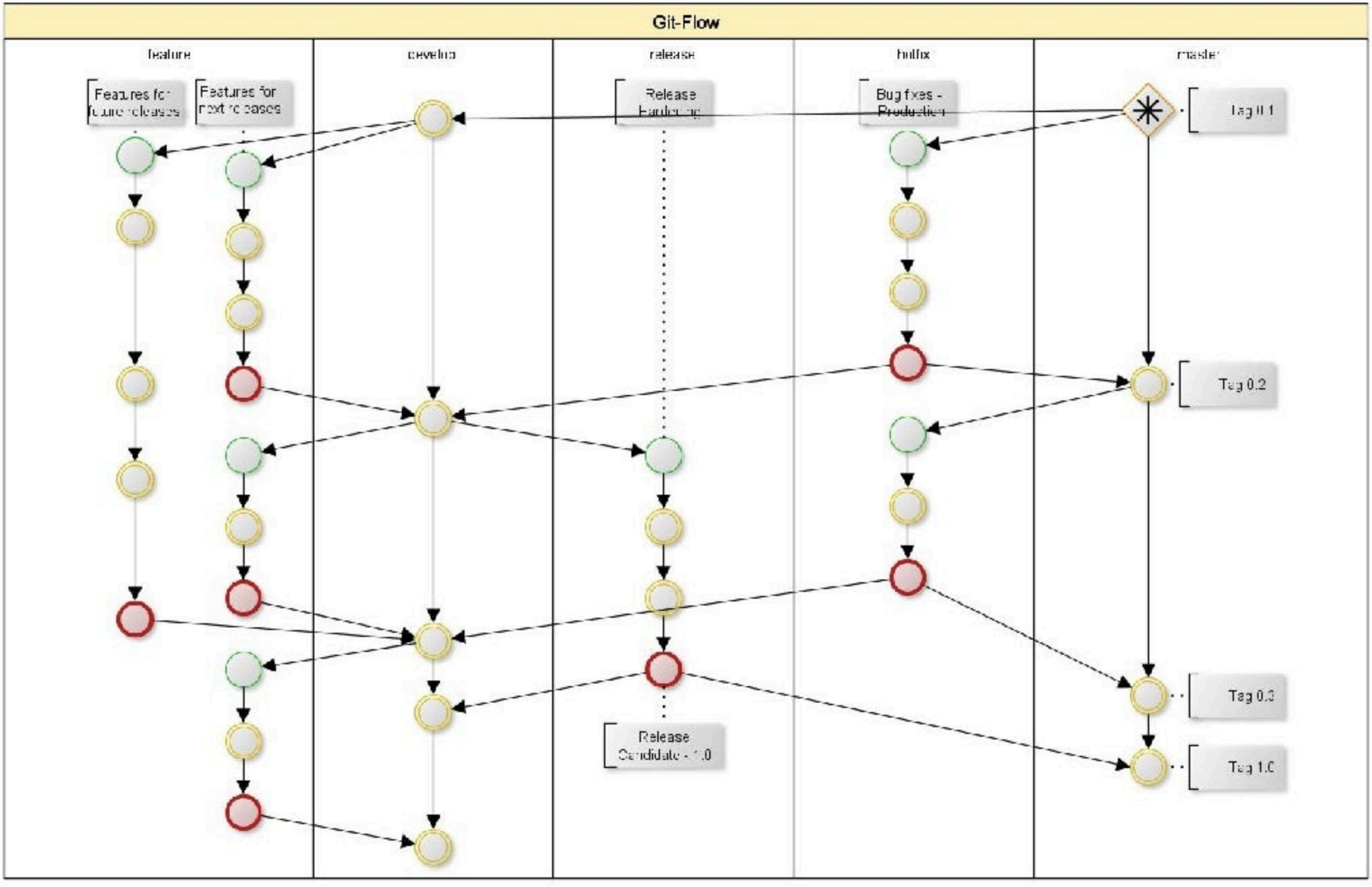
But PRs use branching!

Open Source

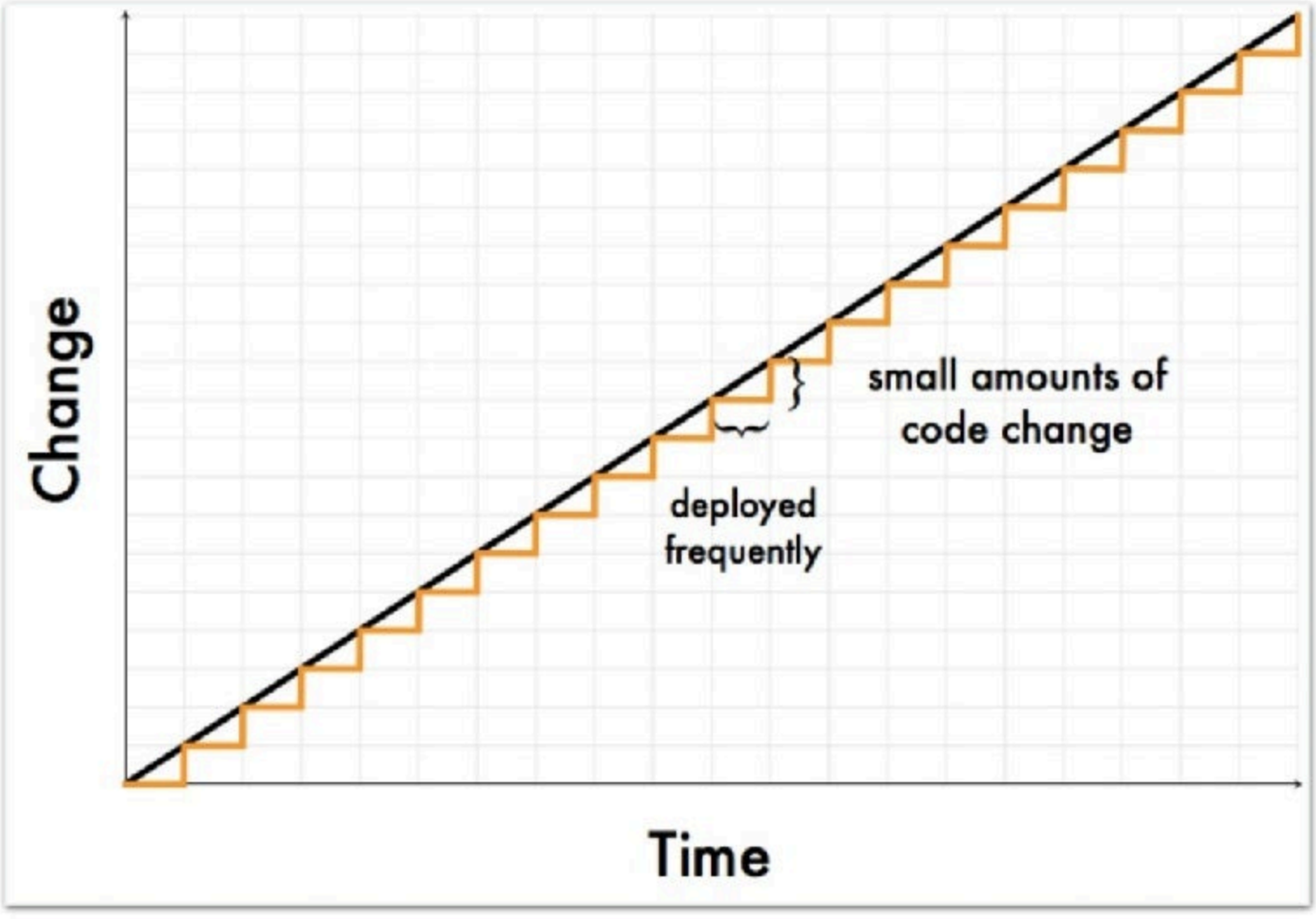
Open Source

"Untrusted" Committers

And then there was gitflow…

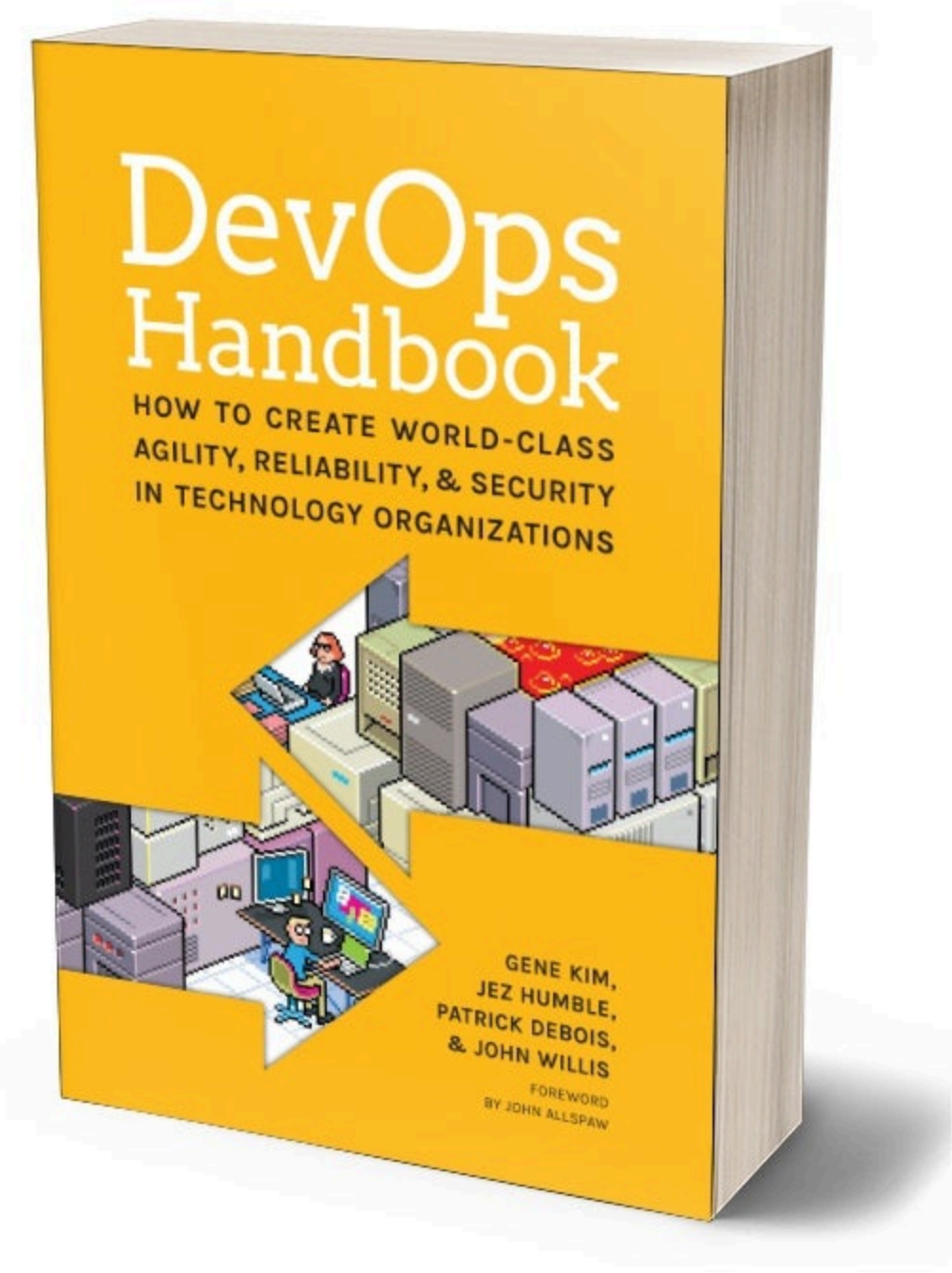

Git-Flow
feature
develop
release
hotfix
master
Features for future releases
Features for next release
Release Candidate - 1.0
Bug fixes - Production
Tag 0.2
Tag 0.3
Tag 1.0

https://www.flickr.com/photos/hackny/8675049276/



Ops Meta-Metrics - John Allspaw

But if I merge in frequently, why do I need GitFlow?


DevOps
Handbook
HOW TO CREATE WORLD-CLASS
AGILITY, RELIABILITY, & SECURITY
IN TECHNOLOGY ORGANIZATIONS
GENE KIM,
JEZ HUMBLE,
PATRICK DEBOIS,
& JOHN WILLIS
FOREWORD
BY JOHN ALLSPAW

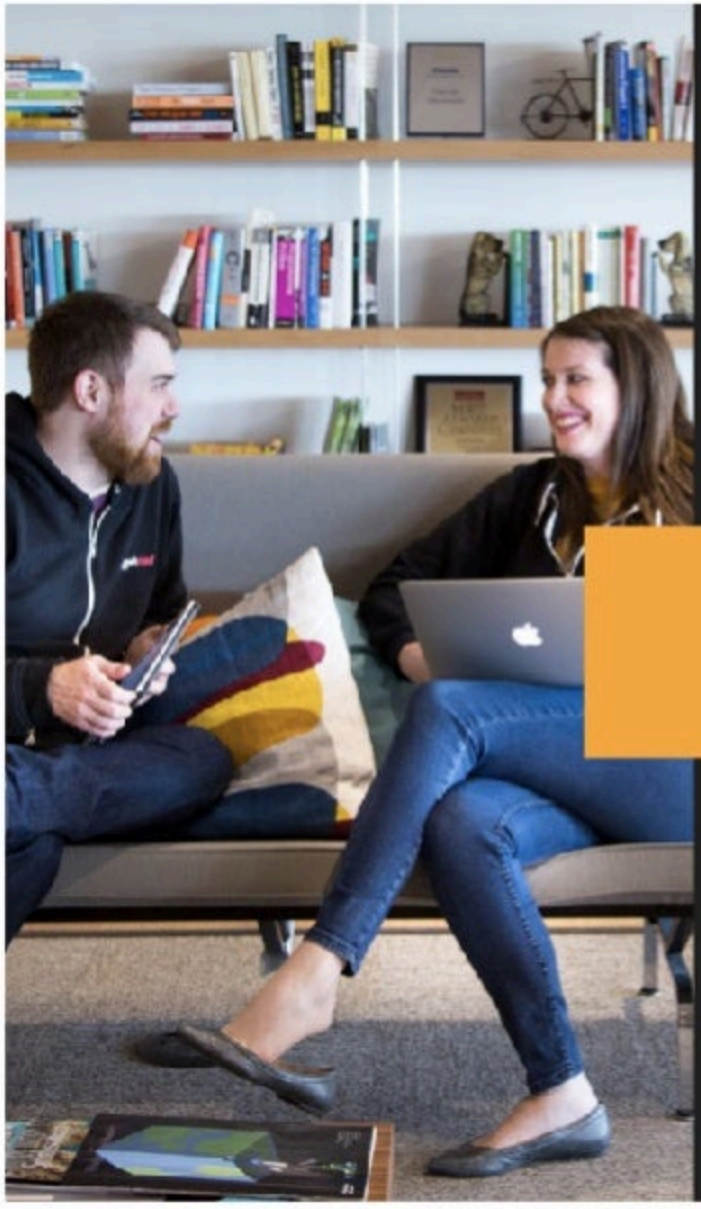
"Trunk-based development is likely the most controversial practice discussed in this book

"Trunk-based development is likely the most controversial practice discussed in this book

[…]

However, the data from Puppet Labs' *2015 State of DevOps Report* is clear: trunk-based development predicts higher throughput and better stability, and even higher job satisfaction and lower rates of burnout."

- Gene Kim, Jez Humble, Patrick Debois & John Willis



2016

State of DevOps Report

Presented by:

puppet + DORA
DEVOPS RESEARCH & ASSESSMENT

Sponsored by:

Hewlett Packard Enterprise · ThoughtWorks · splunk> · ca technologies
Atlassian · Automic · IT REVOLUTION

“We found that having branches or forks with very short lifetimes (less than a day) before being merged into trunk, and less than three active branches in total, are important aspects of continuous delivery, and all contribute to higher performance. So does merging code into trunk or master on a daily basis. “

- State Of Devops Report, 2016

So are branches evil?

Keep batch size small

Keep batch size small

Integrate often

Keep batch size small

Integrate often

Ship often

Thank You!

http://samnewman.io

@samnewman

Also thanks to Don Clark for the people icons
https://commons.wikimedia.org/wiki/File:Gender_neutral.svg CC BY-SA 4.0