A large silhouette of a person's head and shoulders, facing left, with their hand resting against their chin in a thoughtful pose. A thought bubble containing a cartoon-style face with a thinking expression is positioned above the silhouette.

To Build or Not to Build

The Build Lifecycle of a Commit

Bram Adams

Polytechnique Montreal
<http://mcis.polymtl.ca/>



joint work with ...

Mahdis Zolfagharinia

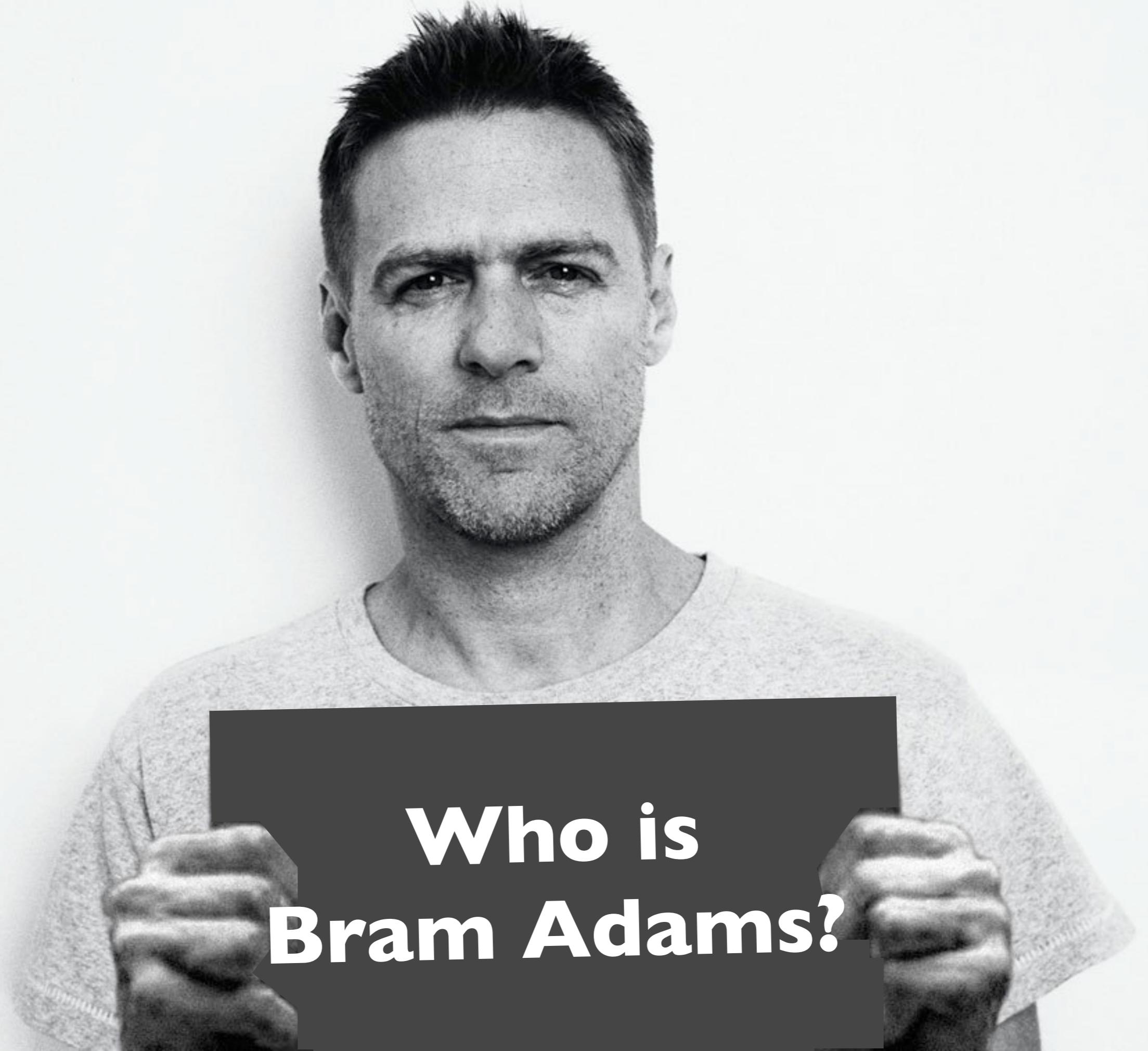


Yann-Gaël Guéhéneuc

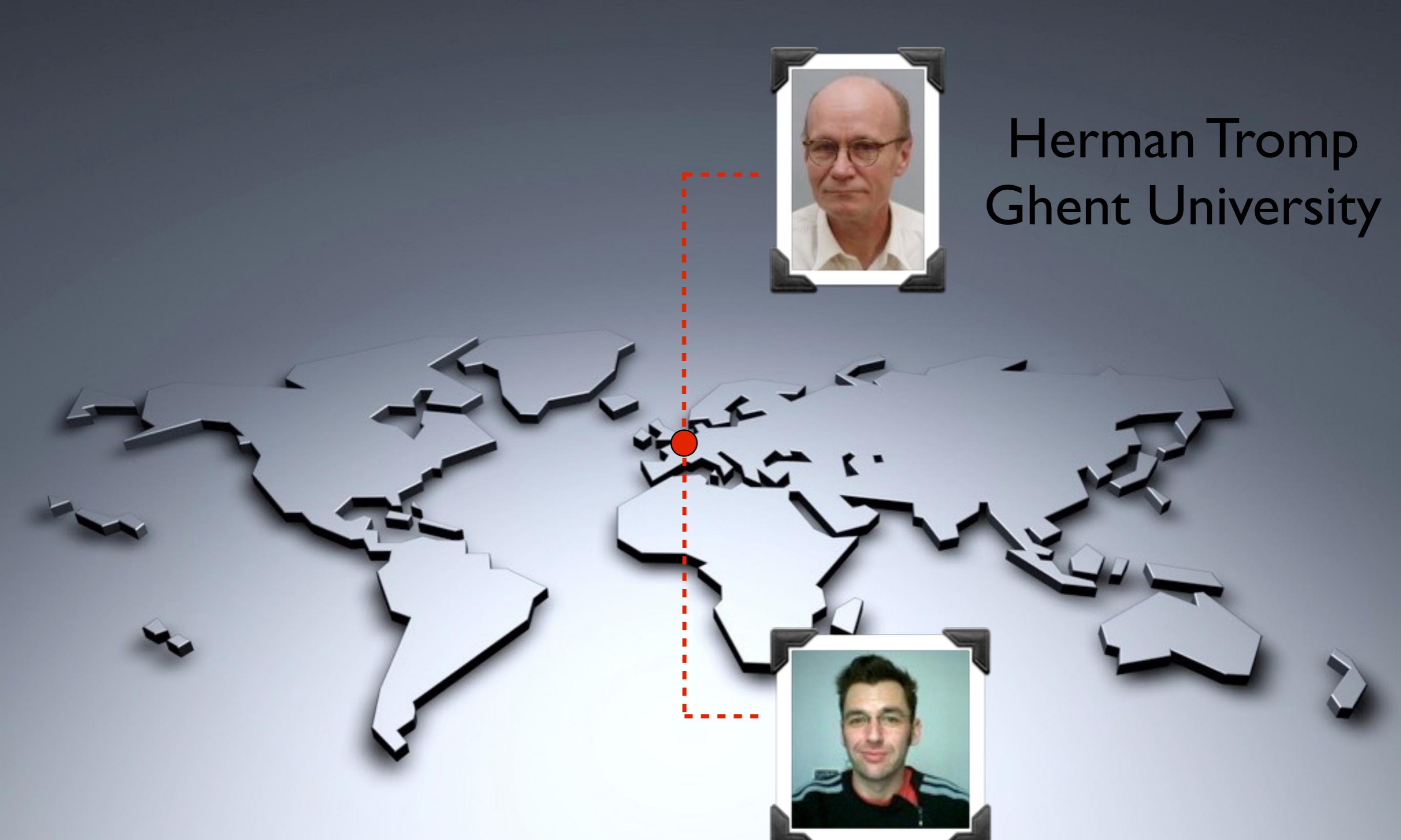
and **special** input by ...



Foutse Khomh



**Who is
Bram Adams?**



Herman Tromp
Ghent University

Wolfgang De Meuter
Vrije Universiteit Brussel



Ahmed E. Hassan
Queen's University



M
C•S
I

(Lab on Maintenance,
Construction and Intelligence
of Software)

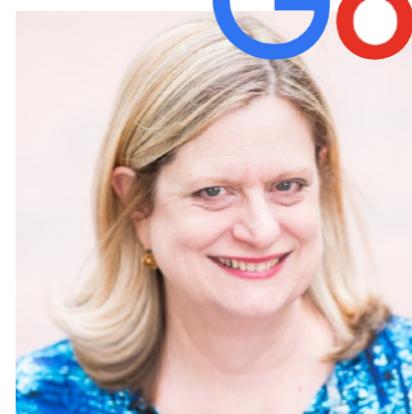
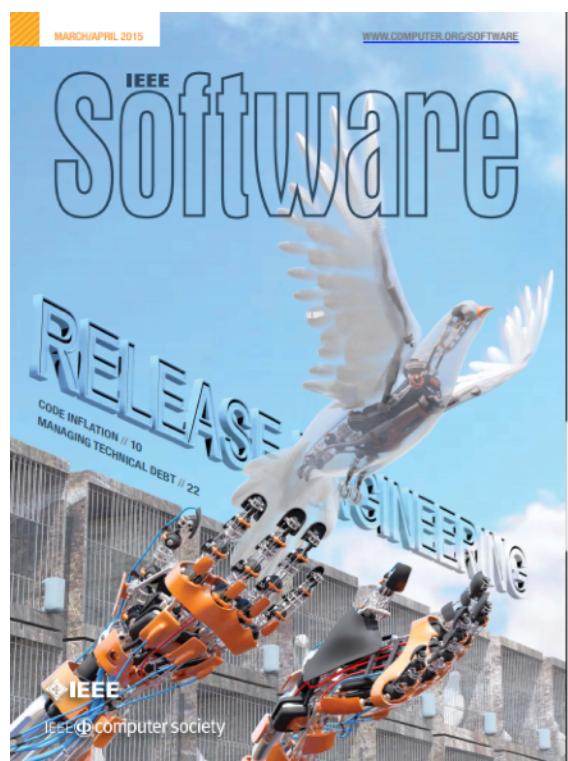


RELENG: International Workshop on Release Engineering

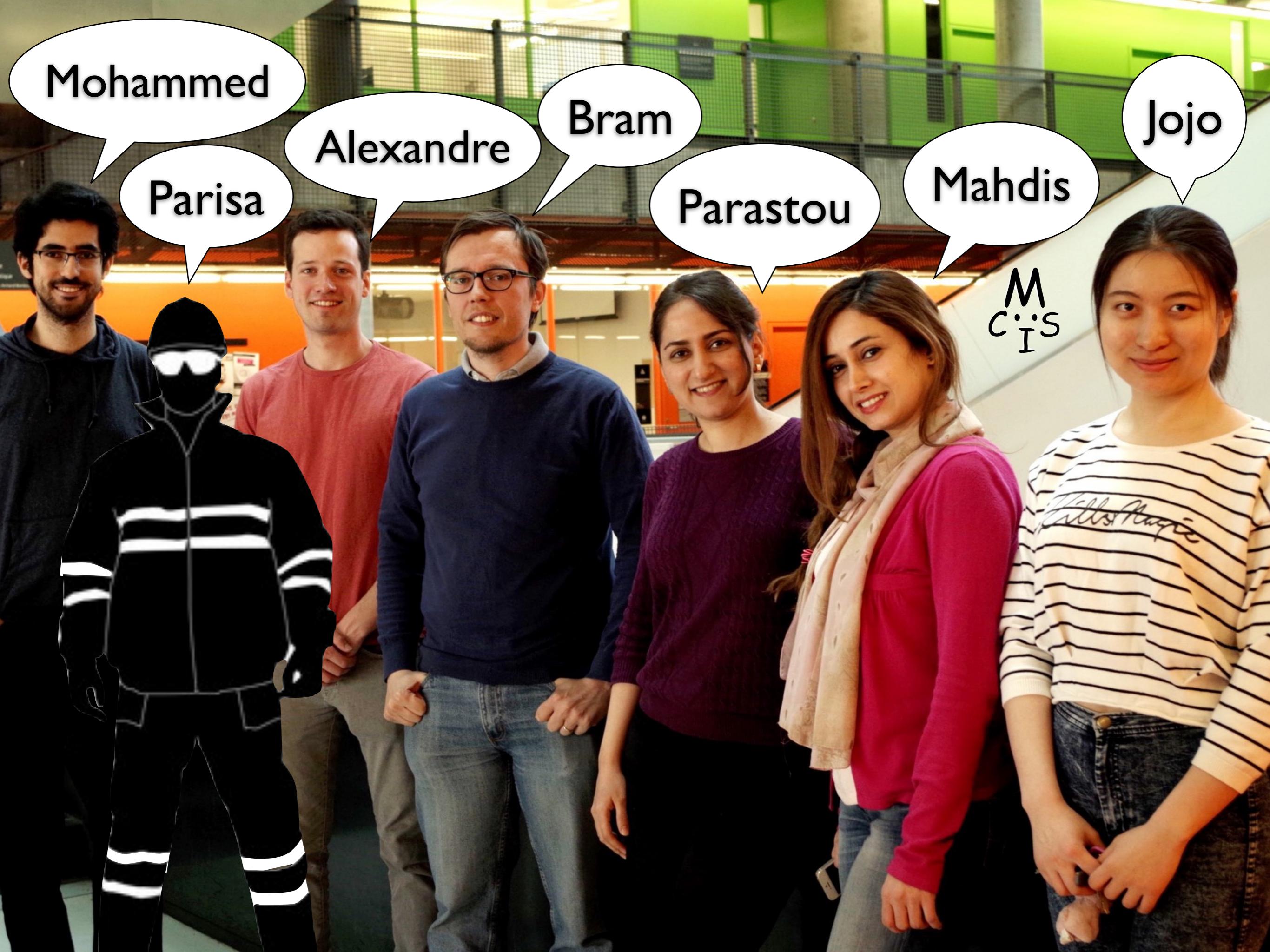
4 editions

>300 participants

dozens of industry
& academic talks



<http://releng.polymtl.ca>



Mohammed

Parisa

Alexandre

Bram

Parastou

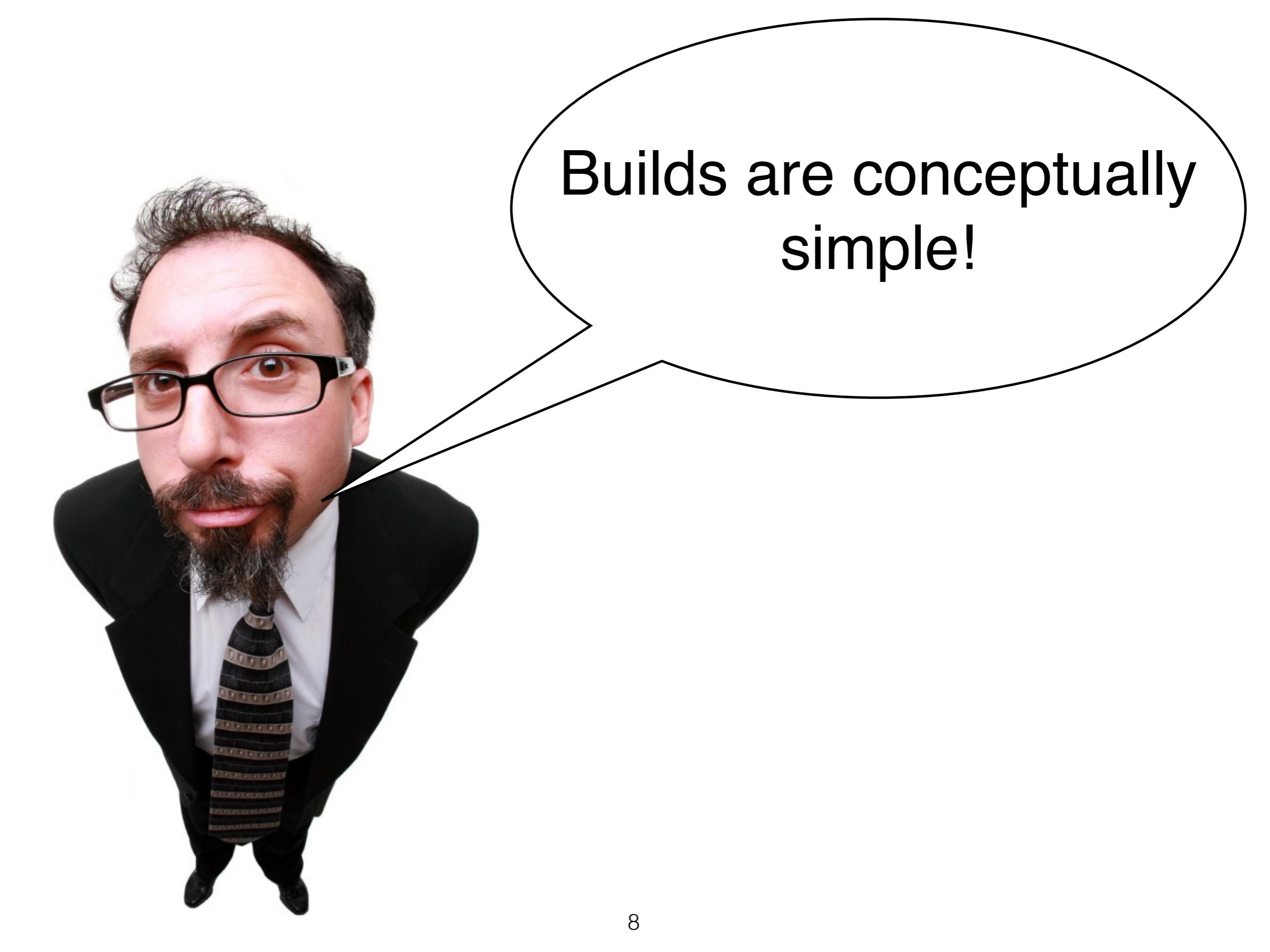
Mahdis

Jojo

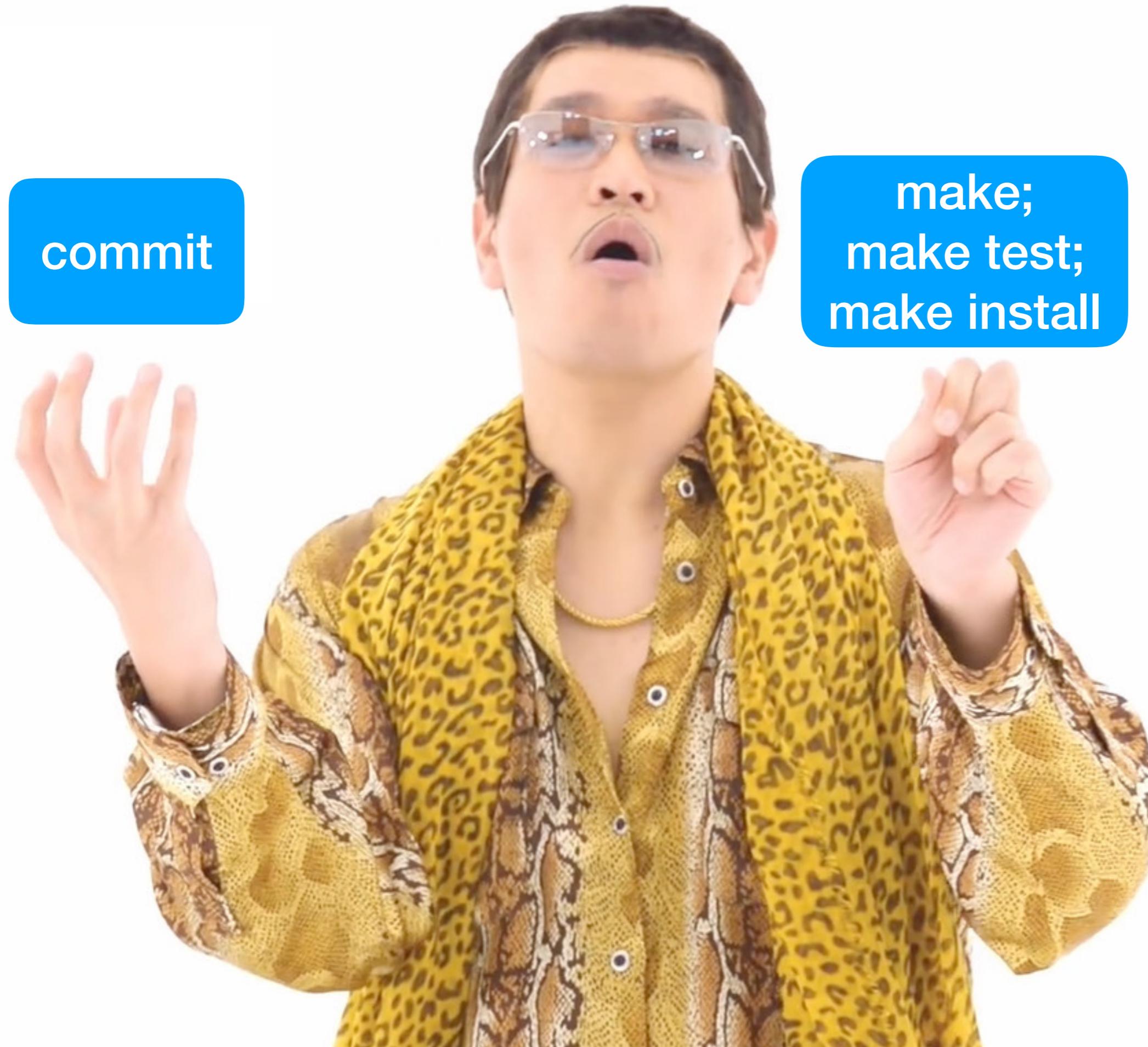
M
C
I
S

Willow

Act 1: “Dawn of the Build Inflation”

A photograph of a man with dark hair, a beard, and glasses, wearing a black suit and a striped tie. He is looking slightly upwards and to his right. A large, thin-lined speech bubble originates from his mouth and extends upwards and to the right, containing the text "Builds are conceptually simple!"

Builds are conceptually
simple!





build

What does a Build do?

compilation

packaging of app
and its
dependencies

signing
executables

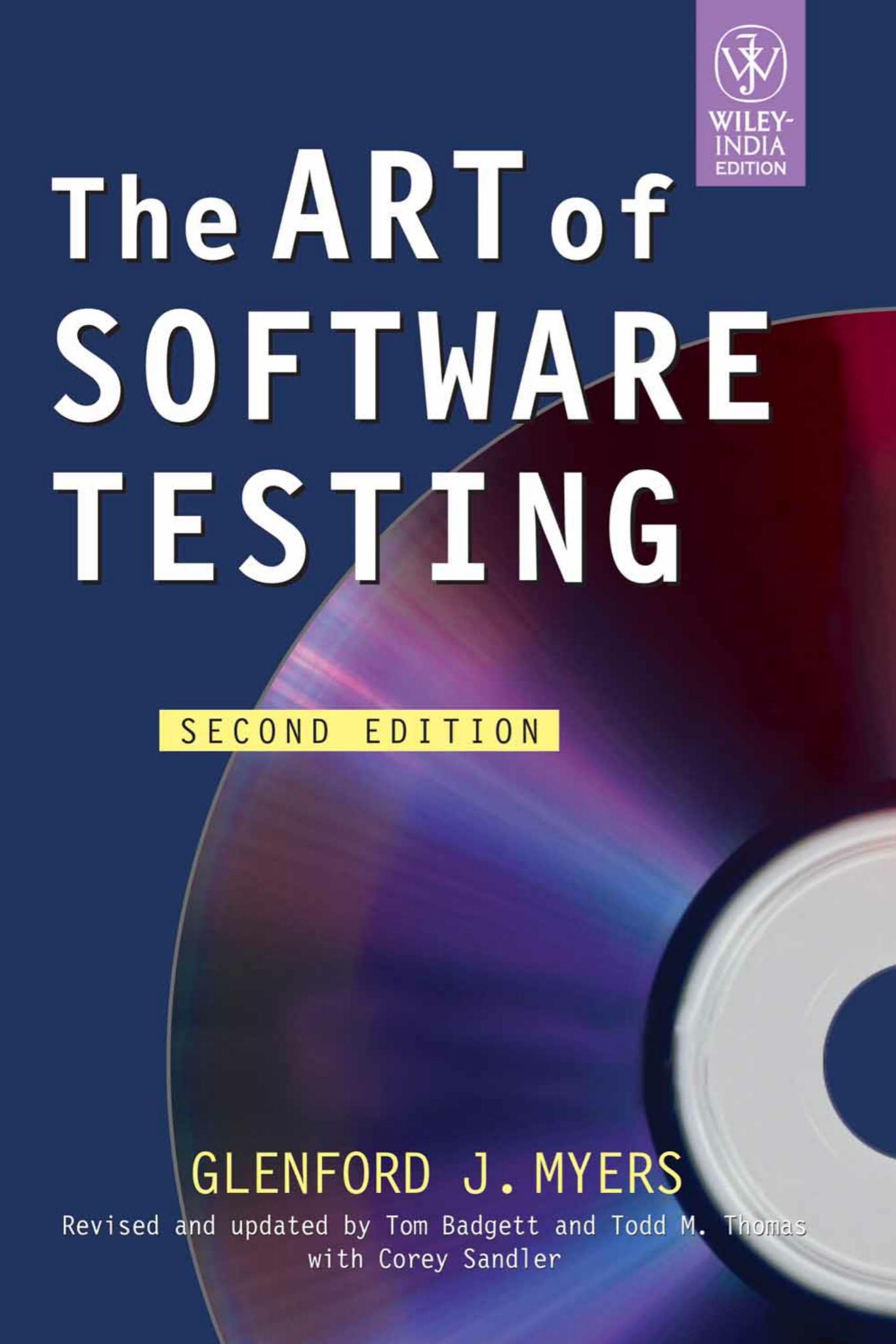
internationalization
of an app

tests

static analysis

generation of
documentation

generation of
partial update
binaries



The ART of SOFTWARE TESTING

SECOND EDITION

GLENFORD J. MYERS

Revised and updated by Tom Badgett and Todd M. Thomas
with Corey Sandler



Testing is the process
of executing a program
with the **intent of**
finding errors.

[Glenford J. Myers]

You mean that build failures
are more **useful** than build
successes as well?



I guess it
depends on the
timing of the
build ...

When to run Builds?

during
development

nightly builds

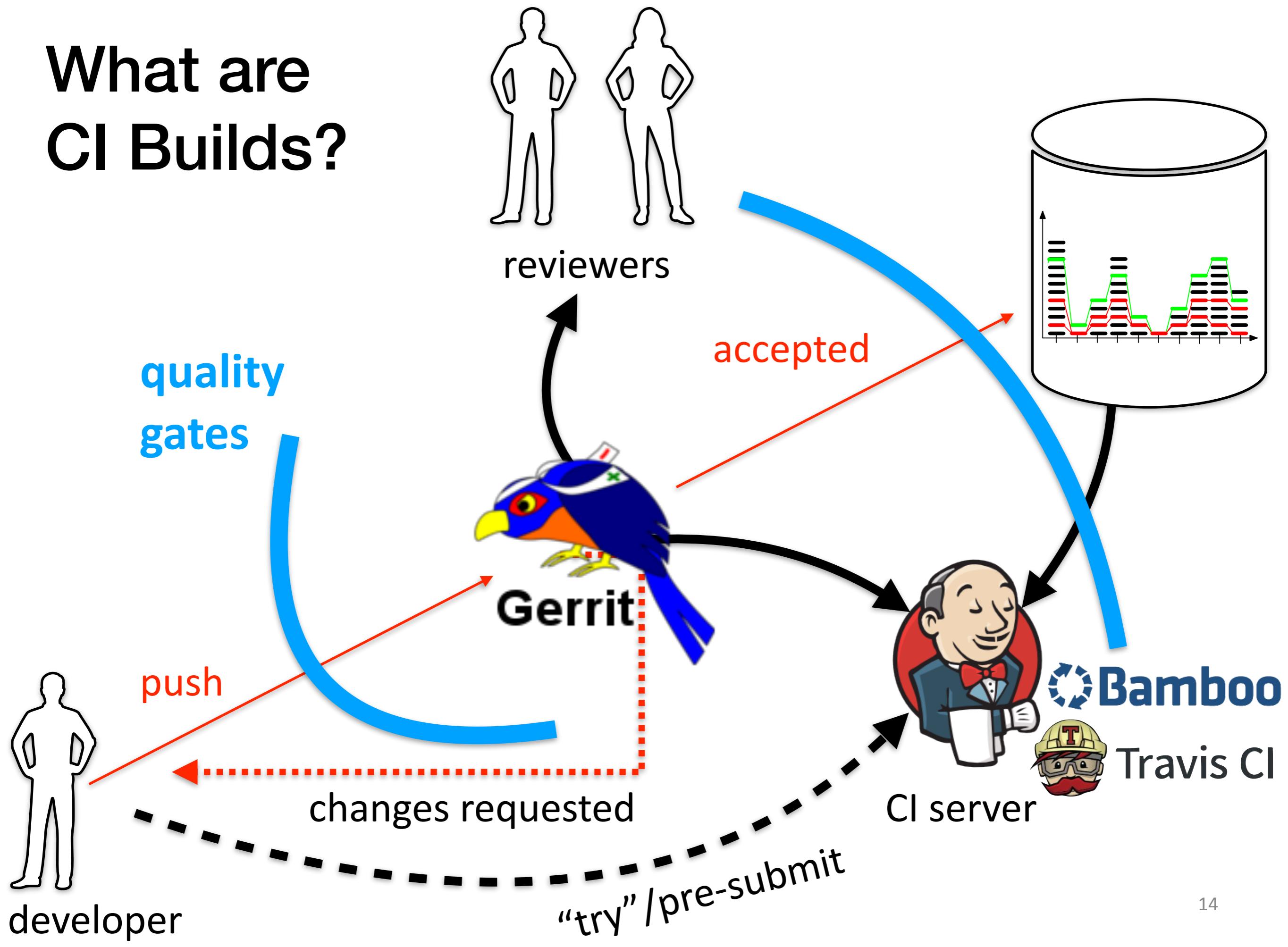
local testing

right before a
release

CI builds!!!



What are CI Builds?



ladle.lbl.gov:8080

Home Most Visited Trackers Cornify Import to Mende... Smart Bookma... Bookmarks

Options Contacts Events Locations Tagspaces Bookmarks Resources

Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools

Jenkins

search cjm | log out

ENABLE AUTO REFRESH

last build history

New Job add description

People

Build History

Project Relationship

Check File Fingerprint

Manage Jenkins

My Views

Build Queue

No builds in the queue.

Build Executor Status

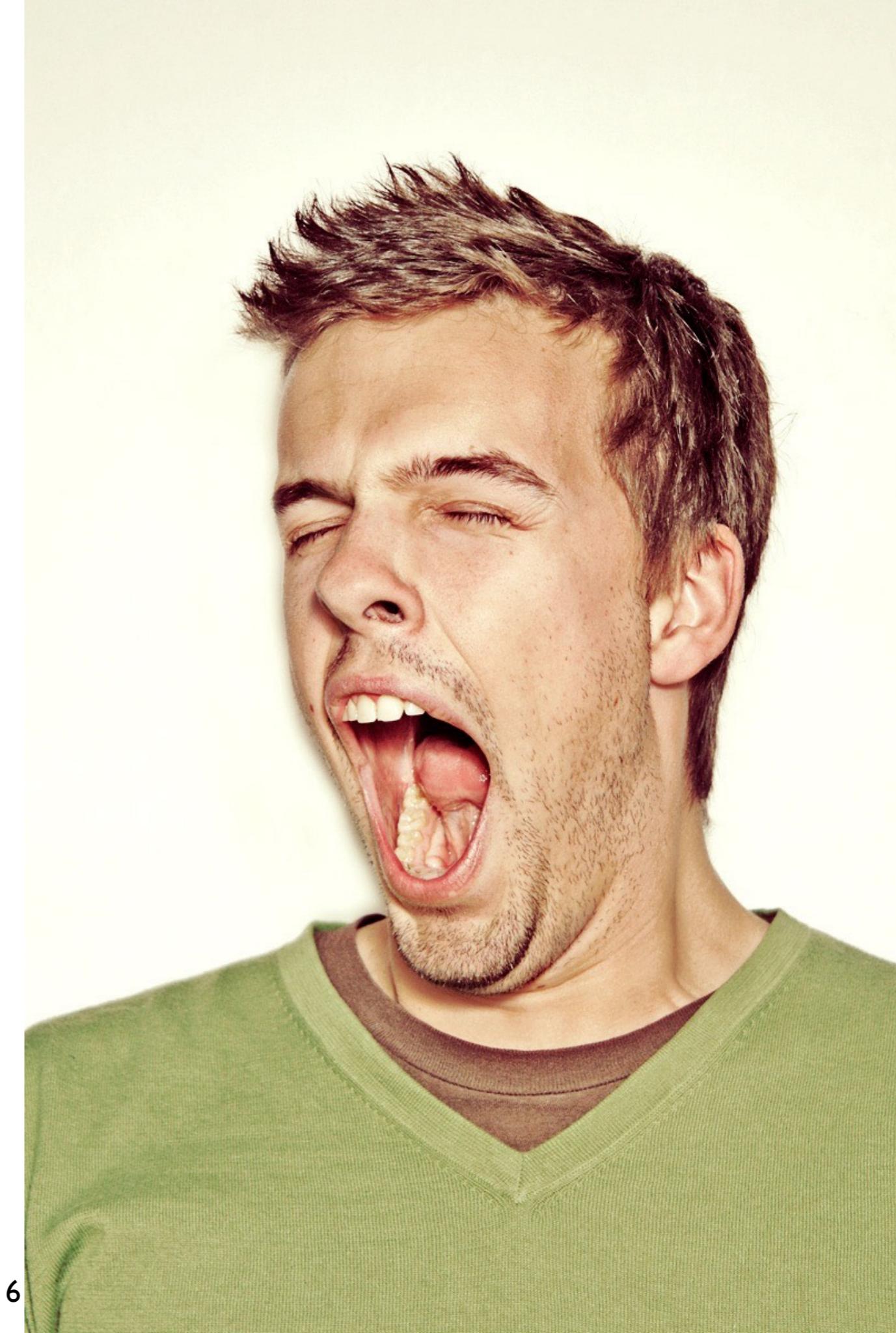
#	Status
1	Idle
2	Idle

All Software

S	W	Name	Last Success	Last Failure	Last Duration	Actions
		build-chebi	5 days 22 hr (#6)	9 days 14 hr (#5)	4 min 49 sec	
		build-cl	3 days 1 hr (#12)	19 days (#10)	1 min 55 sec	
		build-fbbt	5 days 22 hr (#7)	15 days (#6)	2 min 2 sec	
		build-fypo	3 days 9 hr (#7)	7 days 5 hr (#5)	2 min 22 sec	
		build-gaz	1 day 0 hr (#3)	1 day 0 hr (#2)	26 min	
		build-go	4 hr 55 min (#178)	N/A	6 min 48 sec	
		build-go-taxon	1 mo 3 days (#106)	44 min (#243)	3 min 40 sec	
		build-go-xp-chebi	4 hr 48 min (#115)	N/A	2 min 21 sec	
		build-mp	N/A	14 hr (#68)	4 min 1 sec	

<https://douroucouli.wordpress.com/2012/02/16/ontologies-and-continuous-integration/>

We know
all that ...



A photograph of a man with dark hair, a beard, and glasses, wearing a black suit and a striped tie. He is looking slightly upwards and to his right. A large, thin-lined speech bubble originates from his mouth and extends upwards and to the right, containing the text.

Yes, but now things
become interesting!

1

CI does a lot More than just “CI Builds”: OpenStack’s Zuul

Zuul Dashboard

Status

Jobs

Builds

Queue lengths: 0 events, 0 management events, 0 results.

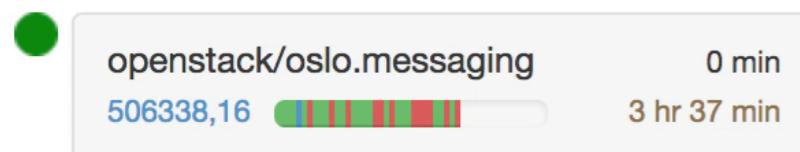
Filters

Expand by default:

check

Newly uploaded patchsets enter this pipeline to receive an initial +/-1 Verified vote.

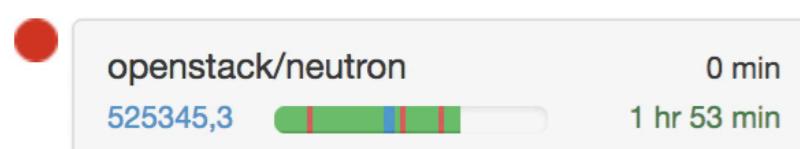
Queue: [openstack/oslo.messaging](#)



Queue: [openstack/tripleo-quickstart-ext...](#)



Queue: [openstack/neutron](#)



Queue: [openstack/cinder](#)



27

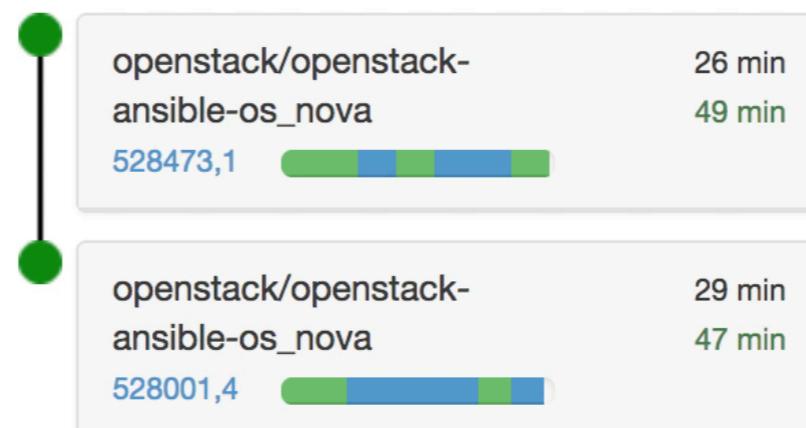
gate

Changes that have been approved by core developers are enqueued in order in this pipeline, and if they pass tests, will be merged.

Queue: [tripleo](#)



Queue: [openstack/openstack-ansible-os_n...](#)



7

post

This pipeline runs jobs that operate after each change is merged.

pre-release

When a commit is tagged with a pre-release tag, this pipeline runs jobs that publish archives and documentation.

release

When a commit is tagged as a release, this pipeline runs jobs that publish archives and documentation.

periodic

Jobs in this queue are triggered on a timer.

release-post

0

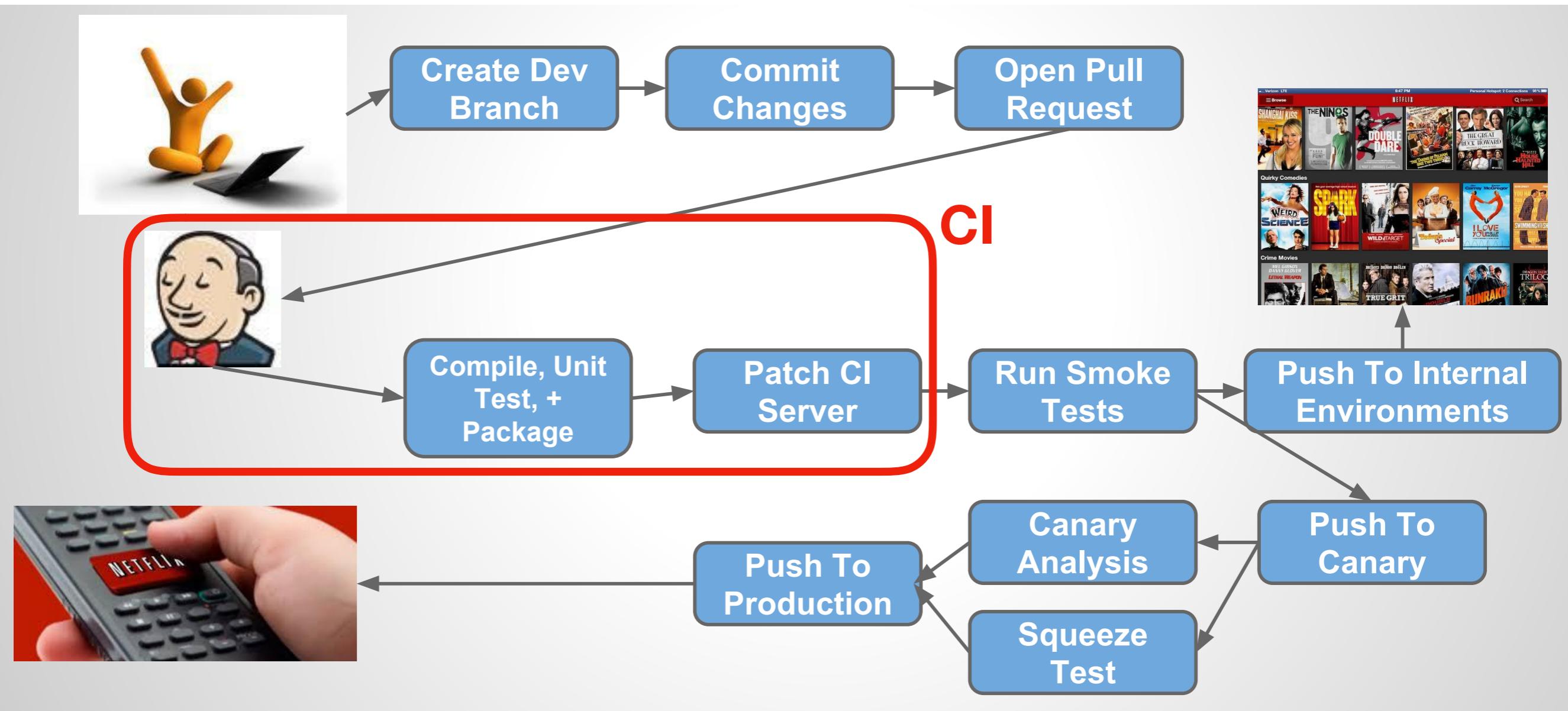
0

0

0

0

E.g., CI forms Backbone of Many Release Engineering Pipelines



Jenkin's Even has a Pipeline DSL!

✓ declarative-ftw 3 Pipeline Changes Tests Artifacts

Branch: — 1s No changes
Commit: — a few seconds ago Started by user admin

Start Build Run Tests Deploy End

Test On Linux

Test On Windows

Steps Test On Linux

> Running tests.. — Print Message

<1s

Some “builds” actually
“deploy” or even
“release”, multiplying the
number of build activities

<https://jenkins.io/blog/2017/09/25/declarative-1/>

2

**CI does All these Builds for All
Variants of a Software System!**



Just Think about how easily a Variant is Created using Feature Toggles ...

```
function reticulateSplines(){
    if( featureEnabled("use-new-SR-algorithm")){
        return enhancedSplineReticulation();
    }else{
        return oldFashionedSplineReticulation();
    }
}
```

```
function oldFashionedSplineReticulation(){
    // current implementation lives here
}
```

```
function enhancedSplineReticulation(){
    // TODO: implement better SR algorithm
}
```

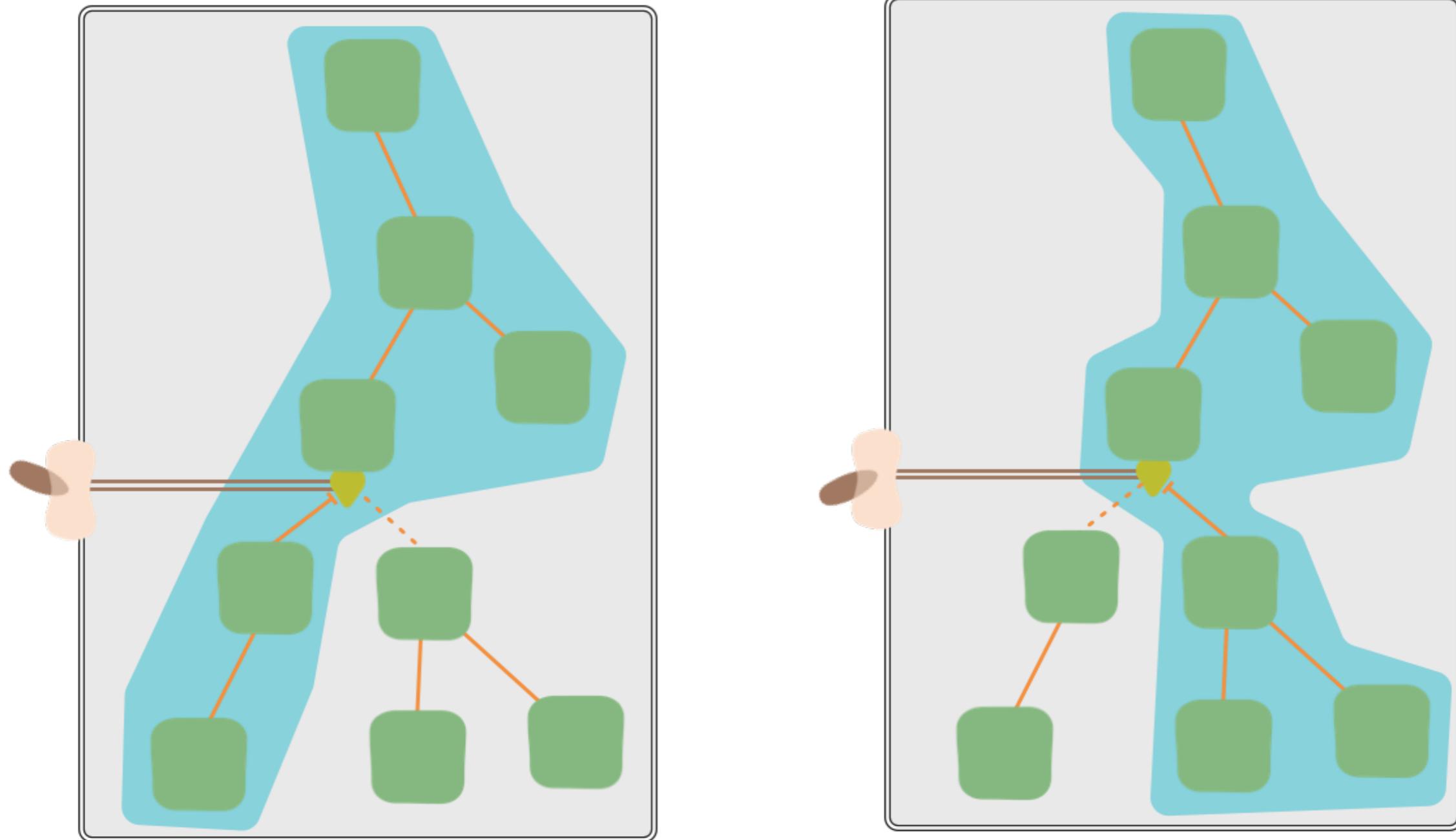
feature initially turned off during testing until stable

new feature being developed for upcoming release

Testing a Toggled Feature



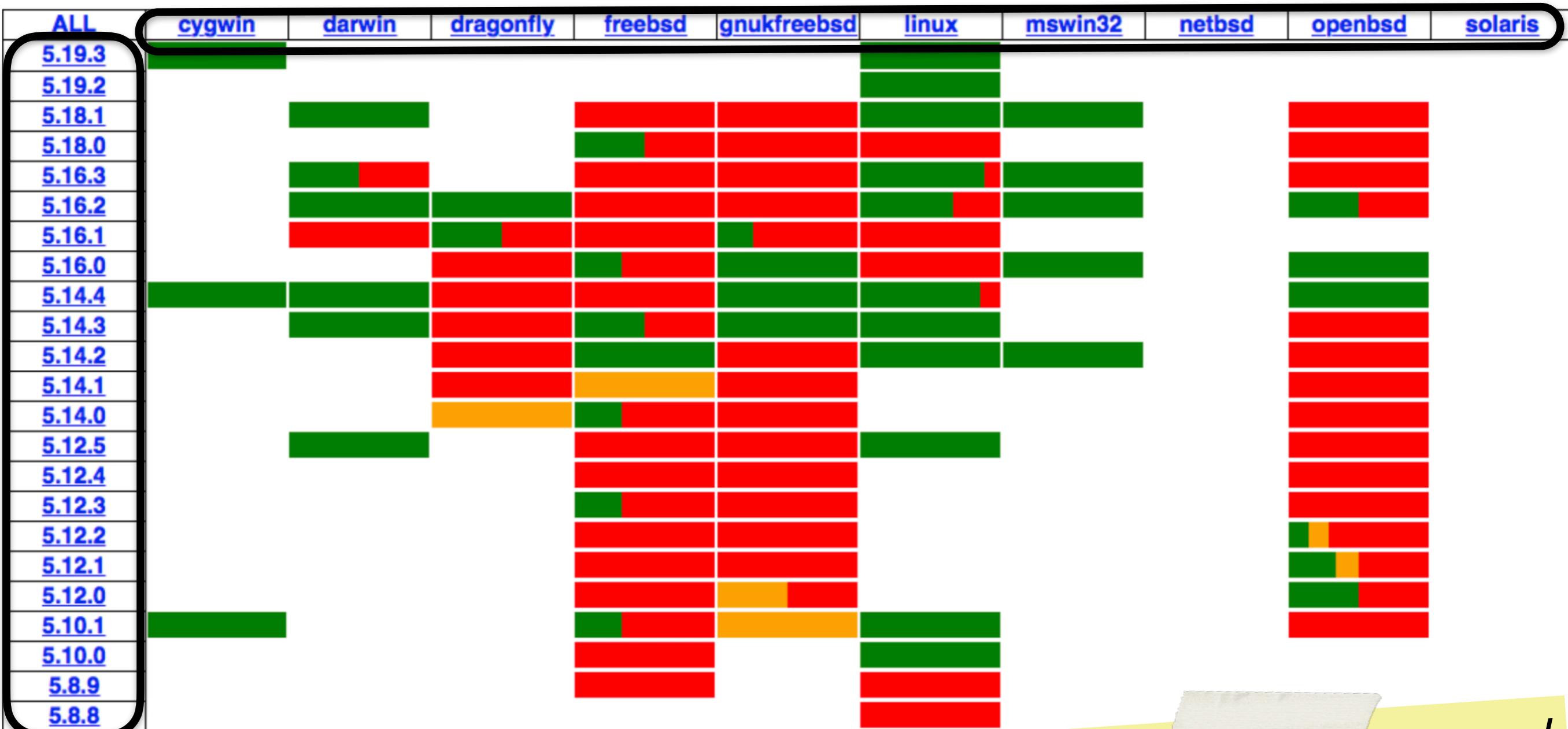
N toggles = 2^N
variants to test!



3

There are also Different Variants of the Software's Environment

different OSes

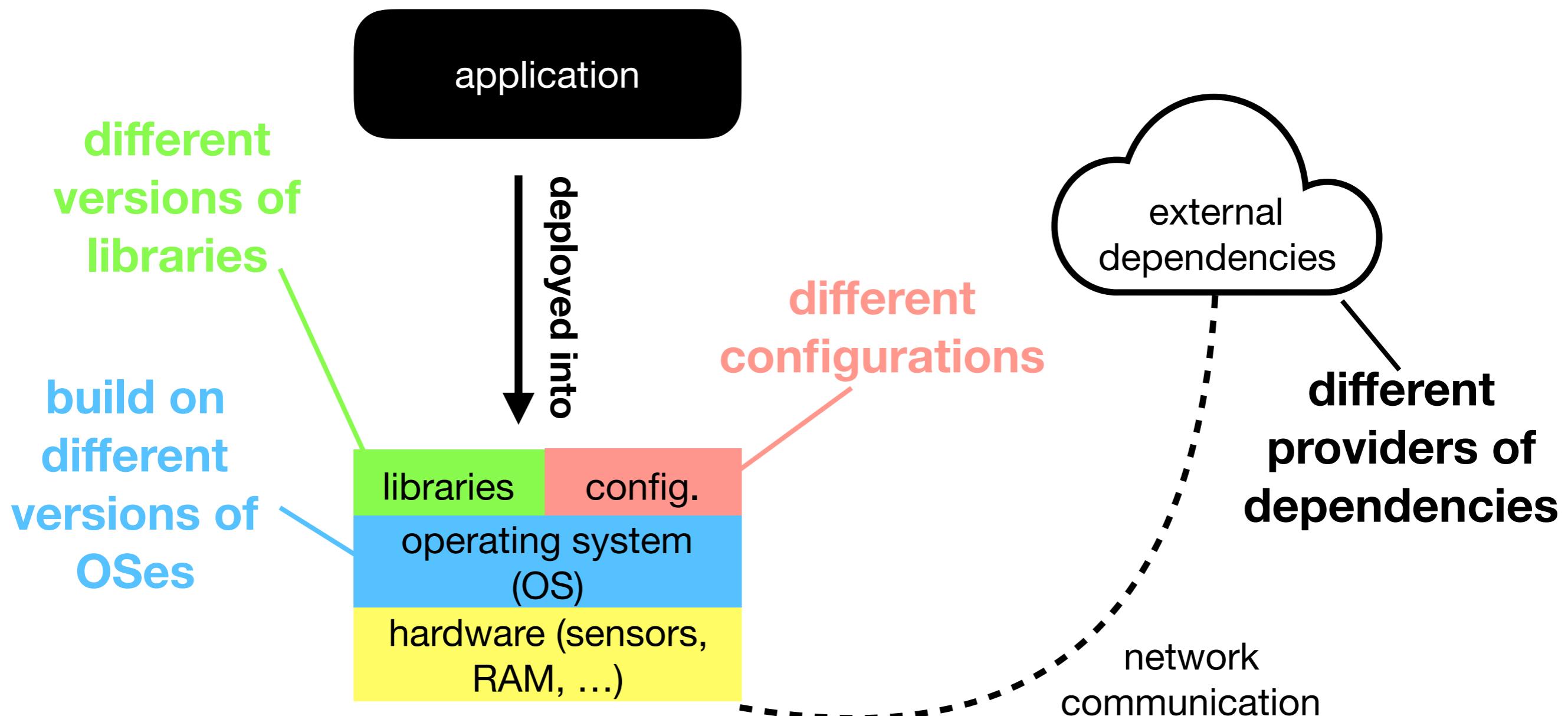


different Perl versions

<http://cpantesters.org/>

Explosion of Additional CI Builds!

Infrastructure-as-Code
makes it easy to specify different variants of environment



mozilla-inbound

i

mozilla-central

i

x

13 unclassified

Y

(+)

○

○

○

○

○

○

Y

Filter platforms & jobs

Sat Dec 16, 17:43:46 - ccoroiu@mozilla.com

99% - 5 in progress

View Tests

▼

454cafb99225 cc Merge mozilla-central to mozilla-inbound
 7cb064f0d25e cc Merge autoland to mozilla-central r=
 29c2618abb00 F No bug, Automated blocklist update
 be7679eee4ab F No bug, Automated HPKP preload list
 625f7e9eb46b F No bug, Automated HSTS preload list
 376a9968d450 MK Bug 1425217 - Remove Encoding::I
 52da1c601408 DT Bug 1377007 - Tweak BoyerMooreF
 0502ddaaaf680 DT Bug 1377007 - autospider.py,runcpt
 d1d7c3af8ee9 DT Bug 1377007 - Tests: Parsing ECMAS
 27a84140b47d DT Bug 1377007 - JS shell bindings for
 1a7b0410b795 DT Bug 1377007 - Implementing basic
 bdbaccfb070e DT Bug 1377007 - GC for binjs-ref pars
 70ae74a3a1b7 DT Bug 1377007 - Implementation of th
 f1eeaf08ba77 DT Bug 1377007 - Introducing BinField
 08d9f4f1fd6e DT Bug 1377007 - Enable/disable build
 ac32aa0058fe JW Bug 1425170. P2 - remove unused
 26b2e4c28172 JW Bug 1425170. P1 - add a member to

Linux opt

tc(+8) tc-M-e10s(+2) tc-R-e10s(+3)
tc-W-e10s(+3)

Linux pgo

tc(B)

Linux debug

SM-tc(arm) tc(+7) tc-M(c3) tc-M-e10s(+3)
tc-R-e10s(+5) tc-W-e10s(Wr1★ +4) tc-X
(+3)

Linux Stylo Disabled debug

tc-M-e10s(gpu) tc-W-e10s(wpt10)

Linux x64 opt

S SM-tc(+5) T-e10s(+4) tc(+9) tc-M-e10s(+10)
tc-R-e10s(+4) tc-W-e10s(+2)
tc-e10s[tier 2](TV) [tier 2](AB)

Linux x64 pgo

T-e10s(+3) tc(+7) tc-M-e10s(bc1)
tc-R-e10s(+3) tc-W-e10s(+2) tc-e10s[tier 2](TV)

Linux x64 asan

tc(+10) tc-M(+2) tc-M-e10s(mda1★ +5)
tc-R-e10s(+4) tc-X(+2) tc-e10s[tier 2](TV)

Linux x64 debug

S SM-tc(cgc +6) SM-tc[tier 2](rust) tc(+10)
tc-M-e10s(+27) tc-R-e10s(+5) tc-X(X8)
tc-e10s[tier 2](TV)

Linux x64 QuantumRender opt

tc-R-e10s(+6)

Linux x64 QuantumRender debug

tc-R-e10s(+7)

Linux x64 Stylo Disabled opt

tc-M-e10s(+2) tc-W-e10s(wpt4)

Linux x64 Stylo Disabled debug

tc-M-e10s(+10)

Linux x64 NoOpt debug

tc[tier 2](B)

OS X 10.10 opt

tc(+2) tc-M-e10s(ss +4) tc-R-e10s(+2)

Travis CI also Builds in Different Environments!

rails / rails  build passing

Current Branches Build History Pull Requests

More options 

 3-2-stable CRON Remove `DEFAULT NULL` for primary key column to support MySQL 5.7.3

Since MySQL 5.7.3 m13 does now allow primary key column is null.

(cherry picked from commit b6655885ef13cf8d1705dc9b5232846f0207febd)

 Commit e17e25c

 Branch 3-2-stable

 Yasuo Honda authored  Andrew White committed

 #43033 failed

 Ran for 25 min 37 sec

 Total time 9 hrs 26 min 35 sec

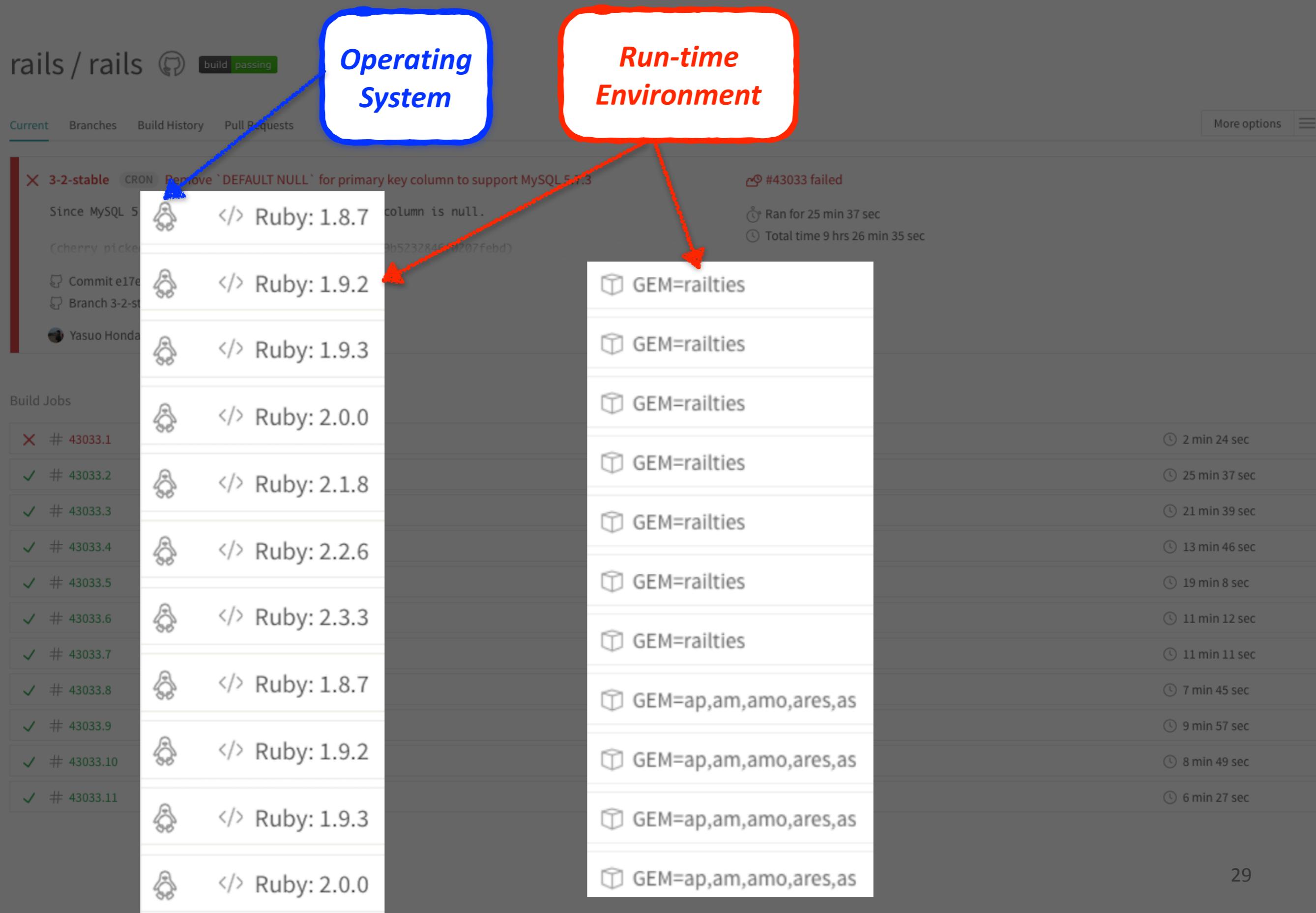
 about an hour ago

Build Jobs

 # 43033.1	 </> Ruby: 1.8.7	 GEM=railties	 2 min 24 sec
 # 43033.2	 </> Ruby: 1.9.2	 GEM=railties	 25 min 37 sec
 # 43033.3	 </> Ruby: 1.9.3	 GEM=railties	 21 min 39 sec
 # 43033.4	 </> Ruby: 2.0.0	 GEM=railties	 13 min 46 sec
 # 43033.5	 </> Ruby: 2.1.8	 GEM=railties	 19 min 8 sec
 # 43033.6	 </> Ruby: 2.2.6	 GEM=railties	 11 min 12 sec
 # 43033.7	 </> Ruby: 2.3.3	 GEM=railties	 11 min 11 sec
 # 43033.8	 </> Ruby: 1.8.7	 GEM=ap,am,amo,ares,as	 7 min 45 sec
 # 43033.9	 </> Ruby: 1.9.2	 GEM=ap,am,amo,ares,as	 9 min 57 sec
 # 43033.10	 </> Ruby: 1.9.3	 GEM=ap,am,amo,ares,as	 8 min 49 sec
 # 43033.11	 </> Ruby: 2.0.0	 GEM=ap,am,amo,ares,as	 6 min 27 sec



Travis CI build environment

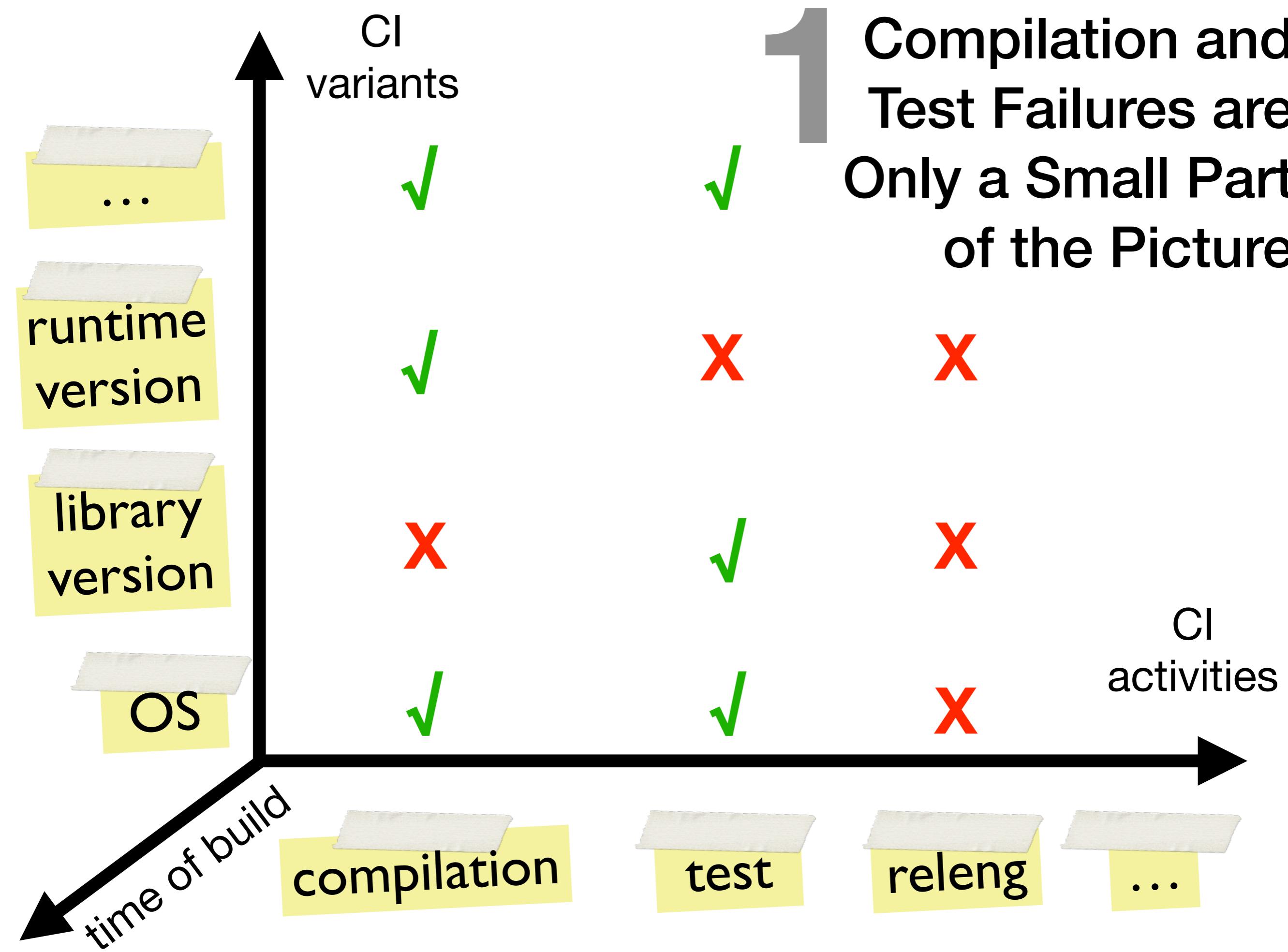




And I guess
that **each failing**
build should probably
be **followed up** by a
code fix, which in turn
needs to be built on
all variants!

Act 2: “What is Wrong with this Build Inflation?”

1 Compilation and Test Failures are Only a Small Part of the Picture



... while Existing Build Research Mostly Focuses on Predicting compilation/test failures, using:

triggering commit

trigger time/ period

trigger author

build history

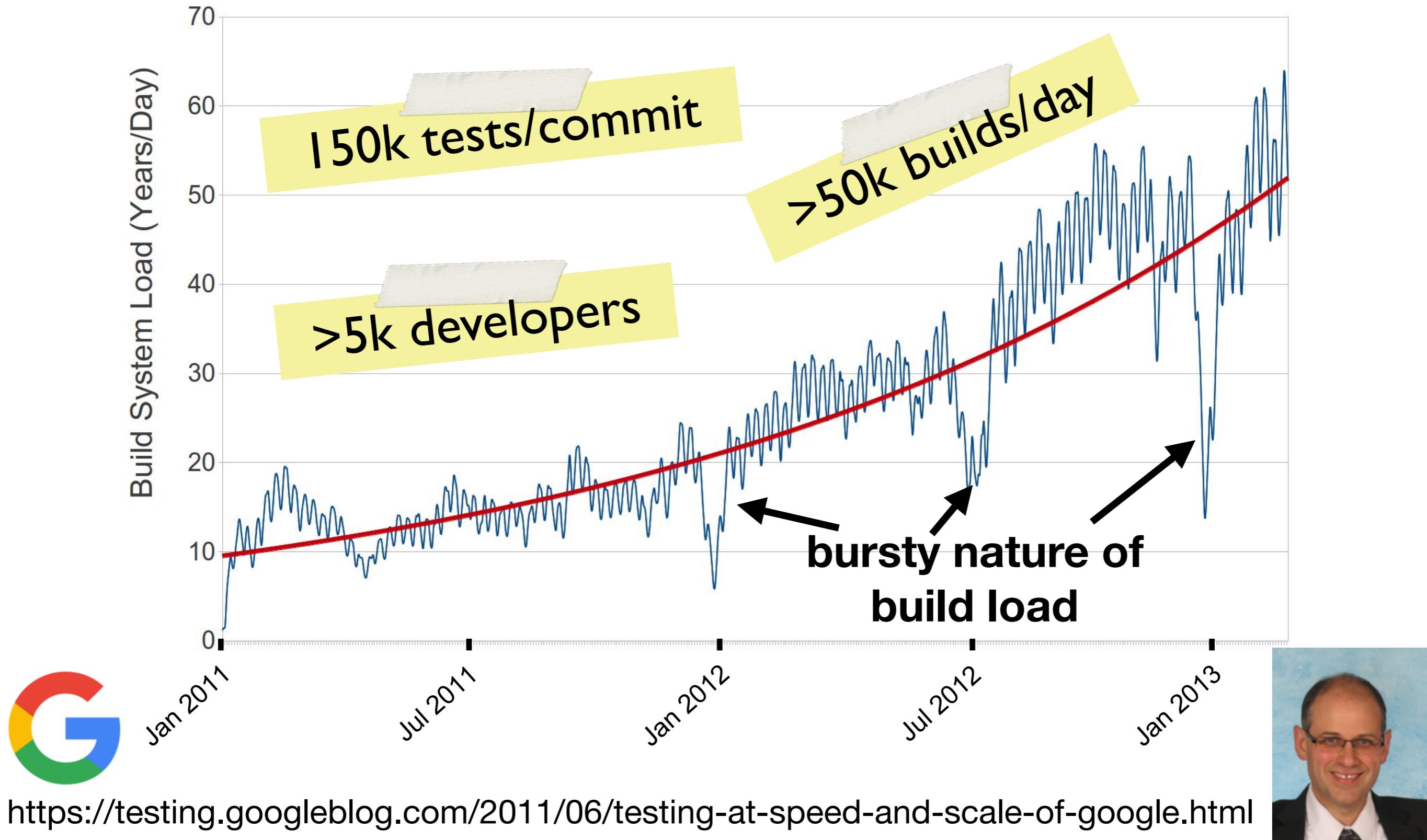
type of VCS action

role who triggered commit

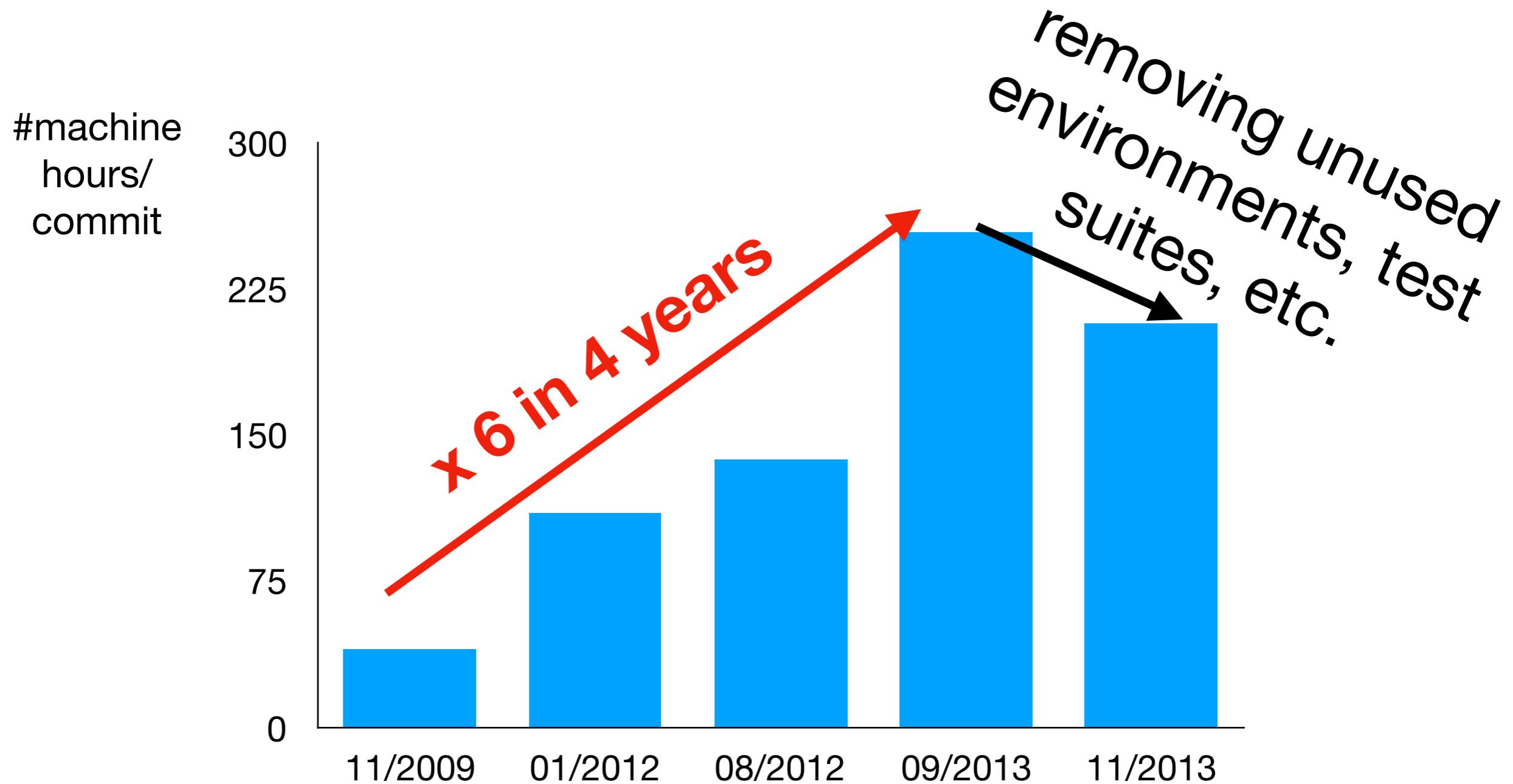
development team

feature vs. bug fix

2 More Builds, More Load



Not Only for Web Apps



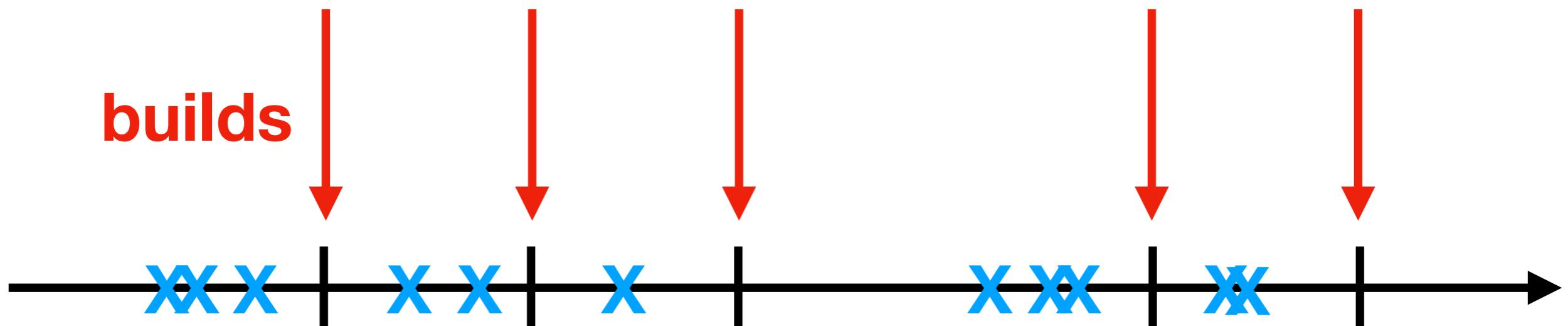
<https://atlee.ca/blog/posts/blog20091102what-happens-when-you-push.html>

<https://atlee.ca/blog/posts/blog20120113what-happens-when-you-push-2012-edition.html>

Load Becomes so High that CI Servers cannot Cope Anymore



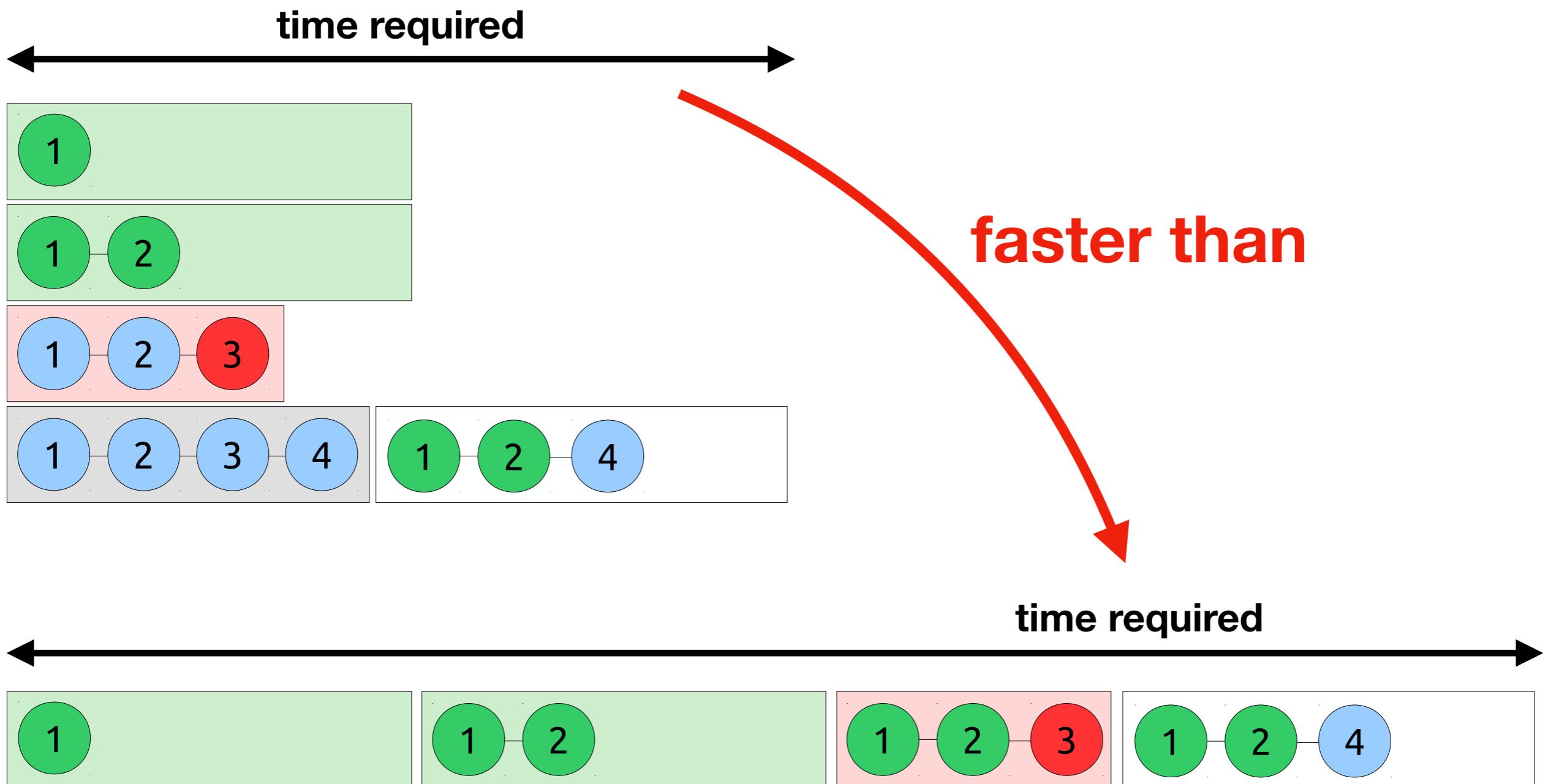
Mitigation #1: Coalescing Commits Arriving Together



problem: if build fails,
not clear which commit
is the root cause

in that case,
expensive bisecting
needed

Mitigation #2: Speculative vs. Serial Gating





Mitigation #3: JIT Scheduling

Continuous Integration:

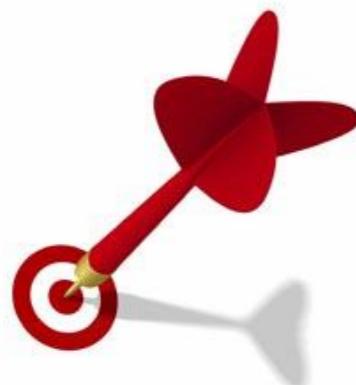
- Run every test affected at every ~~changelist~~.

as often[^] as possible



Schedule tests to run only when system has capacity.

i.e., test case prioritization!



Produce project-wide results at periodic changelists.



Ugly side-effect of
these strategies:
**traceability from commit to
build is screwed up!**

3 Builds = \$\$\$



- assuming that:
 - only AWS build costs money (incorrect => cost estimation is **lower bound**)
 - two cheapest AWS regions are used for daily production load, and a third region on hot-backup
 - healthy mix of “OnDemand”, “Reserved” and “Spot” AWS instances is used
- then: the **build cost of one Mozilla commit is USD 26.40**
- having 7,601 commits in 12/2013, the **total monthly cost is USD 201k**

Let's Revisit our Travis CI Example

rails / rails  build passing

Current Branches Build History Pull Requests

More options 

X 3-2-stable CRON Remove `DEFAULT NULL` for primary key column to support MySQL 5.7.3

Since MySQL 5.7.3 m13 does now allow primary key column is null.

(cherry picked from commit b6655885ef13cf8d1705dc9b5232846f0207febd)

 Commit e17e25c

 Branch 3-2-stable

 Yasuo Honda authored  Andrew White committed

 #43033 failed

 Ran for 25 min 37 sec

 Total time 9 hrs 26 min 35 sec

 about an hour ago

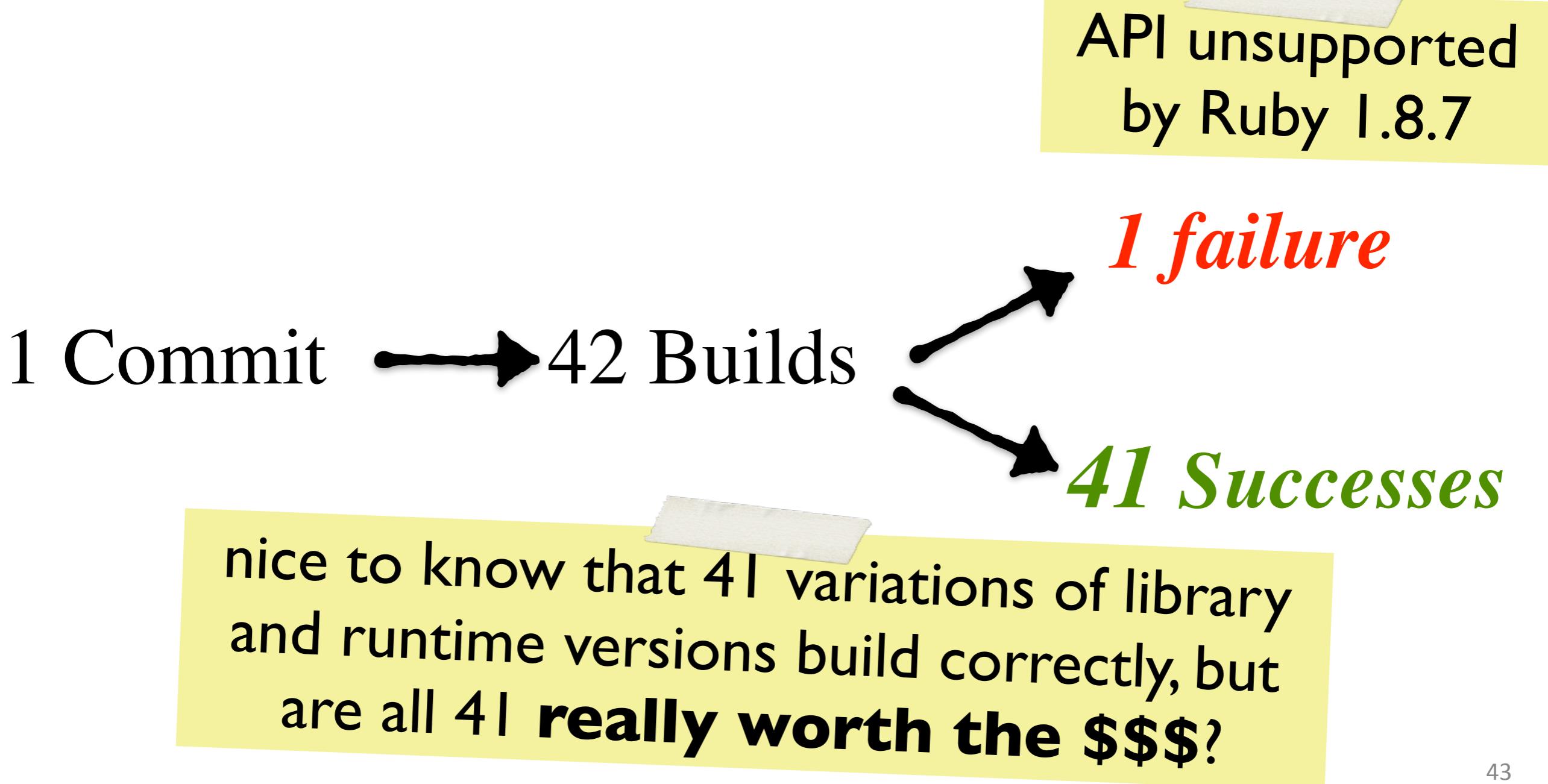
Build Jobs

 # 43033.1	 </> Ruby: 1.8.7	 GEM=railties	 2 min 24 sec
 # 43033.2	 </> Ruby: 1.9.2	 GEM=railties	 25 min 37 sec
 # 43033.3	 </> Ruby: 1.9.3	 GEM=railties	 21 min 39 sec
 # 43033.4	 </> Ruby: 2.0.0	 GEM=railties	 13 min 46 sec
 # 43033.5	 </> Ruby: 2.1.8	 GEM=railties	 19 min 8 sec
 # 43033.6	 </> Ruby: 2.2.6	 GEM=railties	 11 min 12 sec
 # 43033.7	 </> Ruby: 2.3.3	 GEM=railties	 11 min 11 sec
 # 43033.8	 </> Ruby: 1.8.7	 GEM=ap,am,amo,ares,as	 7 min 45 sec
 # 43033.9	 </> Ruby: 1.9.2	 GEM=ap,am,amo,ares,as	 9 min 57 sec
 # 43033.10	 </> Ruby: 1.9.3	 GEM=ap,am,amo,ares,as	 8 min 49 sec
 # 43033.11	 </> Ruby: 2.0.0	 GEM=ap,am,amo,ares,as	 6 min 27 sec

⋮

4

Build Inflation = Diminishing Value of Build Result



Act 3: “An Empirical Study on the Value of a Build”

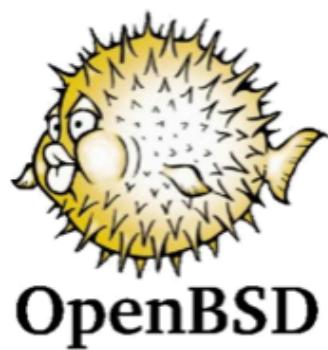


Do not trust build results at face value: an empirical study of 30 million CPAN builds
(Zolfagharia et al., MSR 2017)



Case Study Setup

**Mining the whole history of
CPAN-Testers builds** (39k packages,
10 OSes & 103 Perl versions)



Cygwin



Darwin

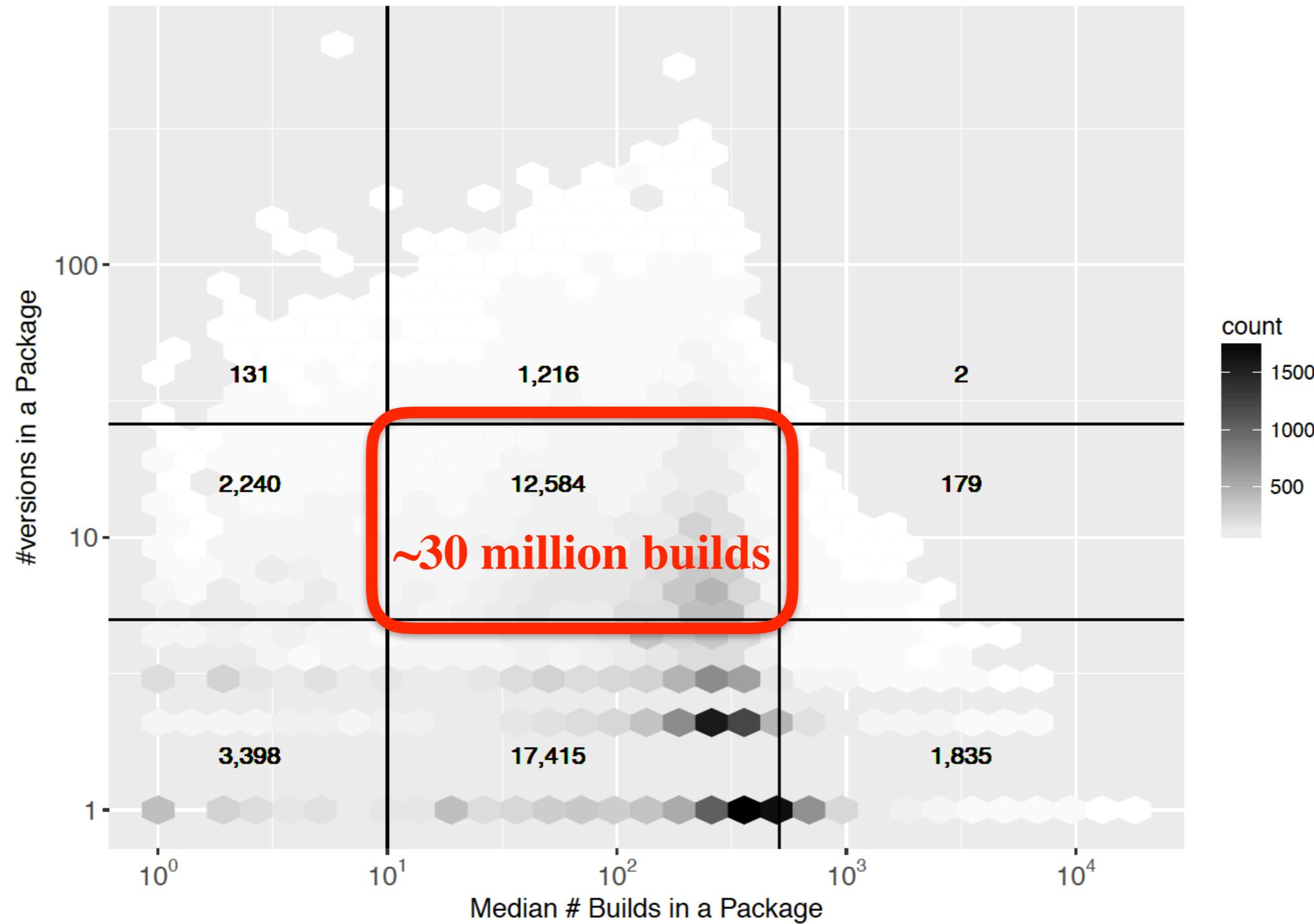


FreeBSD

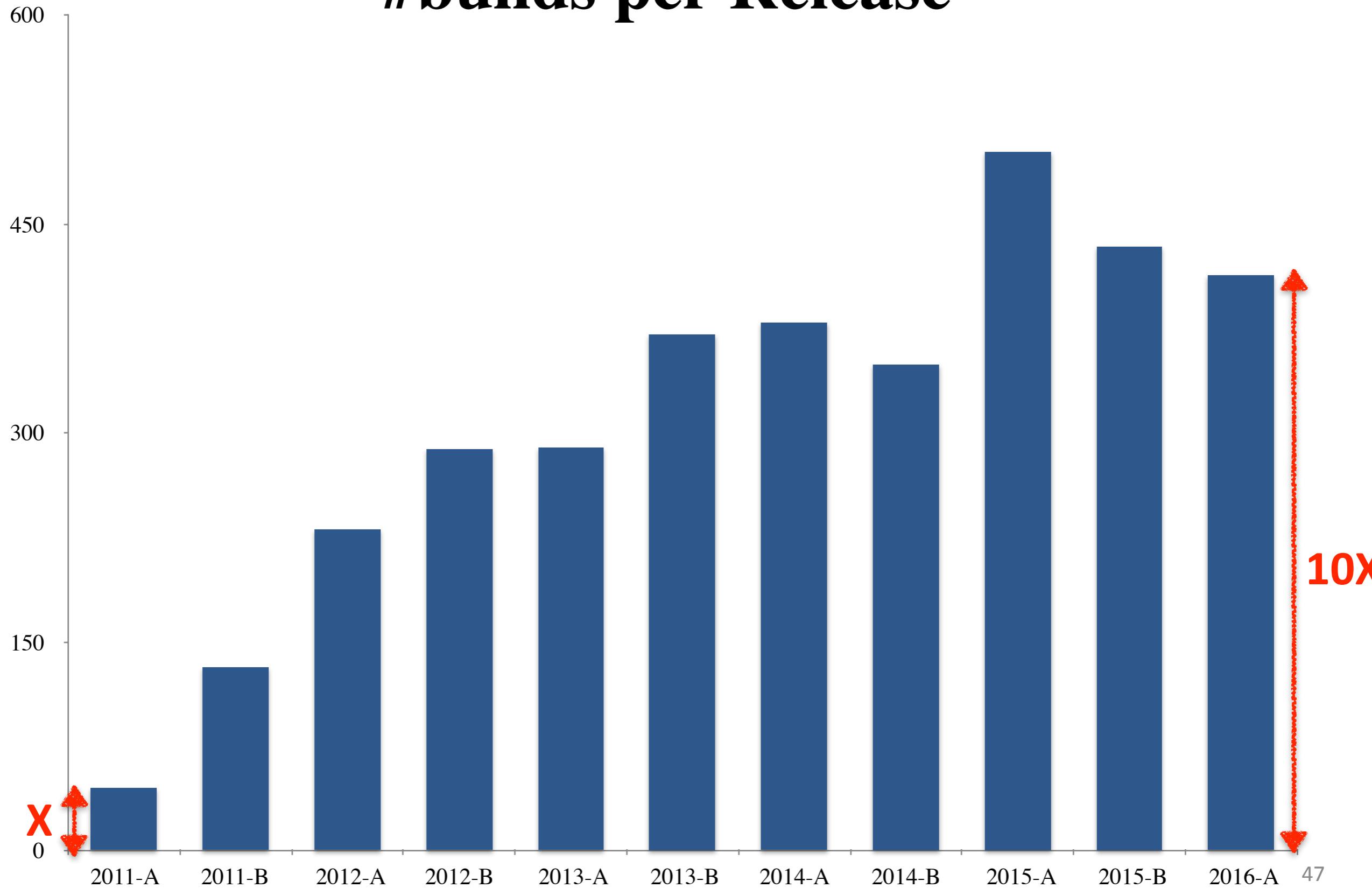


Linux

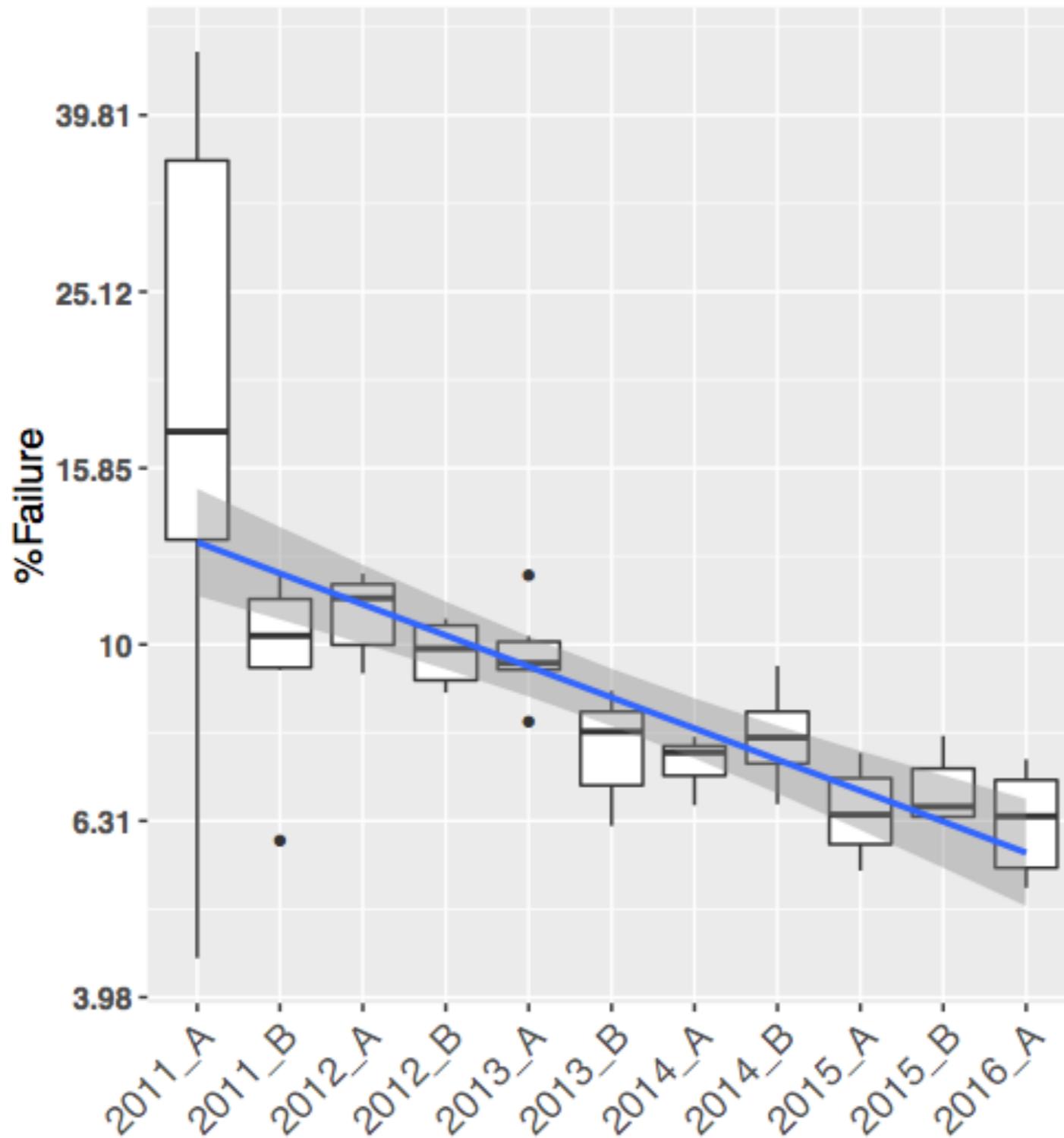
CPAN Packages were Filtered Based on Distribution of #Builds and #Versions



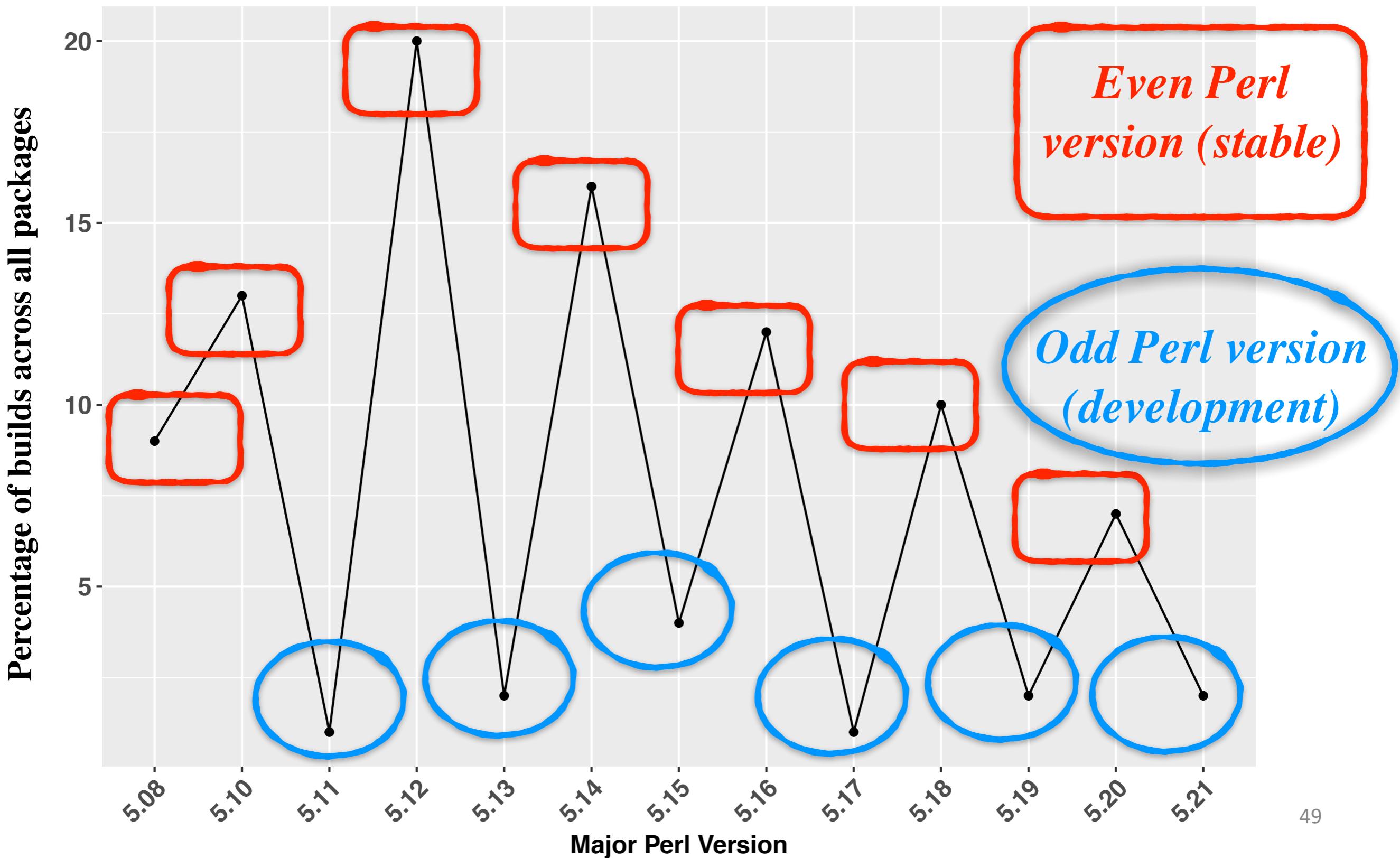
10-fold Increase in Average #builds per Release



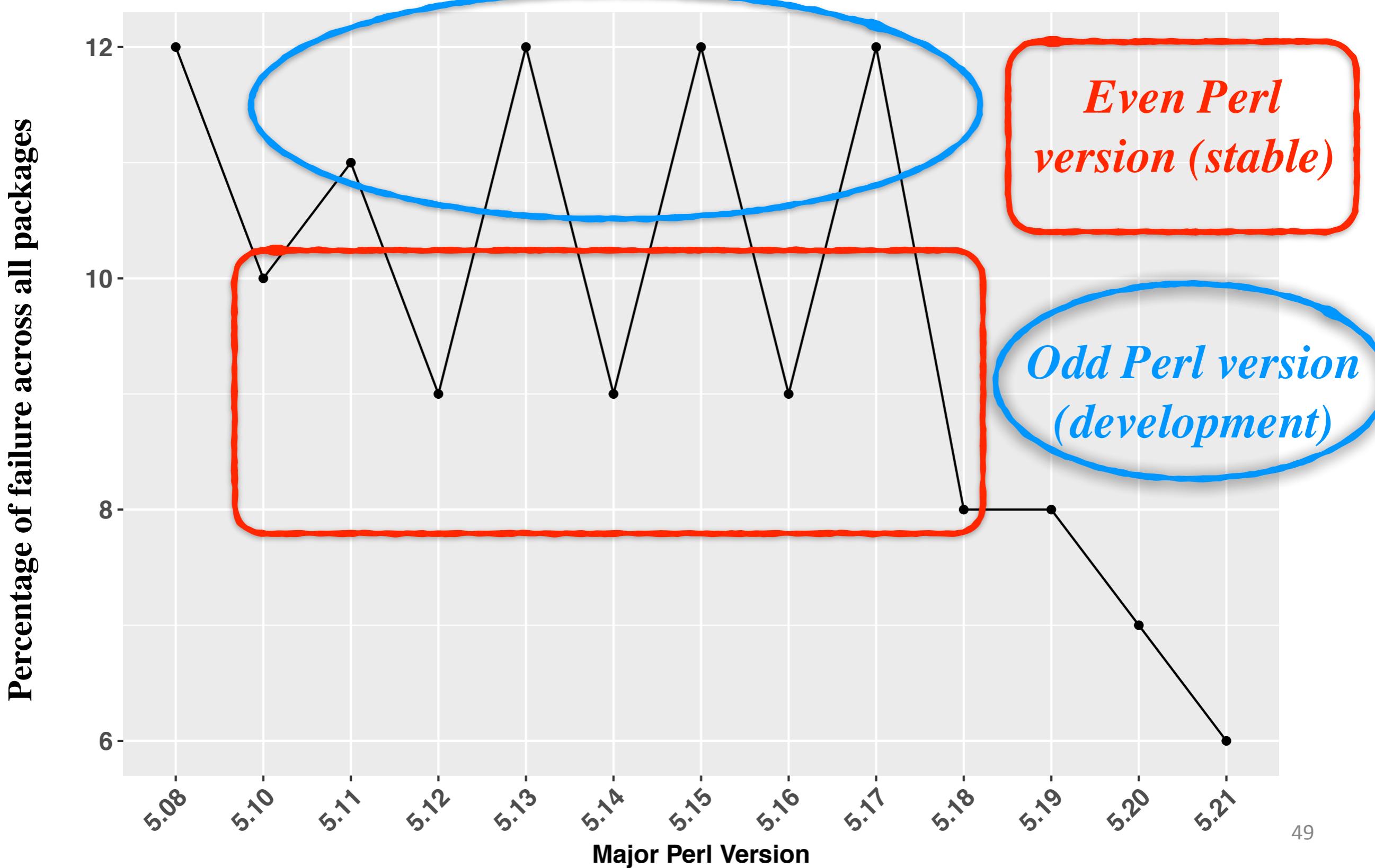
Build Inflation: More Builds Finding Less Failures



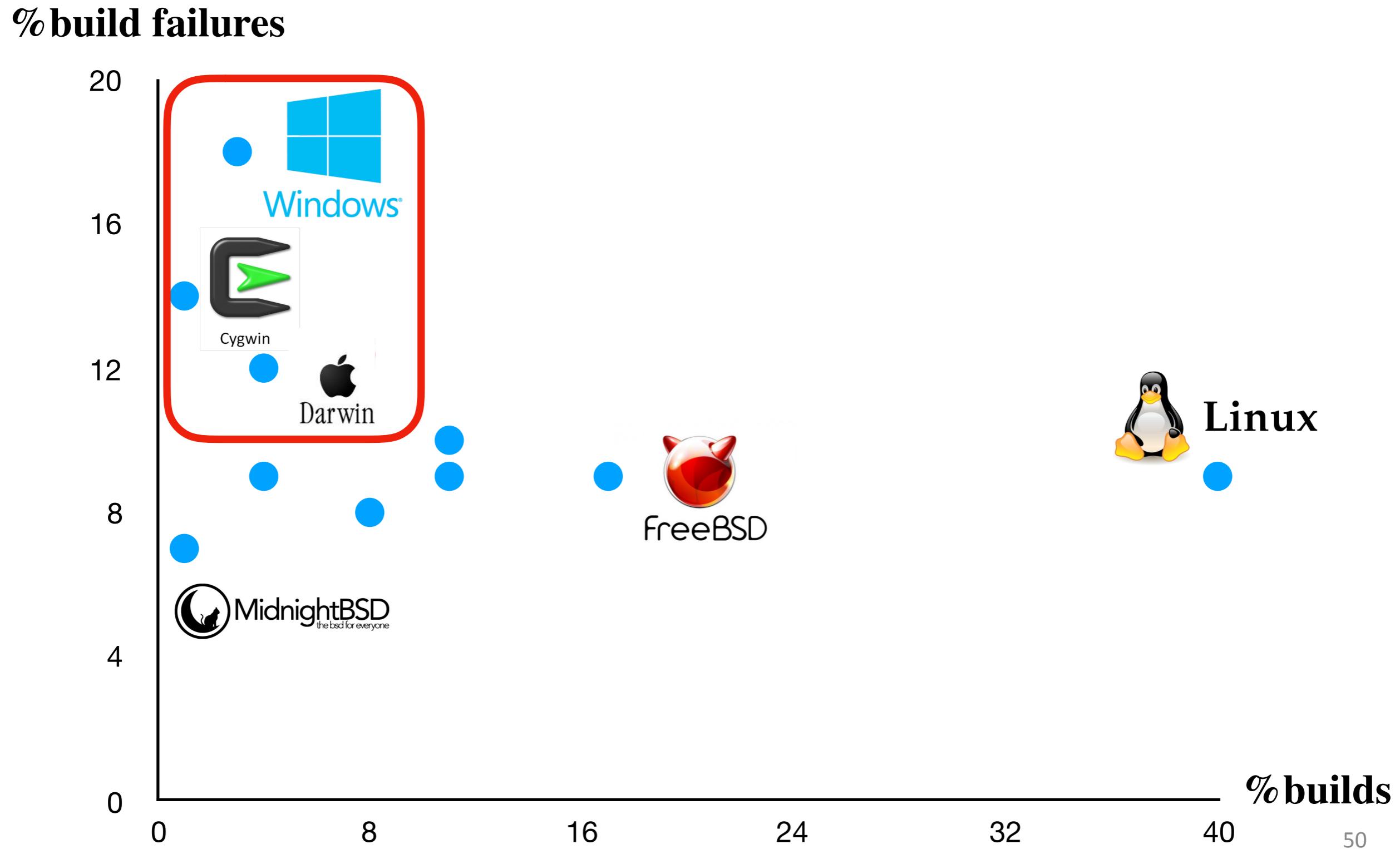
Not Every Environment Yields Equally Reliable Build Results



Not Every Environment Yields Equally Reliable Build Results



Build Results on Some Oses Are Less Reliable Than on Others



How often do Builds Succeed/Fail Consistently across all OSes?

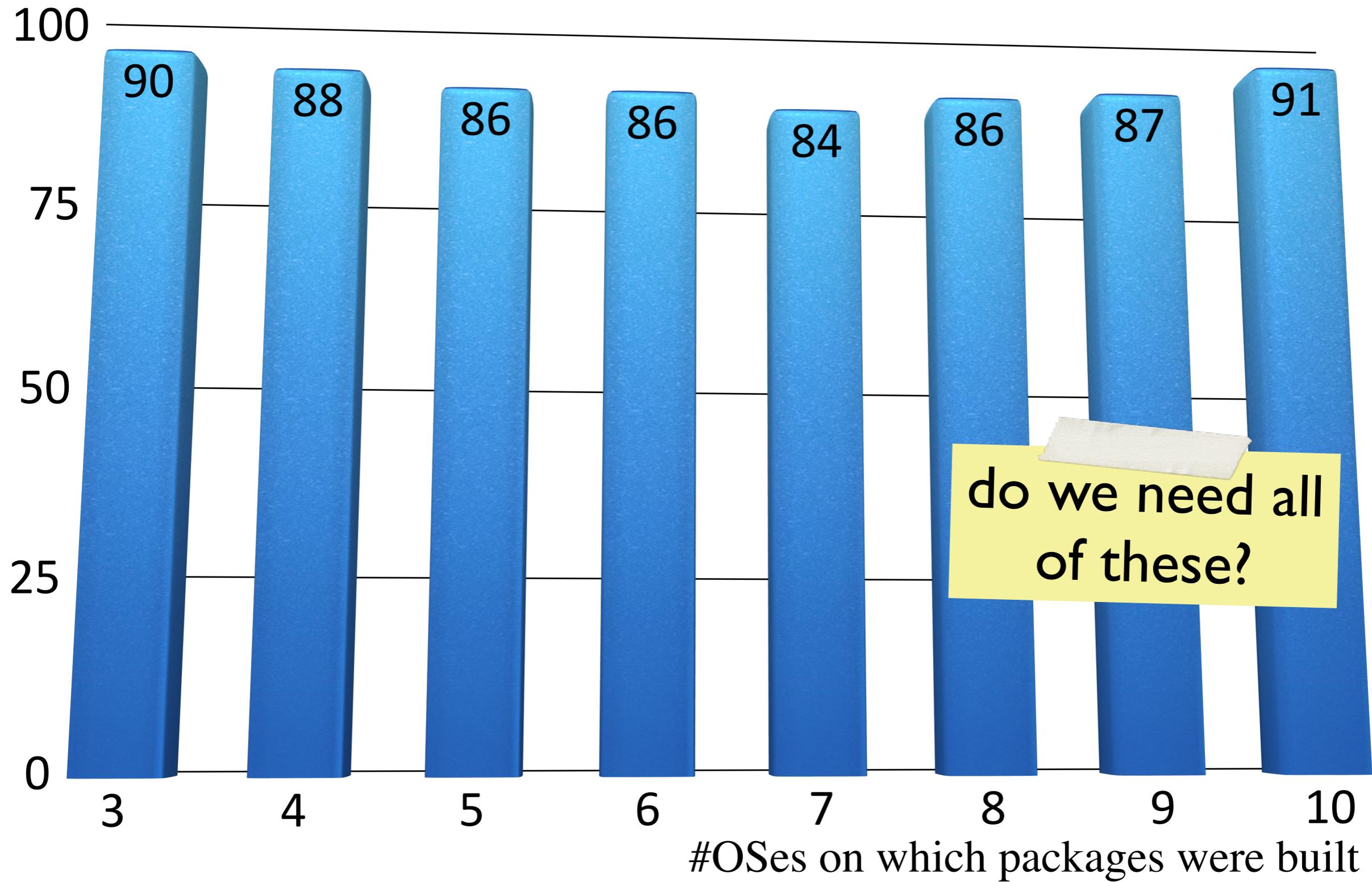


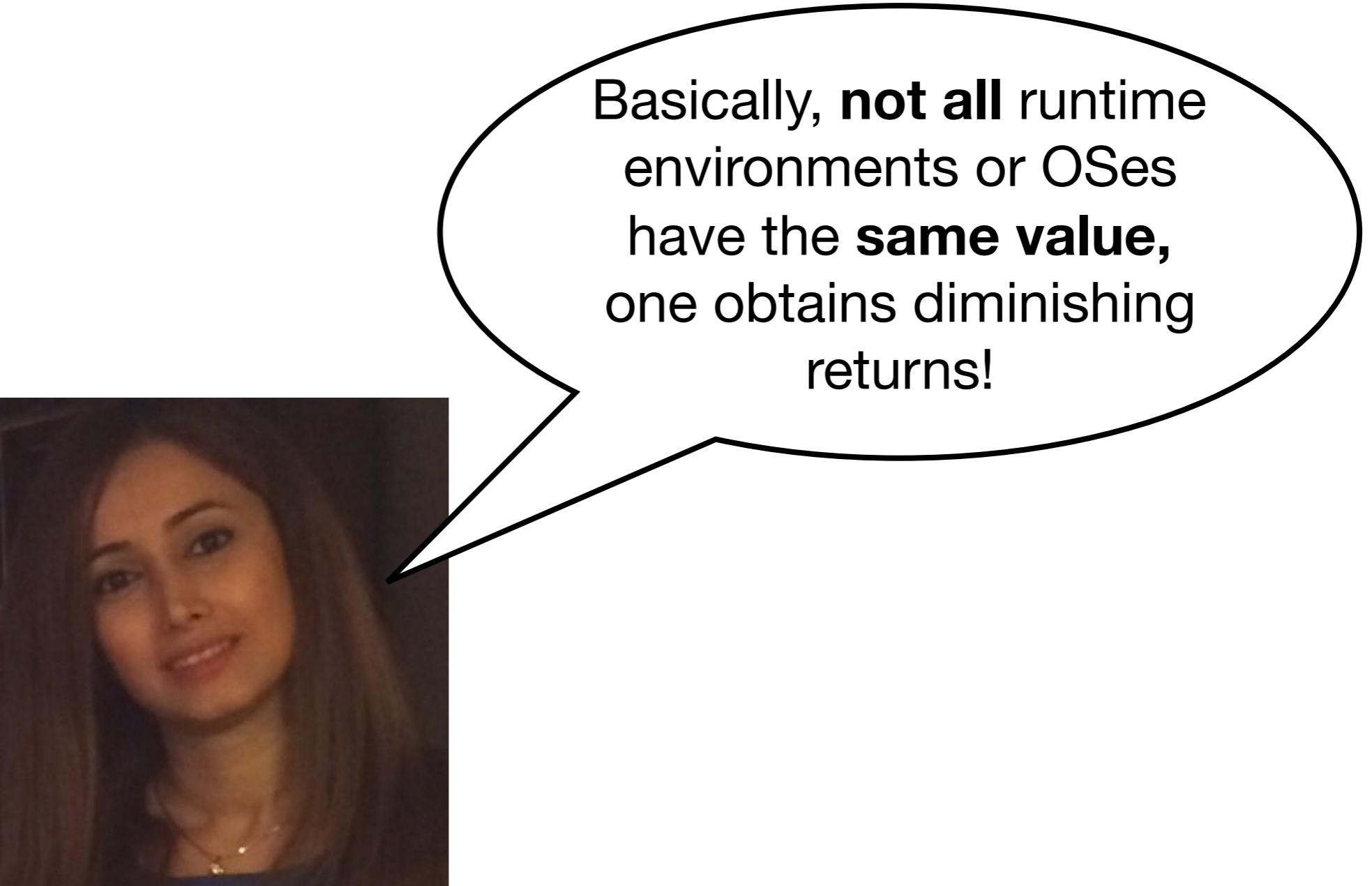


Mac OS



Median of 86.5% of Builds Succeed/ Fail Consistently across All OSes





Basically, **not all** runtime environments or OSes have the **same value**, one obtains diminishing returns!

Act 4: “What You Should Remember from This Talk”



Obviously,
any build provides some
insights...

... and if
your build just broke,
you'll likely need to run all
builds on the bug fix
commit

However, we should Change our Focus

Will this build fail?

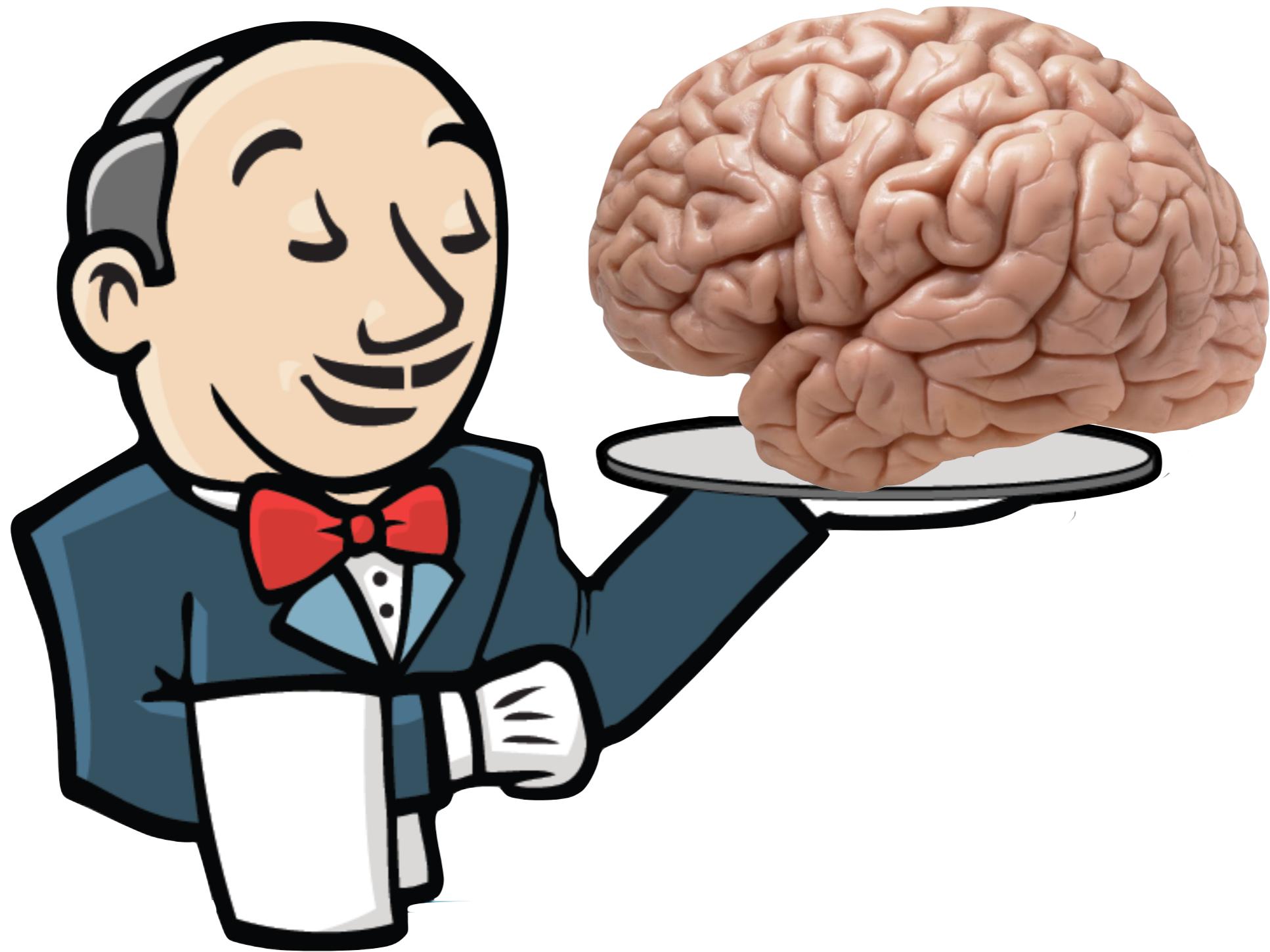
Which tests should be run?

Which files are likely buggy?

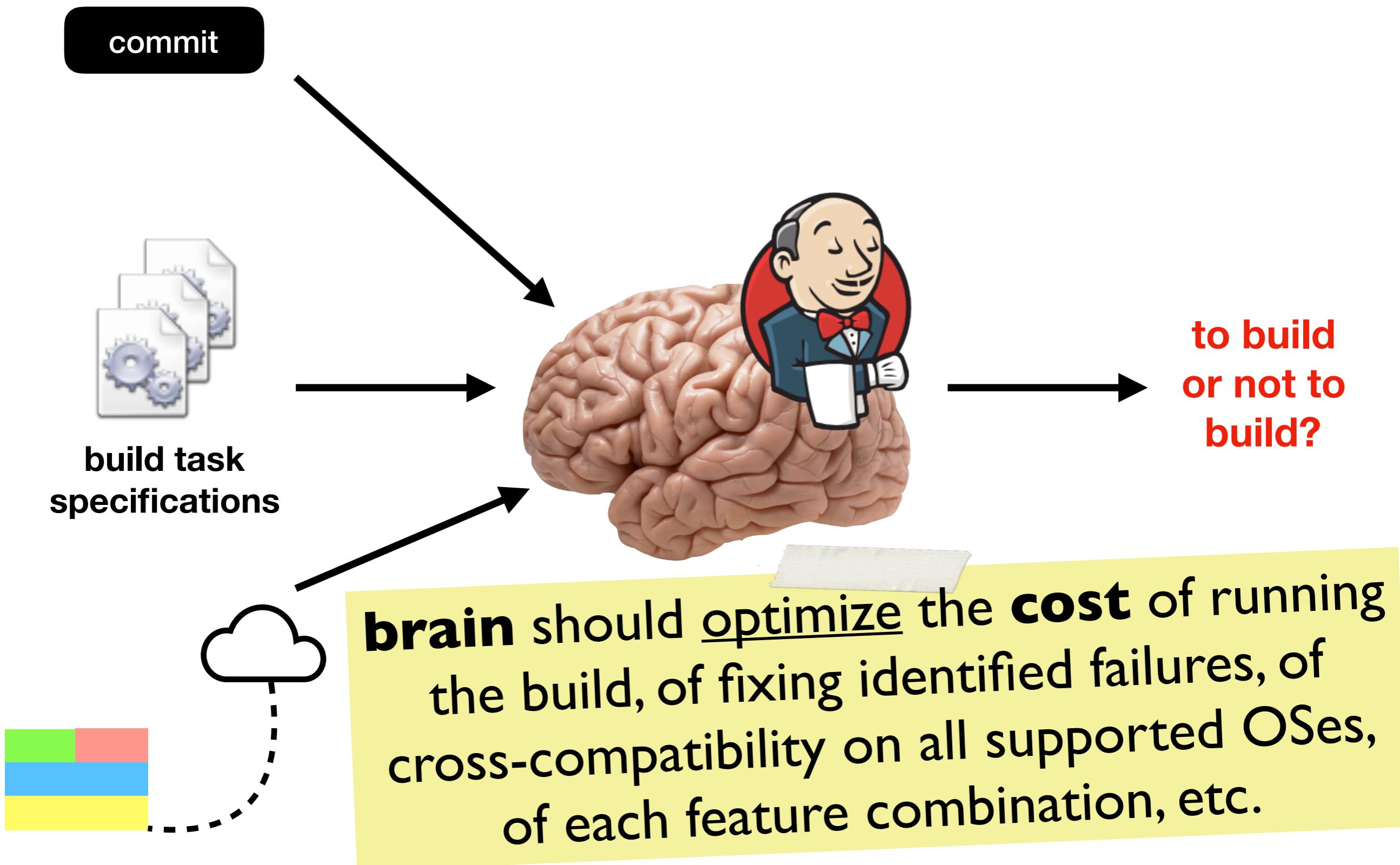
How fast will this build be?

What is the **minimal number of builds** necessary to have sufficient confidence that (1) **the major variants** of the system satisfy their (2) functional and (3) non-functional requirements across (4) **the major targeted environments**?

In other words ...

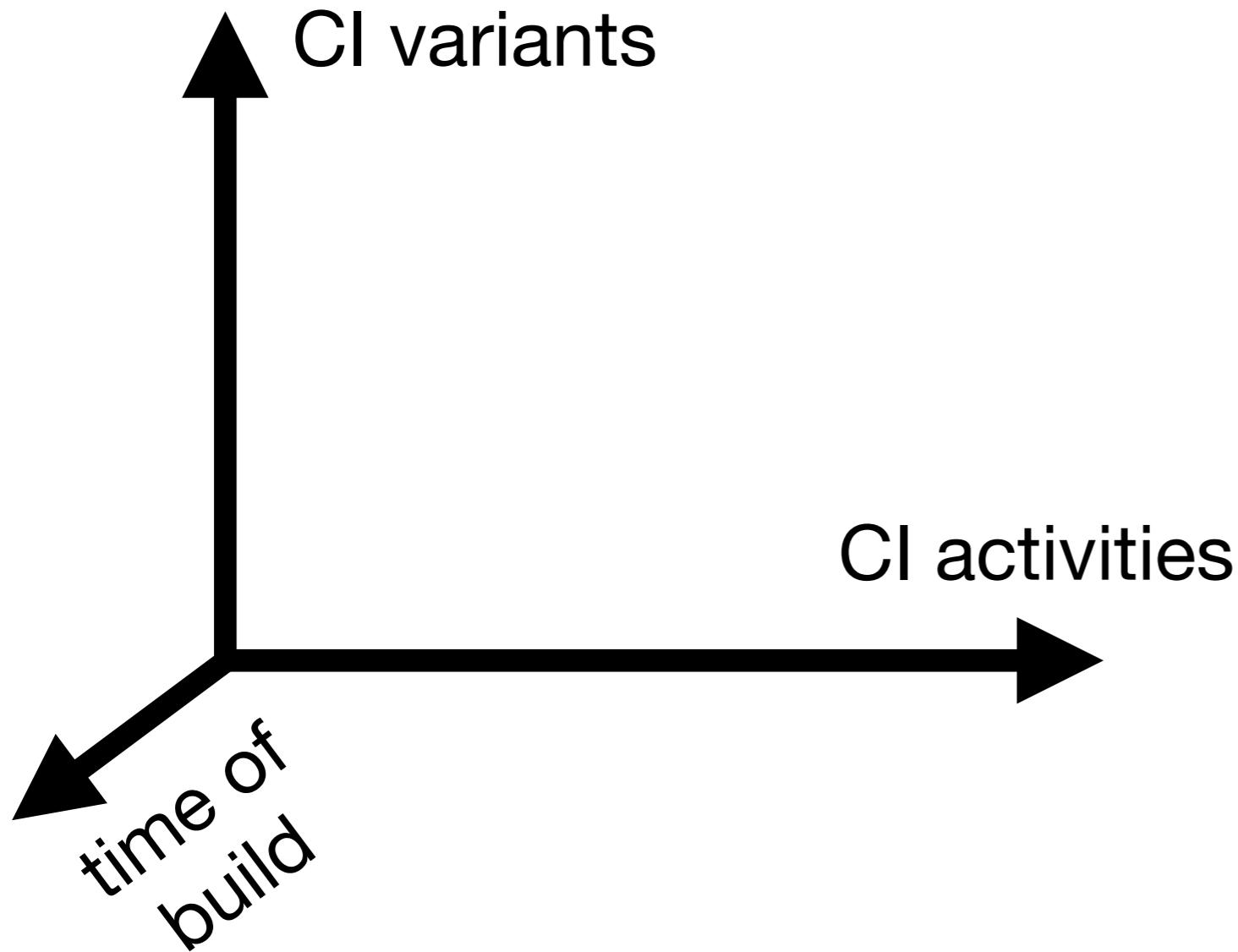


Adding Brains to CI



Finally, Support Manual

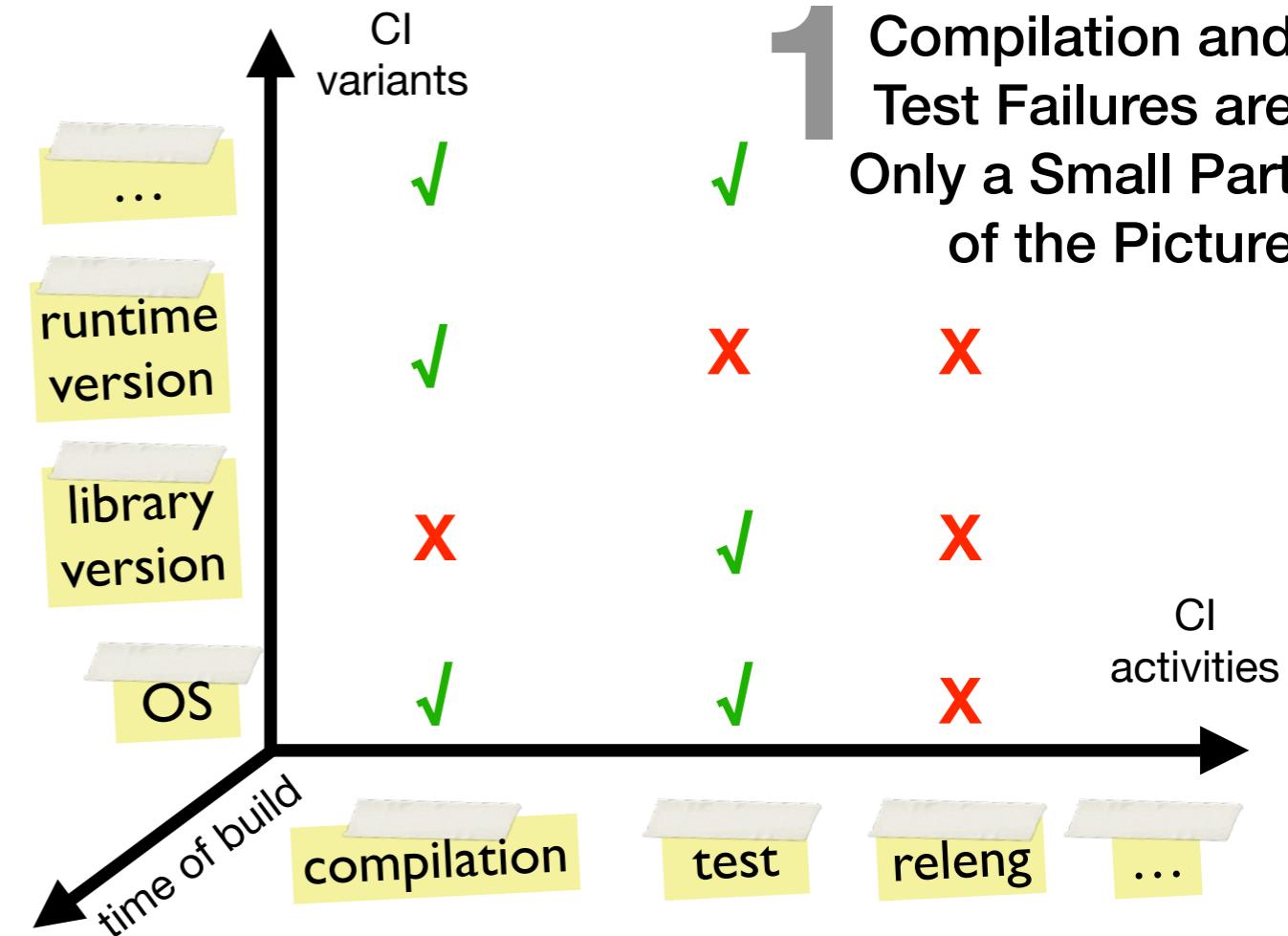
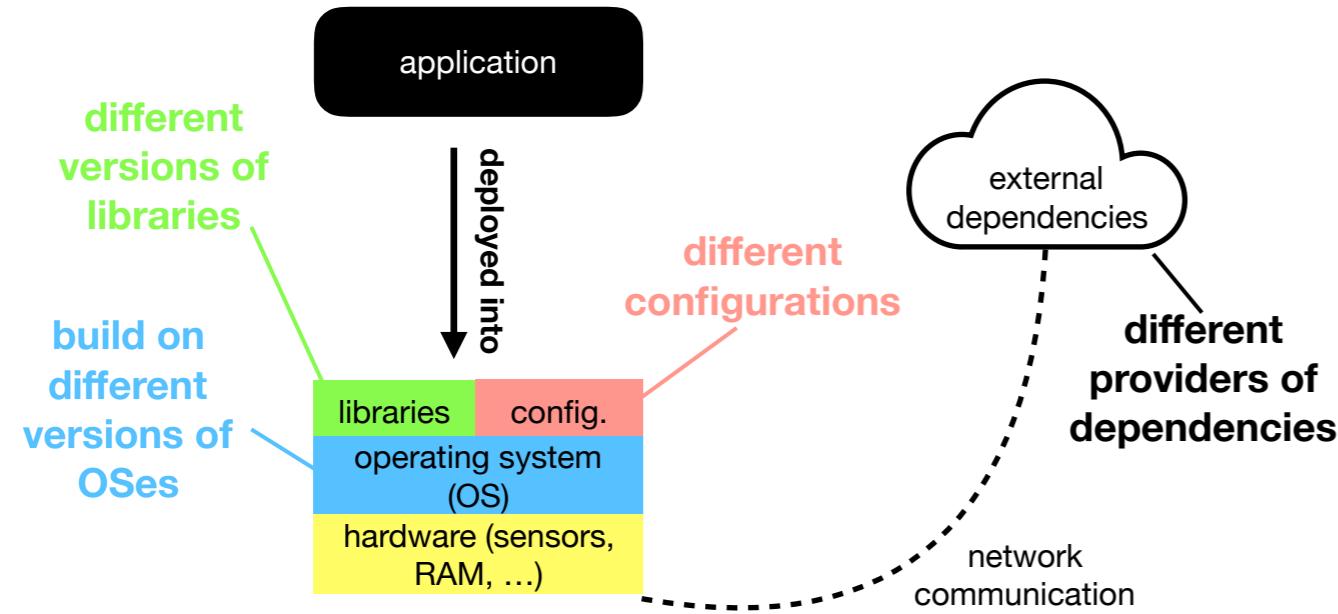
Interpretation of Build Results



techniques like
coalescing/gating, ...

Explosion of Additional CI Builds!

Infrastructure-as-Code
makes it easy to specify different variants of environment



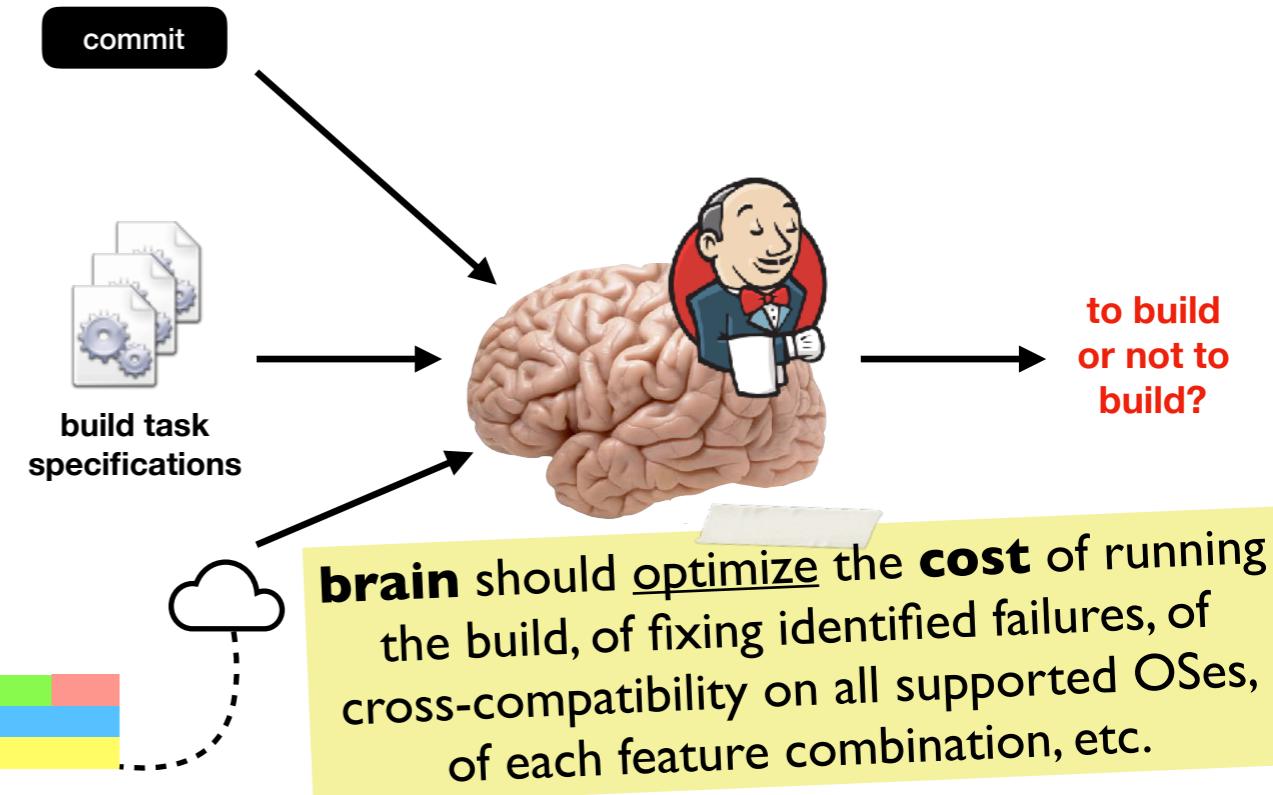
26

3 Builds = \$\$\$

- assuming that:
- only AWS build costs money (incorrect => cost estimation is **lower bound**)
- two cheapest AWS regions are used for daily production load, and a third region on hot-backup
- healthy mix of “OnDemand”, “Reserved” and “Spot” AWS instances is used
- then: the **build cost of one Mozilla commit is USD 26.40**
- having 7,601 commits in 12/2013, the **total monthly cost is USD 201k**



Adding Brains to CI



1 Compilation and Test Failures are Only a Small Part of the Picture