

To Build or Not to Build

The Build Lifecycle of a Commit

Bram Adams

Polytechnique Montreal

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



joint work with ...



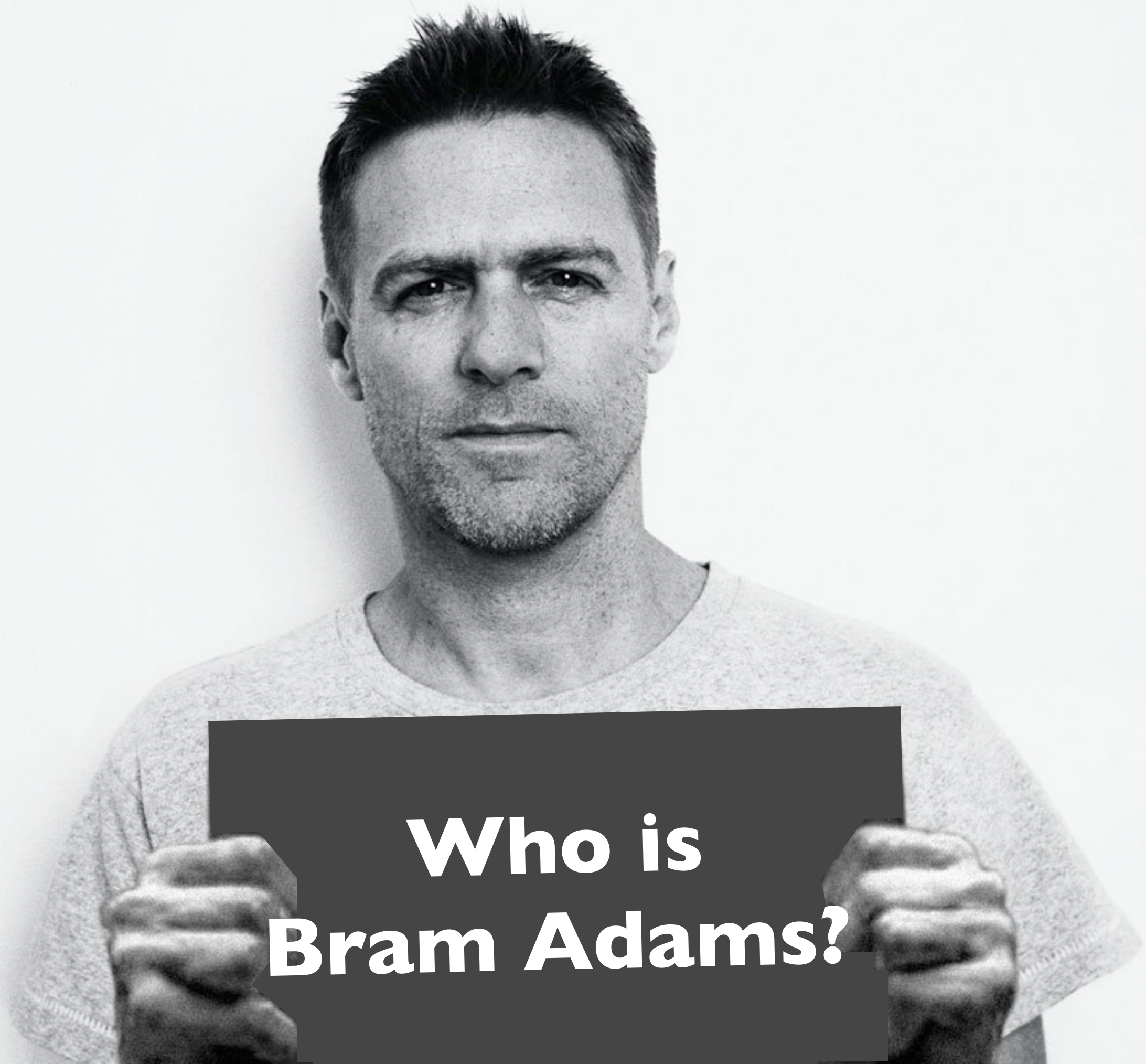
Mahdis Zolfagharinia

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·I·S
I·S

(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

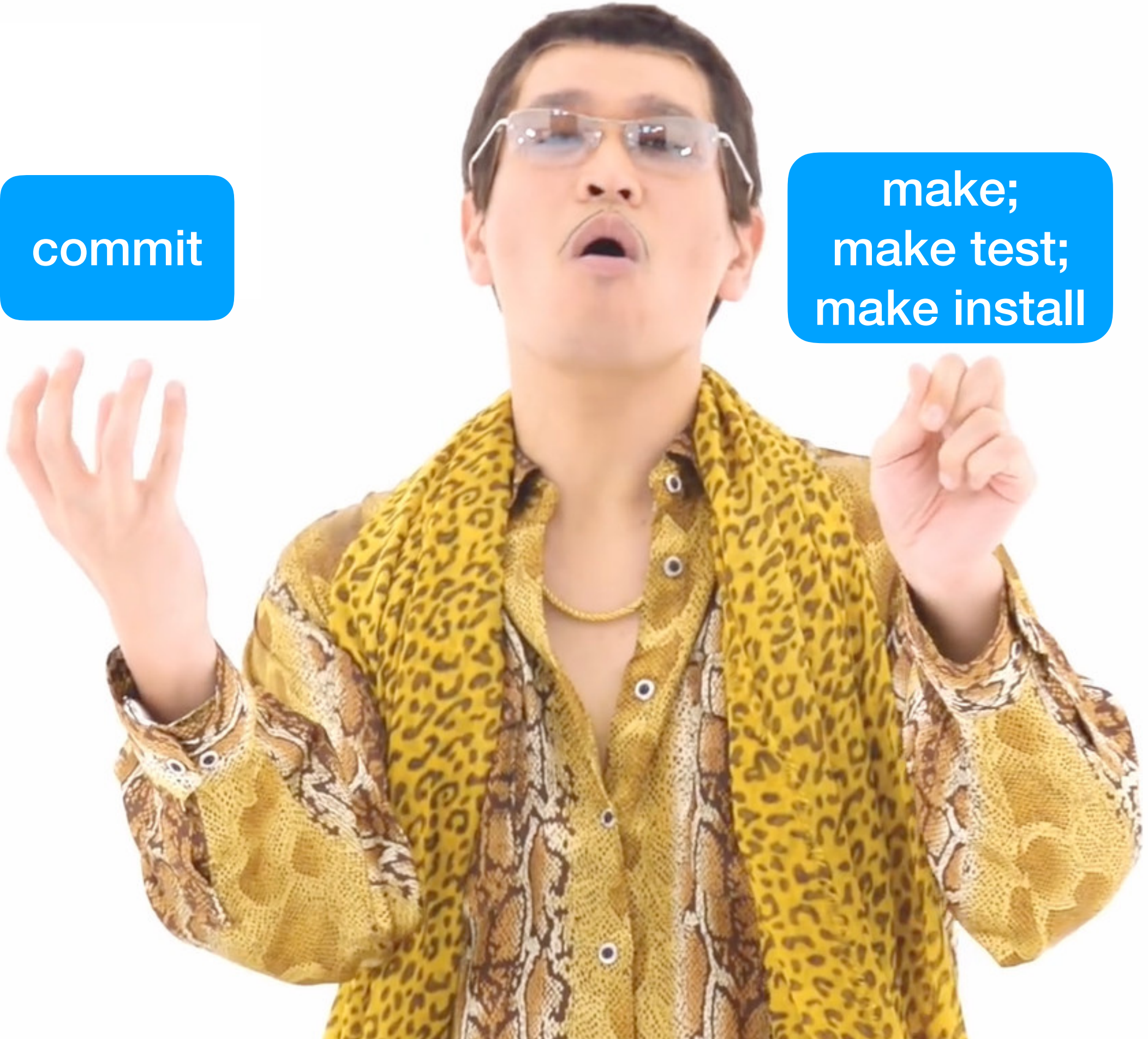
M
C·I·S



Act 1: “Dawn of the Build Inflation”



Builds are conceptually
simple!



commit

make;
make test;
make install



build

What does a Build do?

compilation

packaging of app
and its
dependencies

generation of
documentation

signing
executables

internationalization
of an app

generation of
partial update
binaries

tests

static analysis

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

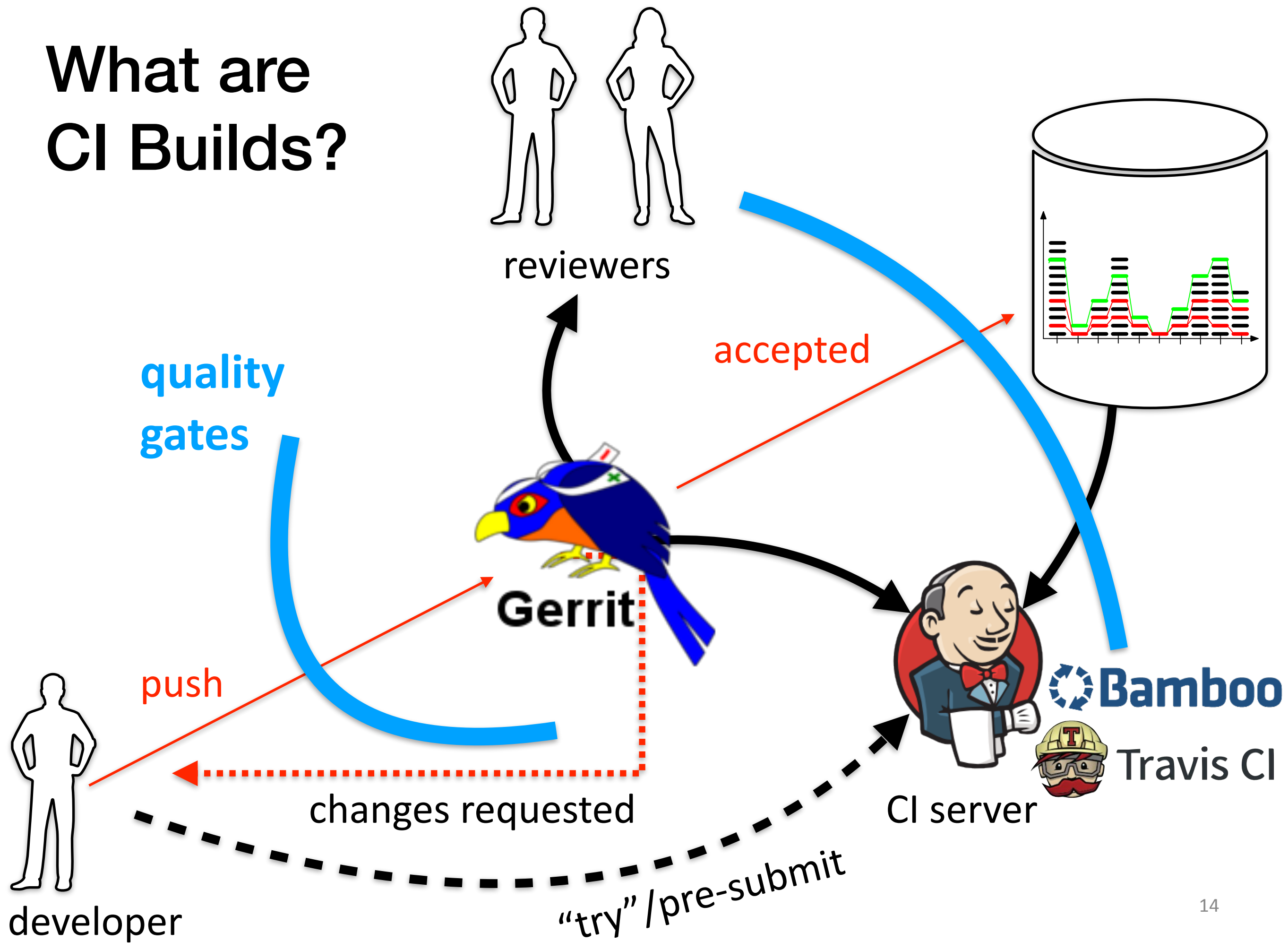
local testing

right before a
release

CI builds!!!



What are CI Builds?



Jenkins

[New Job](#)

[People](#)

[Build History](#)

[Project Relationship](#)

[Check File Fingerprint](#)

[Manage Jenkins](#)

[My Views](#)

Build Queue

No builds in the queue.

Build Executor Status

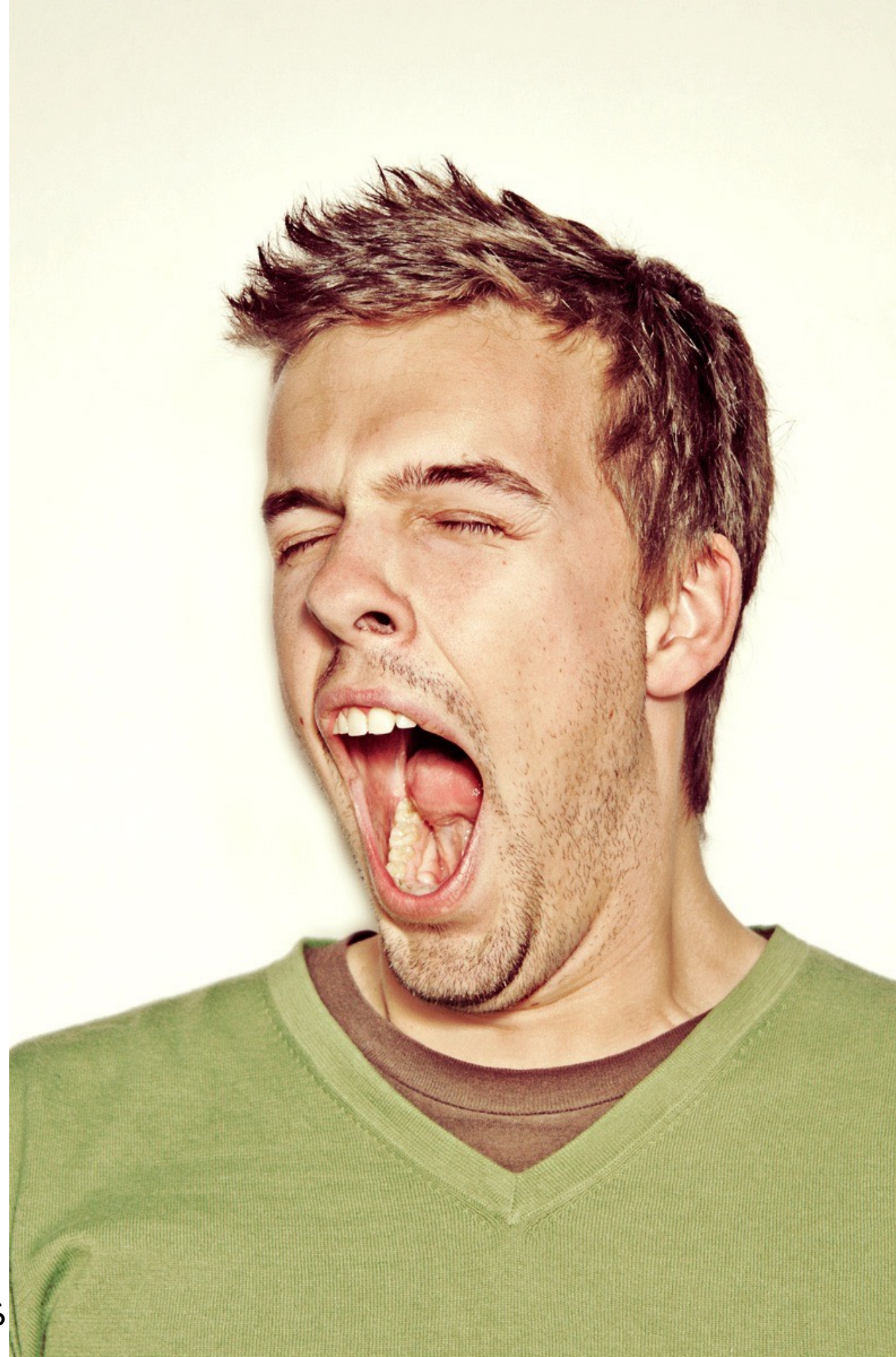
#	Status
1	Idle
2	Idle

last build

build history

All	Software	Name	Last Success	Last Failure	Last Duration
		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-fbdt	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-qaz	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

**We know
all that ...**





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

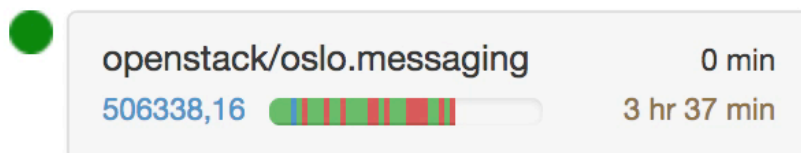
Expand by default: ☐

check

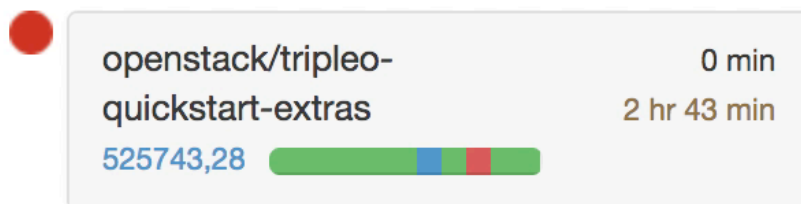
27

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

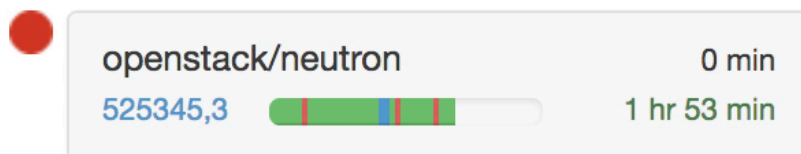
Queue: [openstack/oslo.messaging](#)



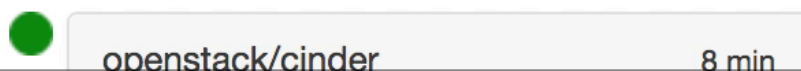
Queue: [openstack/tripleo-quickstart-ext...](#)



Queue: [openstack/neutron](#)



Queue: [openstack/cinder](#)

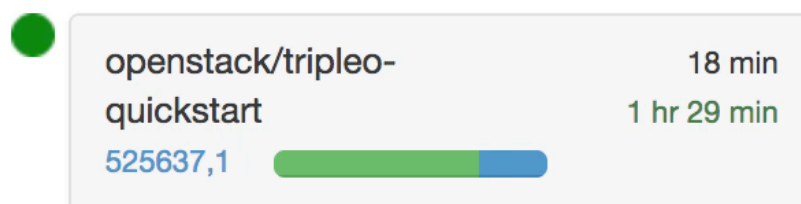


gate

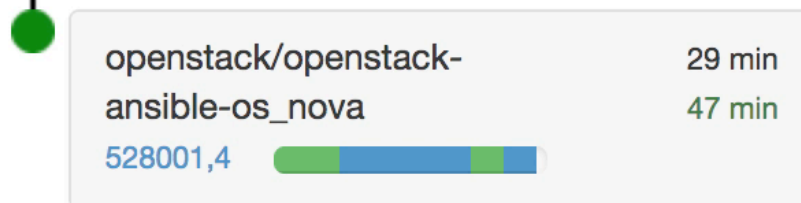
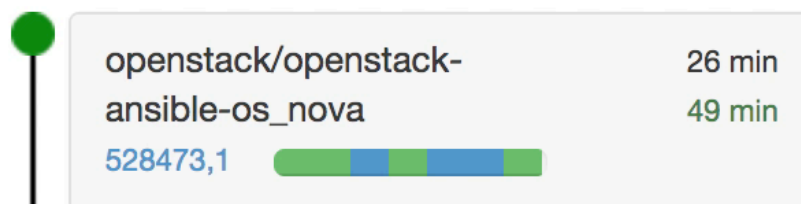
7

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...](#)



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

post

0

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

pre-release

0

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

release

0

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

periodic

0

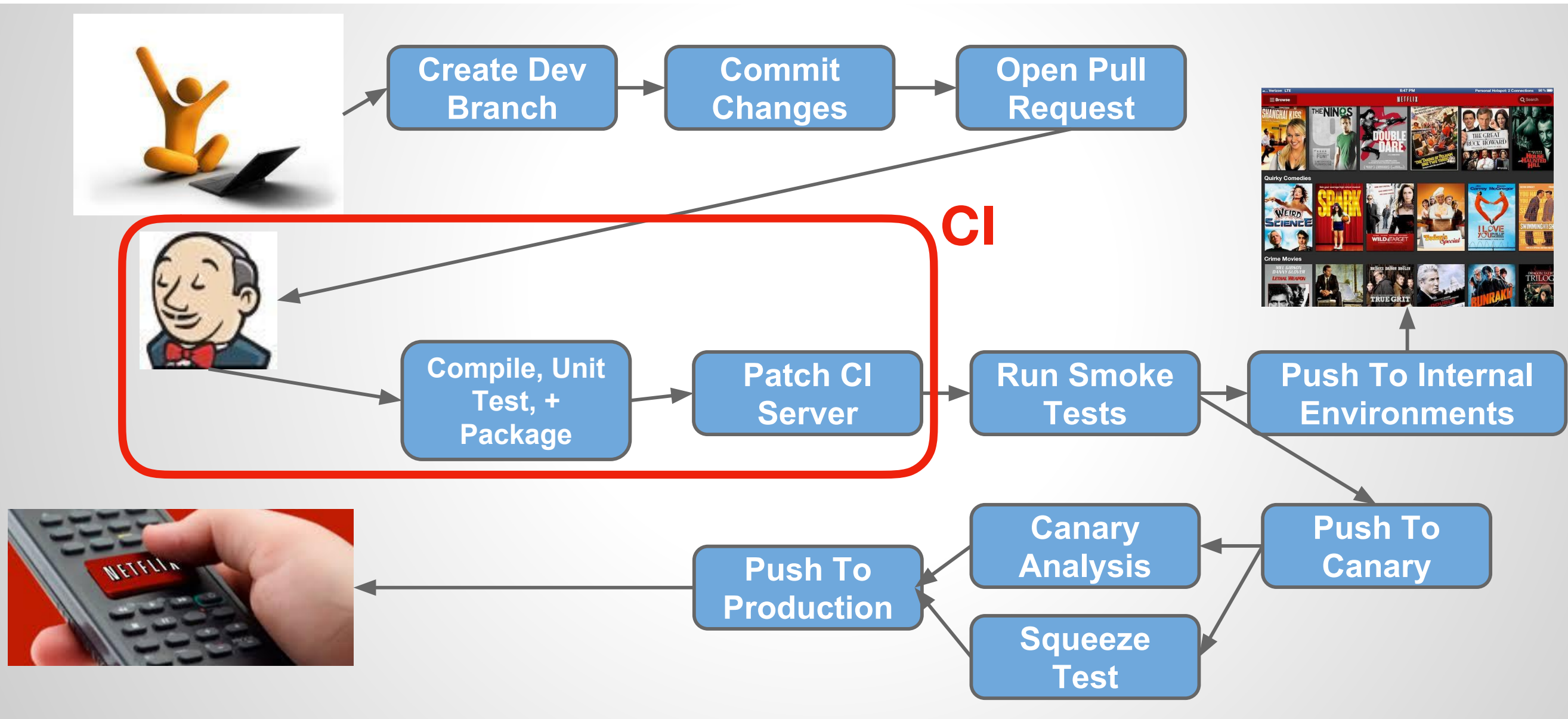
Jobs in this queue are triggered on a timer.

release-post

0

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

NETFLIX



Jenkin's Even has a Pipeline DSL!

✓ declarative-ftw 3 Pipeline Changes Tests Artifacts ↺ ⚙️ ➡️ Logout ✕

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



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

Steps Test On Linux

✓ > Running tests.. — Print Message

✕ ⬇️
<1s



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(featureIsEnabled("use-new-SR-algorithm")){  
    return enhancedSplineReticulation();  
  }else{  
    return oldFashionedSplineReticulation();  
  }  
}
```

feature initially
turned off during
testing until stable

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

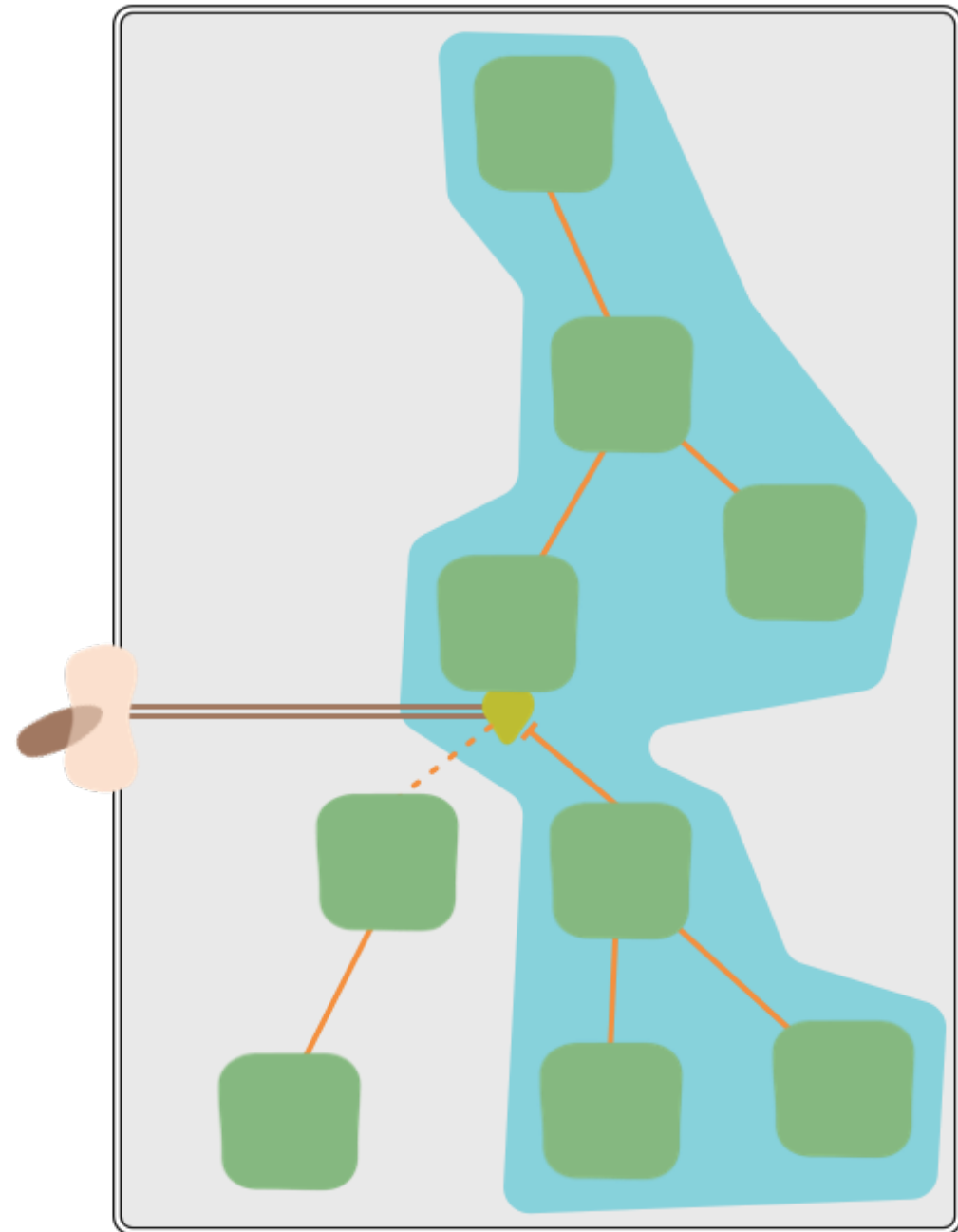
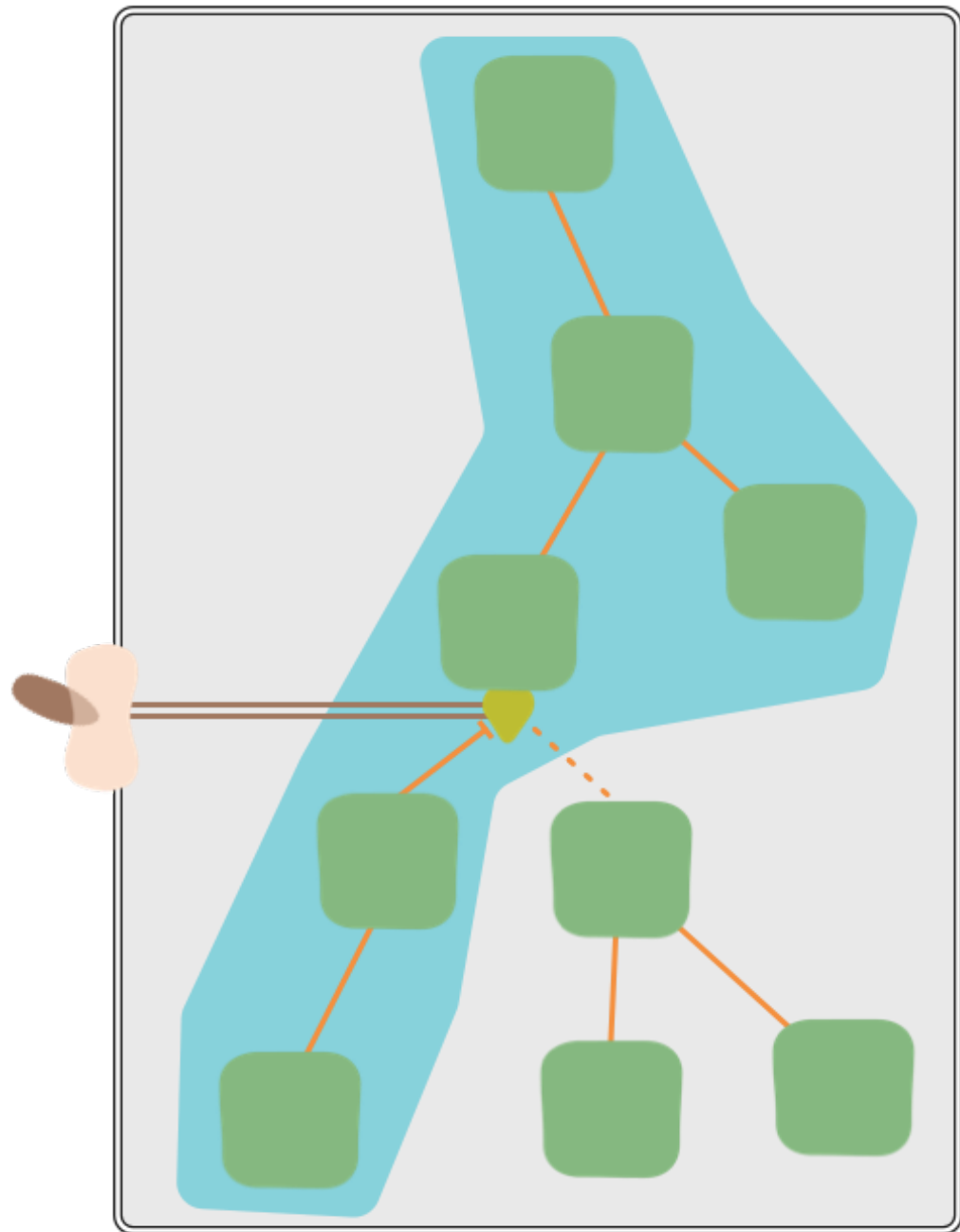
```
function enhancedSplineReticulation(){  
  // TODO: implement better SR algorithm  
}
```

new feature being
developed for
upcoming release

Testing a Toggled Feature



$N \text{ toggles} = 2^N$
variants to test!



3

There are also Different Variants of the Software's Environment

different OSes

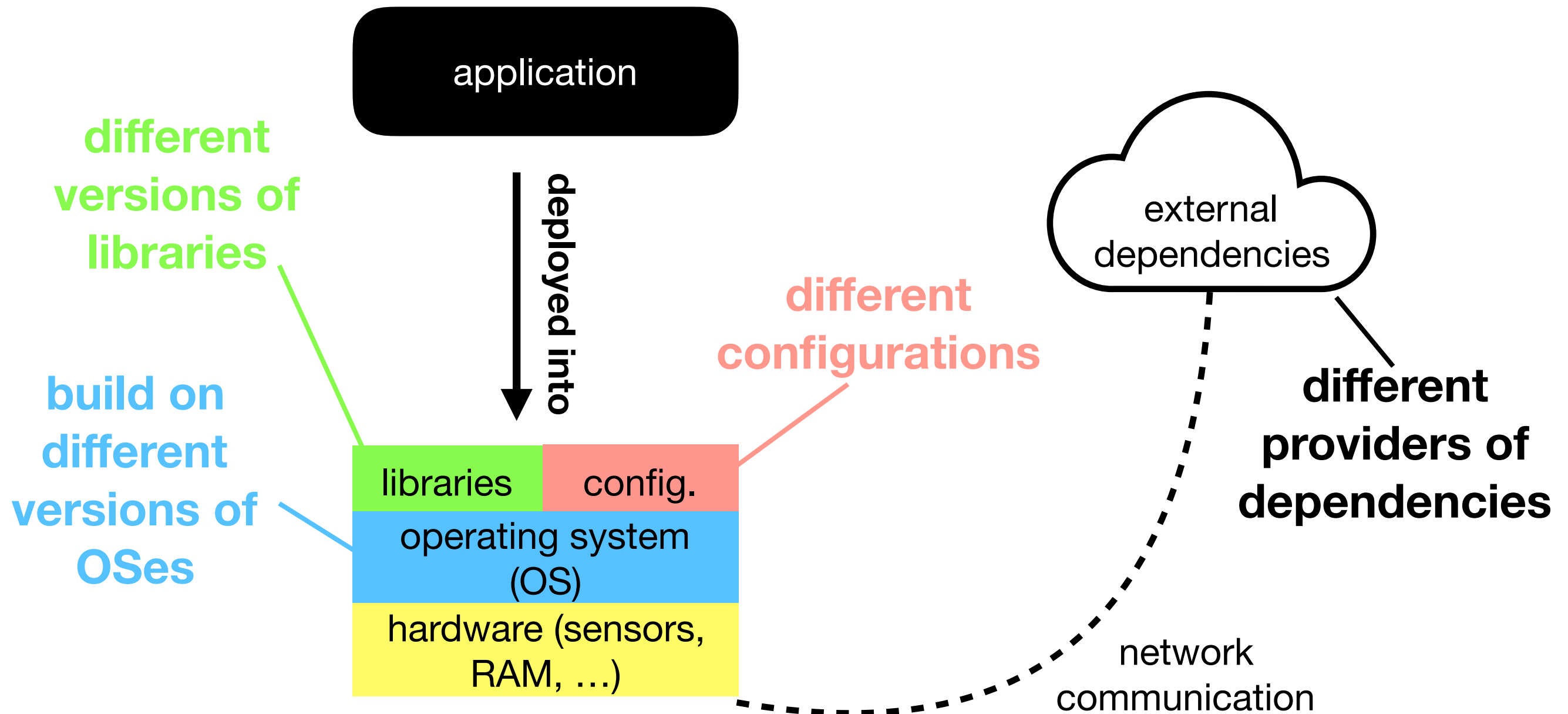
ALL	cygwin	darwin	dragonfly	freebsd	gnukfreebsd	linux	mswin32	netbsd	openbsd	solaris
5.19.3										
5.19.2										
5.18.1										
5.18.0										
5.16.3										
5.16.2										
5.16.1										
5.16.0										
5.14.4										
5.14.3										
5.14.2										
5.14.1										
5.14.0										
5.12.5										
5.12.4										
5.12.3										
5.12.2										
5.12.1										
5.12.0										
5.10.1										
5.10.0										
5.8.9										
5.8.8										

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



mozilla-central



13 unclassified



(+)



Filter platforms & jobs

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

99% - 5 in progress

View Tests



454cafb99225 Merge mozilla-central to mozilla-inb
7cb064f0d25e Merge autoland to mozilla-central r=
29c2618abb00 No bug, Automated blocklist update
be7679eee4ab No bug, Automated HPKP preload l
625f7e9eb46b No bug, Automated HSTS preload l
376a9968d450 Bug 1425217 - Remove Encoding::l
52da1c601408 Bug 1377007 - Tweak BoyerMooreF
0502ddaaf680 Bug 1377007 - autospider.py,runcp
d1d7c3af8ee9 Bug 1377007 - Tests: Parsing ECM
27a84140b47d Bug 1377007 - JS shell bindings for
1a7b0410b795 Bug 1377007 - Implementing basic
bdbaccfb070e Bug 1377007 - GC for binjs-ref pars
70ae74a3a1b7 Bug 1377007 - Implementation of t
f1eeaf08ba77 Bug 1377007 - Introducing BinField
08d9f4f1fd6e Bug 1377007 - Enable/disable build
ac32aa0058fe Bug 1425170. P2 - remove unused
26b2e4c28172 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	 <code></></code> Ruby: 1.8.7	 GEM=railties	 2 min 24 sec
✓ # 43033.2	 <code></></code> Ruby: 1.9.2	 GEM=railties	 25 min 37 sec
✓ # 43033.3	 <code></></code> Ruby: 1.9.3	 GEM=railties	 21 min 39 sec
✓ # 43033.4	 <code></></code> Ruby: 2.0.0	 GEM=railties	 13 min 46 sec
✓ # 43033.5	 <code></></code> Ruby: 2.1.8	 GEM=railties	 19 min 8 sec
✓ # 43033.6	 <code></></code> Ruby: 2.2.6	 GEM=railties	 11 min 12 sec
✓ # 43033.7	 <code></></code> Ruby: 2.3.3	 GEM=railties	 11 min 11 sec
✓ # 43033.8	 <code></></code> Ruby: 1.8.7	 GEM=ap,am,amo,ares,as	 7 min 45 sec
✓ # 43033.9	 <code></></code> Ruby: 1.9.2	 GEM=ap,am,amo,ares,as	 9 min 57 sec
✓ # 43033.10	 <code></></code> Ruby: 1.9.3	 GEM=ap,am,amo,ares,as	 8 min 49 sec
✓ # 43033.11	 <code></></code> Ruby: 2.0.0	 GEM=ap,am,amo,ares,as	 6 min 27 sec

Travis CI build environment

*Operating
System*

*Run-time
Environment*

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

#43033 failed

Since MySQL 5.7.3, the column is null.

Ran for 25 min 37 sec

Total time 9 hrs 26 min 35 sec

(cherry picked from commit 9b5232845b9207febd)

Commit e17e

Branch 3-2-st

Yasuo Honda

Build Jobs

43033.1

43033.2

43033.3

43033.4

43033.5

43033.6

43033.7

43033.8

43033.9

43033.10

43033.11

</> Ruby: 1.8.7

</> Ruby: 1.9.2

</> Ruby: 1.9.3

</> Ruby: 2.0.0

</> Ruby: 2.1.8

</> Ruby: 2.2.6

</> Ruby: 2.3.3

</> Ruby: 1.8.7

</> Ruby: 1.9.2

</> Ruby: 1.9.3

</> Ruby: 2.0.0

GEM=railties

GEM=railties

GEM=railties

GEM=railties

GEM=railties

GEM=railties

GEM=railties

GEM=ap,am,amo,ares,as

GEM=ap,am,amo,ares,as

GEM=ap,am,amo,ares,as

GEM=ap,am,amo,ares,as

2 min 24 sec

25 min 37 sec

21 min 39 sec

13 min 46 sec

19 min 8 sec

11 min 12 sec

11 min 11 sec

7 min 45 sec

9 min 57 sec

8 min 49 sec

6 min 27 sec

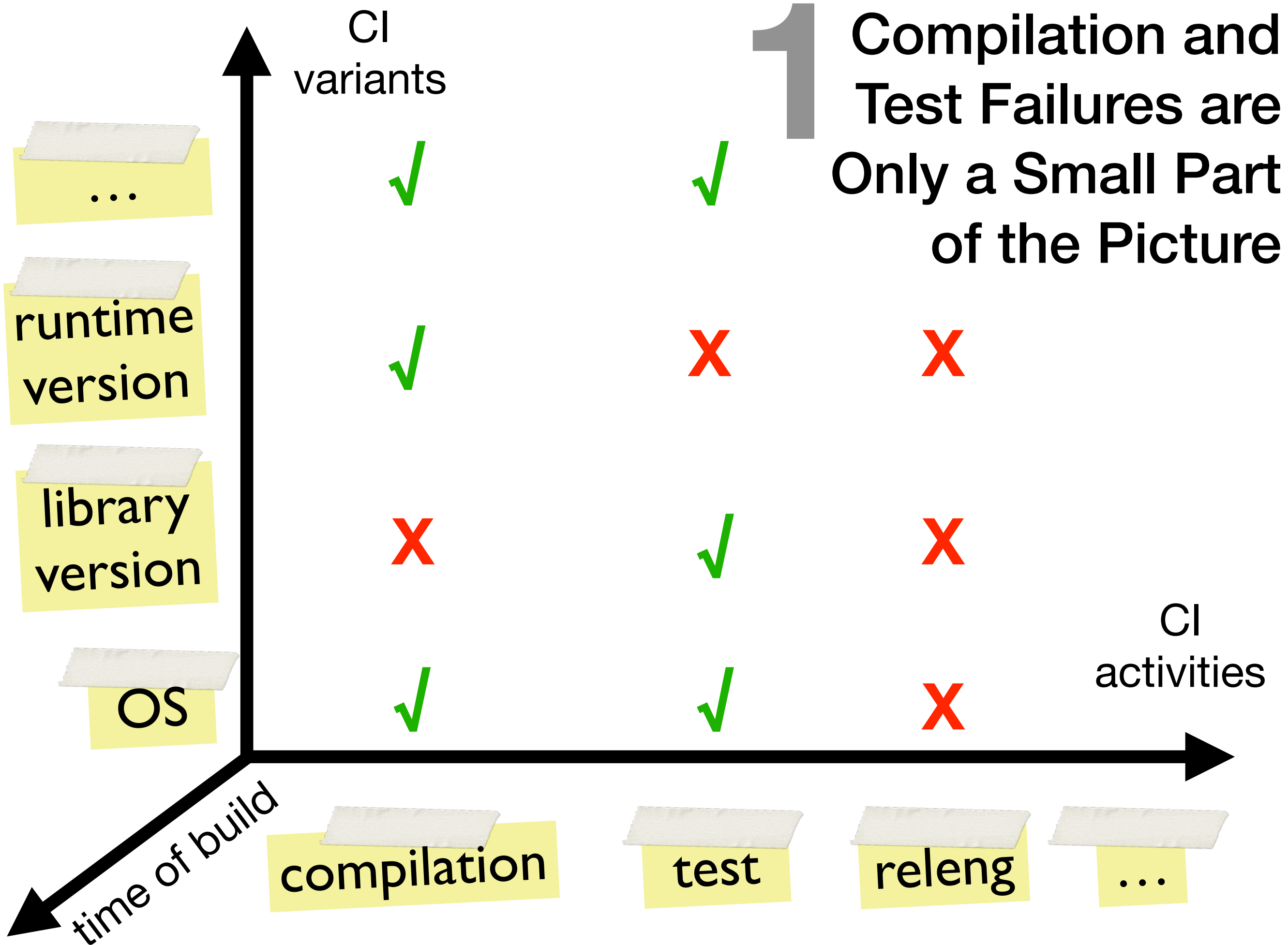


And I guess
that **each failing
build** should probably
be **followed up** by a
code fix, which in turn
needs to be built on
all variants!

**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

build history

development
team

trigger time/
period

type of VCS action

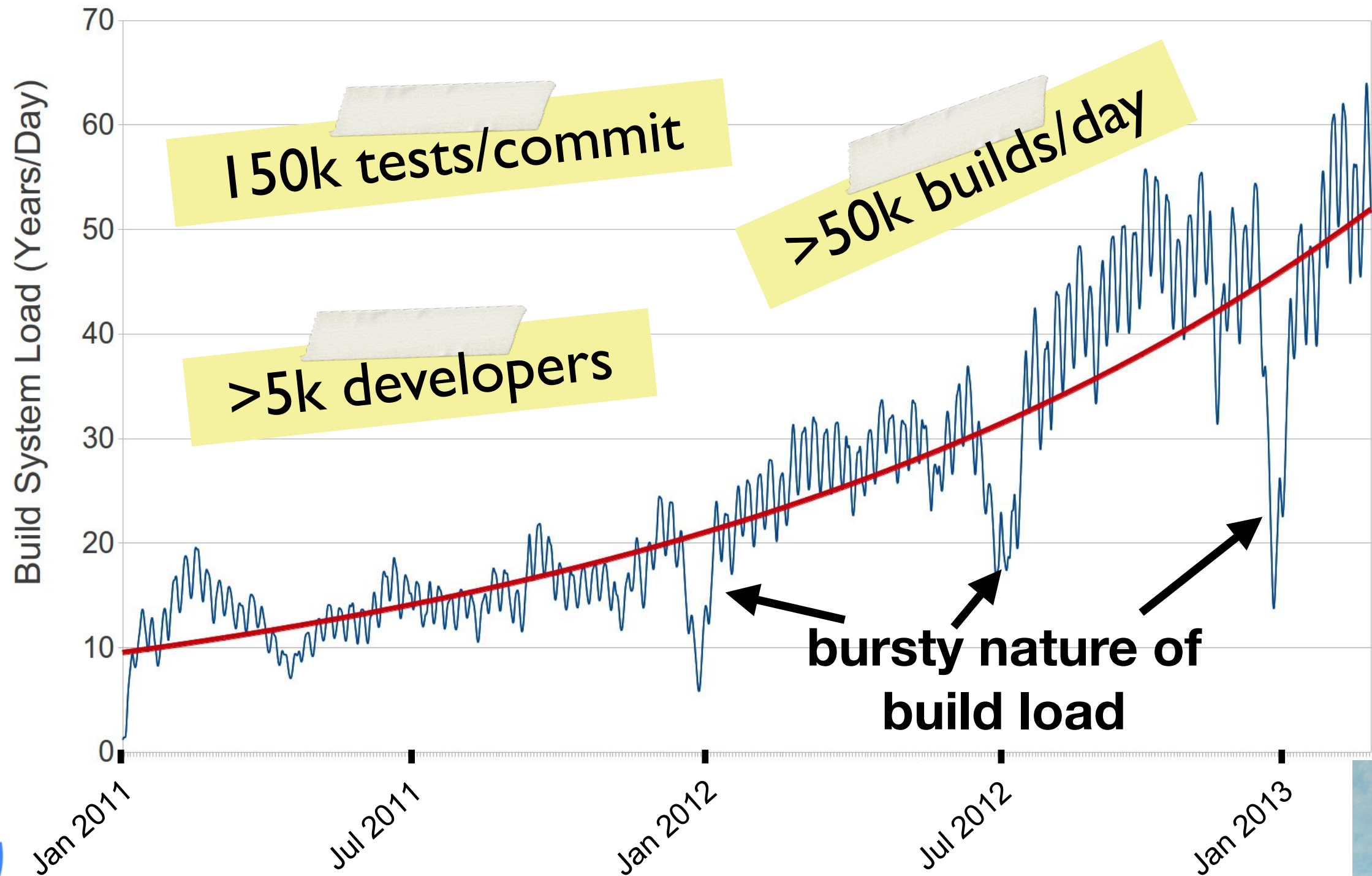
feature vs. bug
fix

trigger author

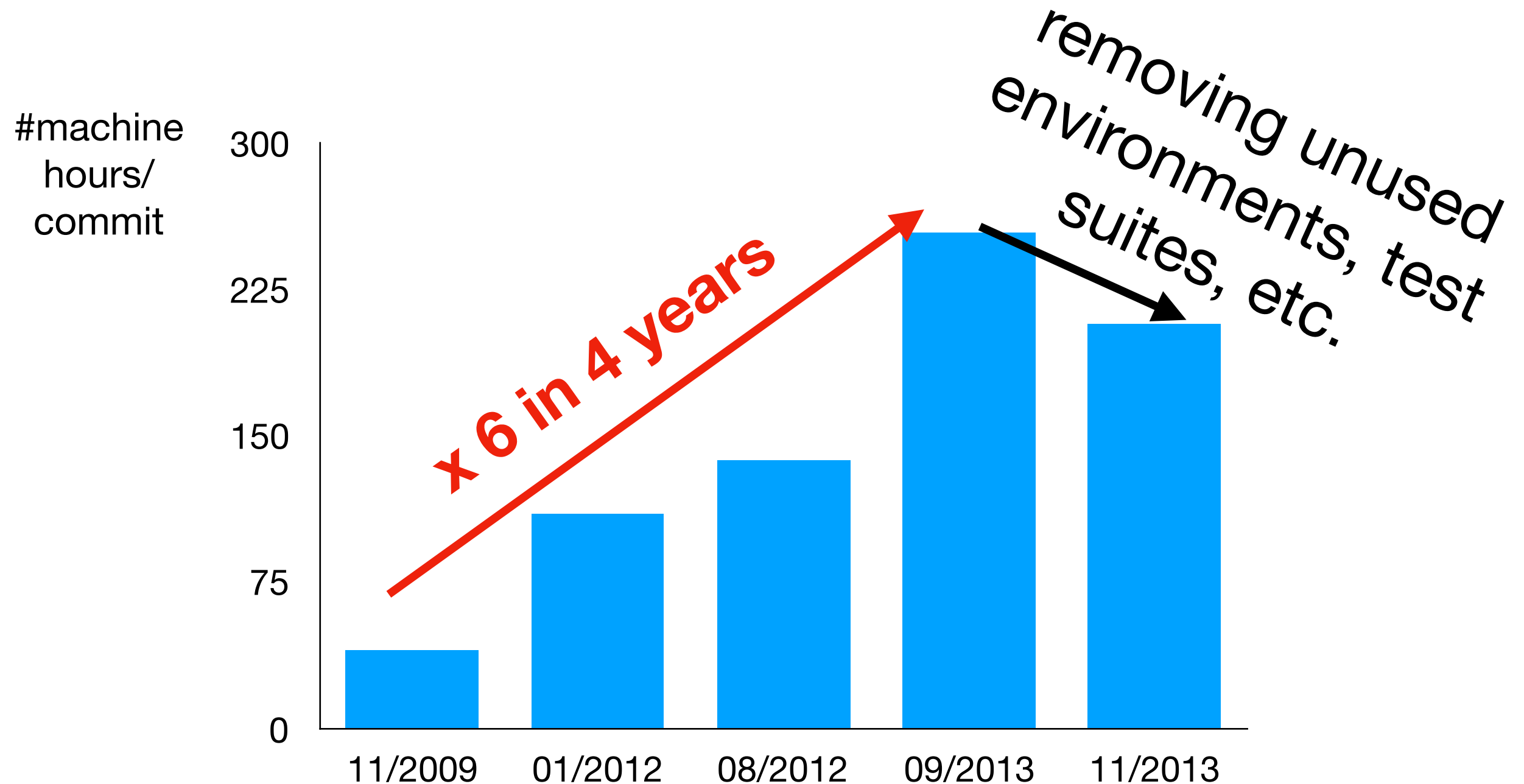
role who
triggered commit

2

More Builds, More Load



Not Only for Web Apps



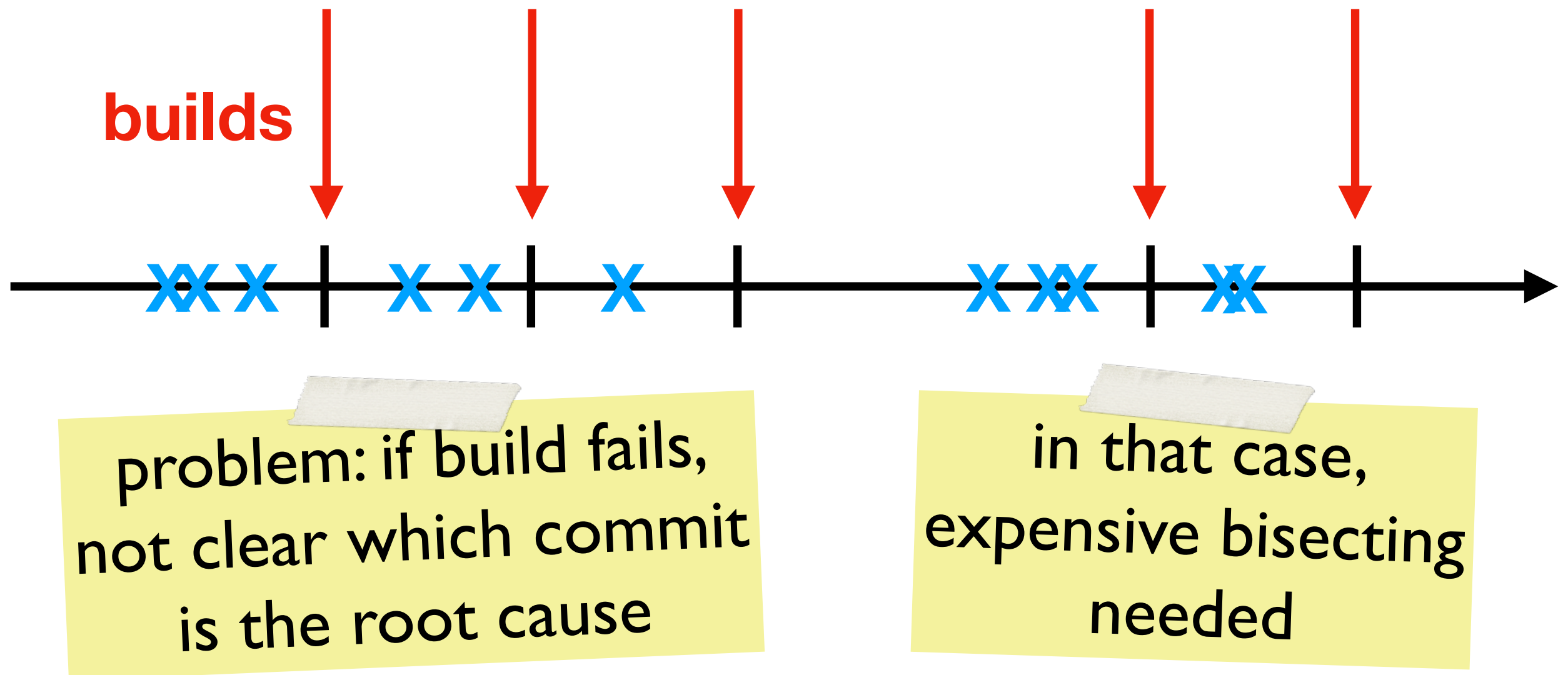
<https://atlee.ca/blog/posts/blog20091102what-happens-when-you-push.html>

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

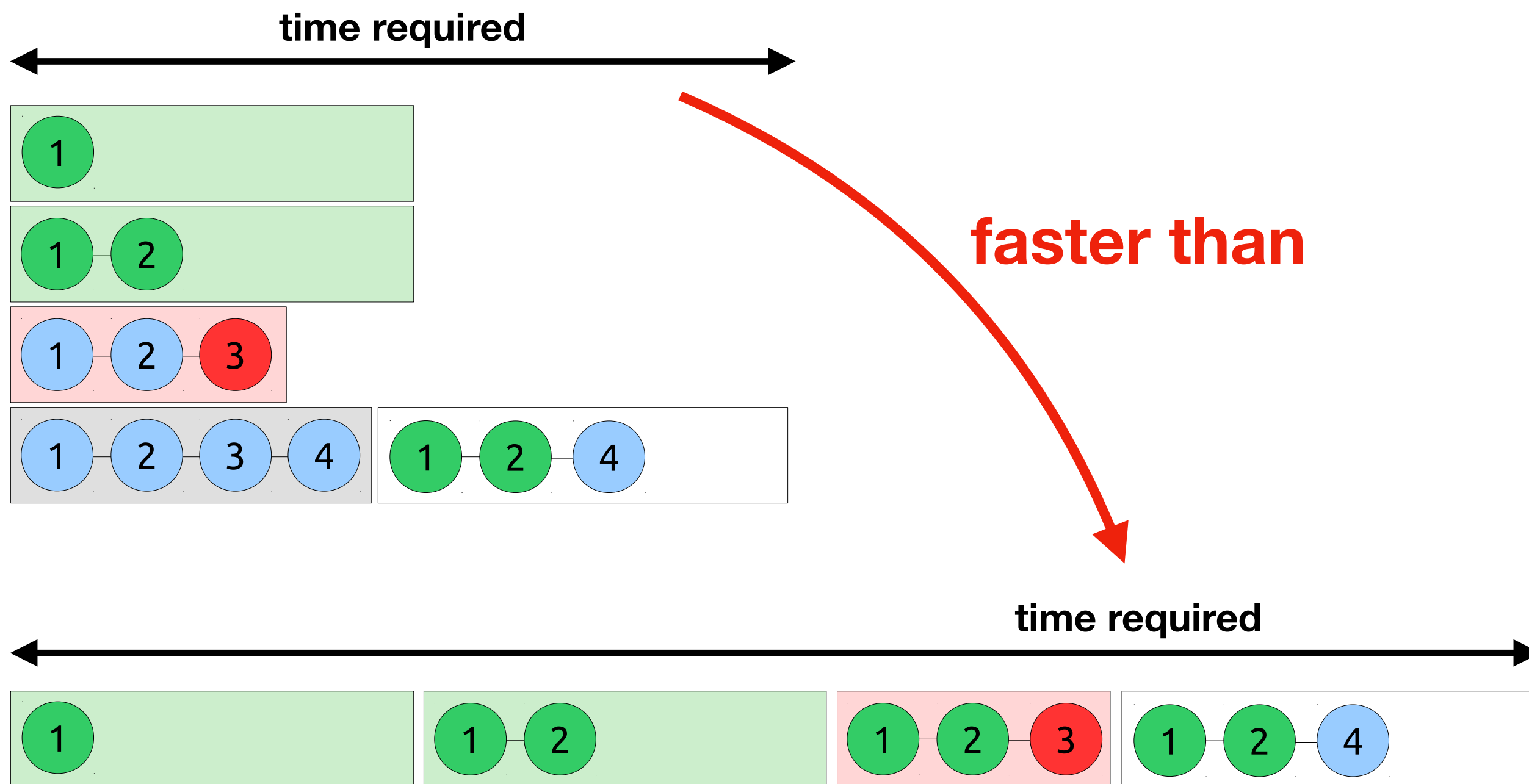
Load Becomes so High that CI Servers cannot Cope Anymore



Mitigation #1: Coalescing Commits Arriving Together



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 = \$\$\$



John O'Duinn

- 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 

✗ **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 $\equiv$ Diminishing Value of Build Result

API unsupported
by Ruby 1.8.7



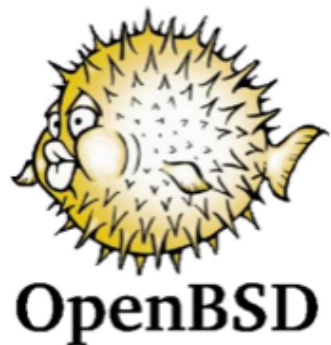
nice to know that 41 variations of library and runtime versions build correctly, but are all 41 **really worth the \$\$\$?**

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



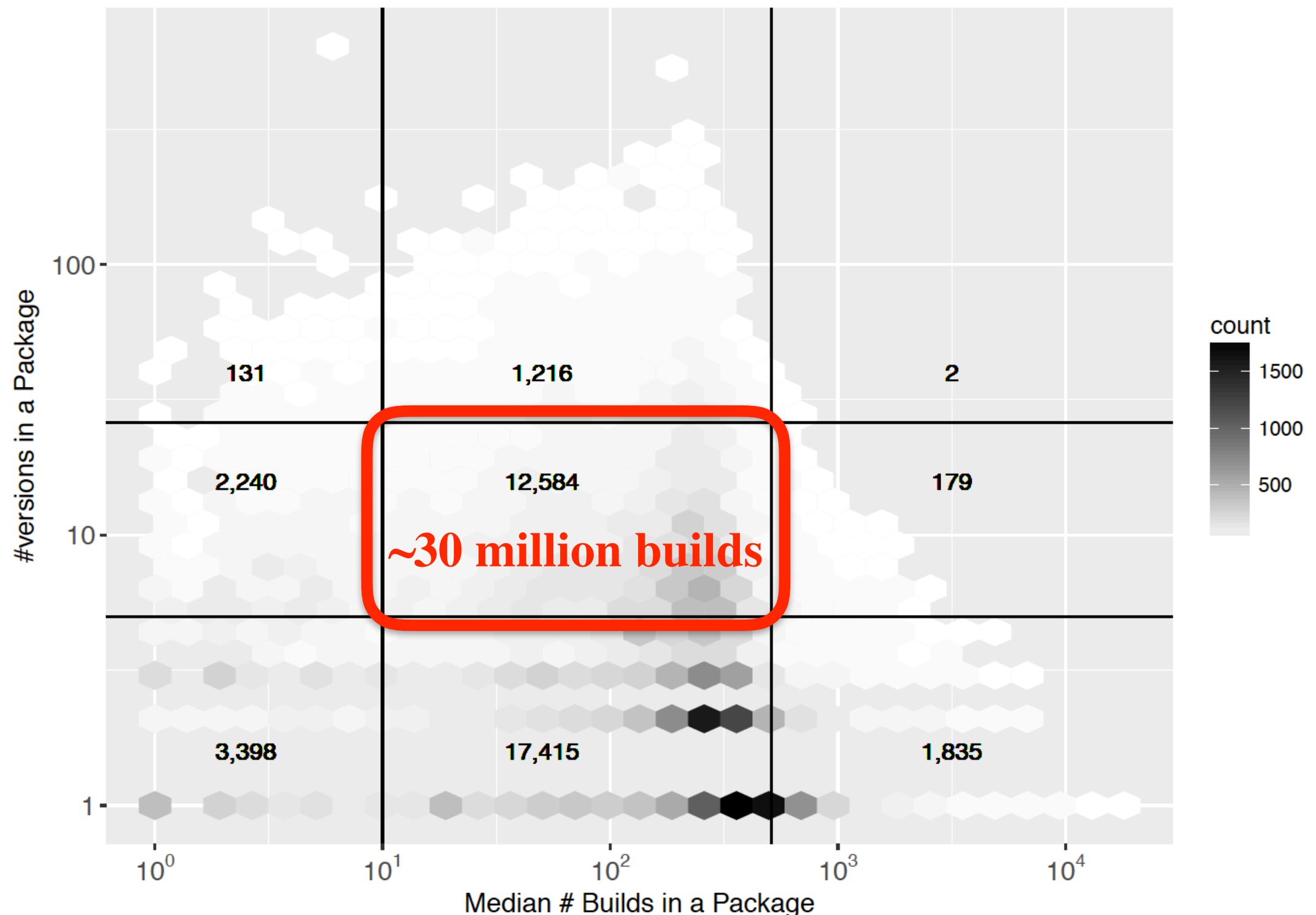
Do not trust build results at face value: an empirical study of 30 million CPAN builds
(Zolfagharinia et al., MSR 2017)

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

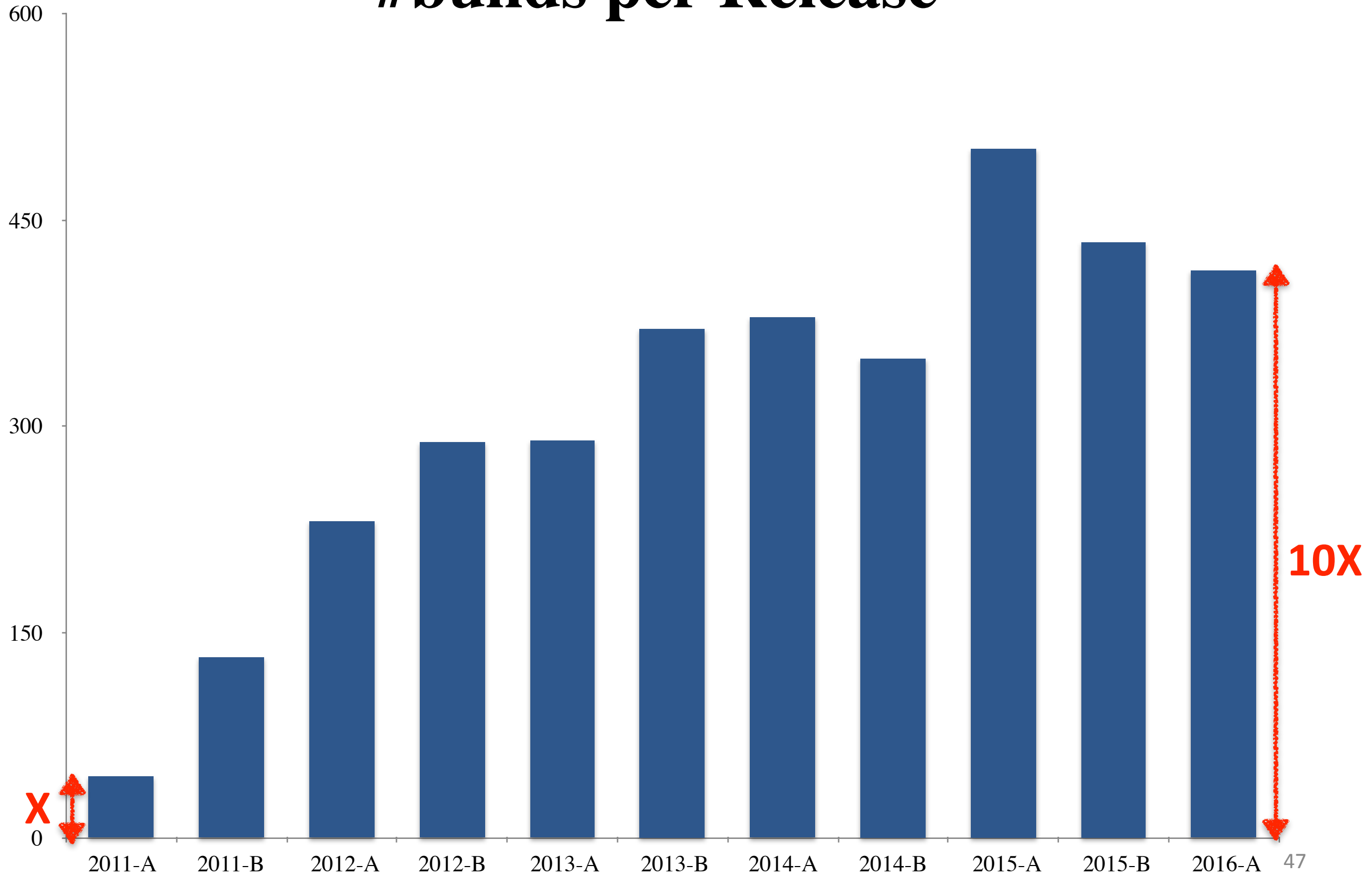


Linux

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

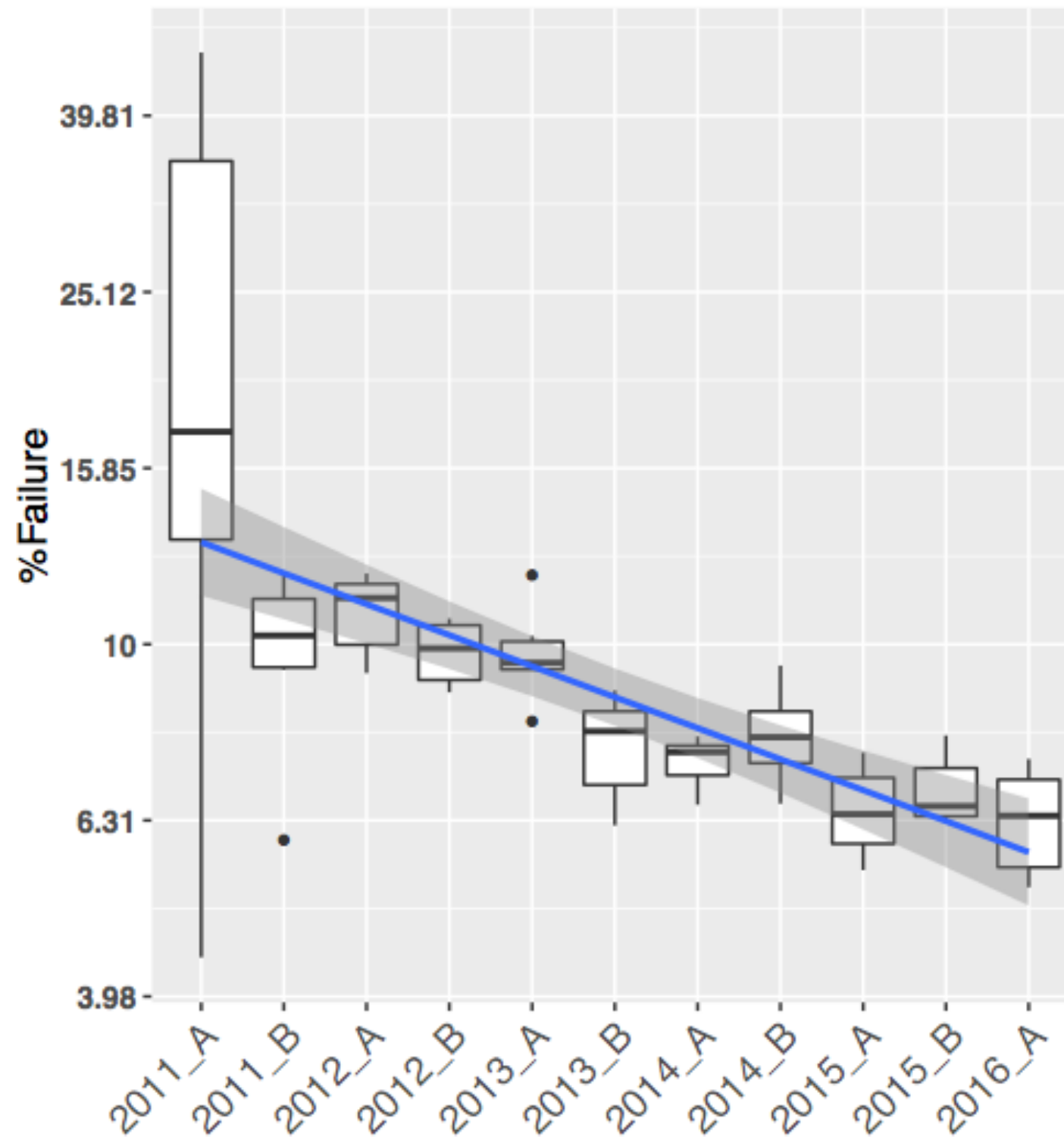


10-fold Increase in Average #builds per Release

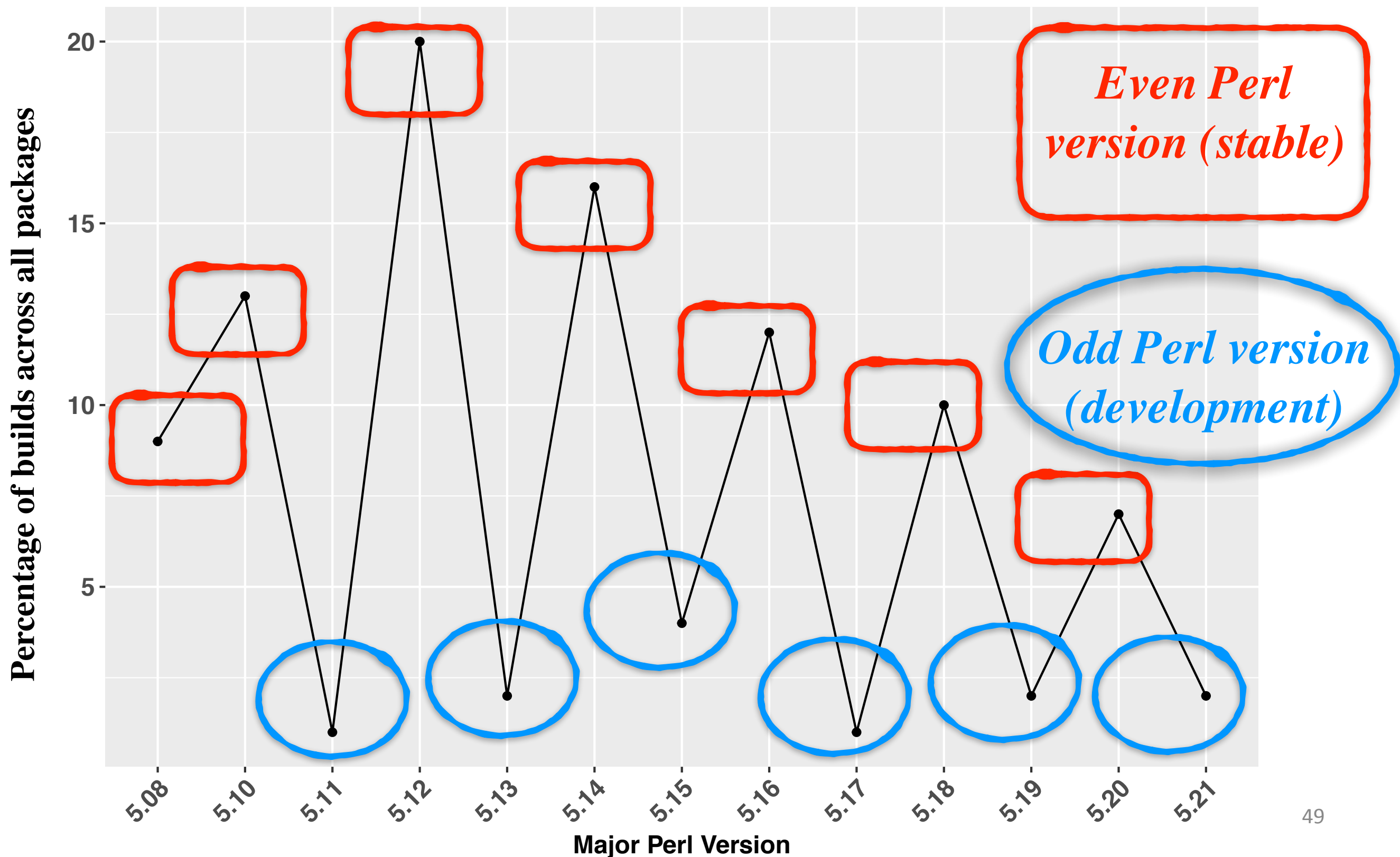


Build Inflation:

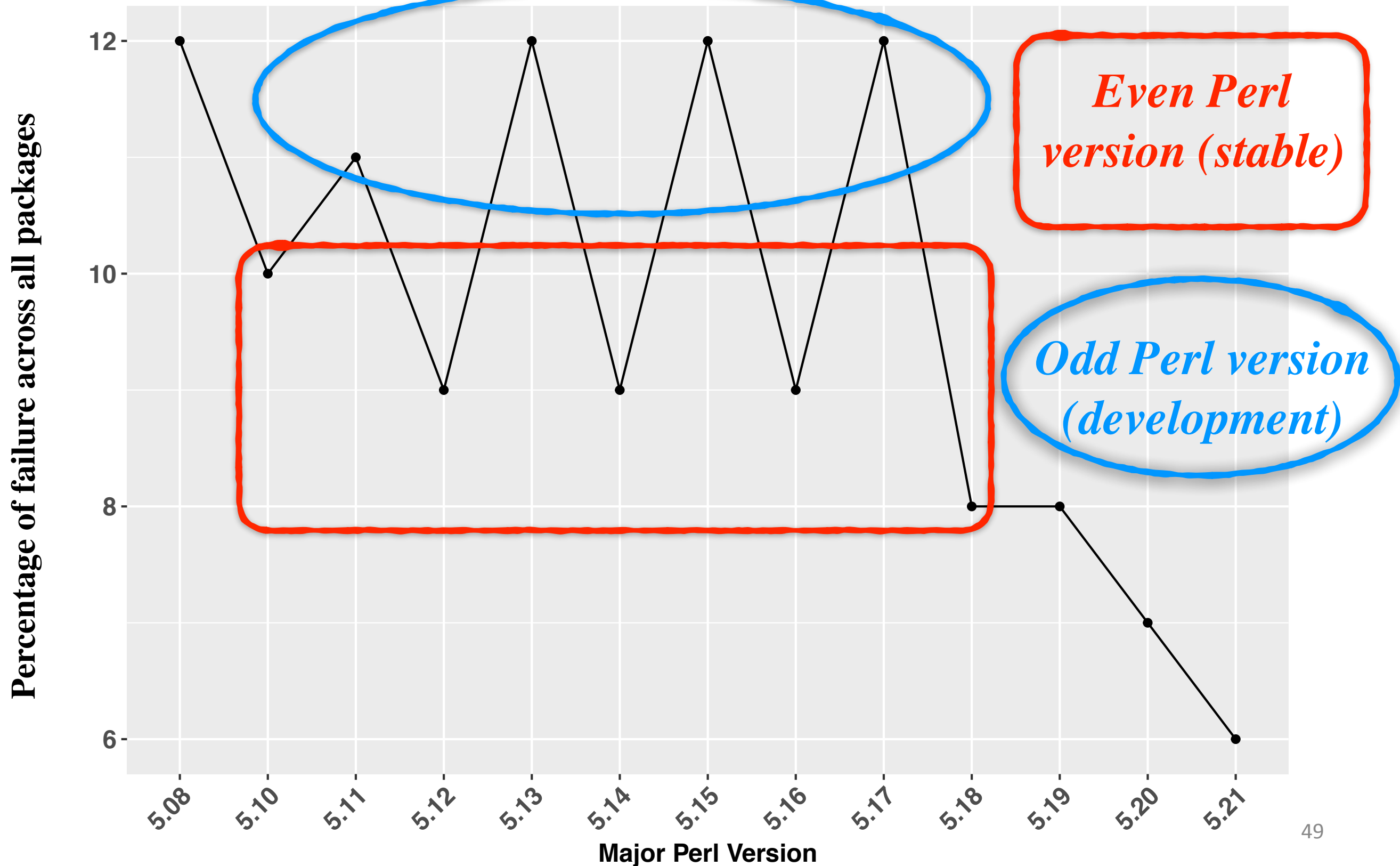
More Builds Finding Less Failures



Not Every Environment Yields Equally Reliable Build Results

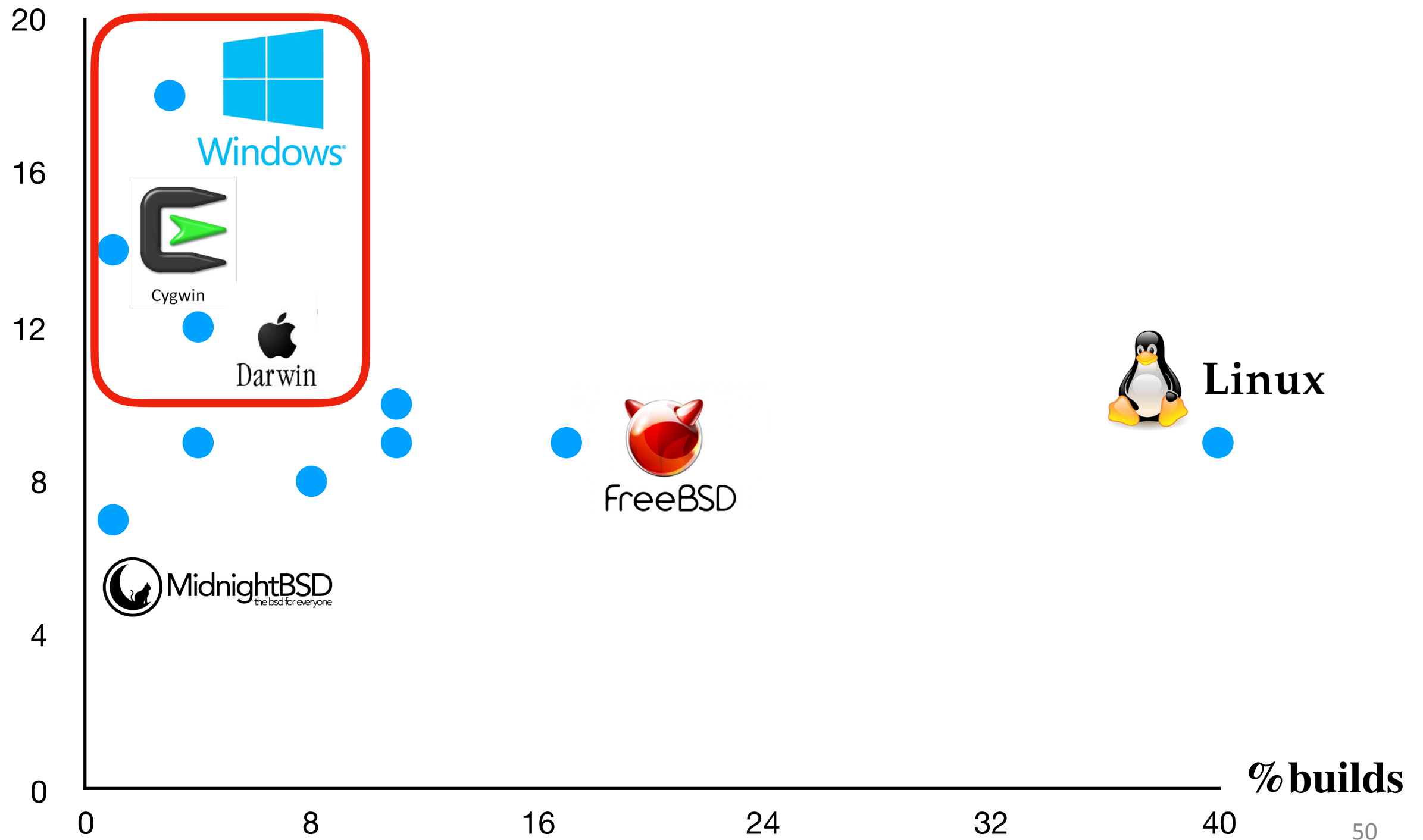


Not Every Environment Yields Equally Reliable Build Results



Build Results on Some Oses Are Less Reliable Than on Others

% build failures

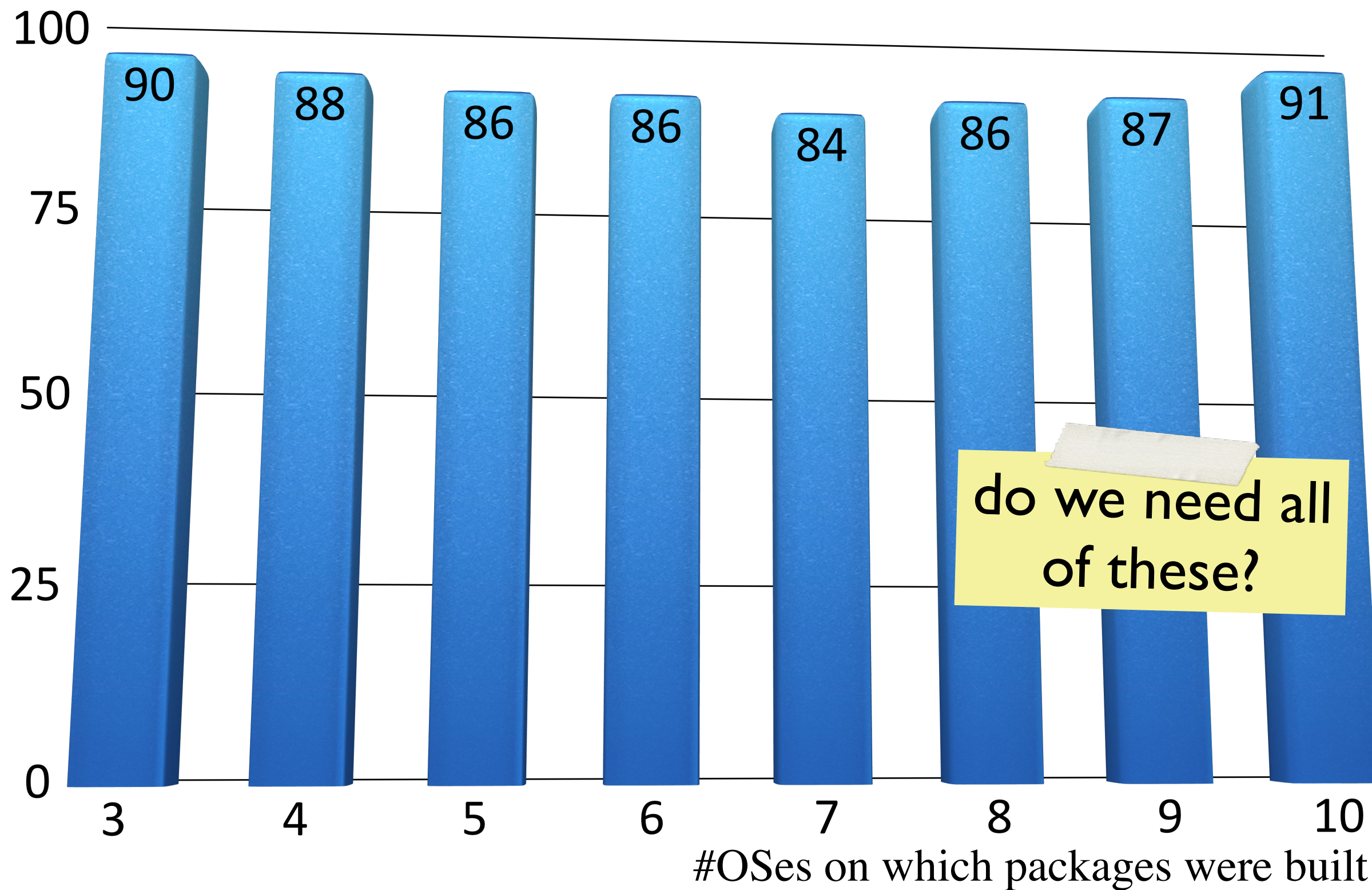


How often do Builds Succeed/Fail Consistently across all OSes?





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?

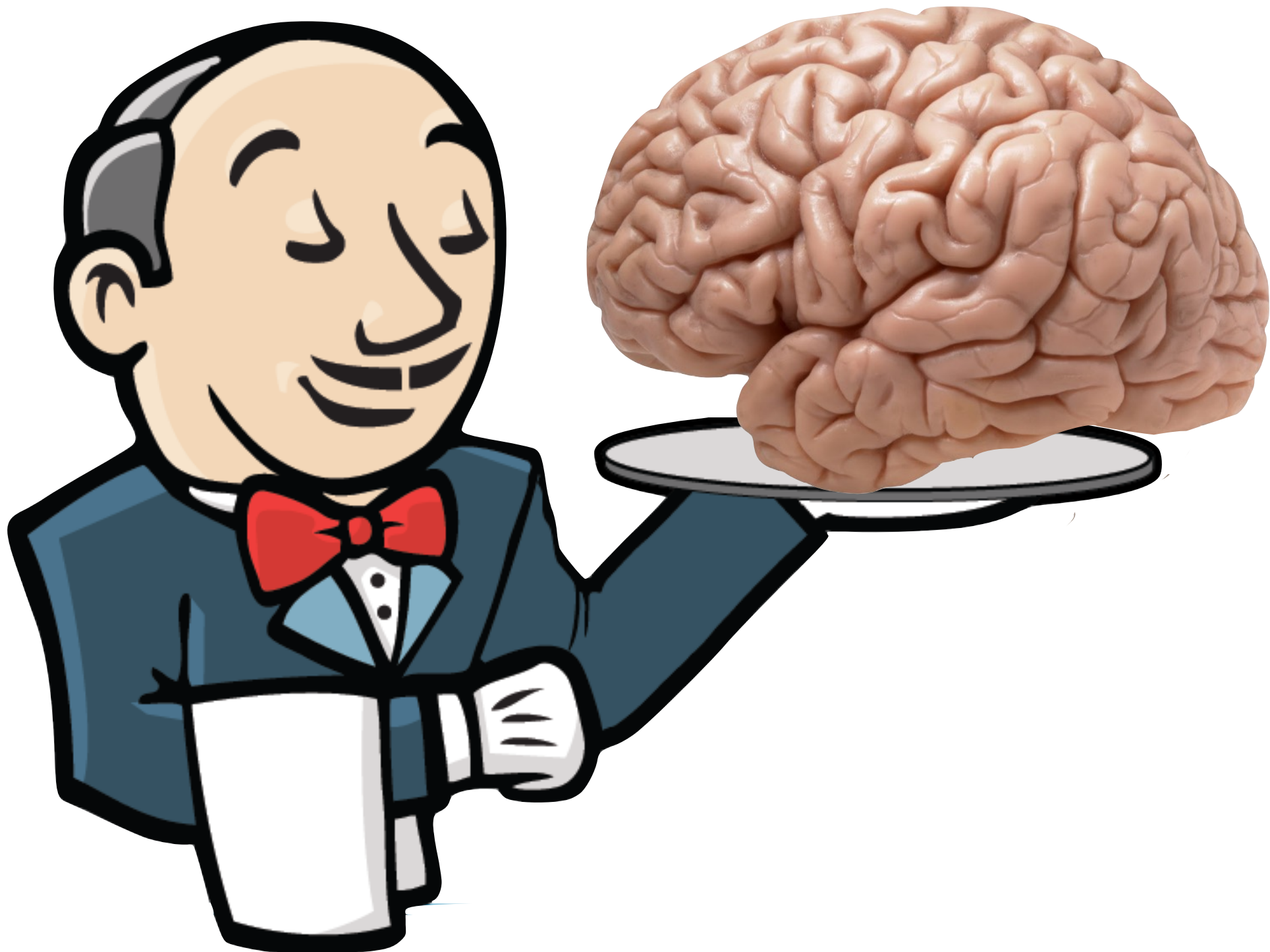
Which tests should be run?

Which files are likely buggy?

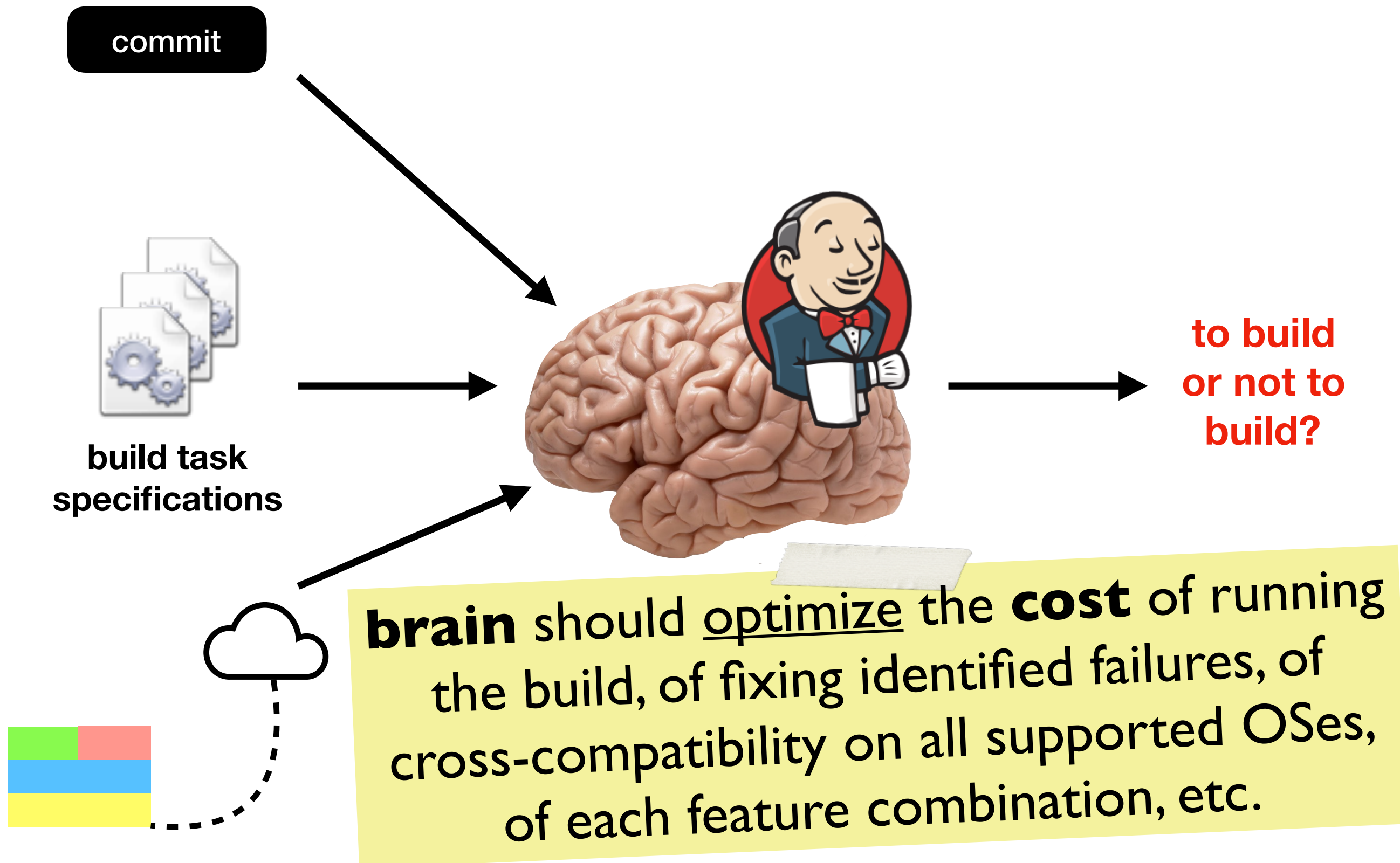
How fast will this build be?

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**?

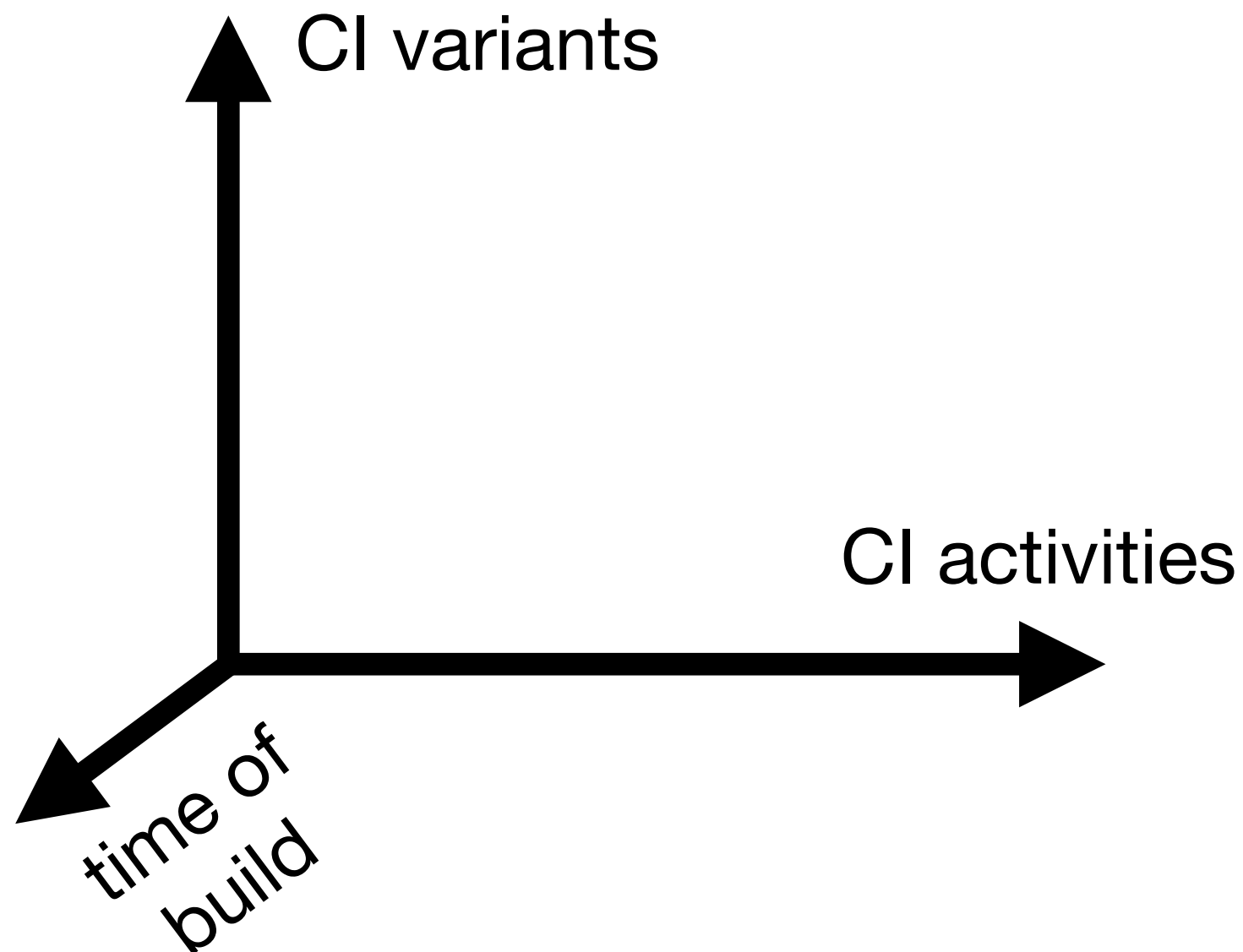
In other words ...



Adding Brains to CI



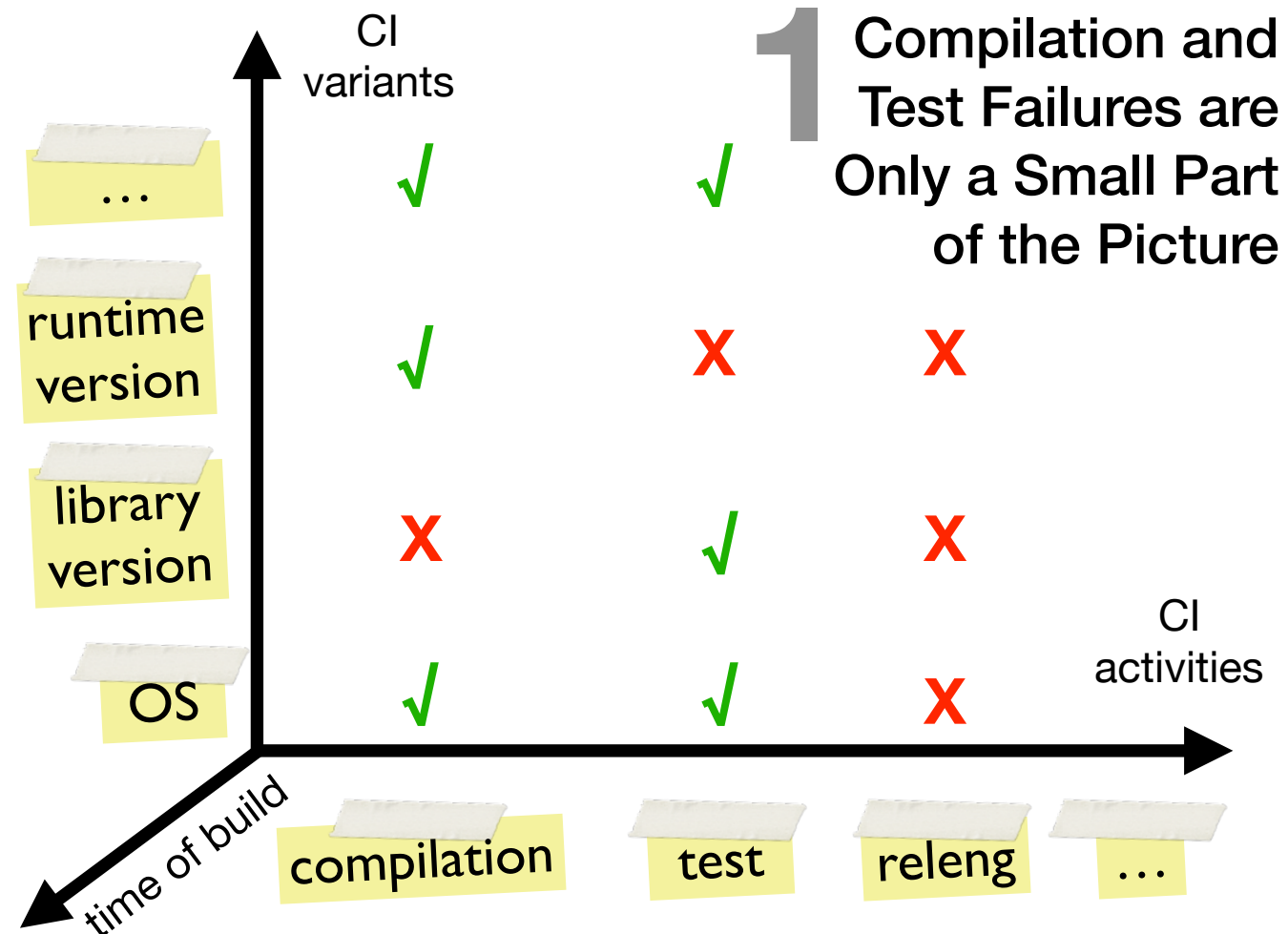
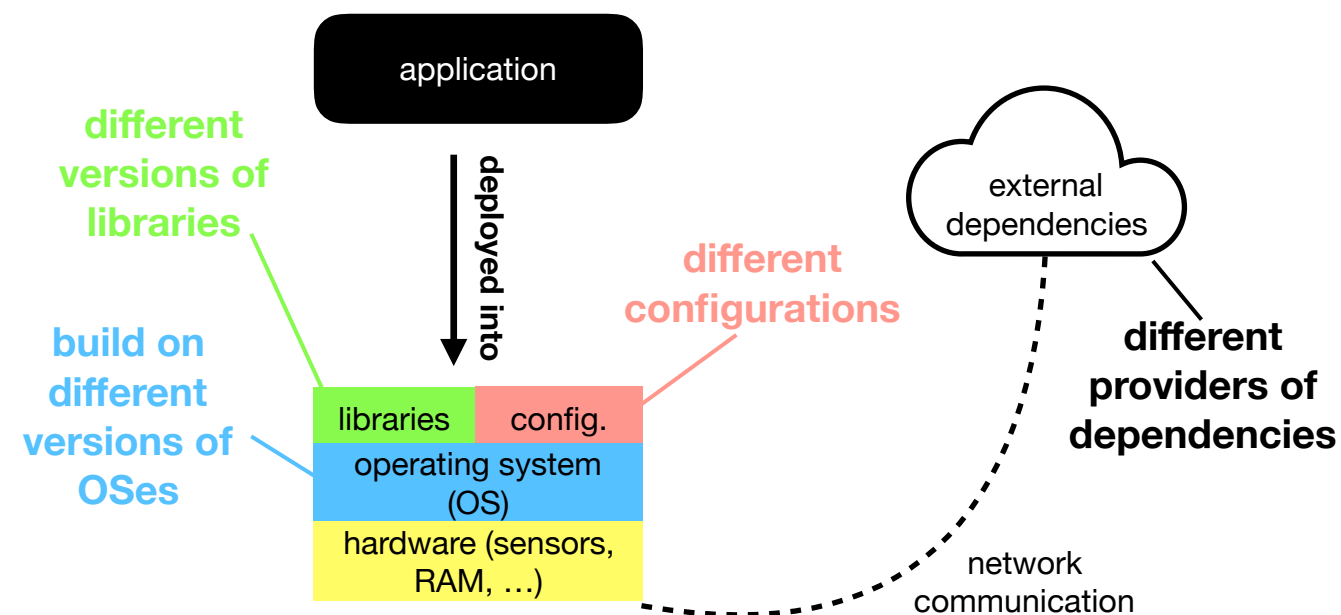
Finally, Support Manual Interpretation of Build Results



techniques like
coalescing/gating, ...

Explosion of Additional CI Builds!

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



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

