

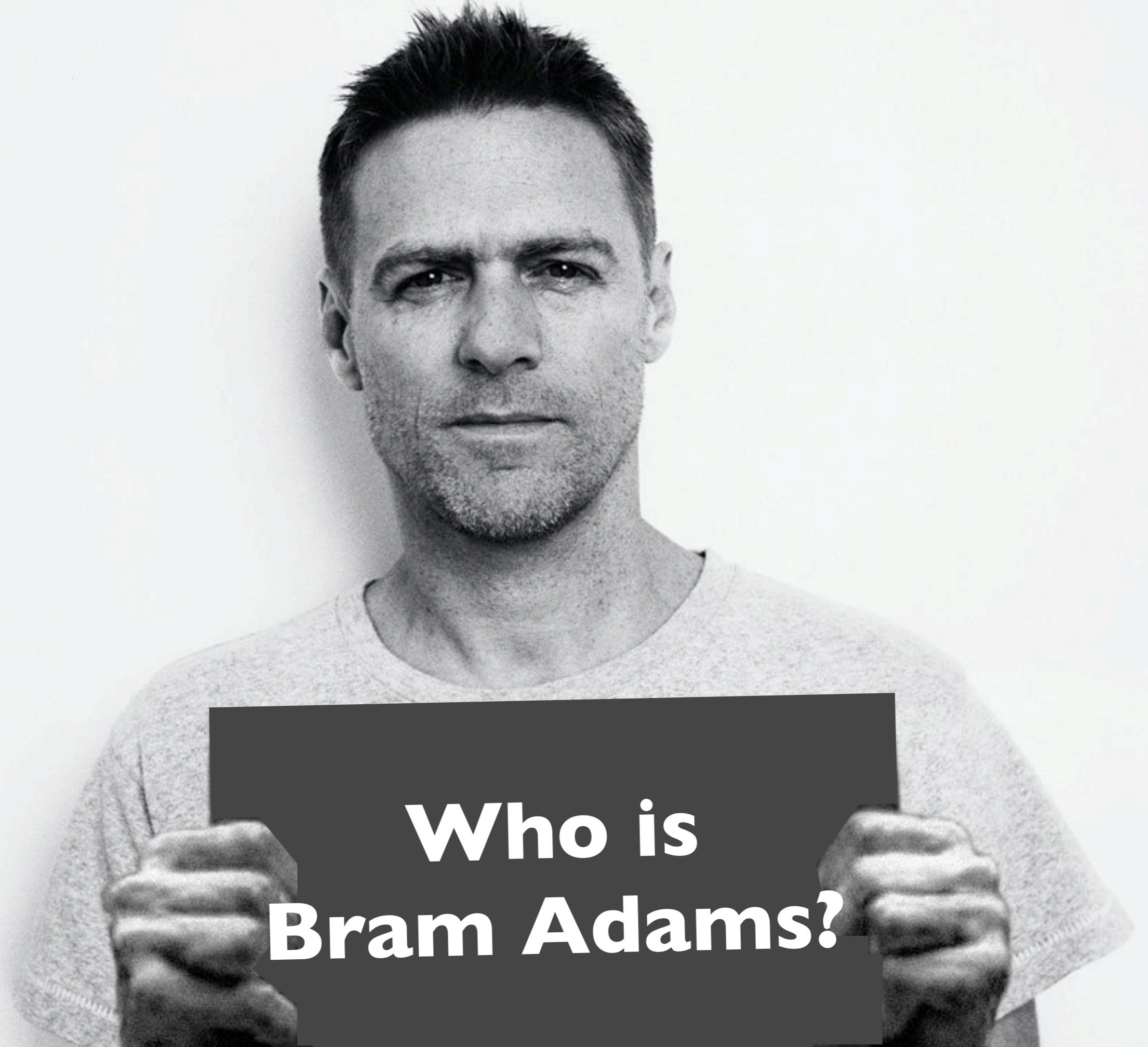
So, you are saying that our software quality was screwed up by ... the release engineer?!



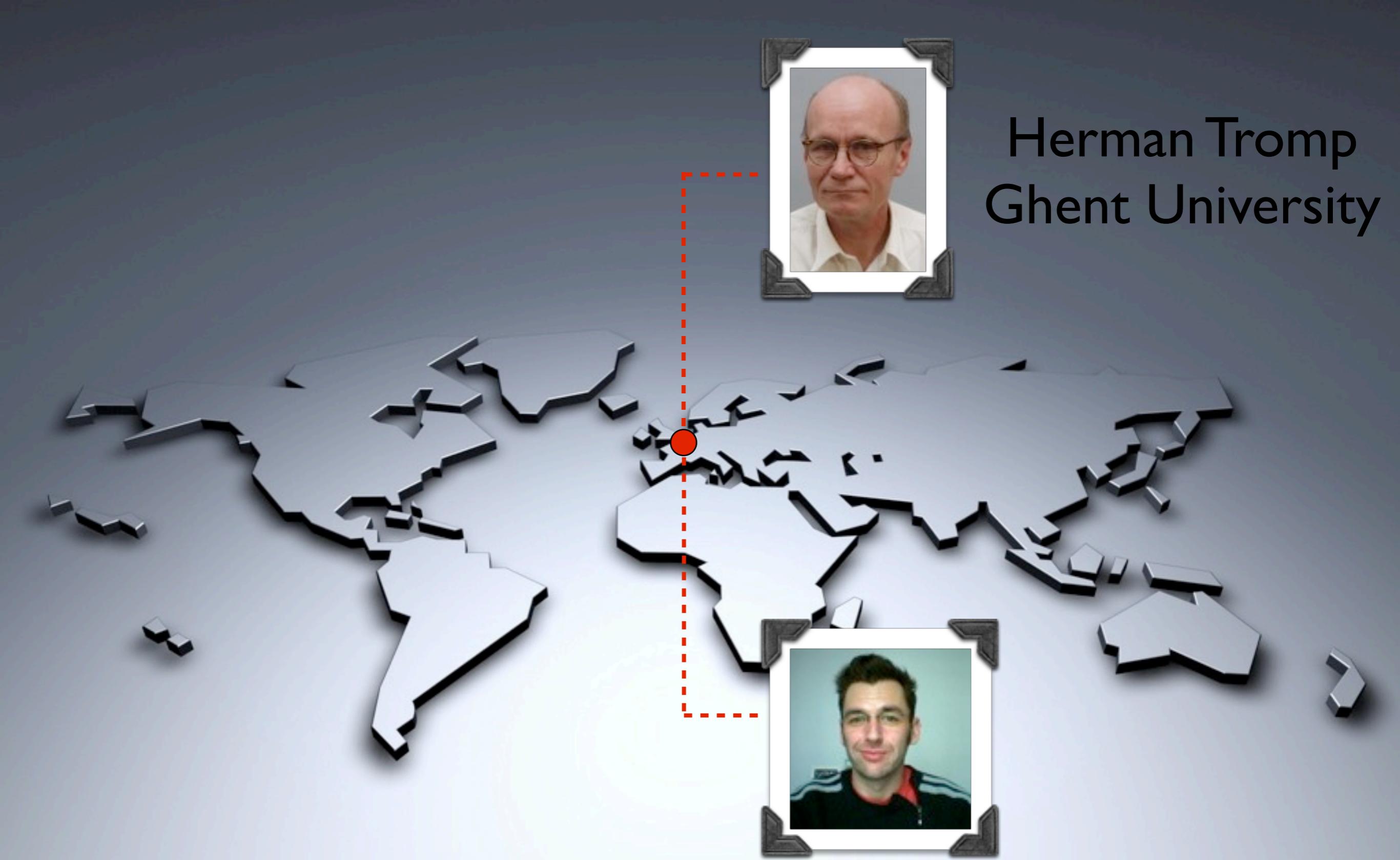
**Bram Adams**

M  
CIS





**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)





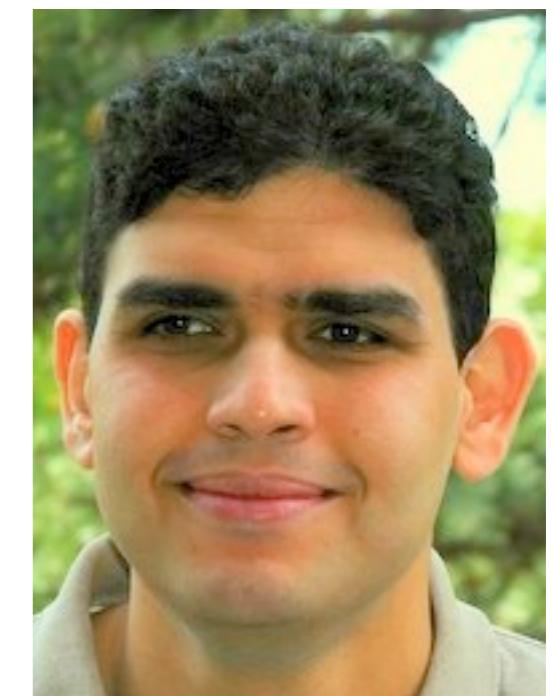
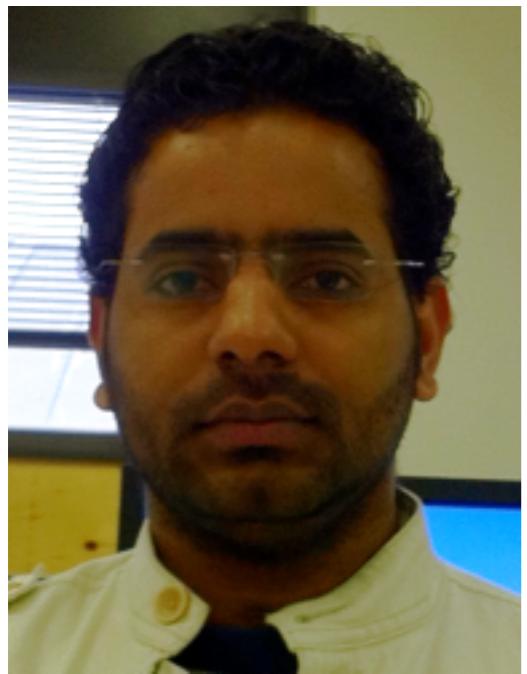
You?

Mini

Parastou

Jojo

M  
C  
I  
S



Tejinder Dhaliwal   Mika Mäntylä   Foutse Khomh   Ahmed E. Hassan

**joint work with**



Daniel German

Emelie Engström

Kai Petersen

Ying Zou

So, you are saying that our software quality was screwed up by ... the release engineer?!



Bram Adams

M  
CIS





Express yourself in the world's largest 3D Chat and Dress-Up community!

[Member Login](#)



**113,653** people online now

in **88** different countries!

Sign in with:



**Choose Your FREE Avatar** ►

Over 2 Million people  
like IMVU on Facebook!

FREE

M  
C  
I  
S



Express yourself in the world's largest 3D Chat and Dress-Up community!

[Member Login](#)

On average we  
deploy new code **fifty** times  
a day.



**113,653** people online now  
in **88** different countries!

Sign in with:



[Choose Your FREE Avatar](#)

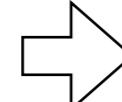
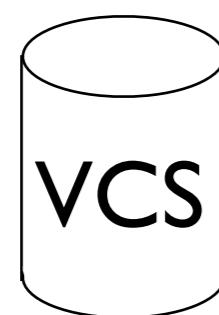
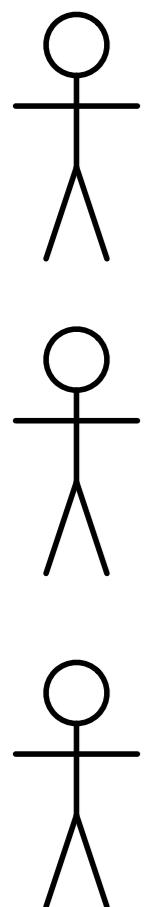
Over 2 Million people  
like IMVU on Facebook!

FREE

M  
C  
I  
S



# Continuous Delivery



**15k tests**



**test**

**9 min.**

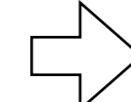
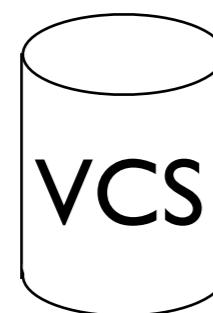
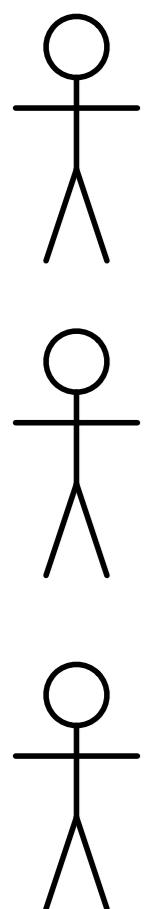
**continuous  
integration**



**staging/production**



# Continuous Delivery



test



9 min.

continuous  
integration

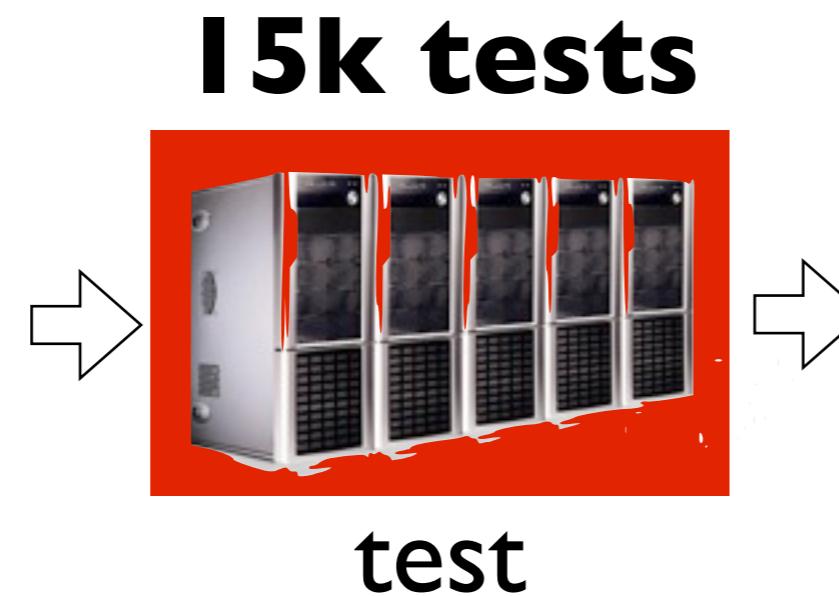
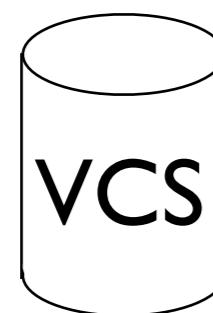
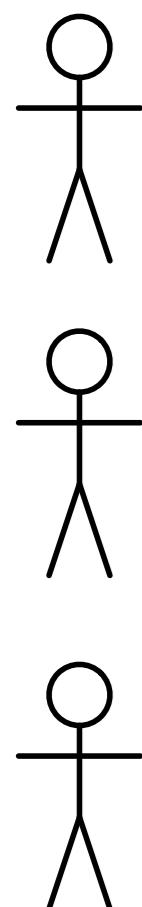
8

staging/production

<http://goo.gl/qPT6>



# Continuous Delivery



continuous  
integration

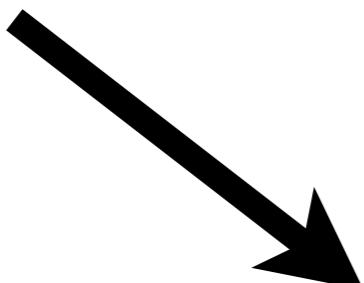
9 min.



staging/production

6 min.

BENEVOL-ers



IMVU, eh?



A photograph of Mark Zuckerberg in profile, facing left. He is wearing a dark t-shirt and holding a microphone. A white speech bubble originates from his mouth, containing the text "Work fast and don't be afraid to break things."

Work fast and  
don't be afraid to break  
things.

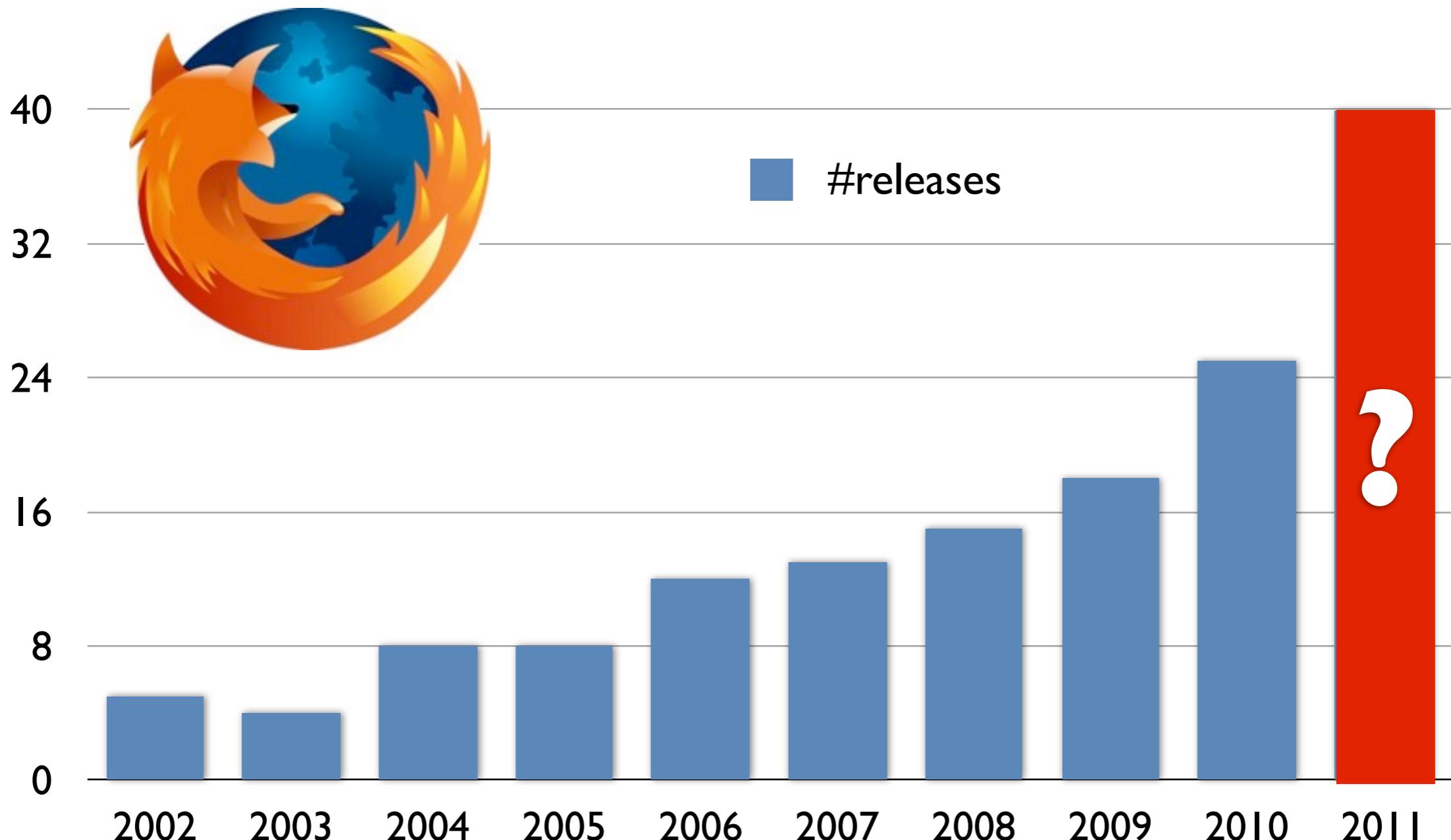
Mark Zuckerberg  
CEO & Founder, Facebook



Build a  
little and then test it.  
Build some more and test  
some more.

James Whittaker

# Even Desktop Apps Release More Frequently



TIMELINE

6 weeks

Awesomeness lands  
on Firefox Nightly

6 weeks

Stabilize on  
Firefox Aurora

More awesomeness  
on Firefox Nightly

6 weeks

Stabilize on  
Firefox Beta

Stabilize on  
Firefox Aurora

*Even more*  
awesomeness  
on Firefox Nightly

6 weeks

Firefox  
Release!

Stabilize on  
Firefox Beta

Stabilize on  
Firefox Aurora

6 weeks

Firefox  
Release!

Stabilize on  
Firefox Beta

... although it's  
Kind of an Illusion!

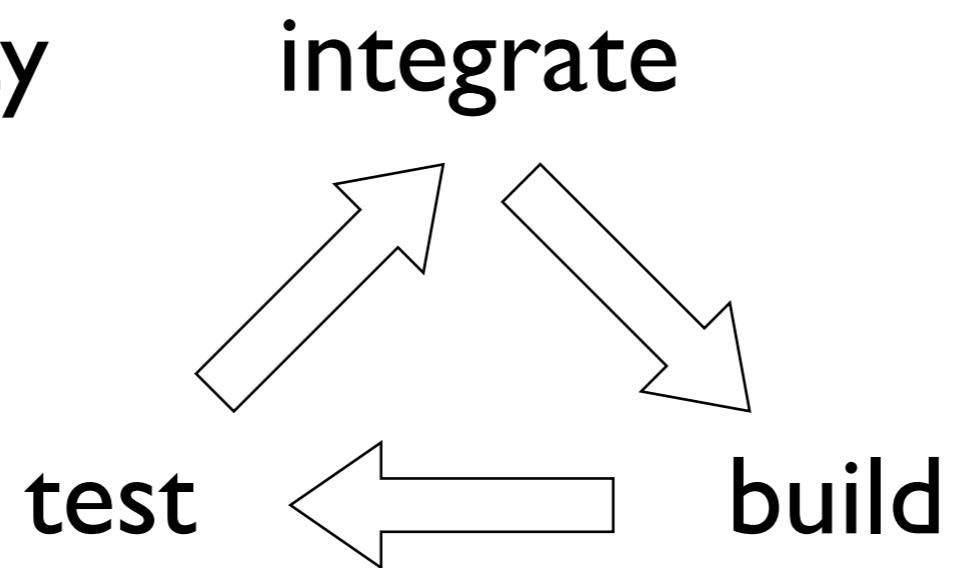


# Rapid Release in a Nutshell



<http://behrns.files.wordpress.com/2008/03/ikea-car.jpg>

in-house/3rd party  
development



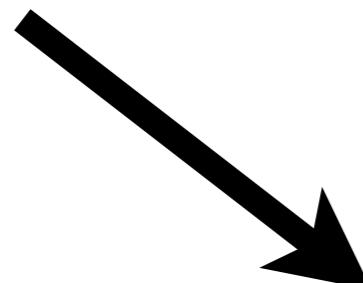
*reduce cycle time!*

14



packaging & delivery

BENEVOL-ers



# Are Rapid Releases Worth the Effort?





rapid release

?  
—  
—

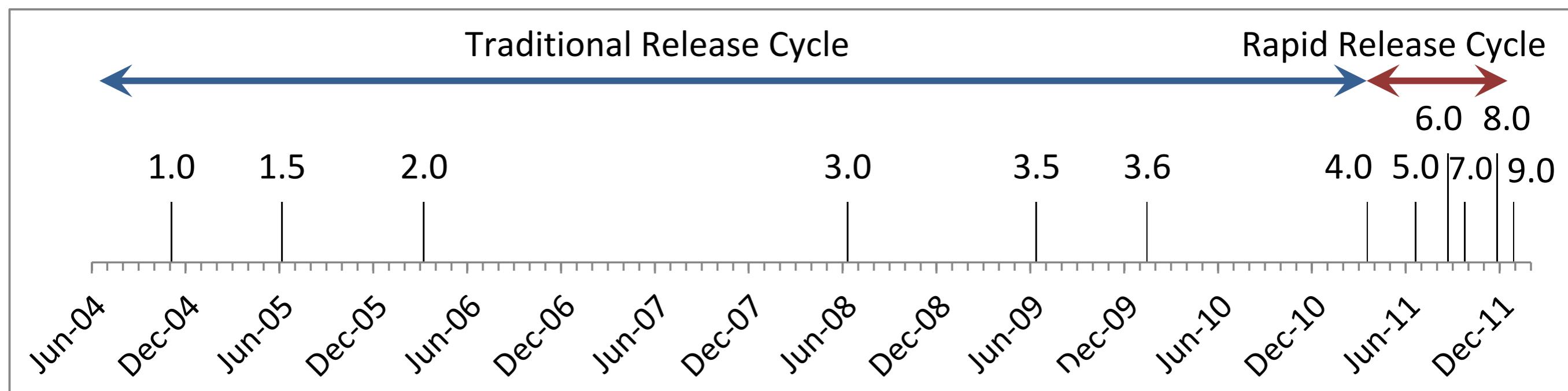


rapid release

?  
—  
—

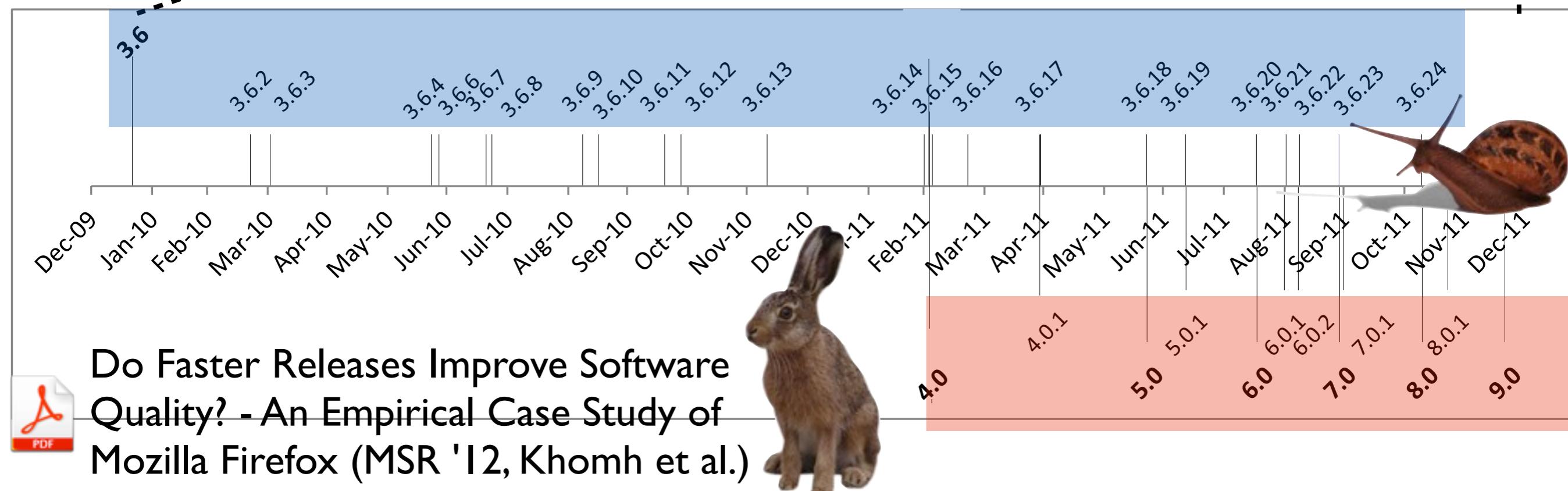
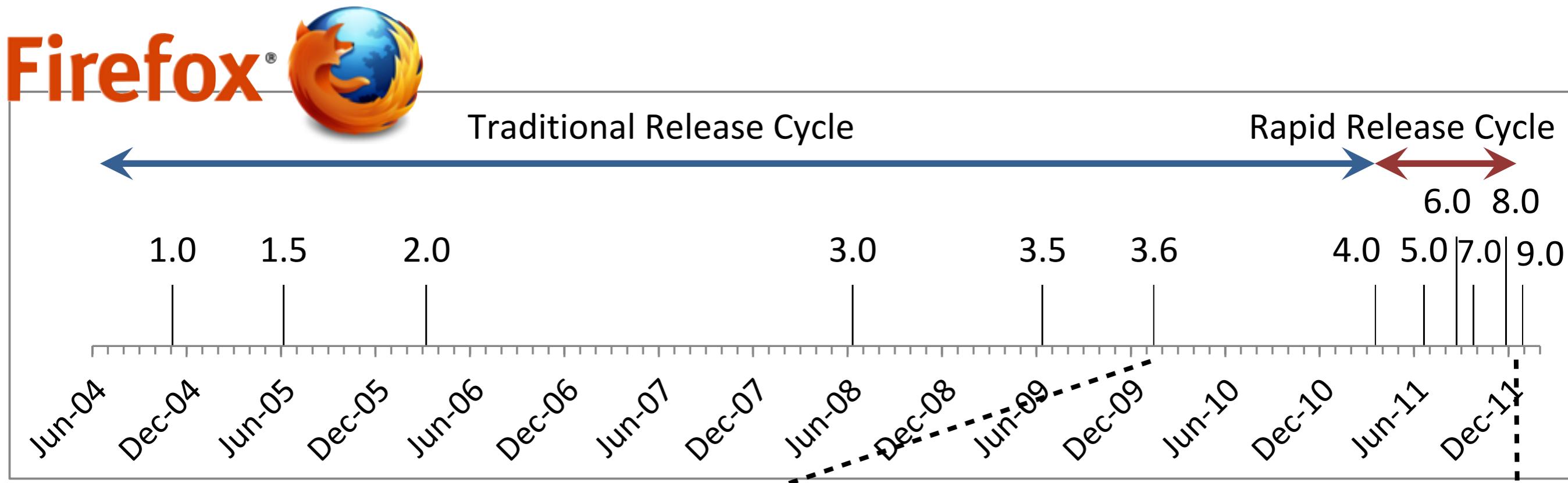


OR



Do Faster Releases Improve Software Quality? - An Empirical Case Study of Mozilla Firefox (MSR '12, Khomh et al.)







# mozilla crash reports

 Find Crash ID or Signature

Product: Firefox

Current Versions

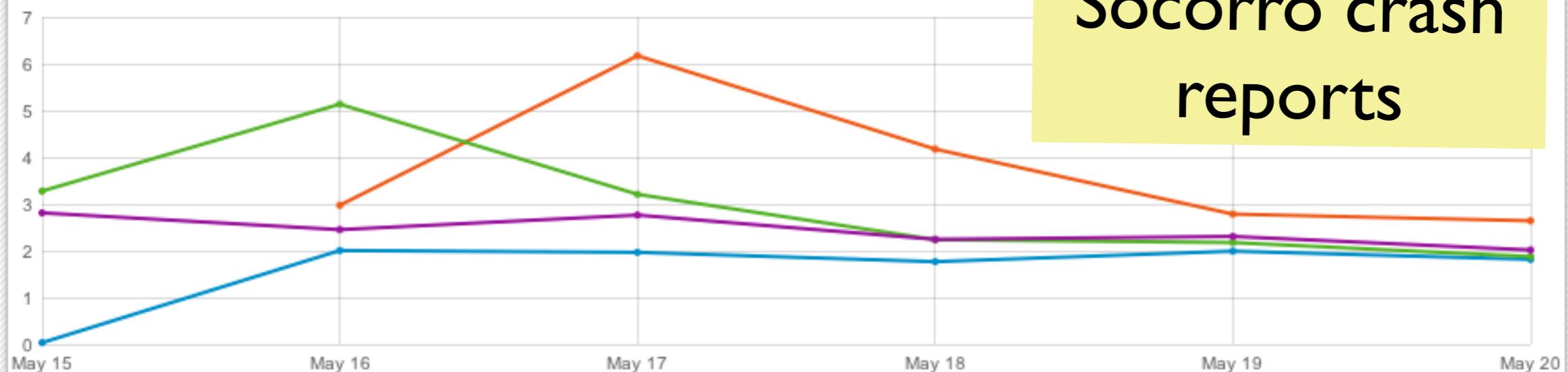
Report: Overview

[Advanced Search](#)

## Firefox Crash Data

[3 days](#)[7 days](#)[14 days](#)Date Range: [By Crash Date](#)[By Build Date](#)

### Crashes per 100 Active Daily Users



## Socorro crash reports

### Crash Reports

[Firefox 24.0a1](#)[Top Crashers](#)[Top Changers](#)[Top Plugin Crashers](#)[Firefox 23.0a2](#)[Top Crashers](#)[Top Changers](#)[Top Plugin Crashers](#)[Firefox 22.0b1](#)[Top Crashers](#)[Top Changers](#)[Top Plugin Crashers](#)[Firefox 21.0](#)[Top Crashers](#)[Top Changers](#)[Top Plugin Crashers](#)



Bugzilla

Bugzilla@Mozilla – Main Page

Home | New | Browse | Search |   [help] | Reports | Requests | Product Dashboard  
| New Account | Log In | Forgot Password

Bugzilla bug  
reports

## Welcome to Bugzilla



[Get Help](#)



[File a Bug](#)



[Search](#)



[Open a New Account](#)

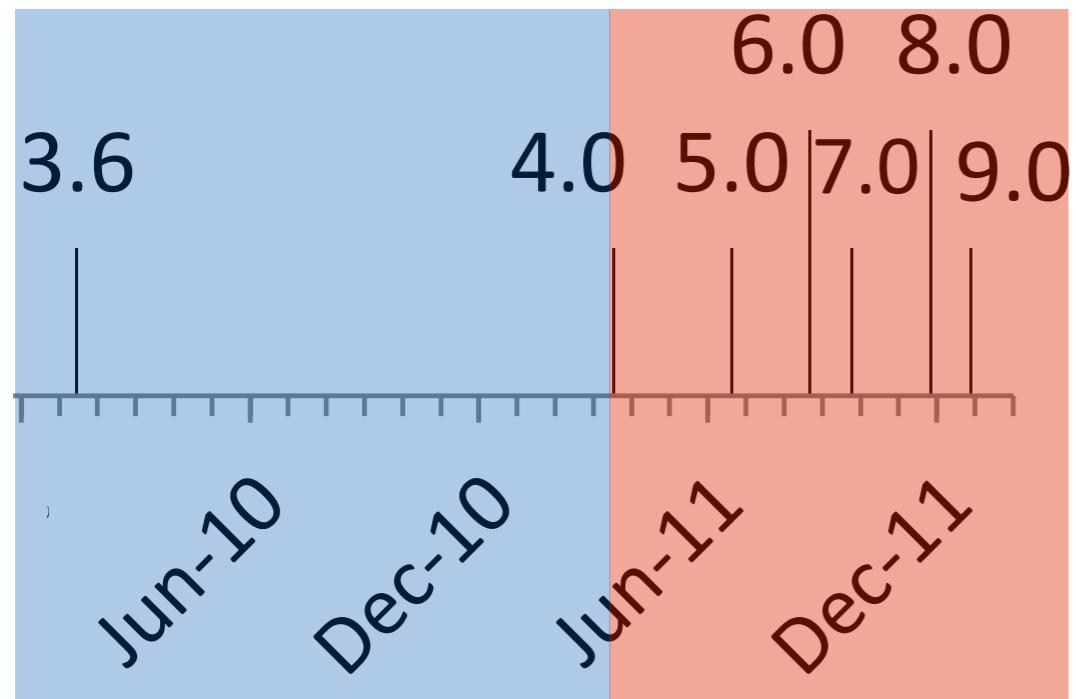
Enter a bug # or some search terms

[Quick Search help](#)

[Bugzilla User's Guide](#) | [Release Notes](#) | [Bugzilla Etiquette](#) | [Bug Writing Guidelines](#) | [Data for Researchers](#)

Home | New | Browse | Search |   [help] | Reports | Requests | Product Dashboard  
| New Account | Log In | Forgot Password

[Privacy Policy](#)

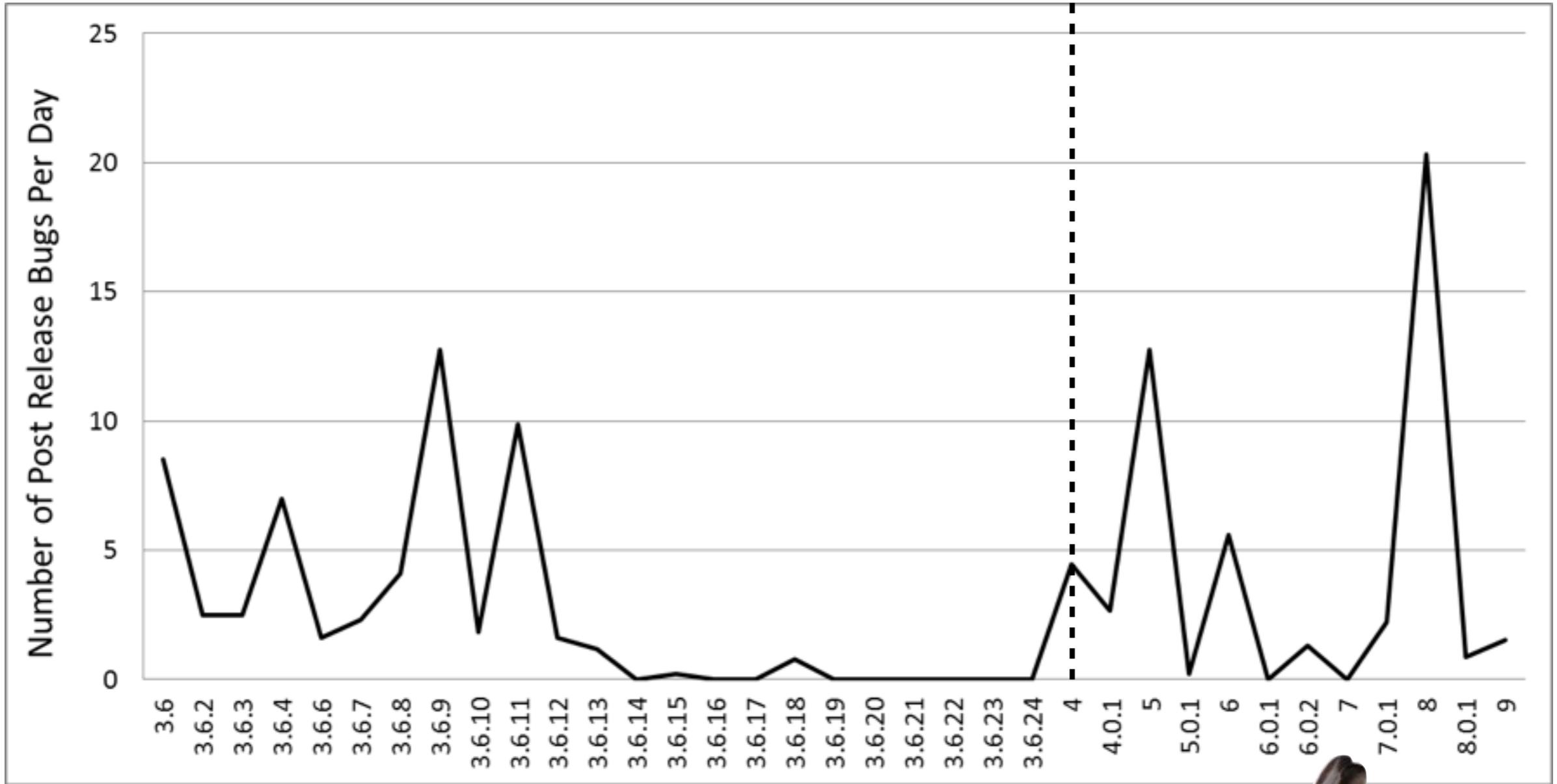


release model

VS.

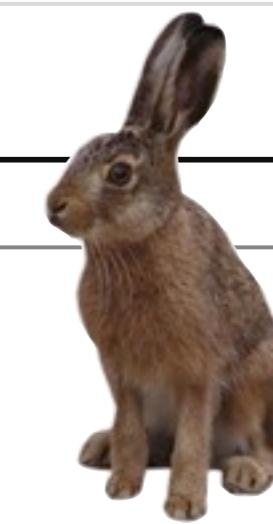
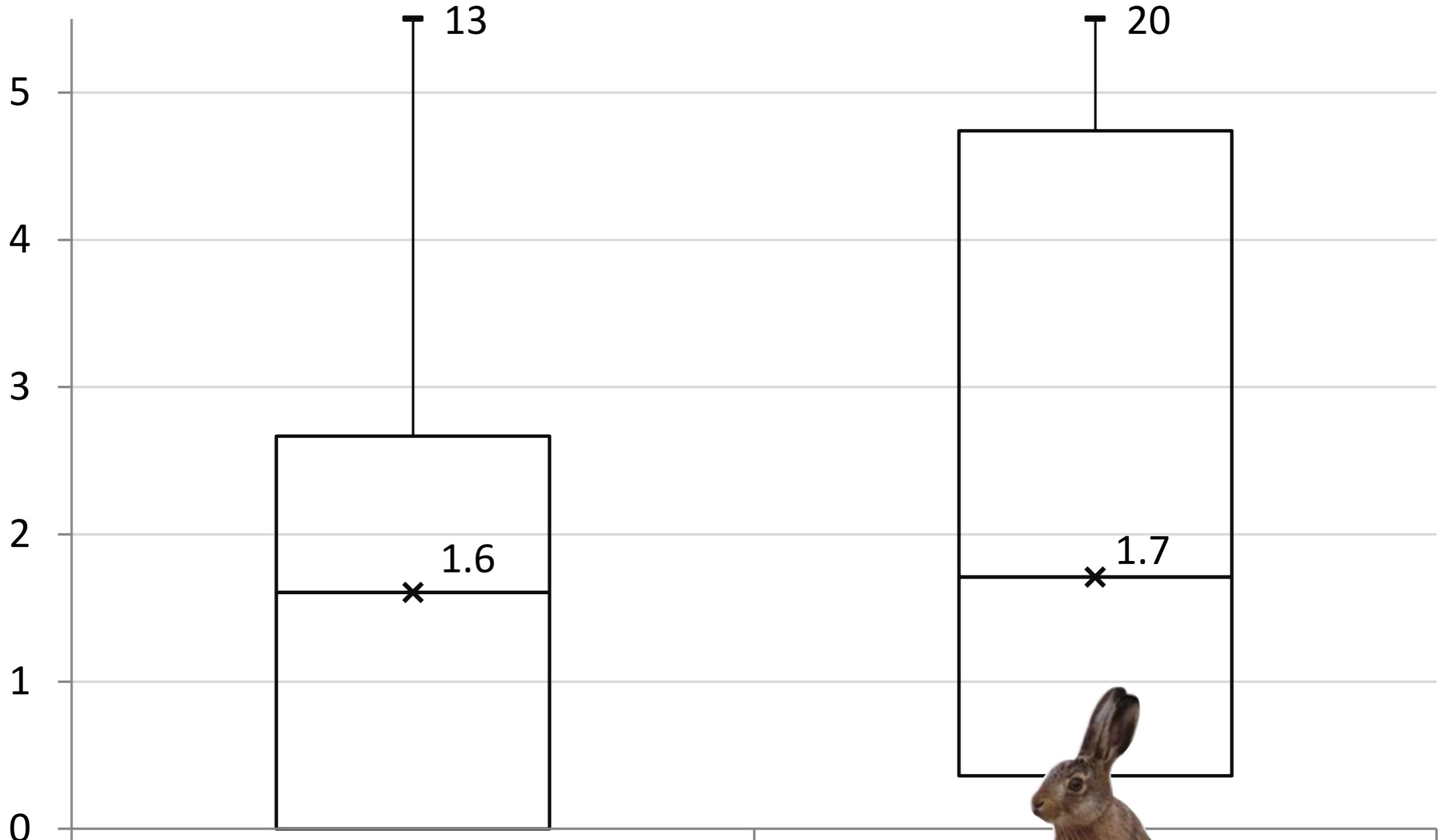


# Same # Post-release Bugs

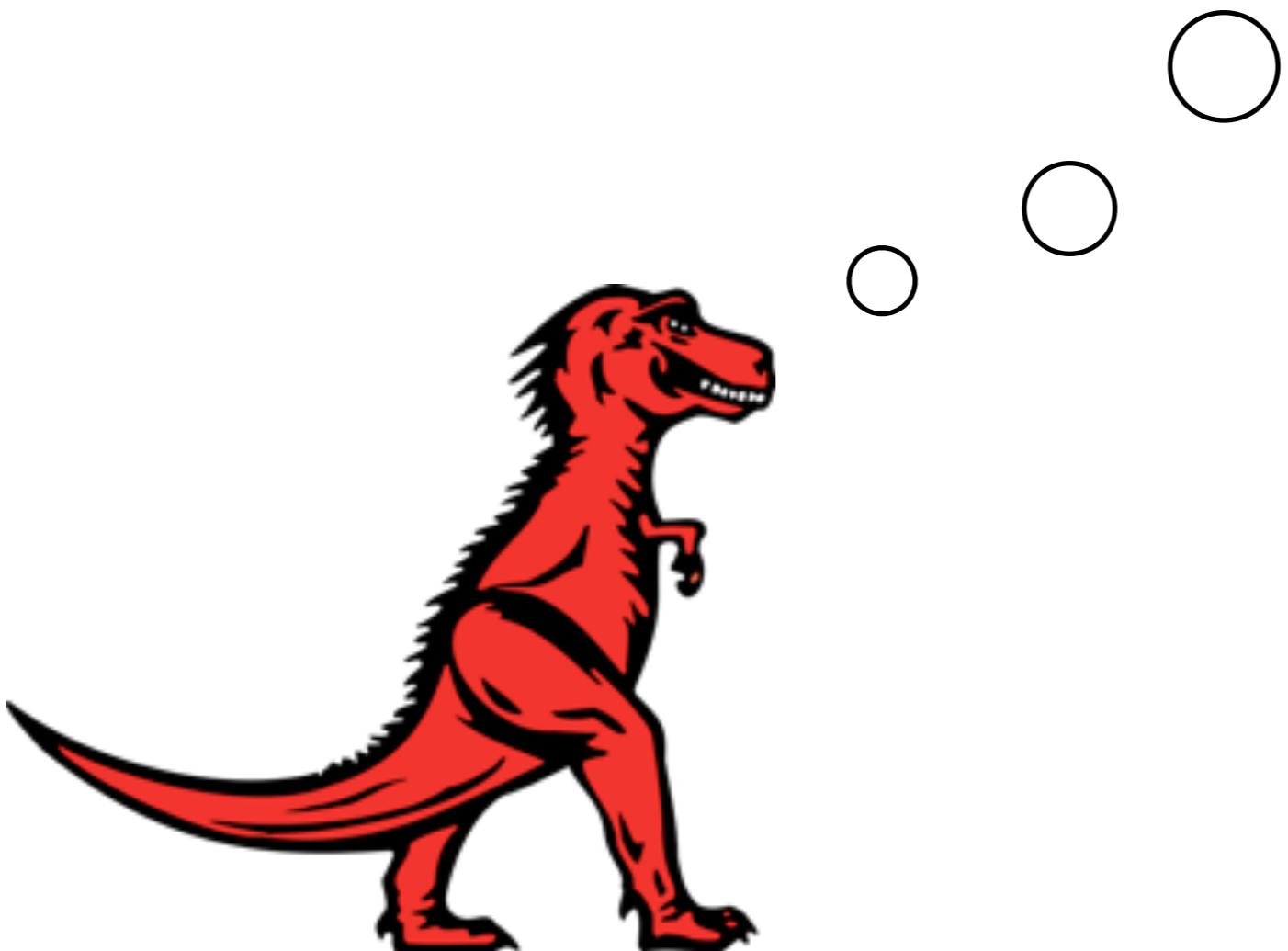


# Same # Post-release Bugs

Number of Post Release Bugs Per Day



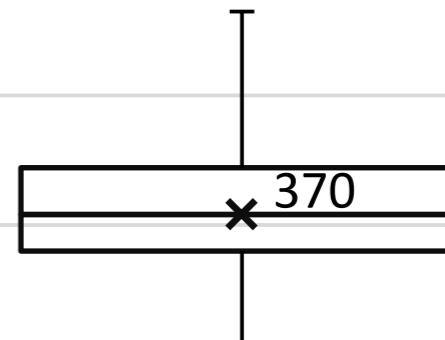
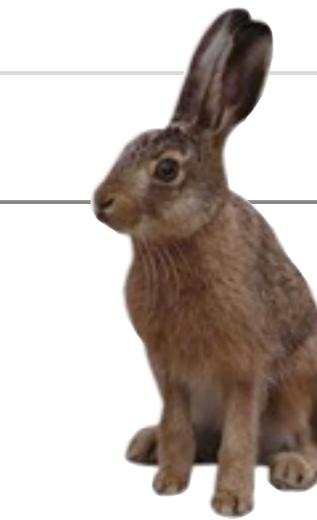
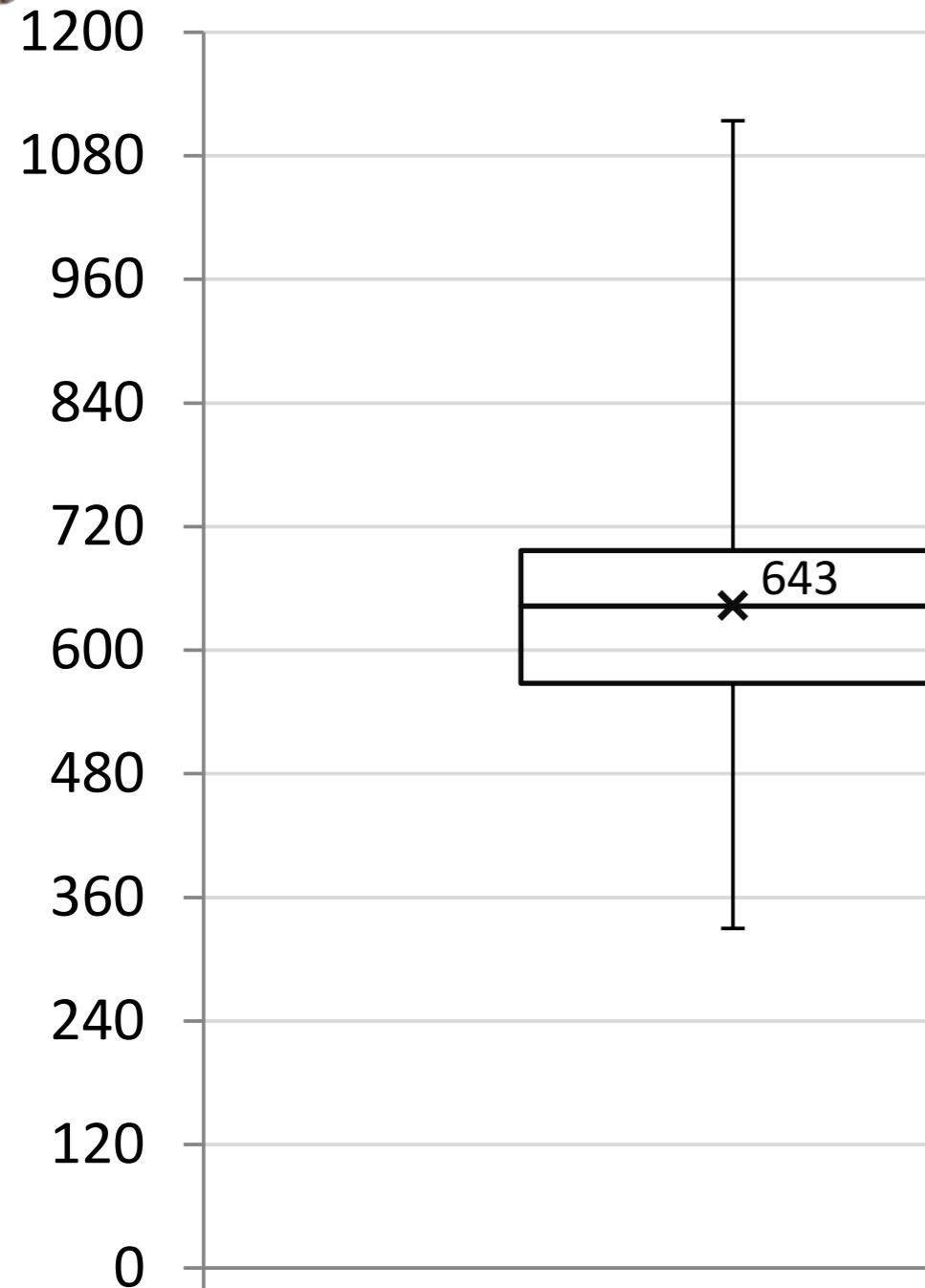
We have about 30 of these project branches active at any one time. That means as a company we can have 30 or more experiments safely running in parallel....I'm guessing that most of the bugs end up being **integration issues** [of these branches into the main trunk] otherwise the project branch wouldn't have merged back in with mainstream devel.



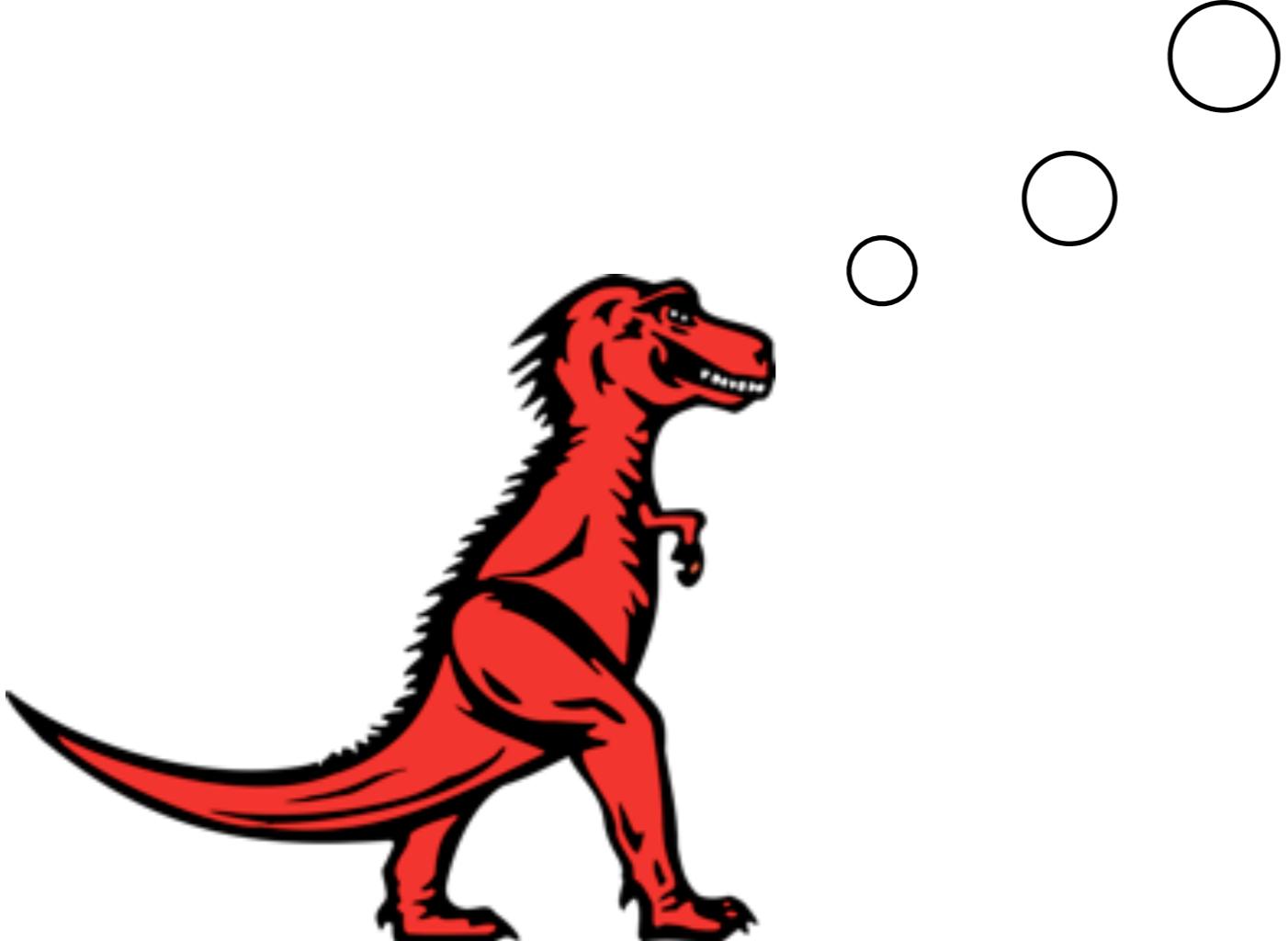


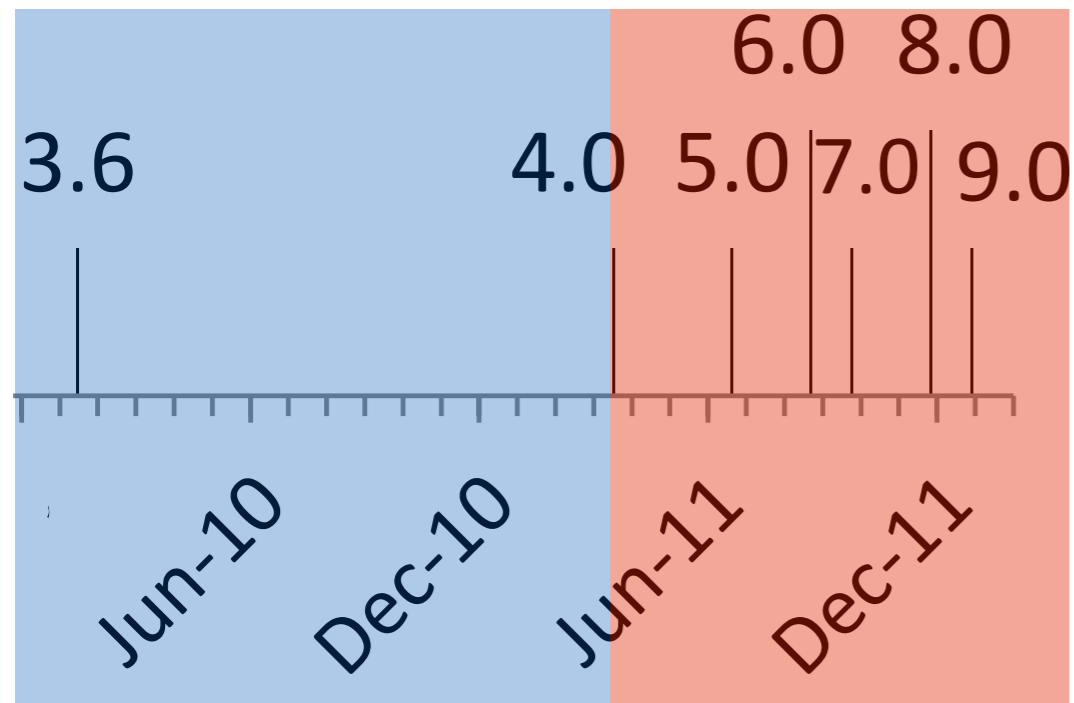
# Crashes Pop Up Earlier

Median UpTime in Seconds



Firefox 4 was a crisis time for Firefox and Mozilla, since it took so long to complete and was a painful engineering effort. We decided to make our development more agile by **switching to rapid release at the same time as we were making a bunch of other changes [...]**, and starting Firefox OS development





release model

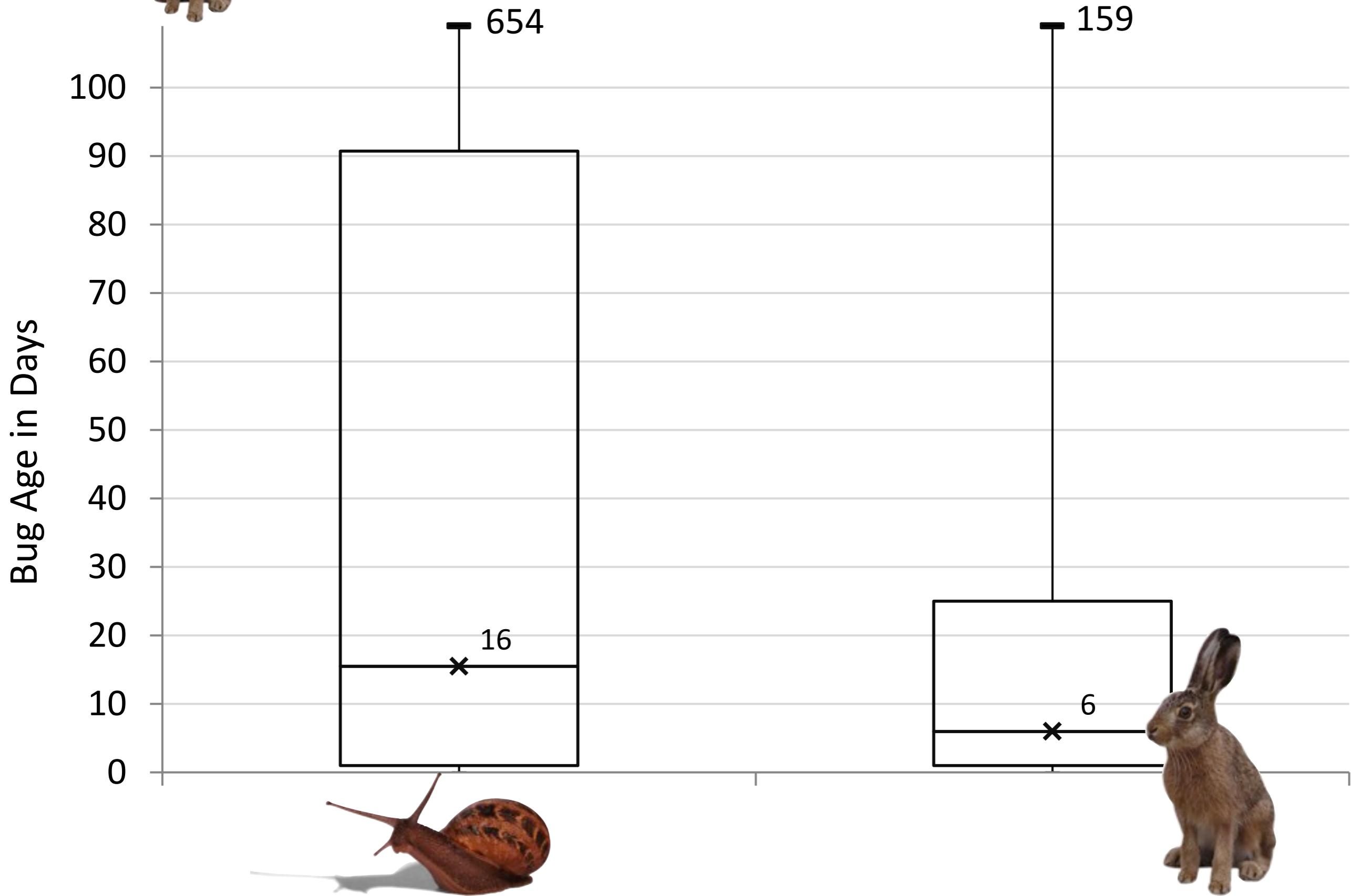
VS.



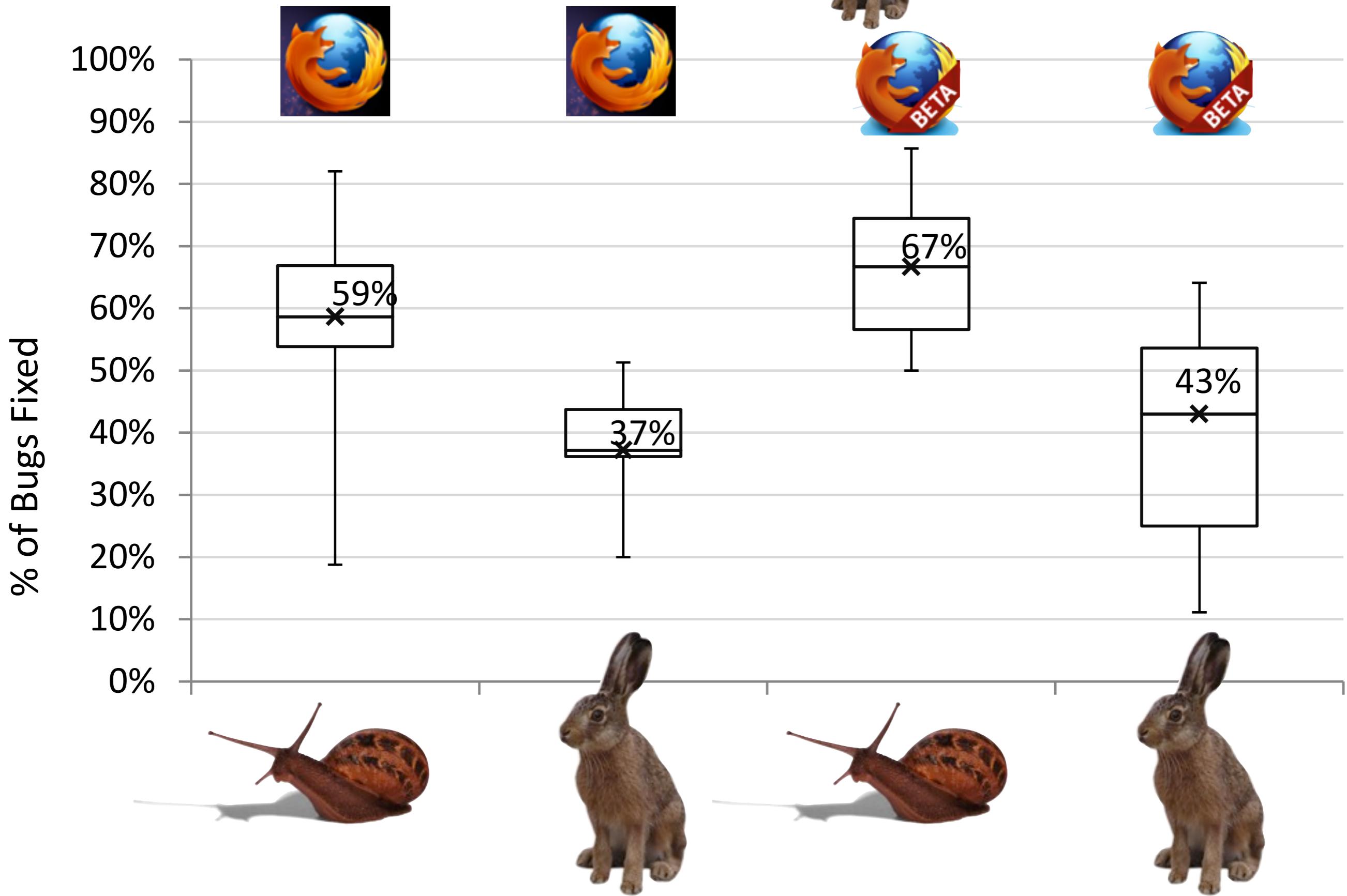
bug fixing



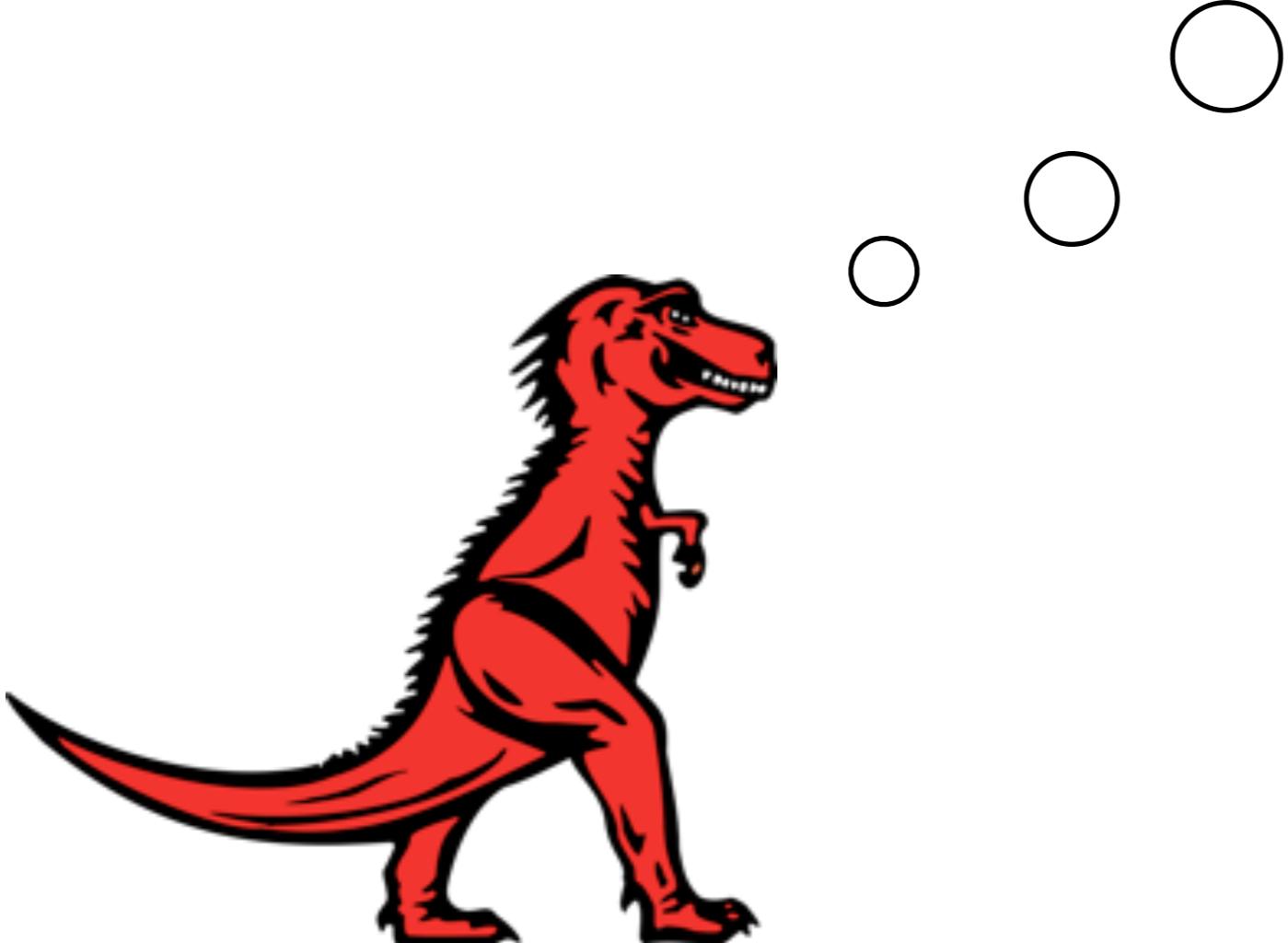
# Bugs are Fixed Faster



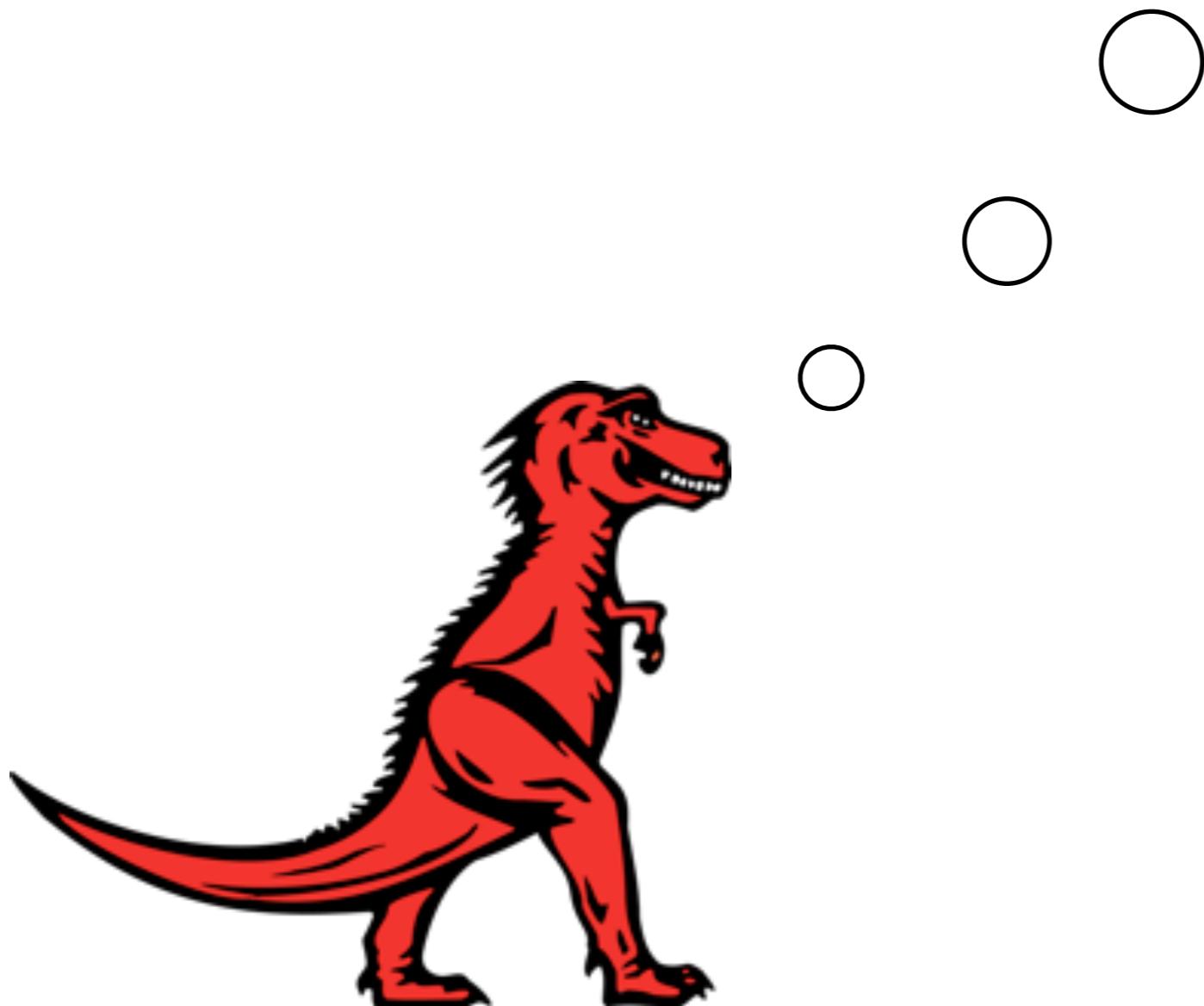
# Proportionally Less Bugs Fixed



I'm not surprised that fewer bugs are fixed: some defects are trivial and some will take weeks of careful code inspection to find and fix. All those bugs would have fallen in the same bucket for a 12+ month release cycle, but now the **harder ones get pushed out to later releases**



can be better interpreted that we are being **less effective at triaging bugs** with rapid release, or that we have more beta users and so the **incoming bug rates are larger**





same  
#bugs

... but faster  
crashes

bugs are  
fixed faster

... but less  
bugs are  
being fixed

# Shall we Blame the Testers?

BENEVOL-ers



# Rapid Release in a Nutshell



in-house/3rd party  
development

integrate

test

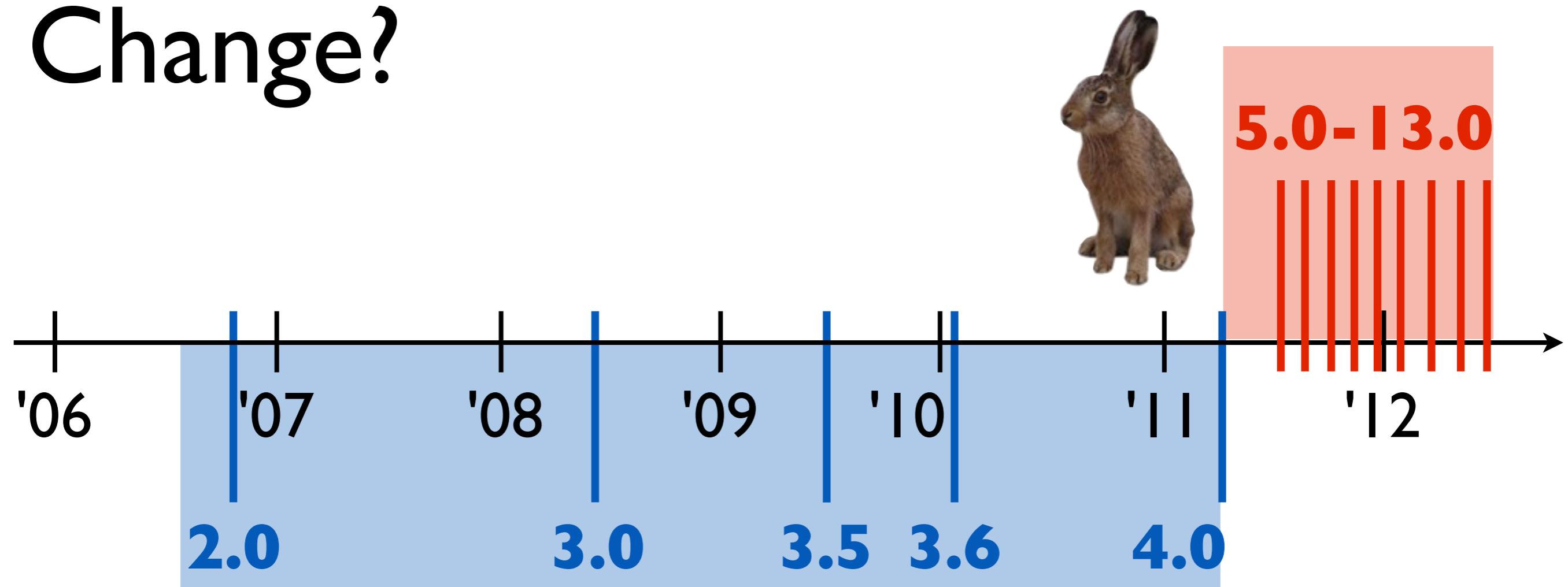
build

*reduce cycle time!*



packaging & delivery

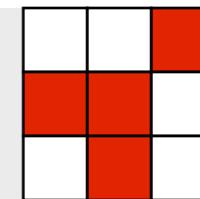
# Did the System Testing Process Change?



Rapid Release Model vs. Software Testing Effort – Study of Mozilla Firefox Project (ICSM '13, Mäntylä et al.)

# Different Test Phases

(selection of) unit tests



**integration + build**

all unit tests

integration tests

capacity tests

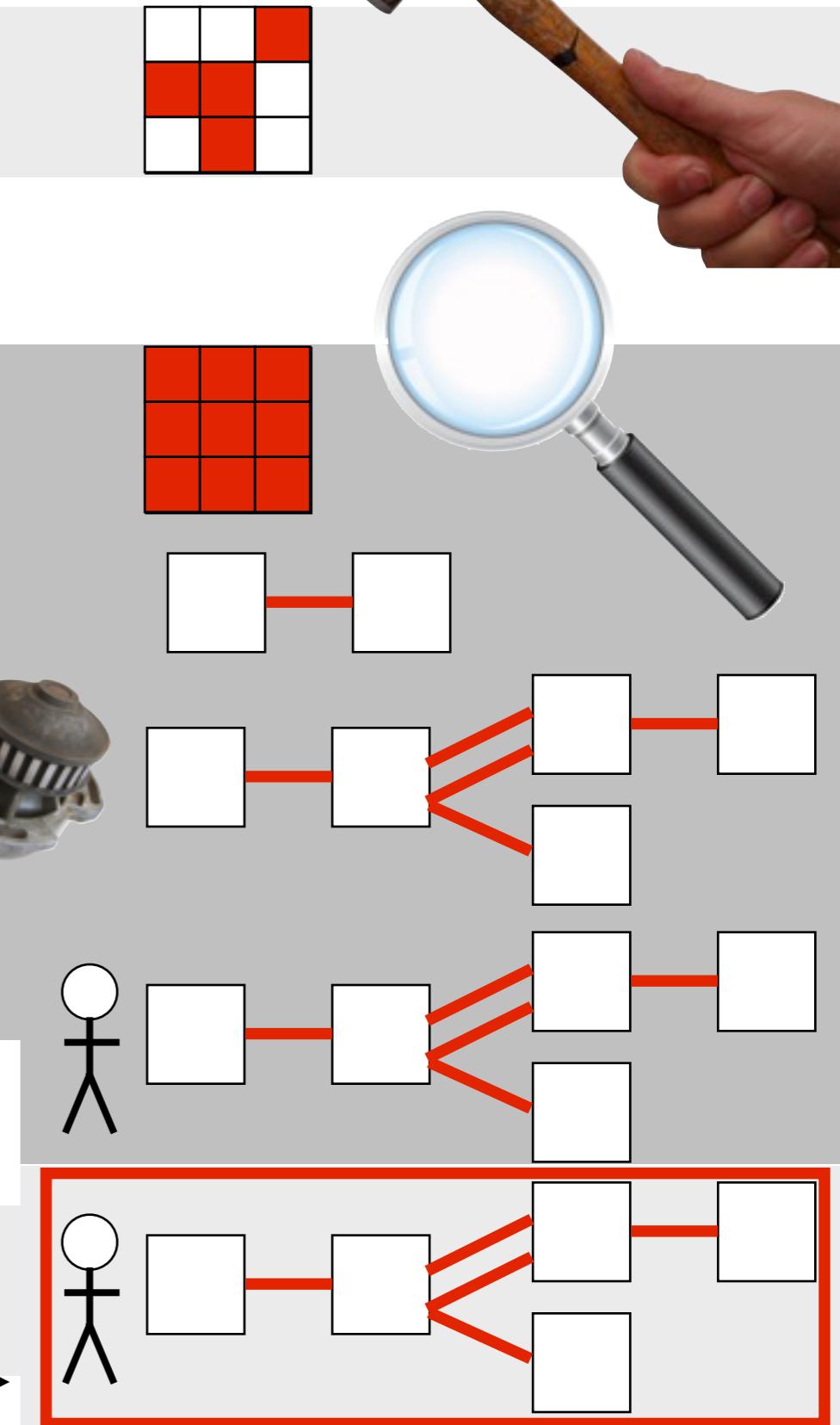
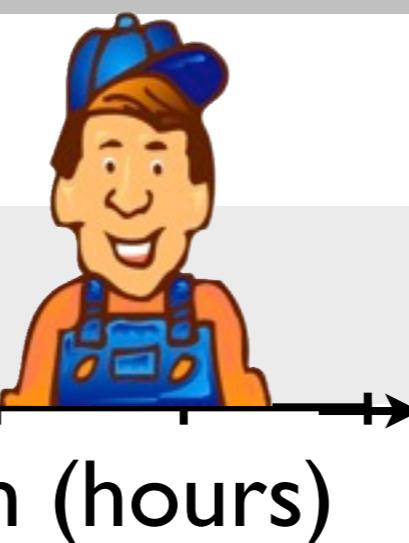
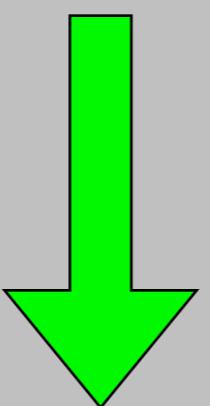
user acceptance/system tests

**release**

smoke tests

test phase

time to run (hours)



# Litmus

welcome

log in

get testing!

run tests

view/search tests

reporting

search results

advanced search

testdays

test runs

statistics

help

litmus tutorial

litmus wiki

## testcase id #8410 - all tab options unchecked

**Testcase ID #:** 8410  
**Summary:** All Tab options unchecked  
**Product:** Firefox  
**Branch:** 3.6 Branch  
**Regression Bug ID #:** None specified  
**Enabled?**   
**Community Enabled?**

## Test Scenario & Expected Outcome

### Steps to Perform:

1. Open Tool | Options (Preferences) | Tabs. Make sure all check boxes are unchecked.

### Expected Results:

- Confirm that all four preferences do not work, since they are disabled:
- A new window is opened instead of a new tab.
  - You will not be warned when closing multiple tabs.
  - You will not be warned when opening multiple tabs (>15 at once, e.g. the default RSS folder in the toolbar).
  - The tab bar is not shown when only one tab is open.
  - Opening a link in a new tab will open it in the background.

### Testgroup(s)

3.6 Full Functional Tests (FFT) (137)

### Subgroup(s)

FX 3.6 FFT - Tabbed Browsing (1242)

## recent test results for testcase id #8410 - all tab options unchecked

submission date	result id#	testcase id#: summary	product	platform	branch	locale
2010-11-02 02:16:00	<a href="#">336691</a>	<b>8410:</b> All Tab options unchecked	Firefox		3.6 Branch	en-US
2010-07-07 10:07:17	<a href="#">279135</a>	<b>8410:</b> All Tab options unchecked	Firefox		3.6 Branch	en-US
2010-05-12 11:20:12	<a href="#">257854</a>	<b>8410:</b> All Tab options unchecked	Firefox		3.6 Branch	en-US
2010-01-17 02:21:29	<a href="#">232877</a>	<b>8410:</b> All Tab options unchecked	Firefox		3.6 Branch	en-US
2010-01-11 20:56:35	<a href="#">230165</a>	<b>8410:</b> All Tab options unchecked	Firefox		3.6 Branch	en-US
2010-01-07 10:50:21	<a href="#">228669</a>	<b>8410:</b> All Tab options unchecked	Firefox		3.6 Branch	en-US
2009-10-22 03:22:31	<a href="#">213485</a>	<b>8410:</b> All Tab options unchecked	Firefox		3.6 Branch	en-US

search test results: [recent failures](#) | [recently unclear](#) | [most common failures](#) | [testcases most frequently marked as unclear](#) | [testgroup popularity](#) | [results with comments](#)

search testcases: **testcases needed!** | [recently added](#) | [recently changed](#) | by id:

Recorded  
Test  
Results

# Litmus

welcome

log in

get testing!

run tests

view/search tests

06/2006 - 06/2012

test runs  
statistics

help

litmus tutorial

10% of test cases are automated

## testcase id #8410 - all tab options unchecked

<b>Testcase ID #:</b>	8410
<b>Summary:</b>	All Tab options unchecked
<b>Product:</b>	Firefox
<b>Branch:</b>	3.6 Branch
<b>Regression Bug ID #:</b>	None specified
<b>Enabled?</b>	<input checked="" type="checkbox"/>

1. Open Tool | Options (Preferences) | Tabs. Make sure all check boxes are unchecked.

147 releases

Confirm that all your preferences do

- A new window is opened instead of a new tab.
- You will not be warned when closing tabs.
- You will not be warned when opening multiple tabs (>10 at once), e.g. the default RSS folder in the toolbar.
- The tab bar is not shown when only one tab is open.
- Opening a link in a new tab will open it in the background.

312,502 test executions

6,058 different testers

2,009 tested builds

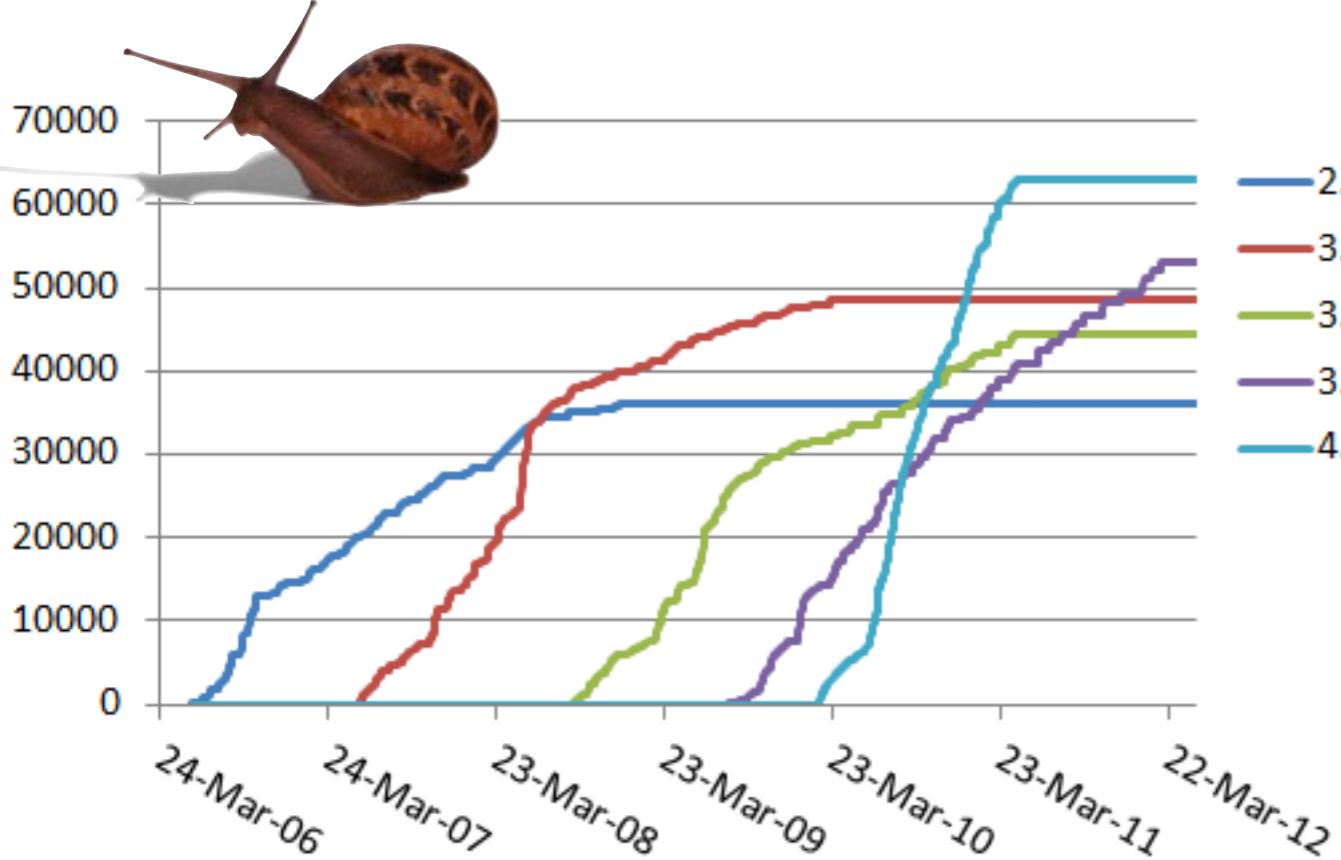
date	testcase id	summary	platform	branch	language
2010-07-07 10:07:17	336691	8410: All Tab options unch	Windows	3.6 Branch	en-US
2010-05-12 11:20:12	279135	8410: All Tab options unchecked	Firefox	3.6 Branch	en-US
2010-01-17 02:21:29	257854	8410: All Tab options unchecked	Firefox	3.6 Branch	en-US
2010-01-11 20:56:35	232877	8410: All Tab options uncheck	Firefox	3.6 Branch	en-US
2010-01-07 10:50:21	230165	8410: All Tab options uncheck	Firefox	3.6 Branch	en-US
2009-10-22 03:22:31	229120	8410: All Tab options uncheck	Firefox	3.6 Branch	en-US
	213420	8410: All Tab options uncheck	Firefox	3.6 Branch	en-US

search test results: recent failures | re

cases most frequently marked as unclear | testgroup popularity |

results with comments

search testcases: testcases needed! | recently added | recently changed | by id:



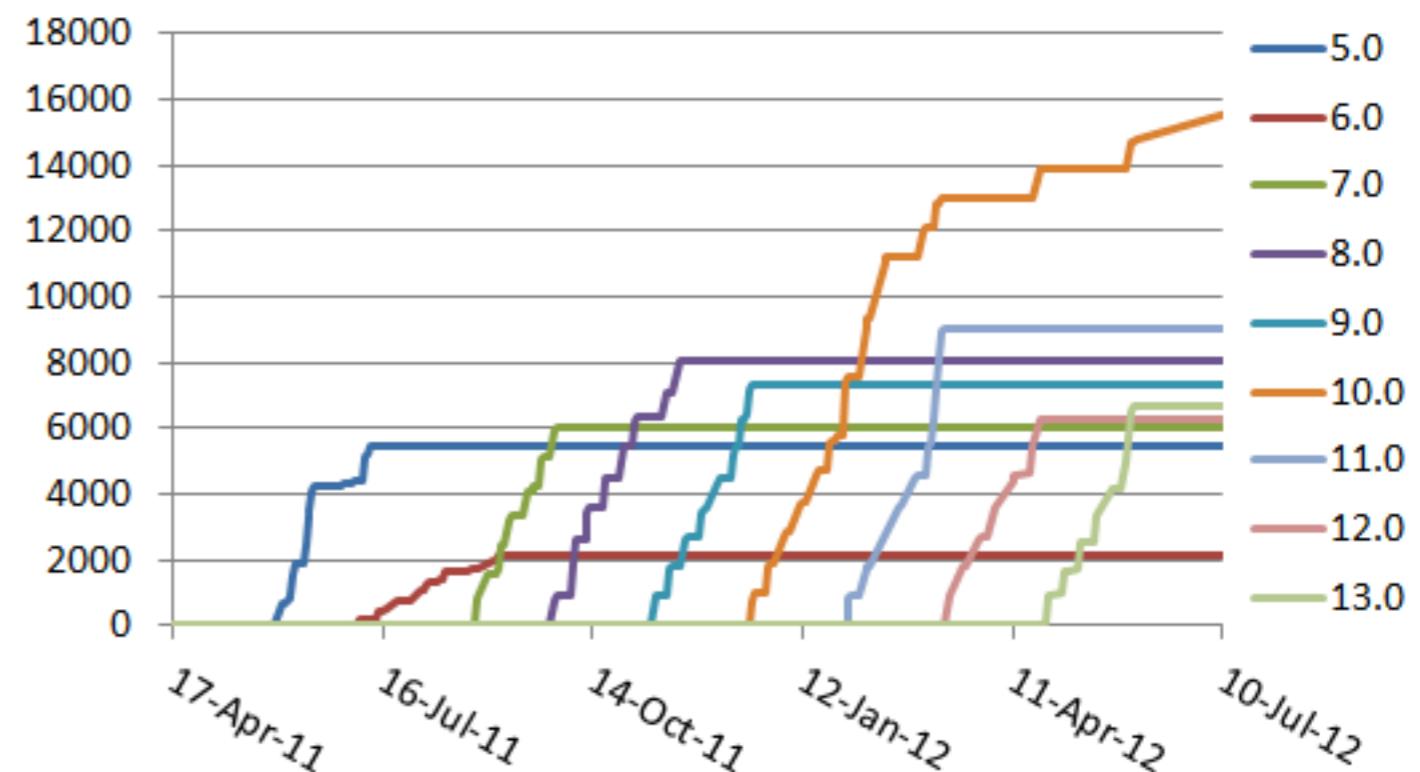
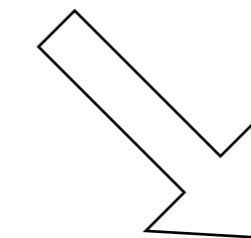
... but twice as many  
test executions per day



6,000 - 9,000  
executions/release

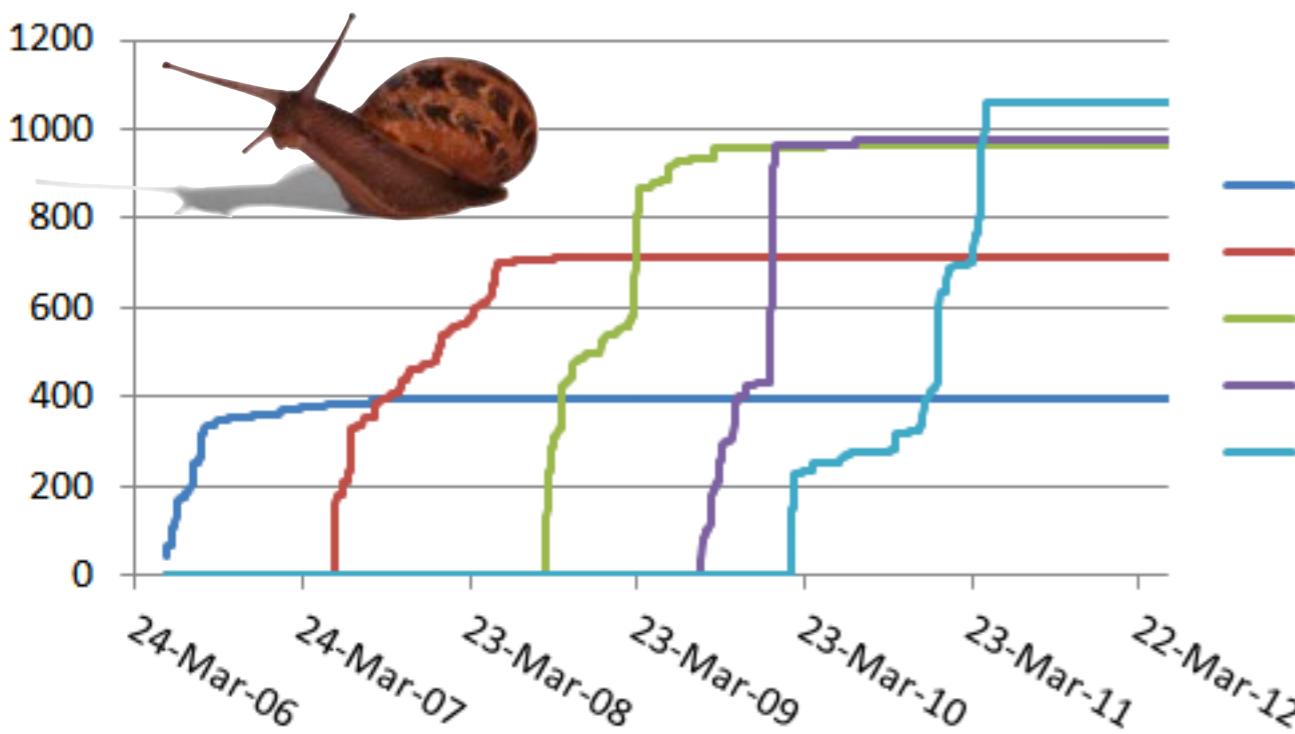
35,000 - 55,000  
executions/release

# Testing More Intense



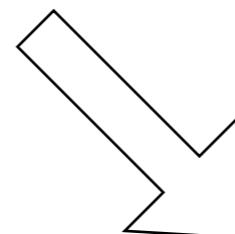
To survive under the time constraints of a rapid release we've had to cut the fat and **focus on those test areas which are prone to failure**, less on ensuring legacy support



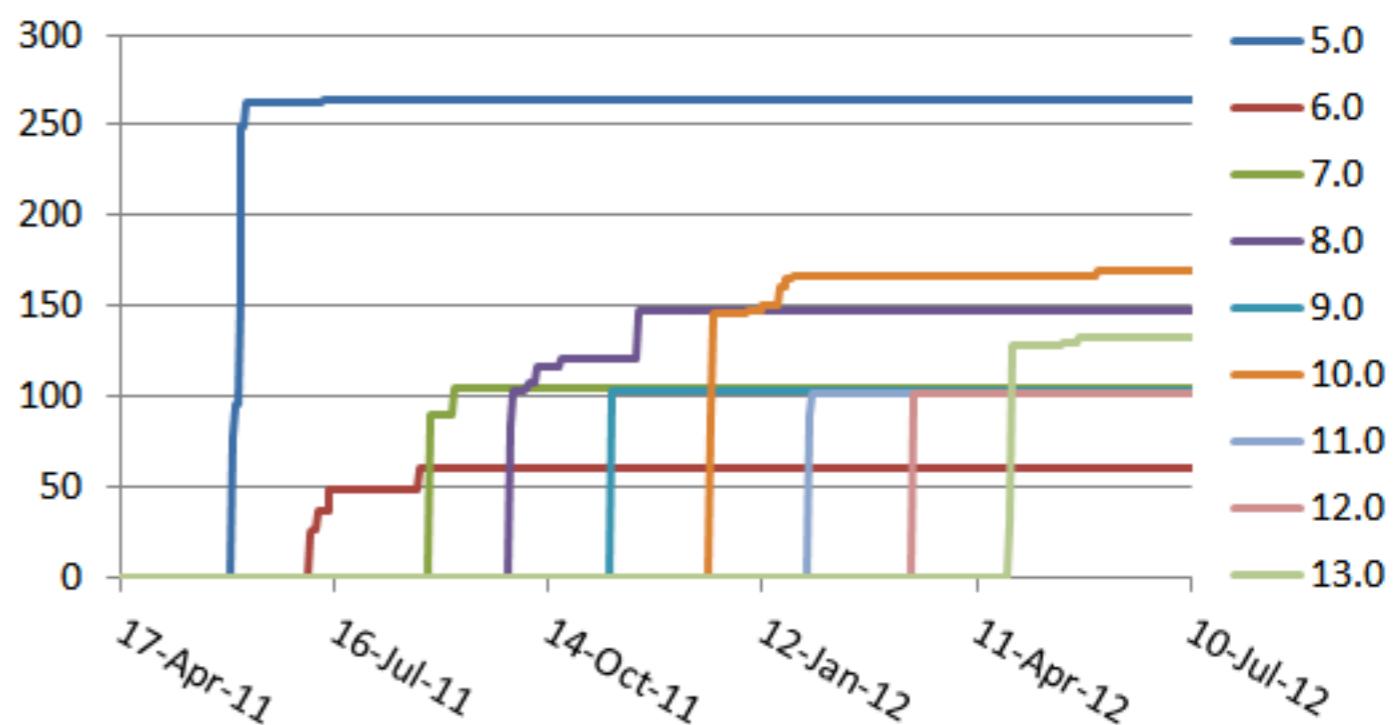


700 - 1,100  
different test cases

# Smaller Scope of Testing

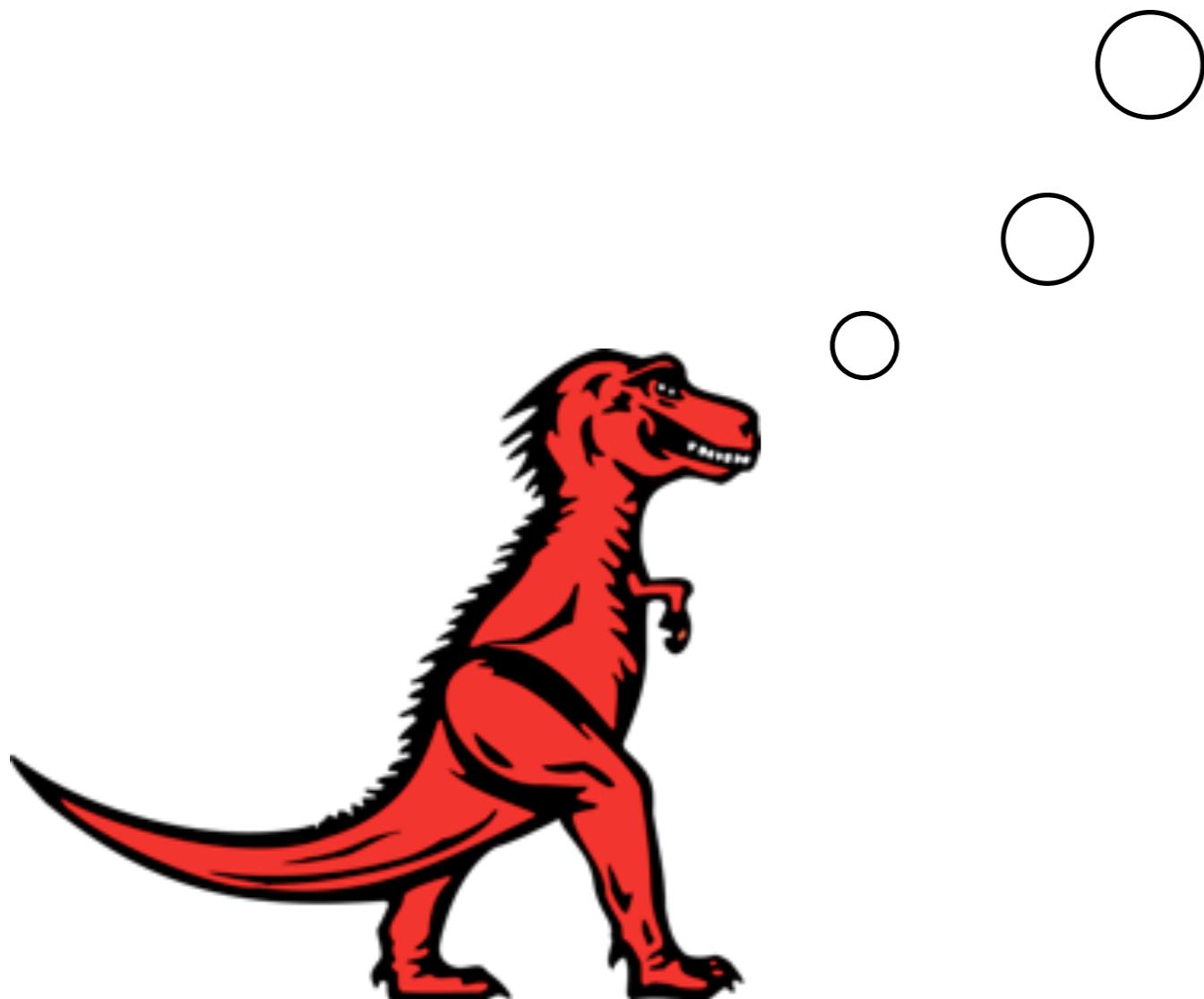


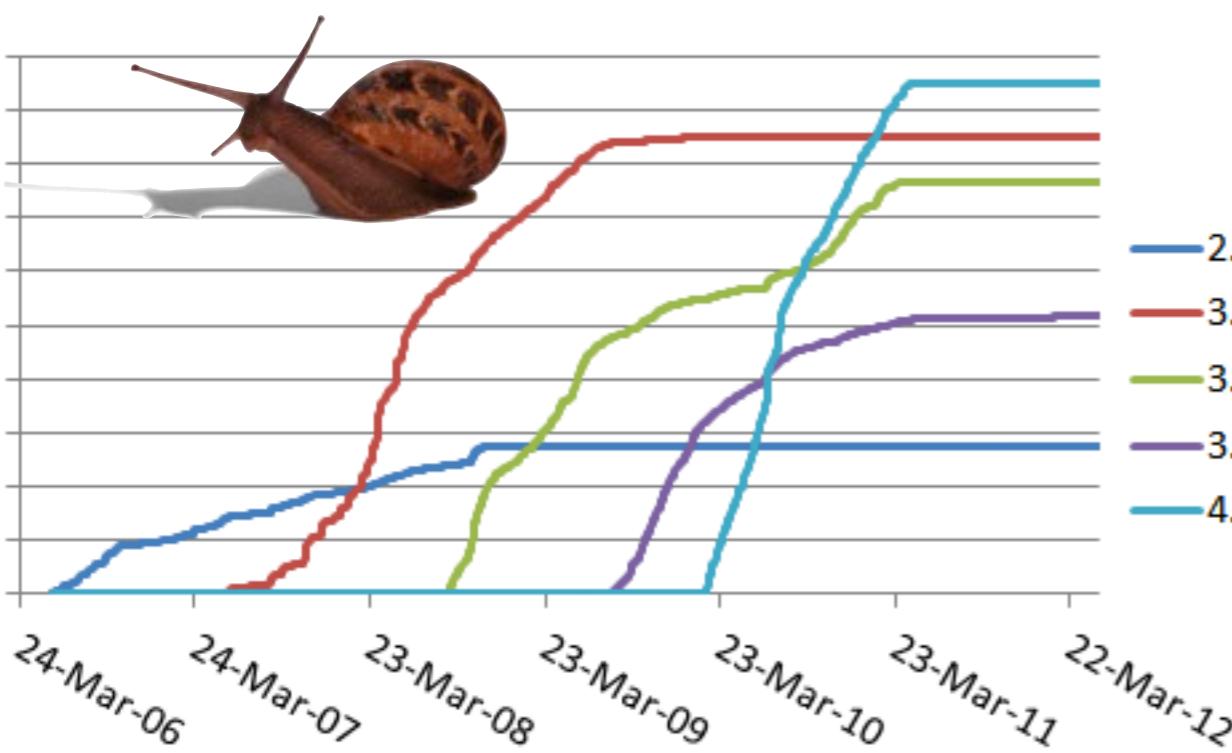
100 - 170 different  
test cases



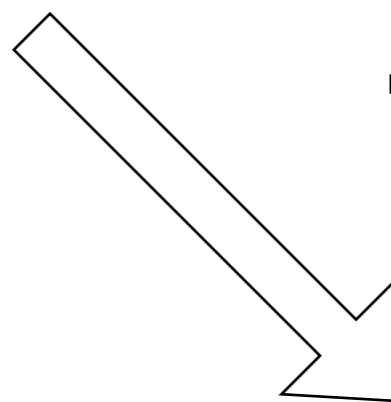
a **fixed** set of tests for **areas prone to regression**  
(Flash plugin testing for example)  
and

a **dynamically** changing set of tests to cover recent  
regressions we chemspilled for and **high risk** features





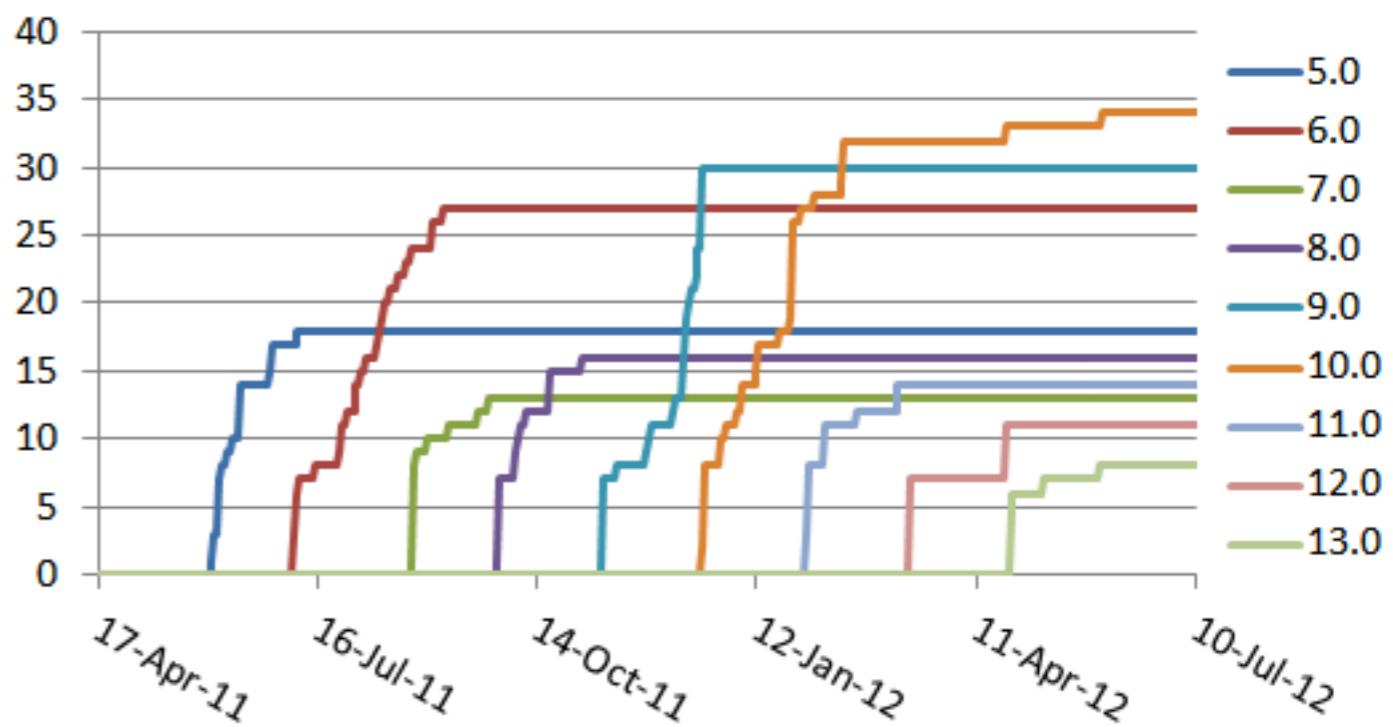
1,000 - 1,900  
testers



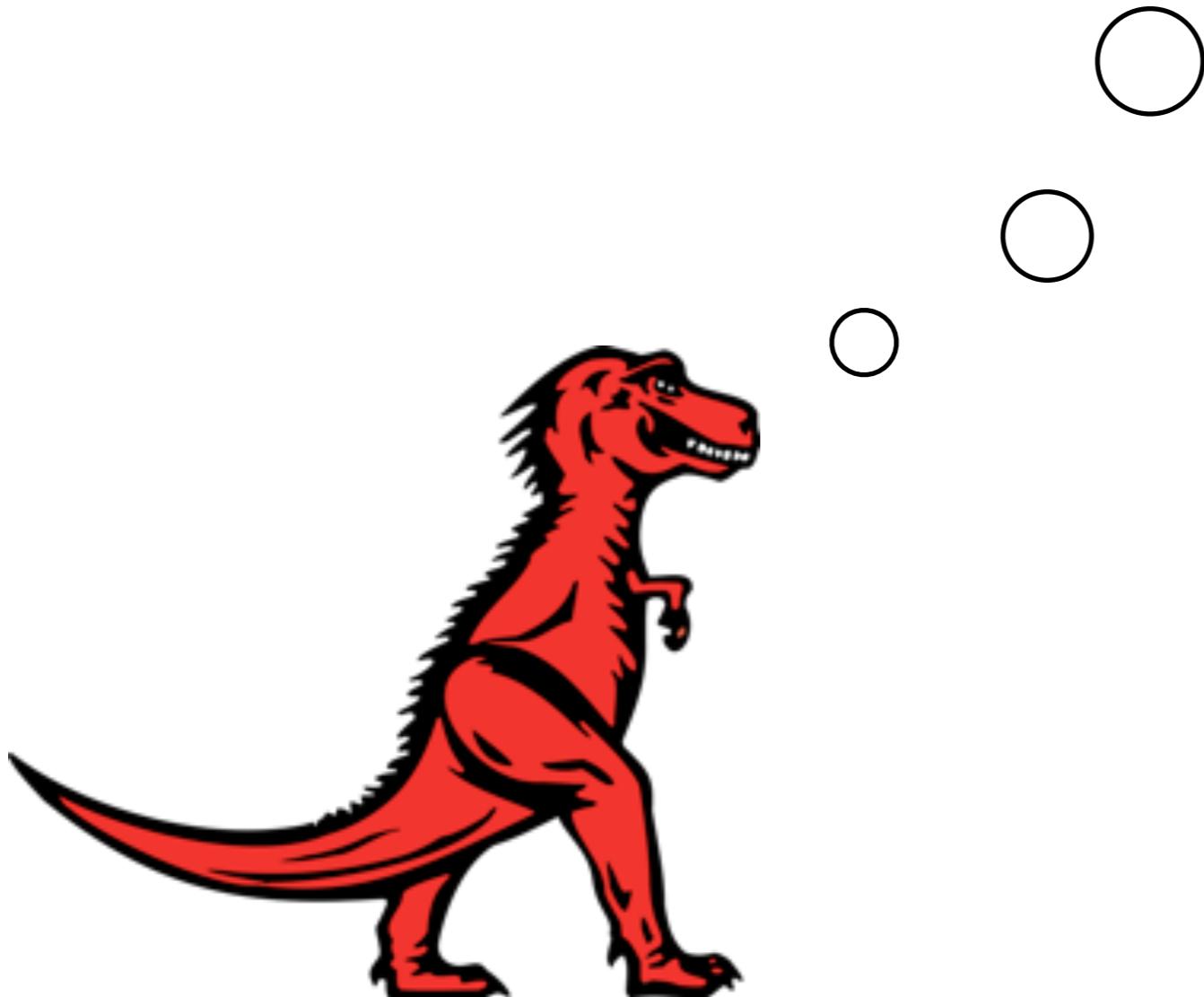
**Smaller  
Testing Team**



8 - 34 testers



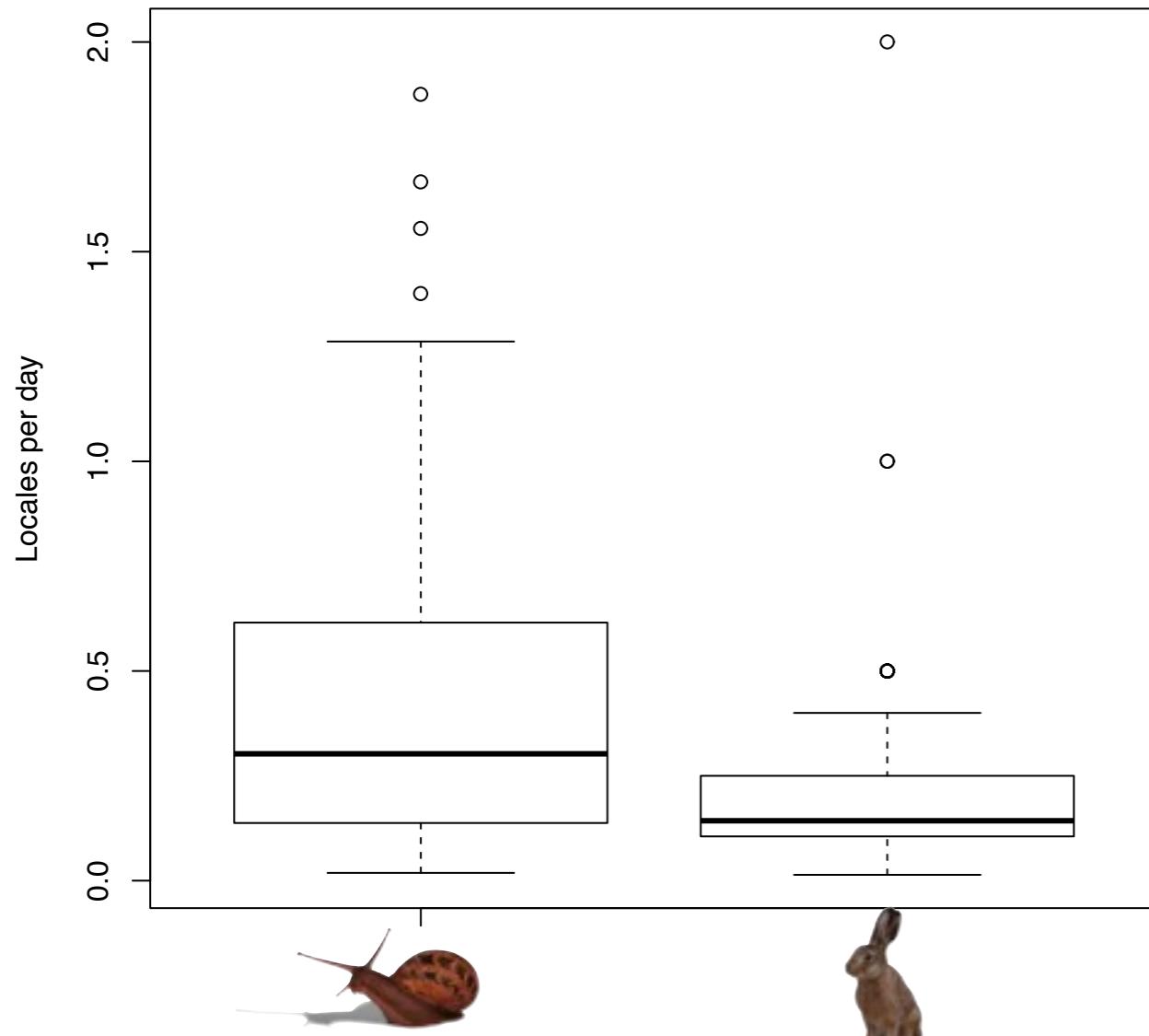
The weakest point [in rapid release] is that it's **harder to develop a large community** which more accurately represents the scale and variability of the general population. Frequently this means that we don't hear about issues until after release, in effect **turning our release early adopters into beta testers**



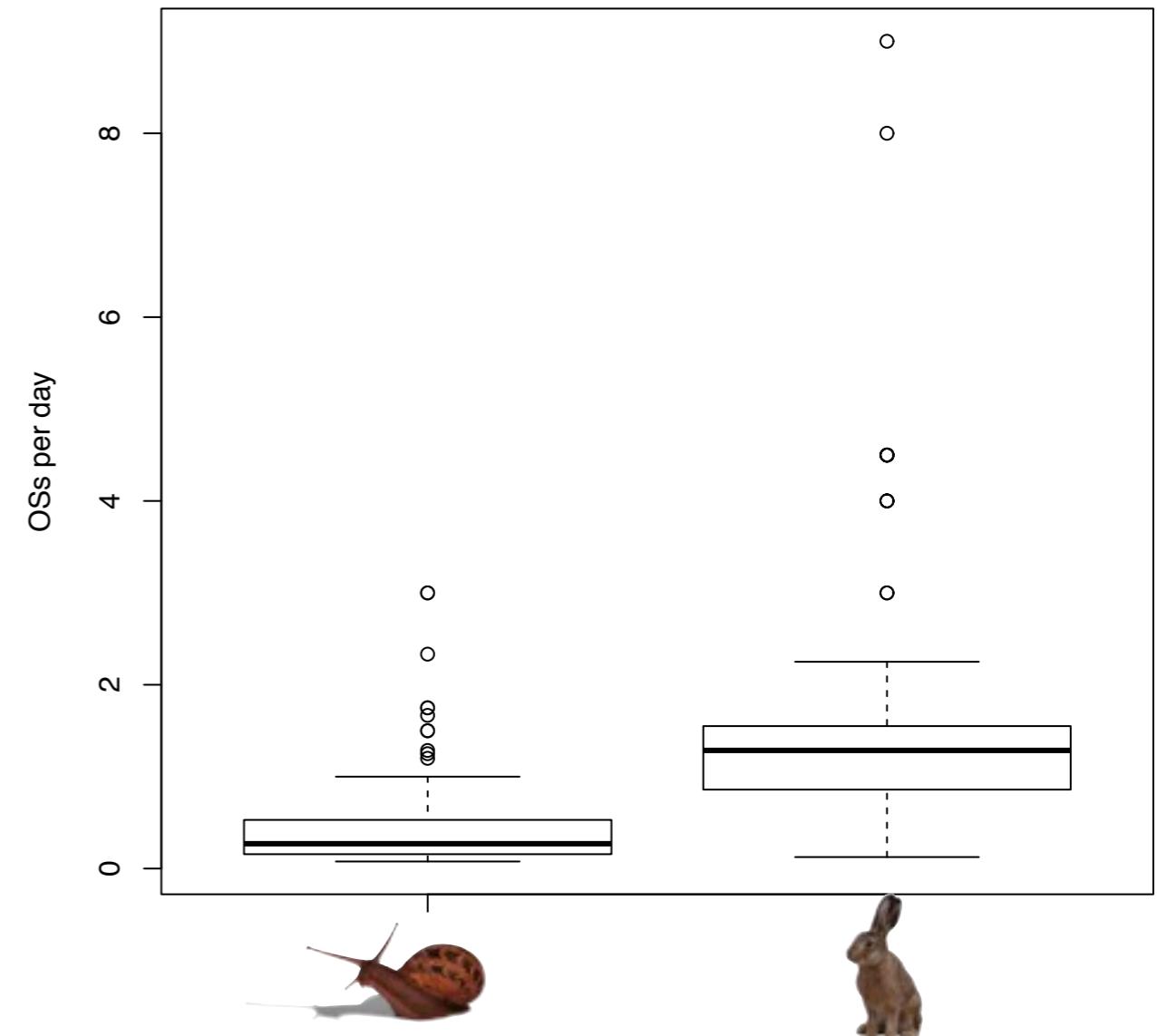
1) the core team has remained largely unchanged since adopting rapid release 2) the contract team has nearly doubled ... **We can scale up our team much faster through contractors than through hiring.** The time afforded to us to make the switch to rapid releases left little room for failure which is why we took that approach.



# Test Priorities



less locales being  
tested ...



... but more  
operating systems

we now **distribute across Betas**.  
For example, we might test Windows 7,  
OSX 10.8, and Ubuntu in one Beta then  
Windows XP, Mac OSX 10.7, and  
Ubuntu in another Beta

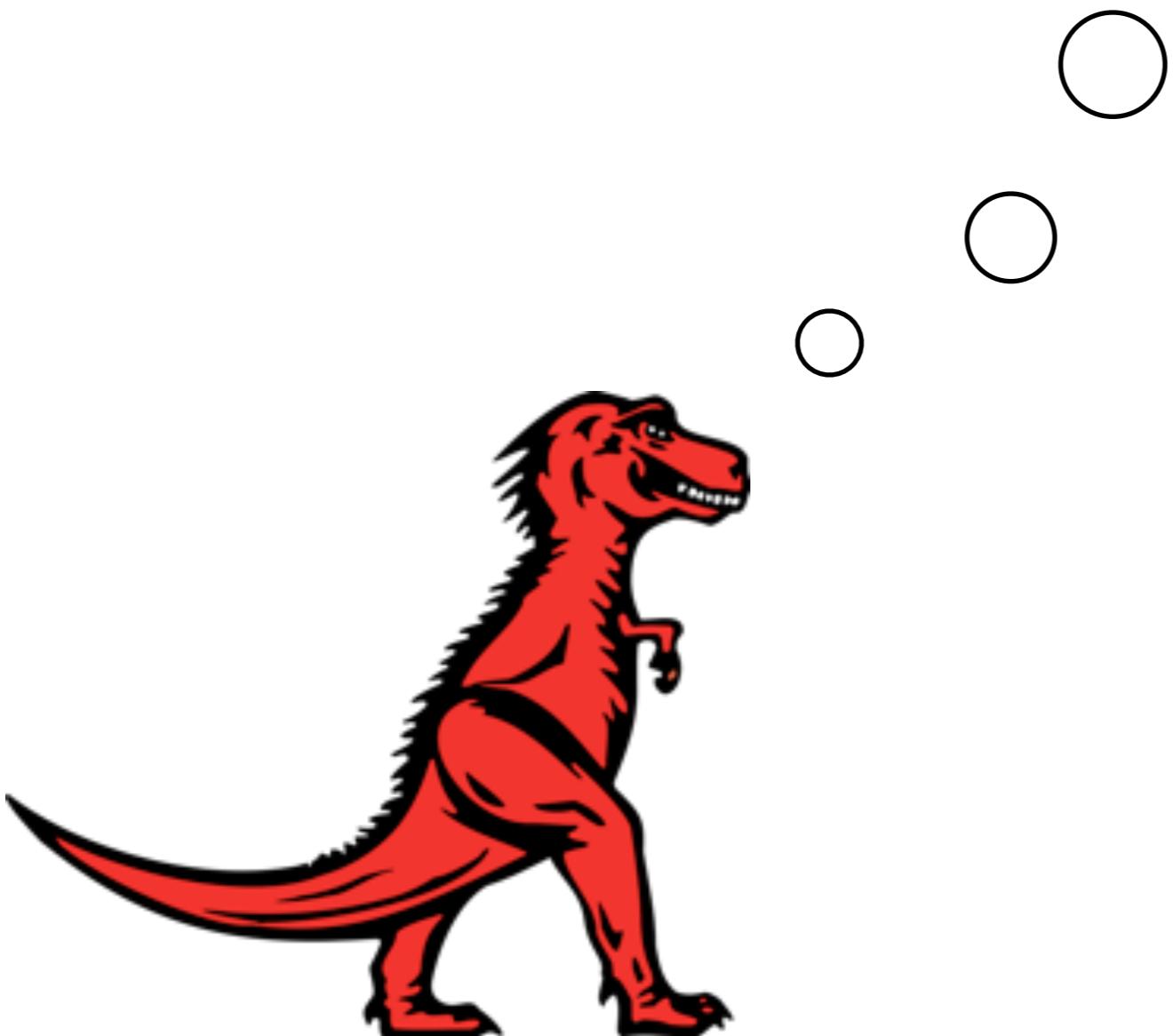


In Other Words ...



**Testing has to  
become  
more  
Focused**

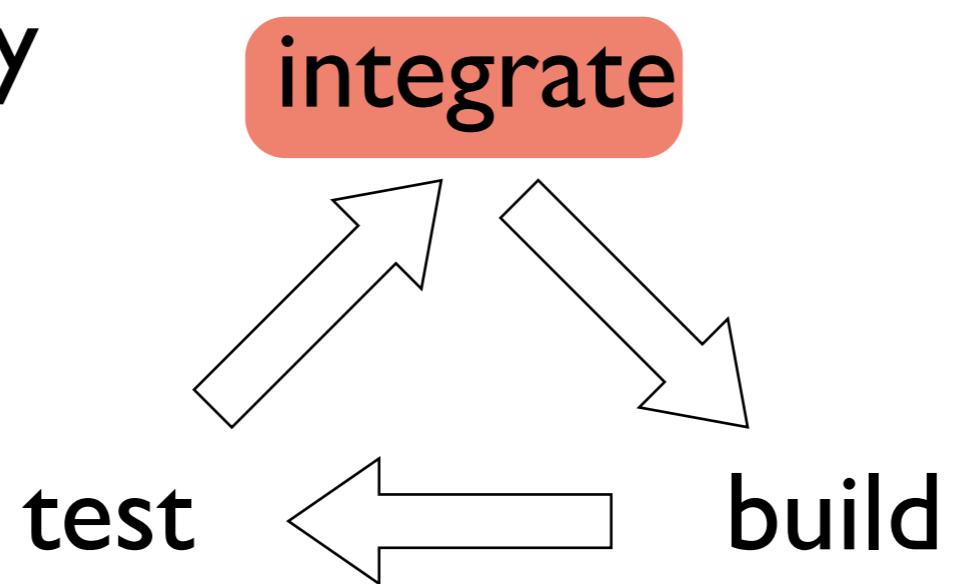
The greatest strength [of rapid releases] is that **the scope of what needs to be tested is narrow** so we can focus all of our energy on deep diving into a few areas



# Release Engineering



in-house/3rd party  
development



*reduced cycle time!*

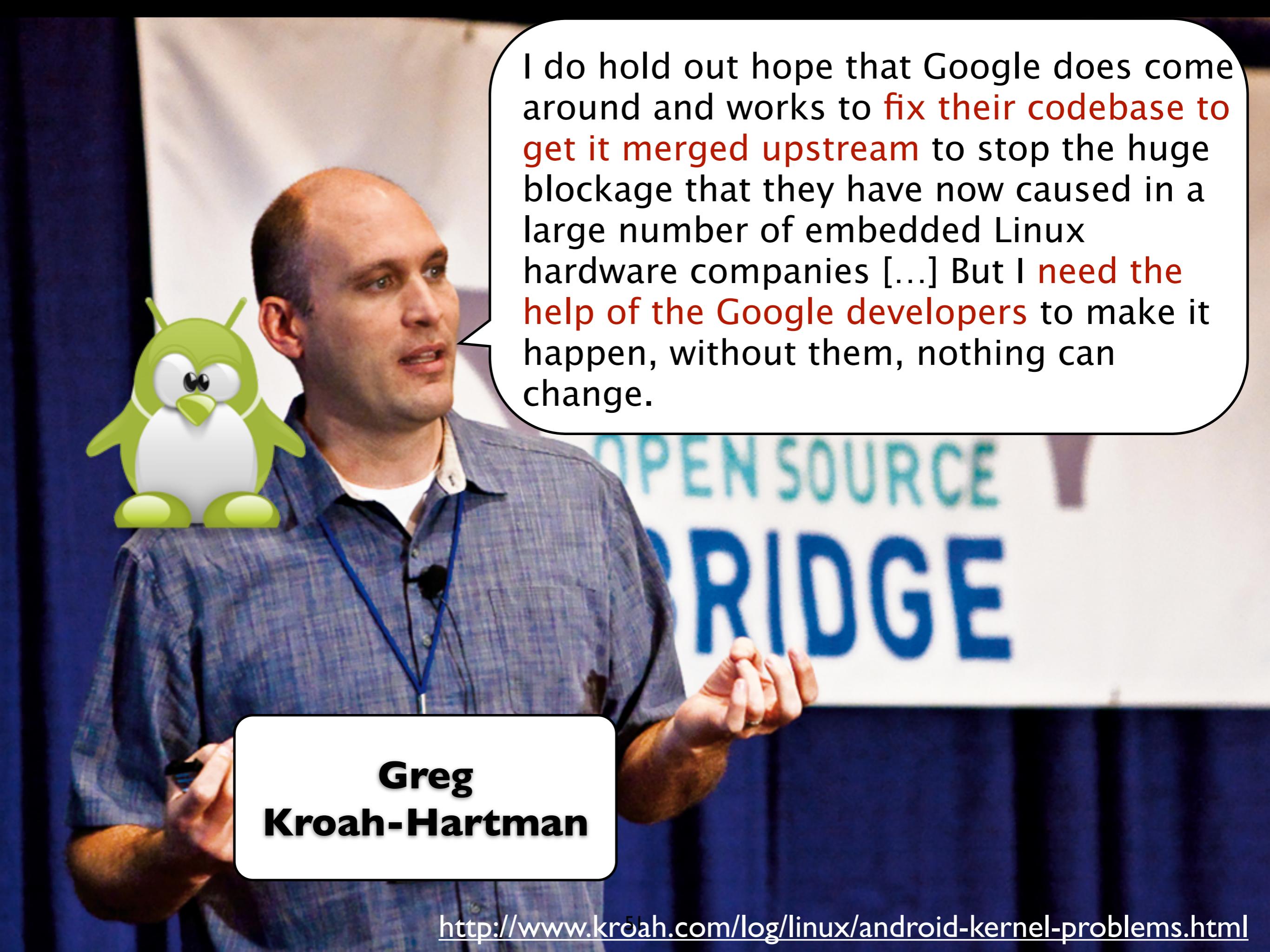


deployment

# Will My Patch Make It?

(And How Fast?)

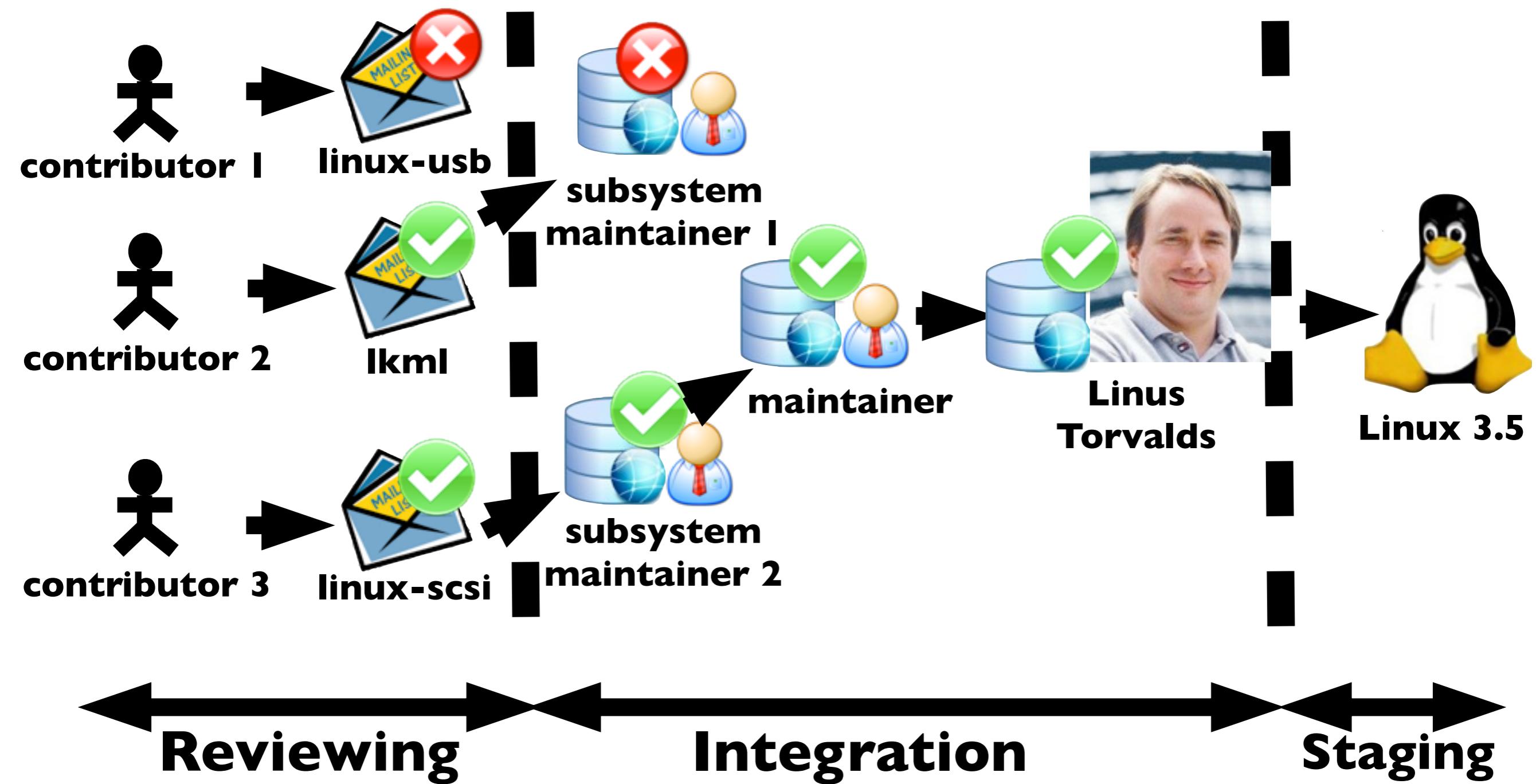




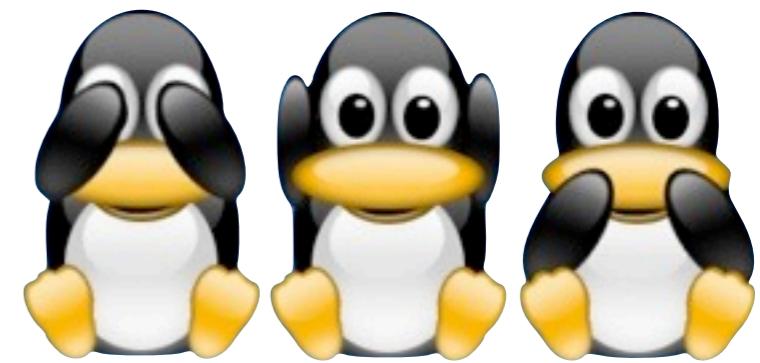
I do hold out hope that Google does come around and works to fix their codebase to get it merged upstream to stop the huge blockage that they have now caused in a large number of embedded Linux hardware companies [...] But I need the help of the Google developers to make it happen, without them, nothing can change.

**Greg  
Kroah-Hartman**

# The Linux Integration Process



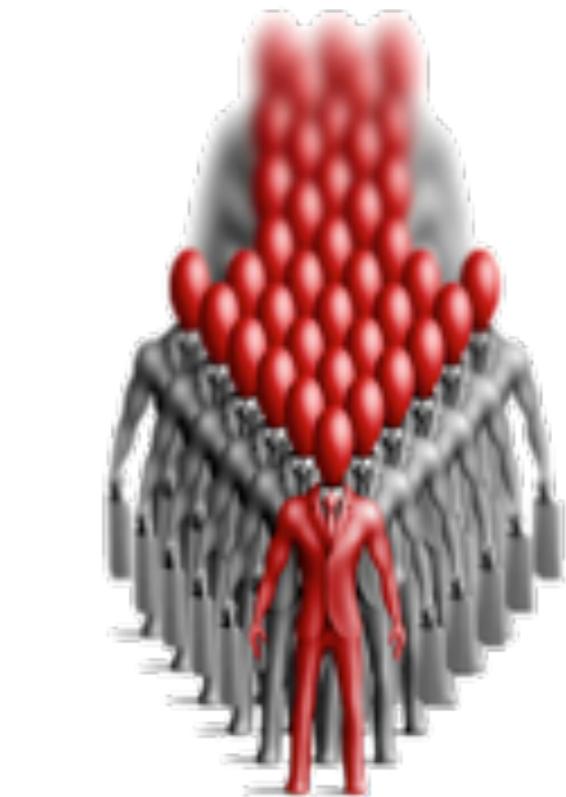
# Research Questions



**How many patches are merged?**

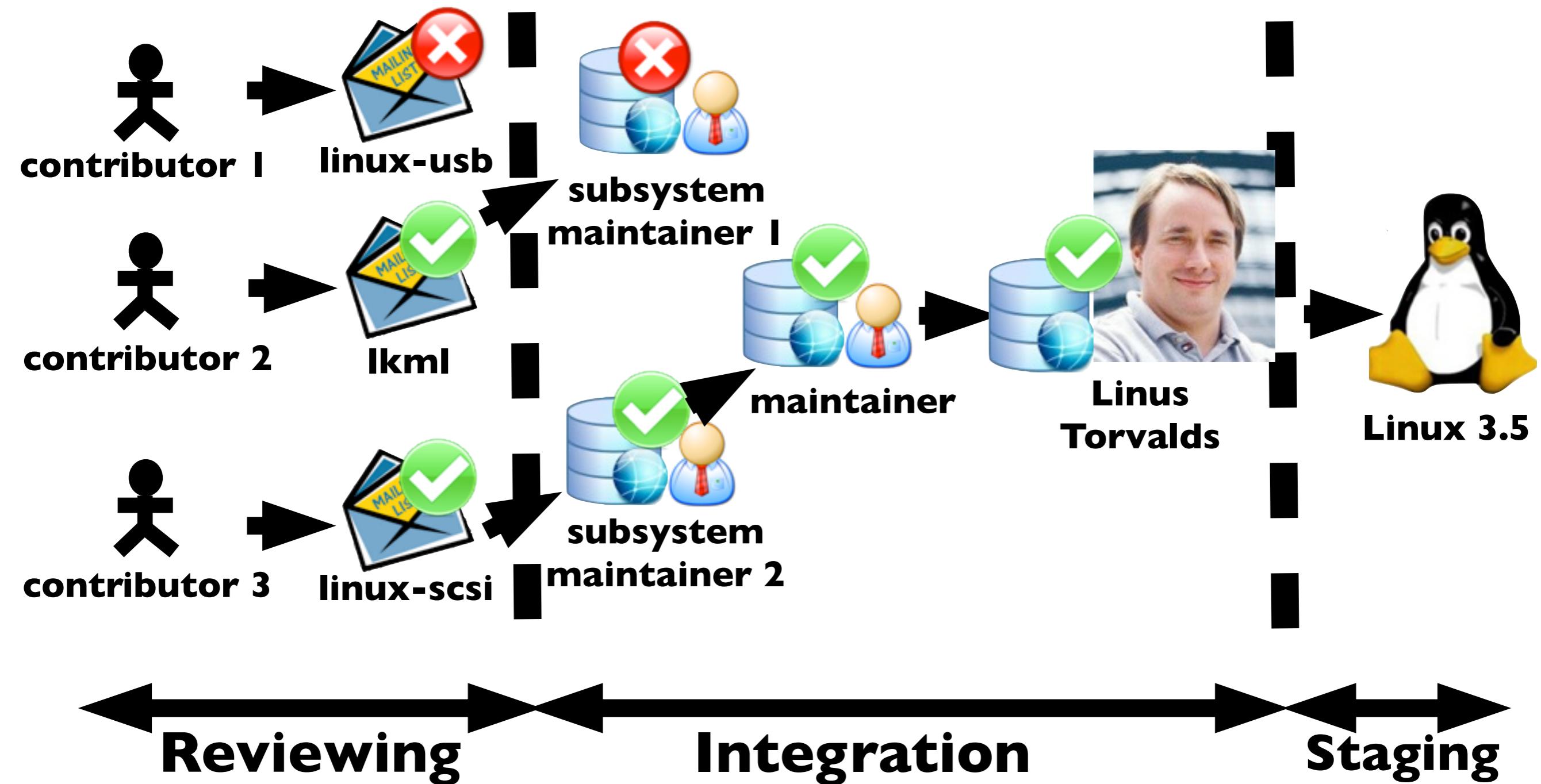


**What kind of patch is merged more likely?**

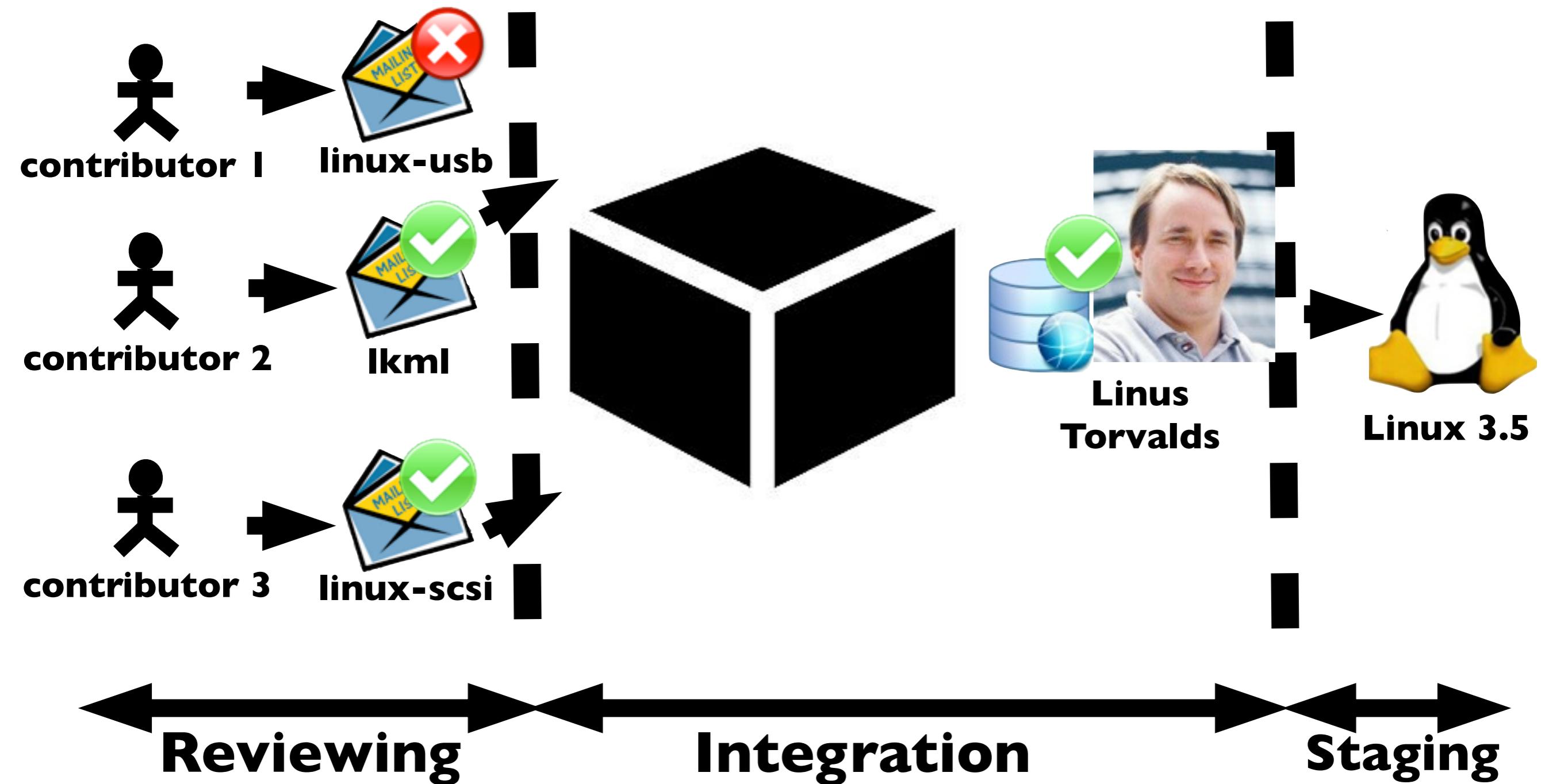


**What kind of patch is accepted faster?**

# Case Study Setup



# Case Study Setup



# Case Study Setup



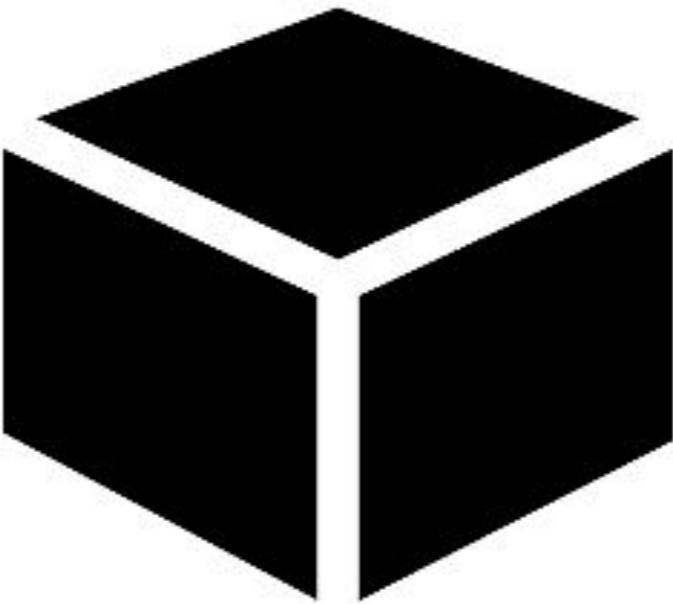
**linux-usb**



**lkml**



**linux-scsi**



**Linus  
Torvalds**

# Case Study Setup



**linux-usb**



**lkml**

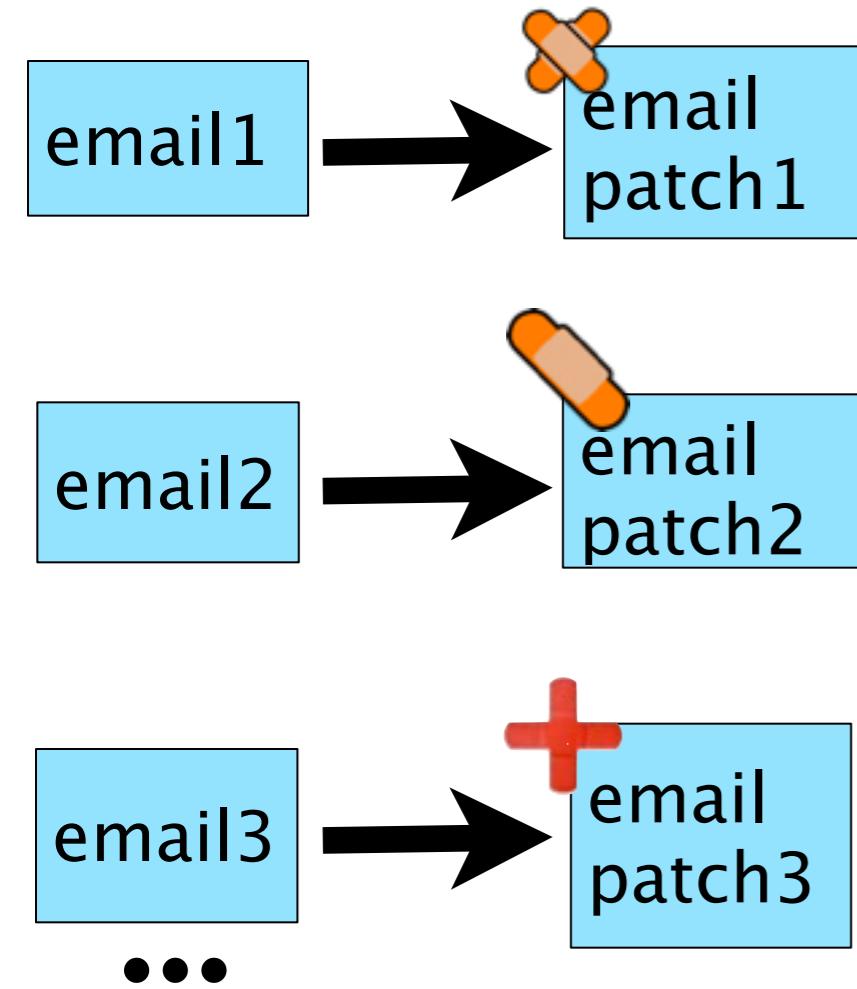


**linux-scsi**



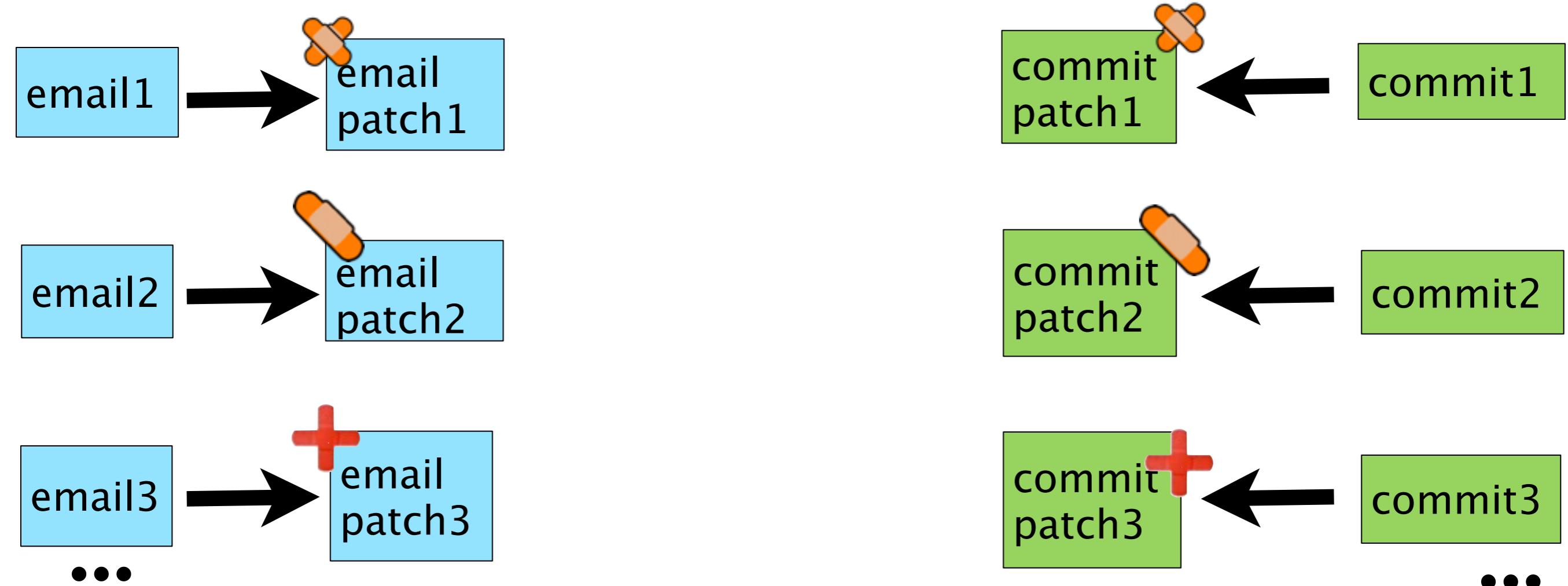
**Linus  
Torvalds**

# Case Study Setup

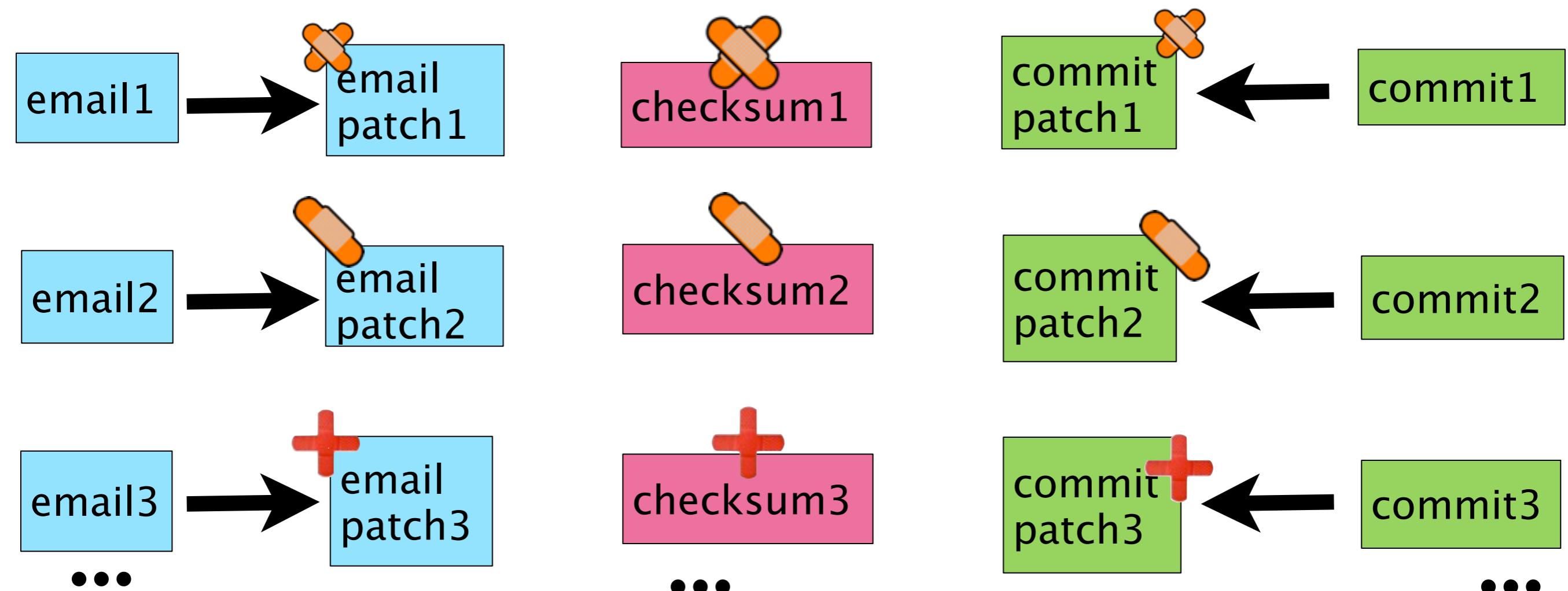


**Linus  
Torvalds**

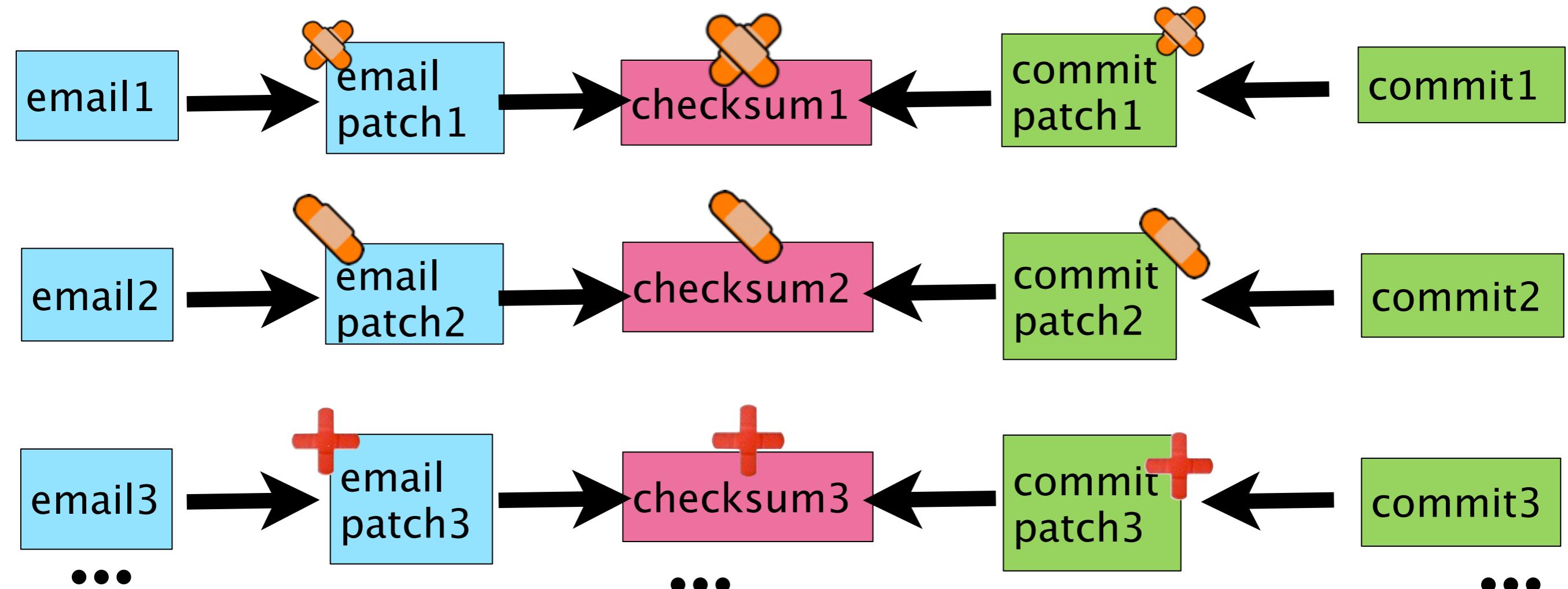
# Case Study Setup



# Case Study Setup



# Case Study Setup





**Commit**



**Experience**

**5 Dimensions  
(29 Patch Metrics)**



**Email**

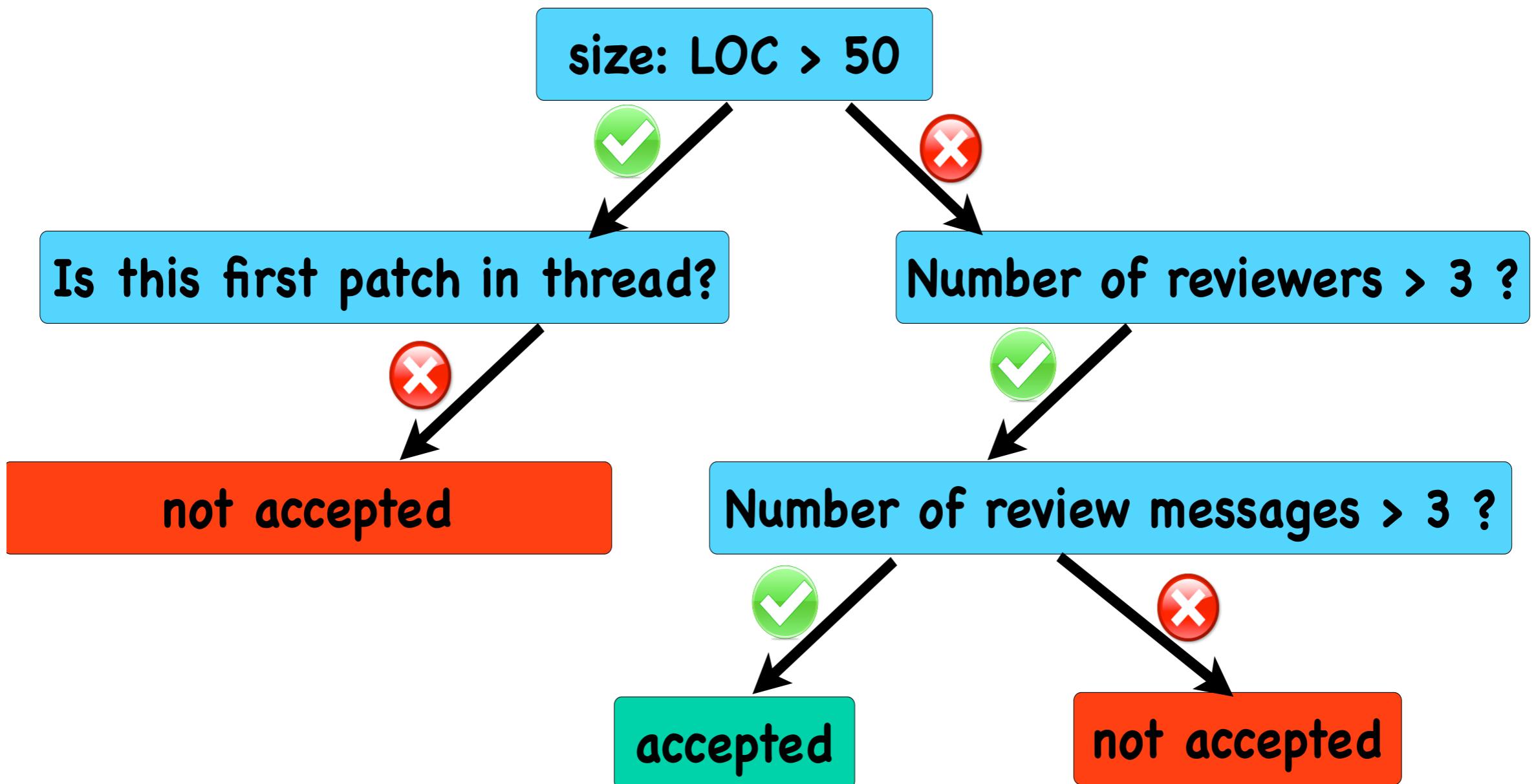


**Patch**



**Review**

# Building Decision Trees

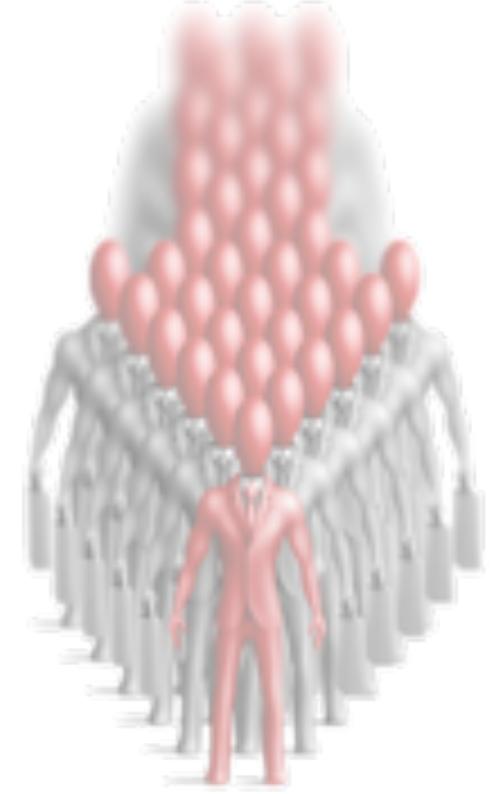




**How many patches are merged?**



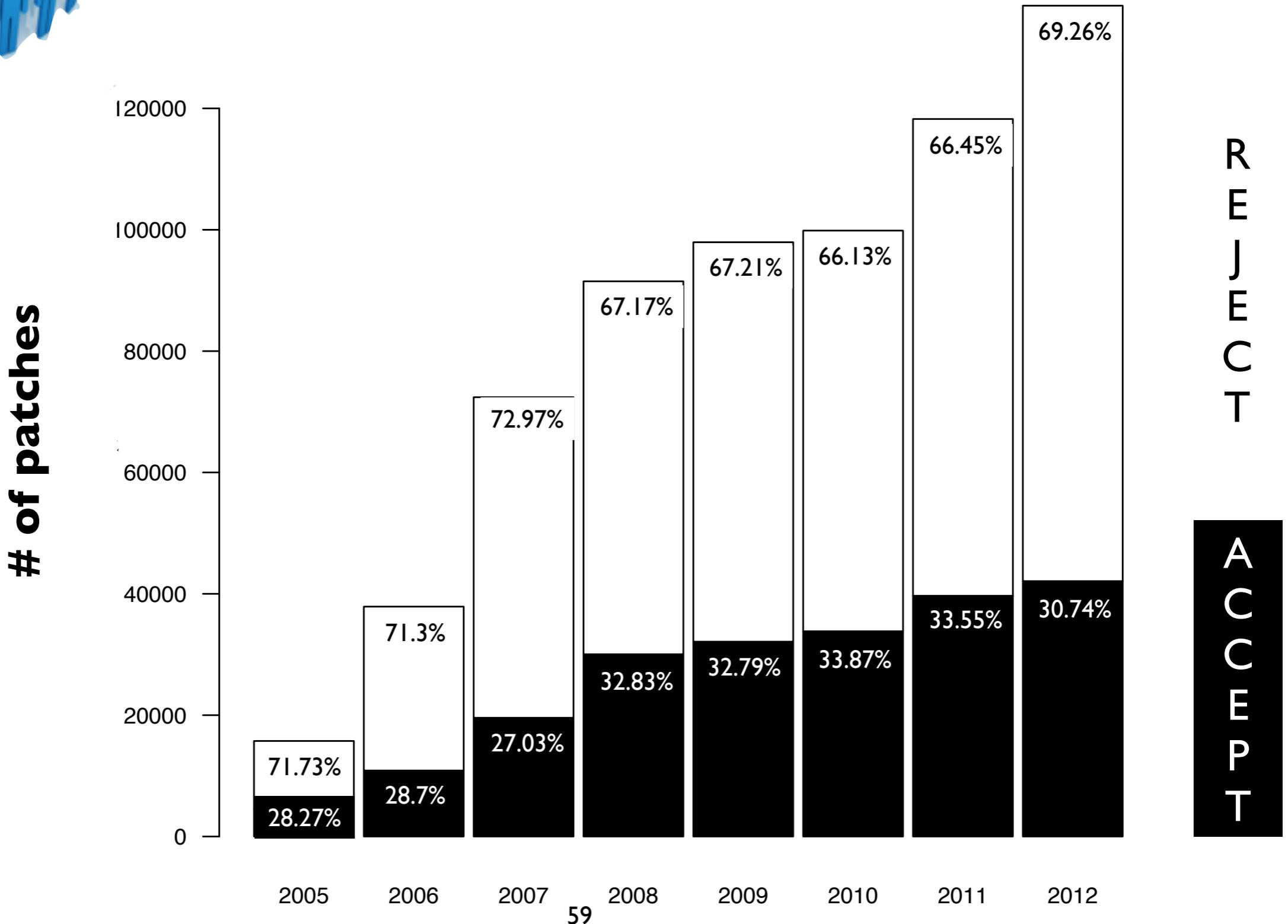
**What kind of patch is merged more likely?**



**What kind of patch is accepted faster?**

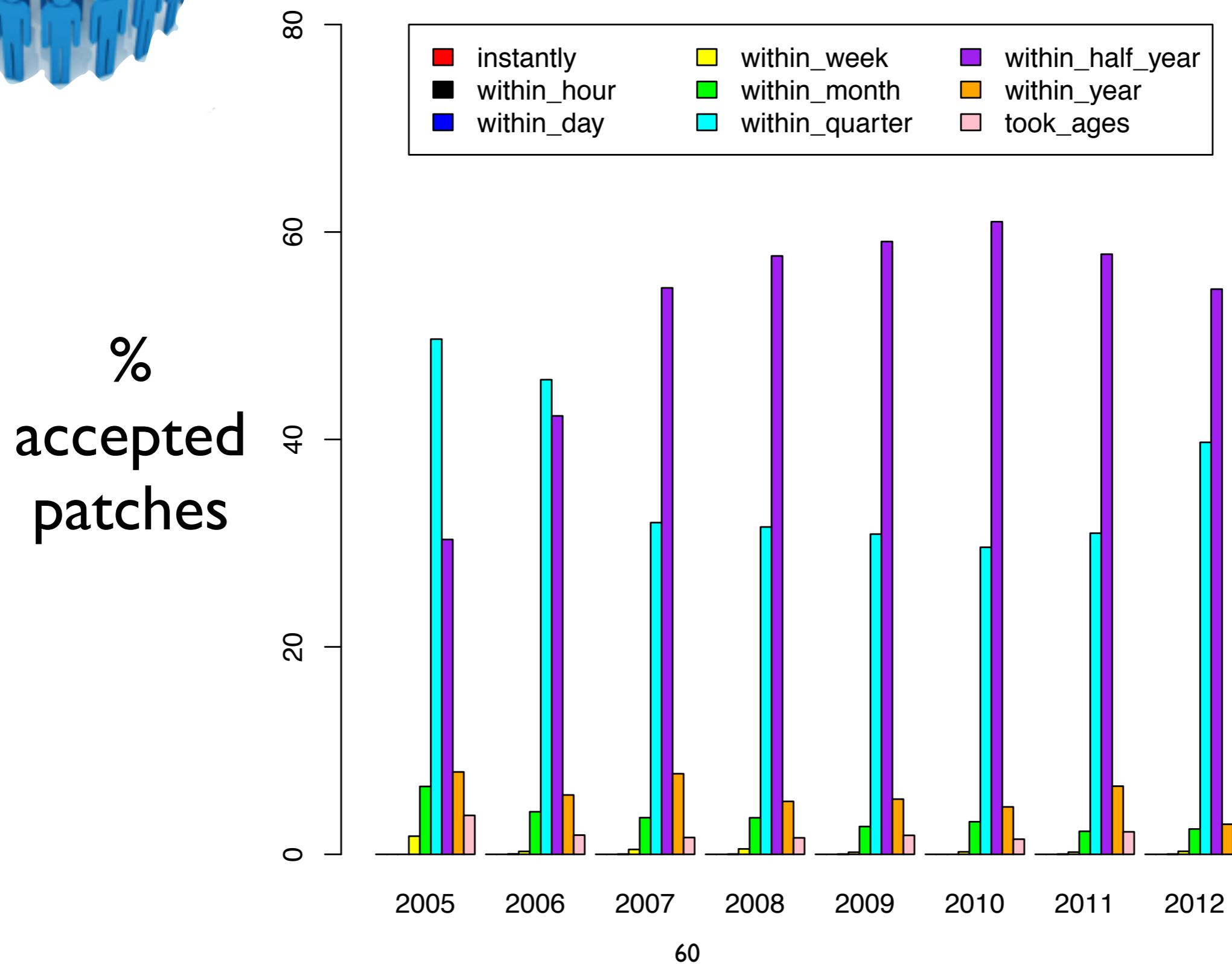


# 33% of Patches Make it ...



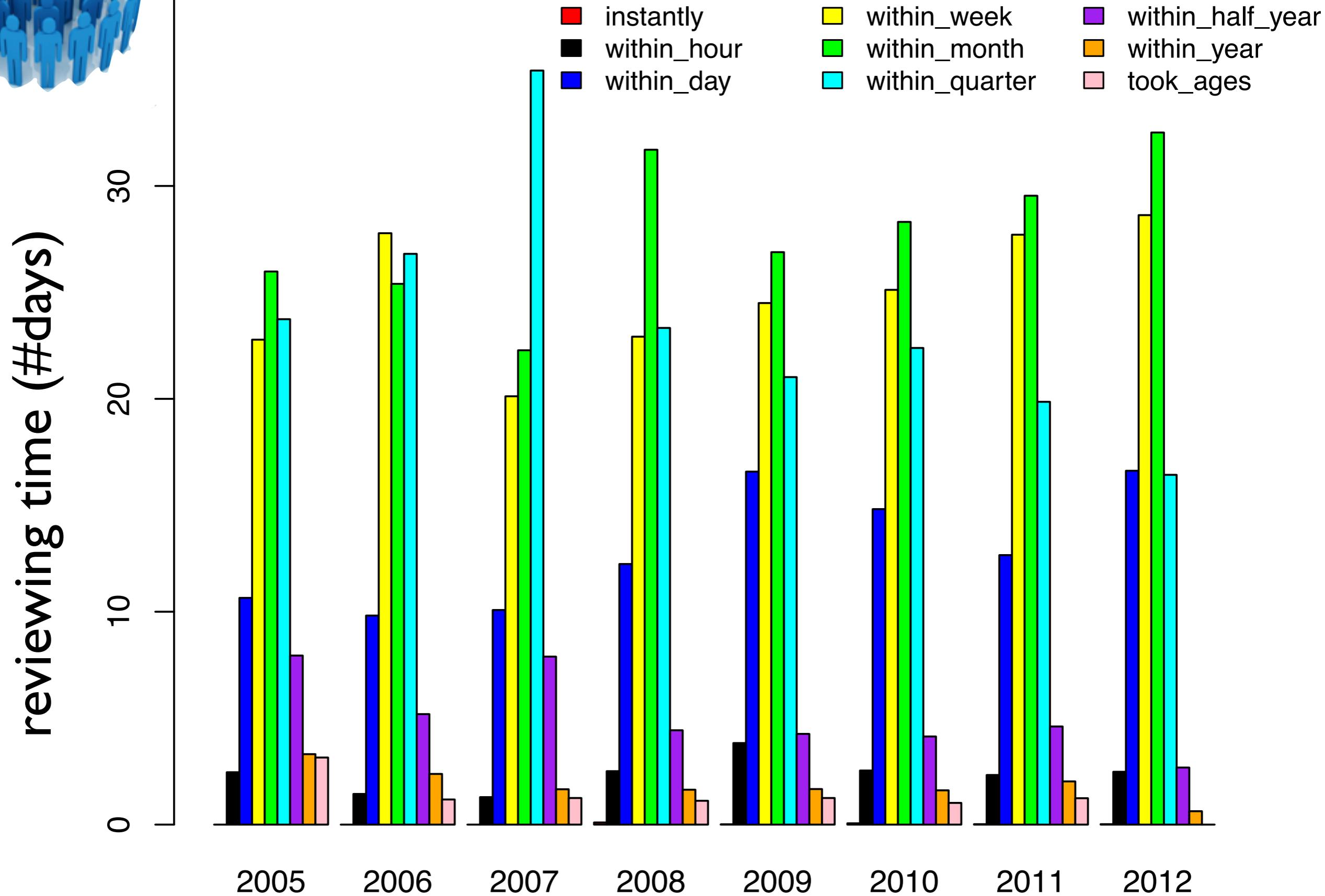


# ... and it takes I~6 Months!



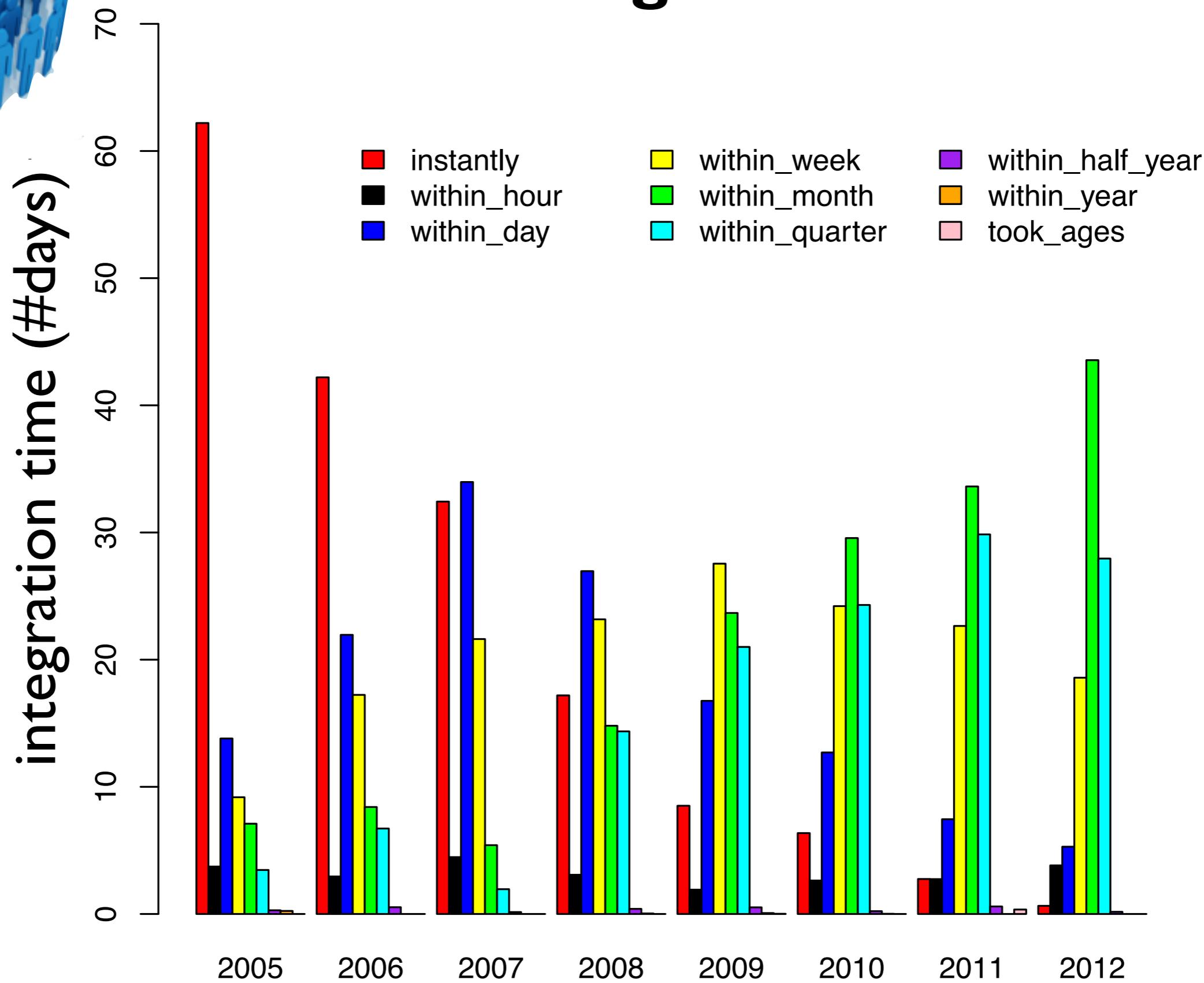


# Reviewing Speeds Up ...





# ... while Integration Slows Down



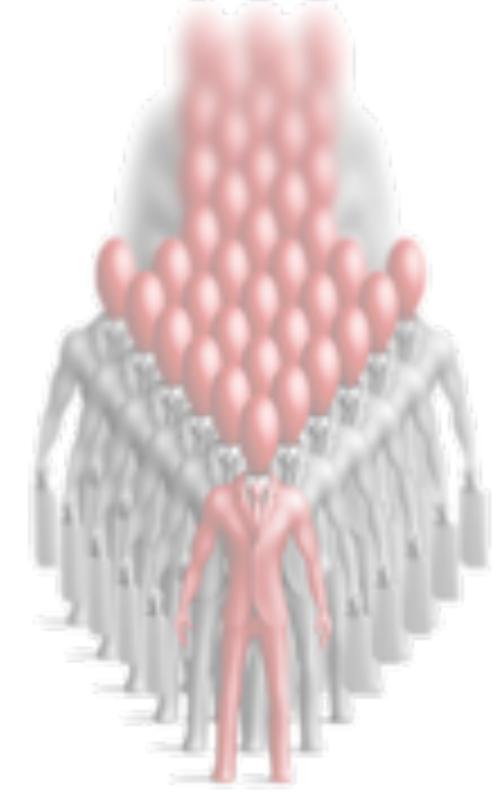


**How many patches are merged?**

33% patch acceptance

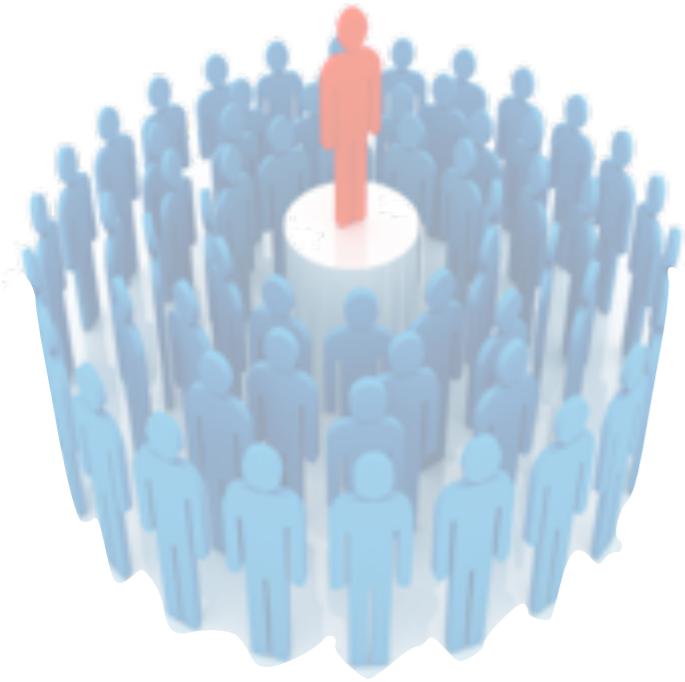


**What kind of patch is merged more likely?**



**What kind of patch is accepted faster?**

integration becomes slower

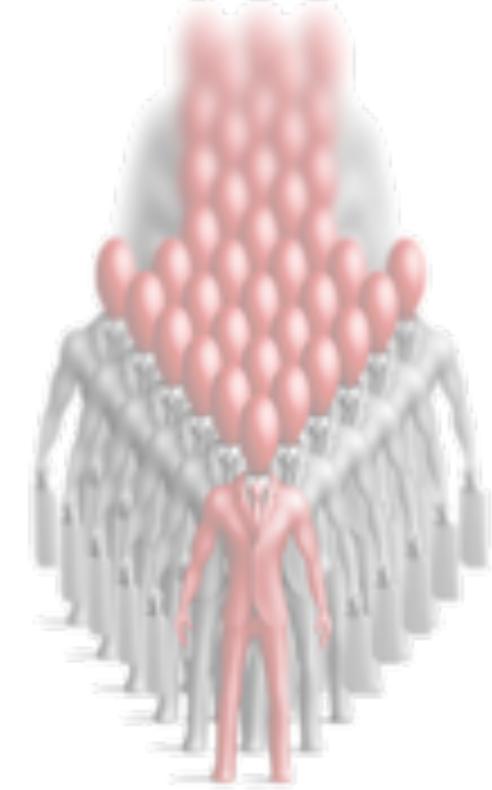


**How many patches are merged?**

33% patch acceptance



**What kind of patch is merged more likely?**

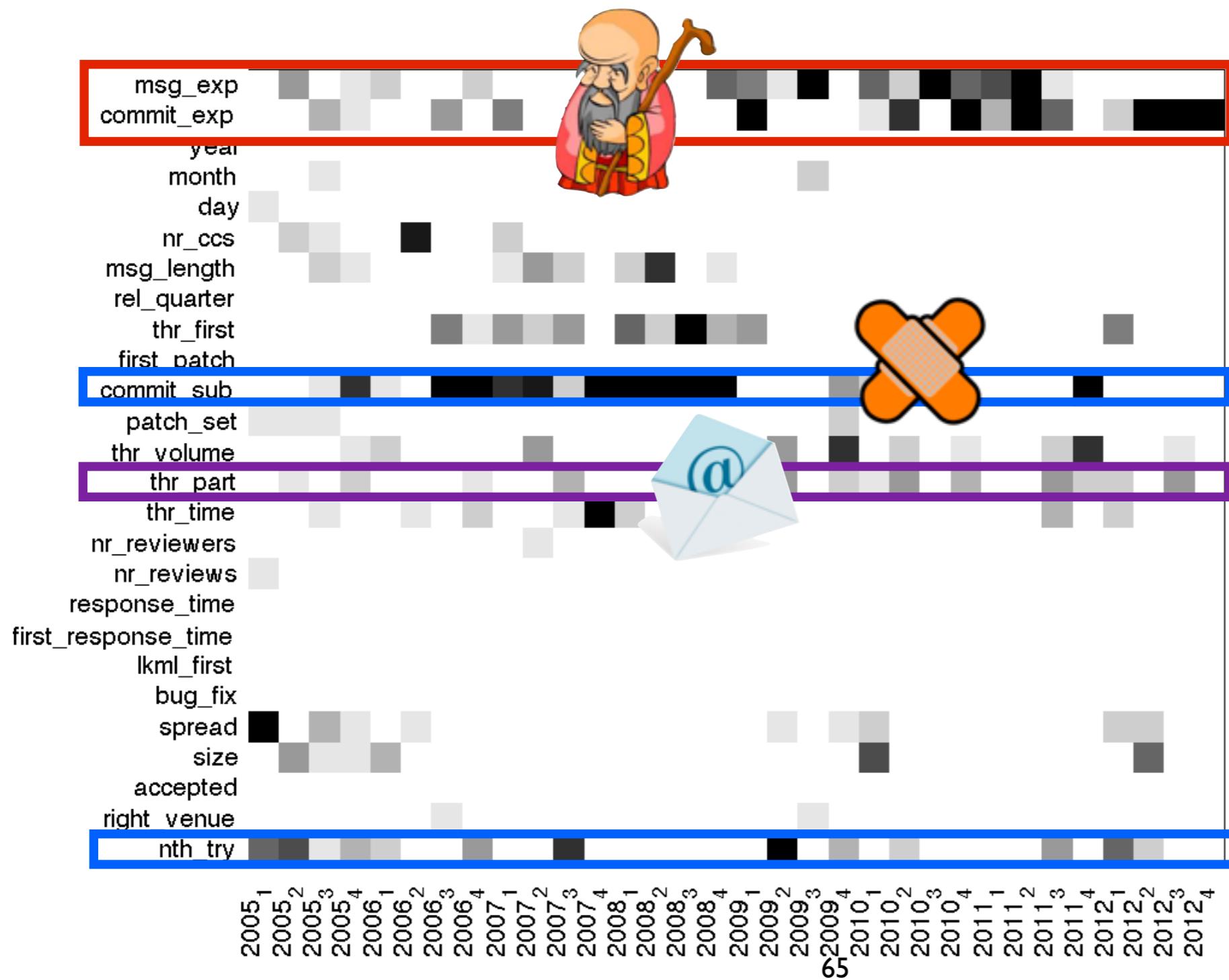


**What kind of patch is accepted faster?**

integration becomes slower

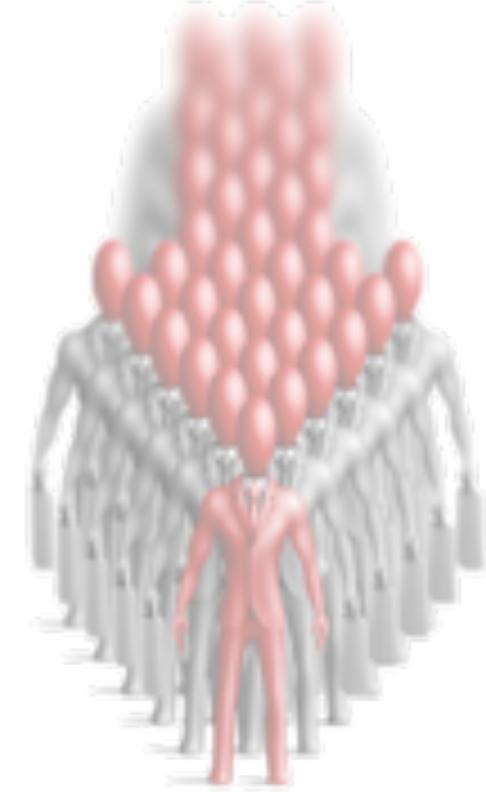
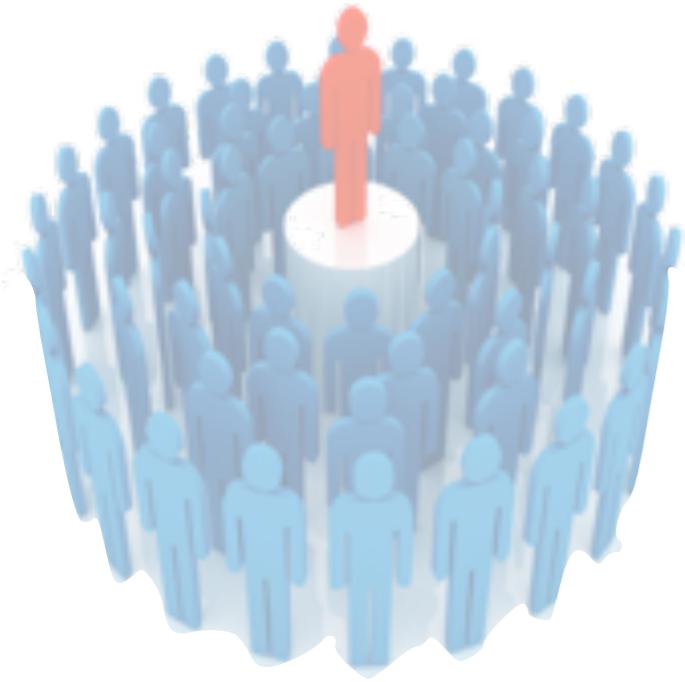


# Experience Matters Most for Patch Acceptance



precision: 73%  
recall: 68.47%





**How many patches are merged?**

33% patch acceptance

integration becomes slower

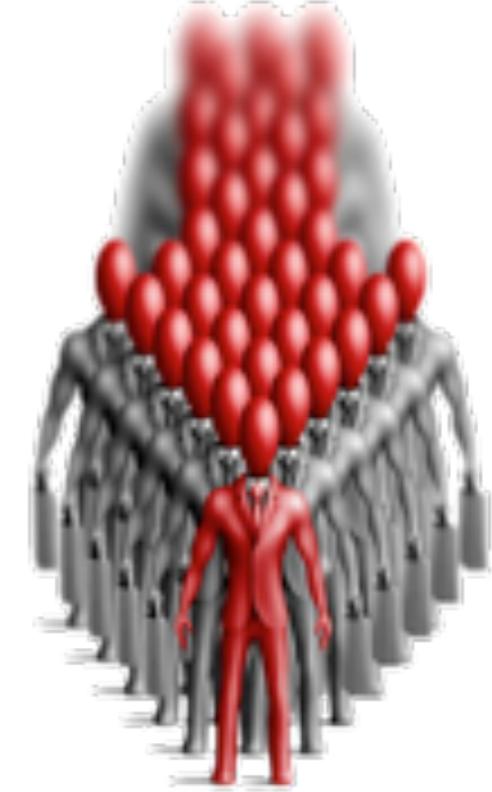
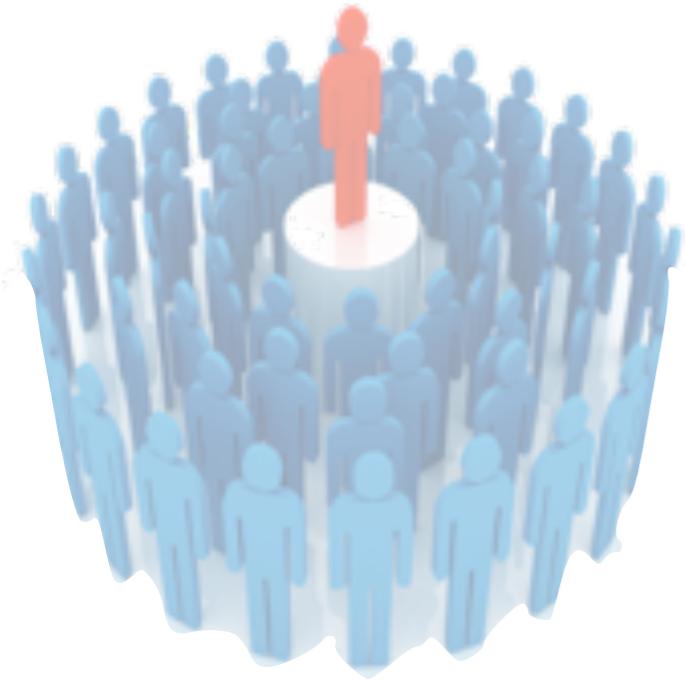
**What kind of patch is merged more likely?**

by experienced developers

mature patch

specific subsystem

**What kind of patch is accepted faster?**



## How many patches are merged?

33% patch acceptance

integration becomes slower

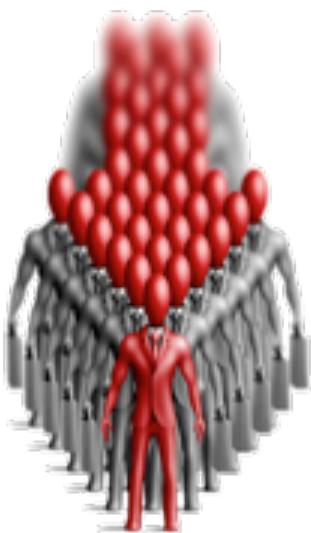
## What kind of patch is merged more likely?

by experienced developers

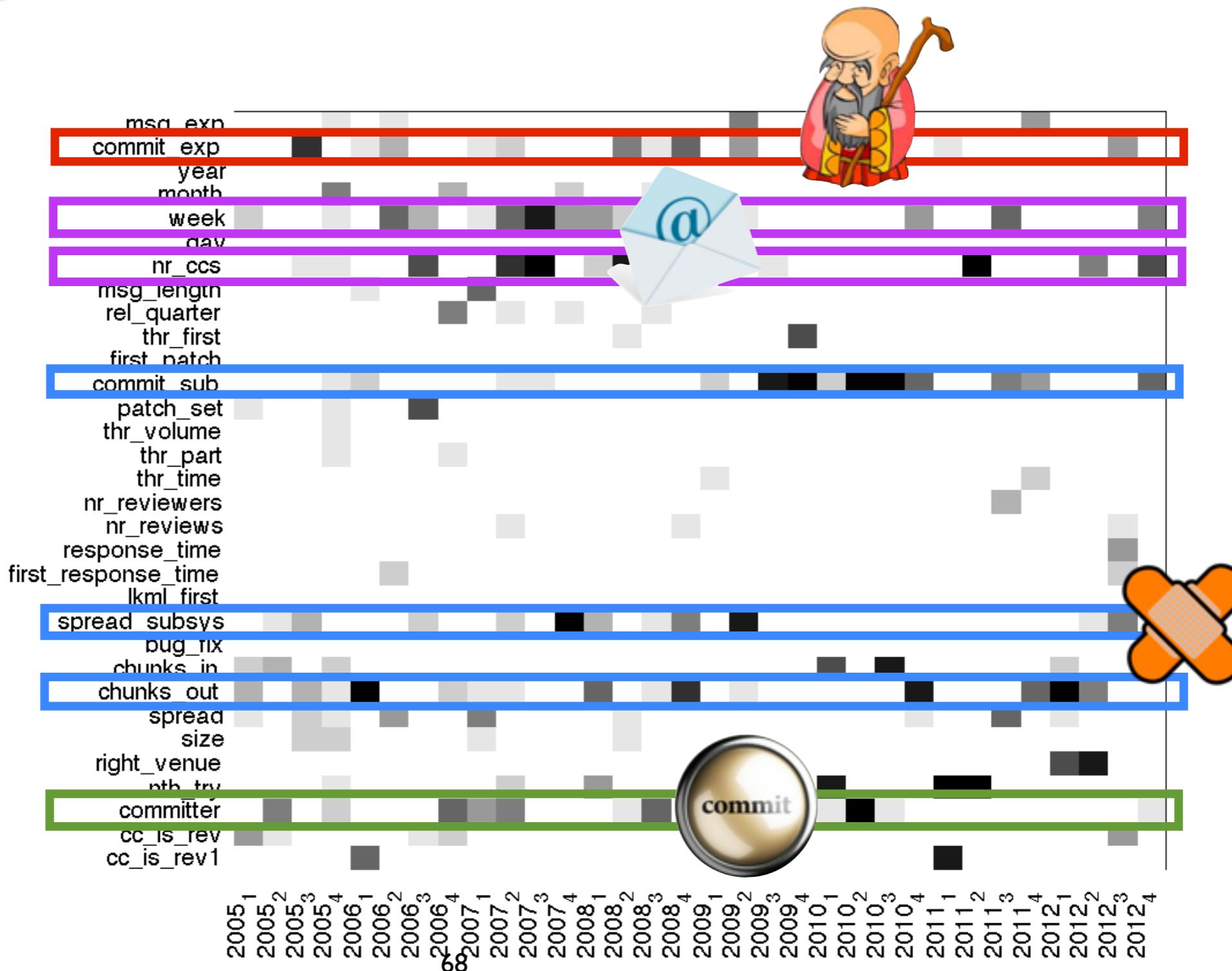
mature patch

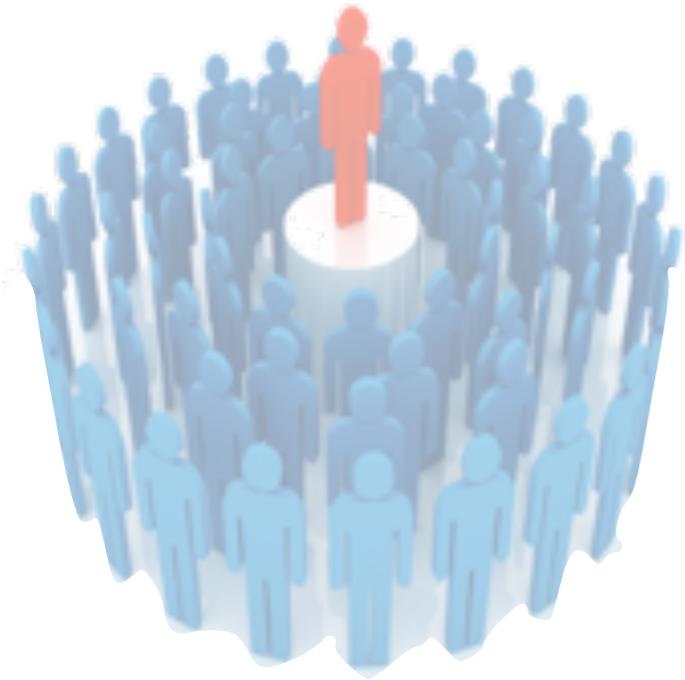
specific subsystem

## What kind of patch is accepted faster?



# Experience, Reviewers and Patch Spread Matter for Reviewing Time





## How many patches are merged?

33% patch acceptance

integration becomes slower

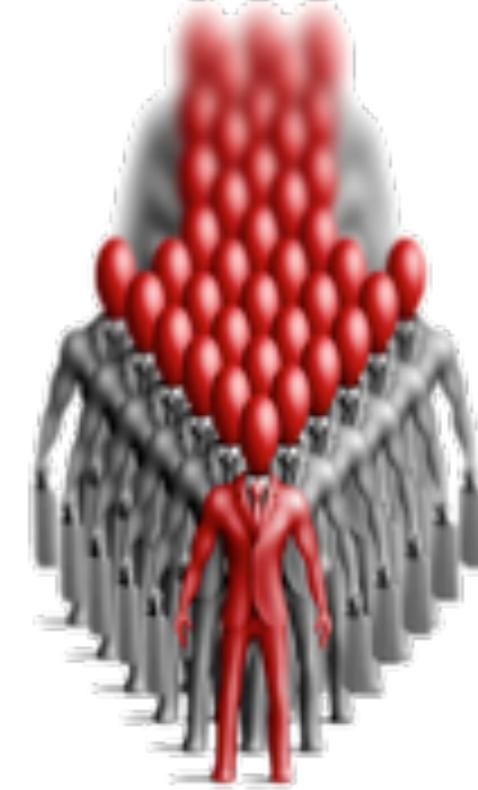


## What kind of patch is merged more likely?

by experienced developers

mature patch

specific subsystem



## What kind of patch is accepted faster?

less spread out

by more experienced developers

# Release Engineering

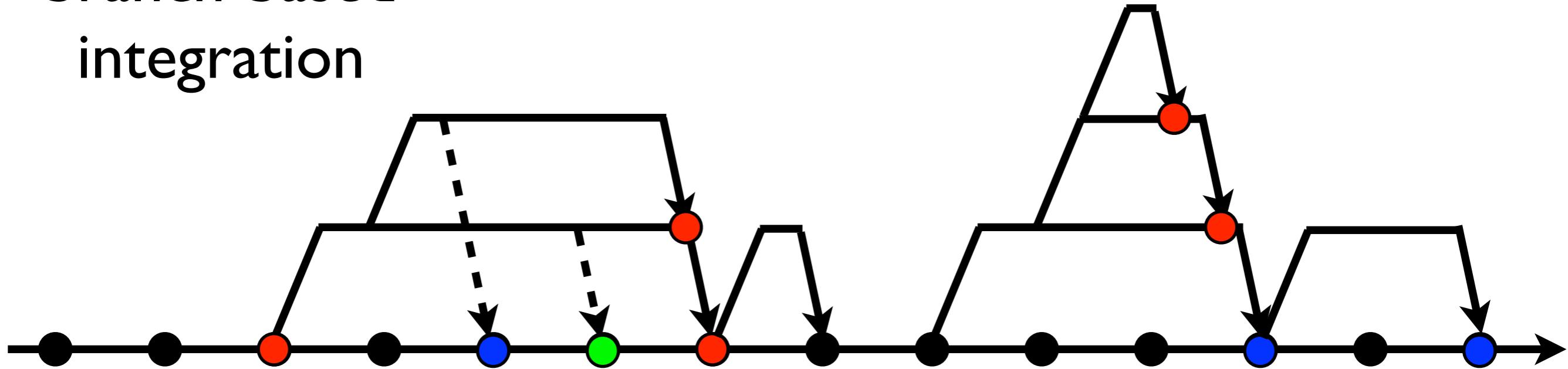
?

Software  
Quality

# Rapid Release vs. Quality?

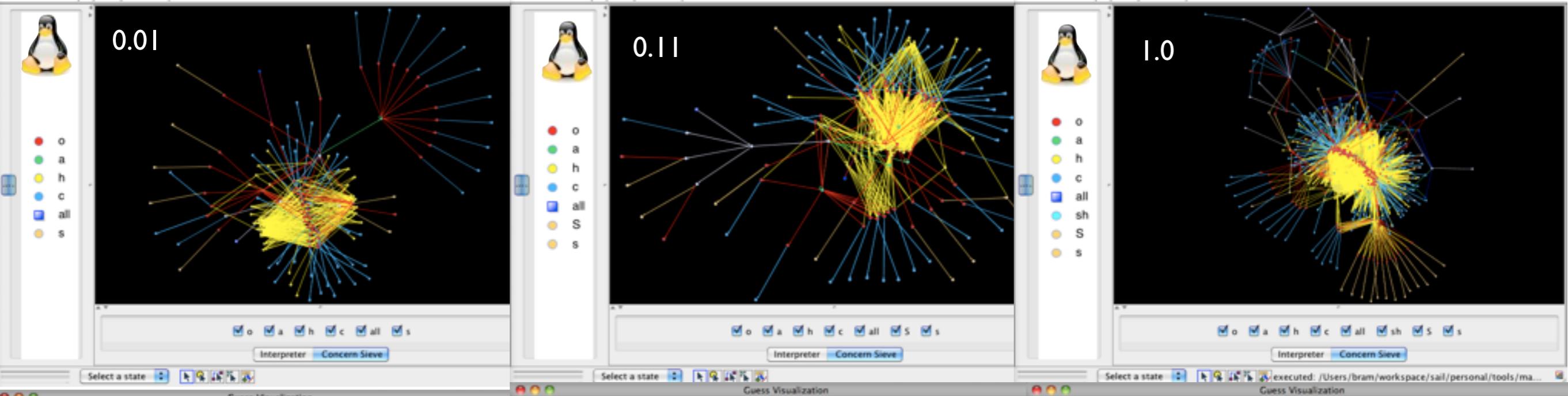


branch-based  
integration



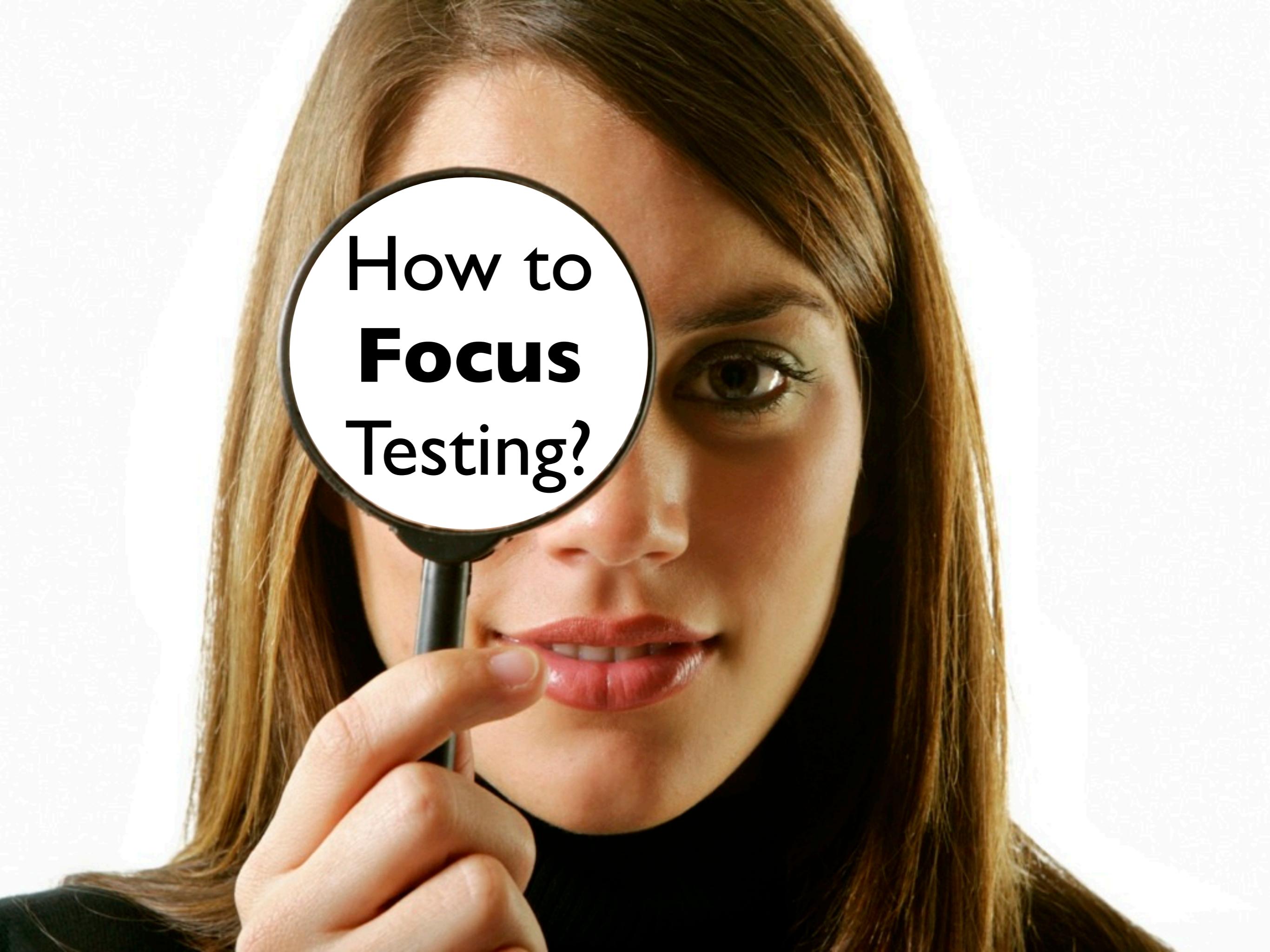
vs.

trunk-based  
integration



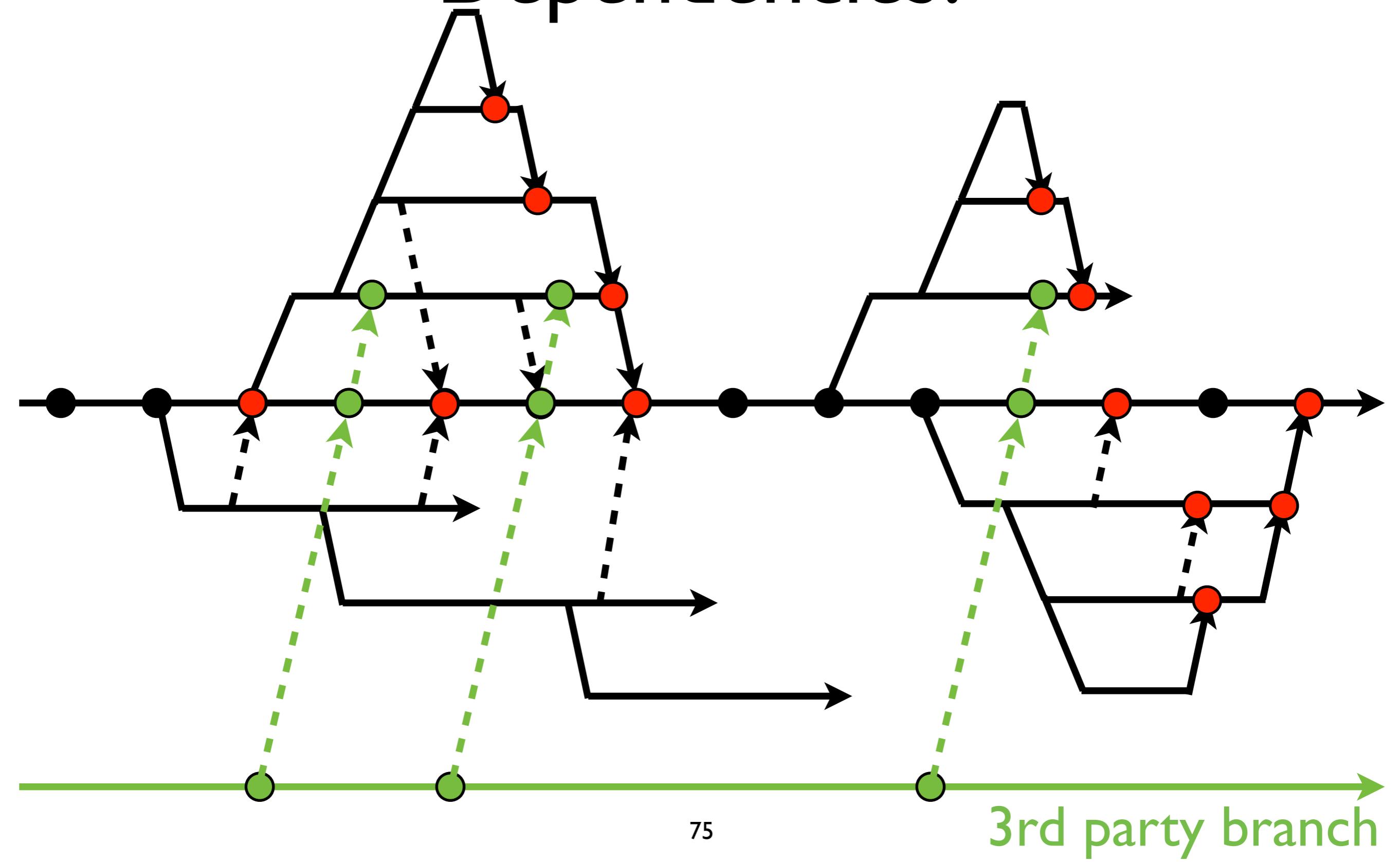
# How to Evolve Build Systems in a Healthy Way?



A close-up photograph of a woman's face. She has long, straight brown hair and is looking directly at the camera with a neutral expression. A magnifying glass is held up to her eye level, focusing on her right eye. Inside the circular frame of the magnifying glass, the text "How to Focus Testing?" is written in a bold, black, sans-serif font.

**How to  
Focus  
Testing?**

# How to Keep up with 3rd Party Dependencies?



# Where are the Release Engineering Tools?

focus your  
testing efforts

test your  
build!

what did the  
other team break  
now ;-)

keep poking those  
upstream guys until they  
give in :-)





# How to Align Release Engineers with the Rest of the Organization?



Express yourself in the world's largest 3D Chat and Dress-Up community!

[Member Login](#)

On average we  
deploy new code **fifty** times  
a day.

in 00 different countries!

Sign in with: [f](#) [g](#) [Y!](#)

Choose Your FREE Avatar >

Over 2 Million people  
like IMVU on Facebook!

FREE  
M  
C  
I  
S

same  
#bugs

... but faster  
crashes

bugs are  
fixed faster

... but less  
bugs are  
being fixed



How many patches are merged?

33% patch acceptance

integration becomes slower

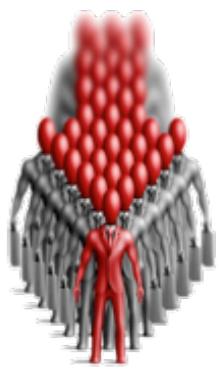


What kind of patch is merged more likely?

by experienced developers

mature patch

specific subsystem



What kind of patch is accepted faster?

less spread out

by more experienced developers

# 1st International Workshop on Release Engineering

RELENG 2013

6 practitioner  
talks

84  
participants

Home  
Call for Papers  
Organizers  
PC Members  
**Program**  
Registration  
RELENG Links  
Important dates:  
Submission: February

... and 10  
academic talks

Keynote: **Release Engineering as a "force-multiplier"**

by **John O'Duinn**

(Director of Release Engineering at Mozilla Corporation)

2 keynotes



Keynote: **Against All Odds – Completely Overhauling  
LinkedIn's Release Process**

**Roman Scheiter**

(Director of Engineering Services at LinkedIn)



**LOGICBLOX®**

**Google™**

**NETFLIX**

**IBM®**

**puppet  
labs**



Express yourself in the world's largest 3D Chat and Dress-Up community!

[Member Login](#)

On average we  
deploy new code **fifty** times  
a day.

in 60 different countries!

Sign in with:

**Choose Your FREE Avatar** >

Over 2 Million people  
like IMVU on Facebook!

**FREE**

M  
C  
I  
S

same  
#bugs

... but faster  
crashes

bugs are  
fixed faster

... but less  
bugs are  
being fixed

Testing has to  
become  
more  
**Focused**



How many  
patches are  
merged?

33% patch  
acceptance

integration  
becomes slower

What kind of  
patch is merged  
more likely?

by experienced  
developers  
mature patch  
specific subsystem

What kind of  
patch is accepted  
faster?

less spread out  
by more experienced  
developers