

# Lessons Learned from the International Workshop on Release Engineering



**Bram Adams, MCIS  
Stephany Bellomo, SEI  
Foutse Khomh, SWAT**

**<http://releng.polymtl.ca>**

# PART I: RELENG?!

**Back in 2009 ...**



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:

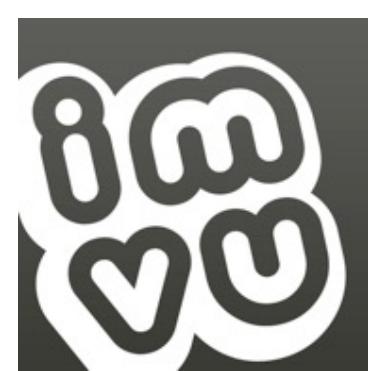


Choose Your FREE Avatar ➤

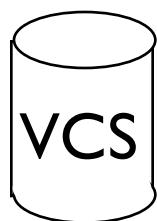
Over 2 Million people  
like IMVU on Facebook!

FREE

M  
C  
I  
S



# Continuous Delivery: the Early Days



15k tests

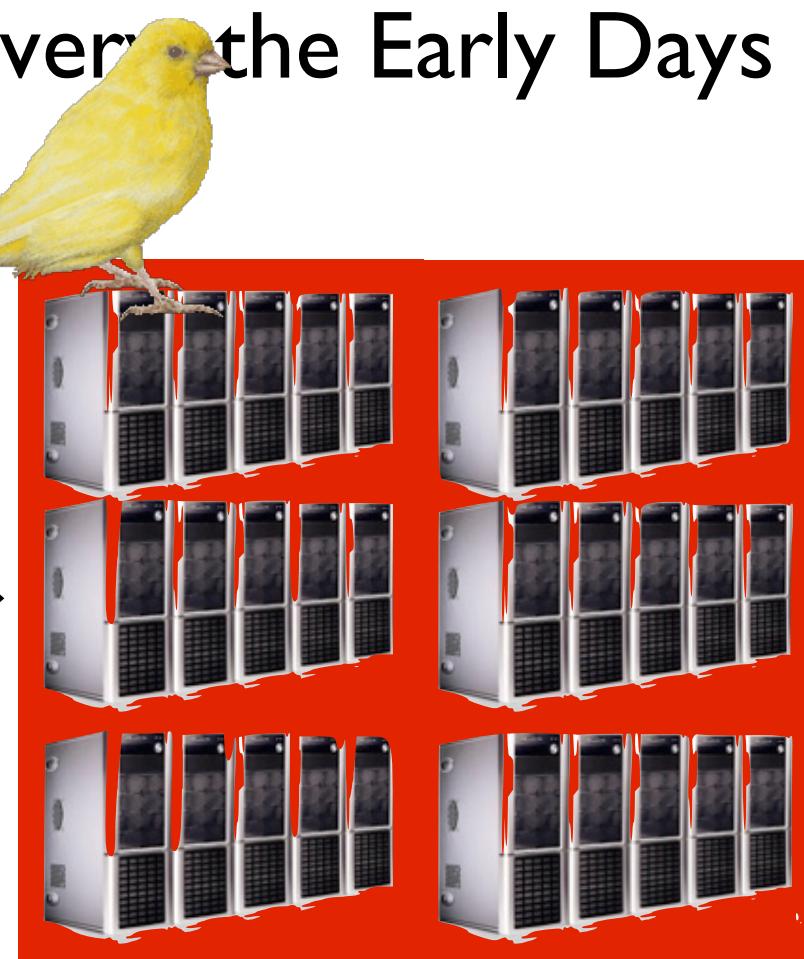


test

5

continuous  
integration

9 min.



6 min.

<http://goo.gl/qPT6>

Nowadays ...

# feature-based releases (i.e., when it's done)



vs.

# regular, time-based releases



openstack™

6 months



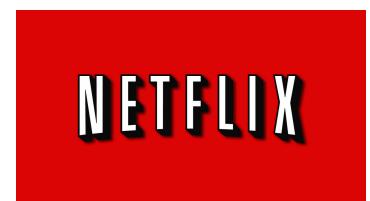
6 weeks



2 weeks (mobile)



twice/day  
(web)



dozens of  
times/day

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

Cycle Time !=  
Development Time

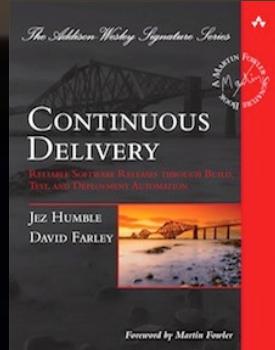
A man with dark hair and large black-rimmed glasses is shown from the chest up. He is wearing a dark suit jacket over a white shirt. His right hand is resting against his chin, with his fingers partially hidden in his pocket, suggesting a thoughtful or confused state. A large, hollow speech bubble originates from his mouth and extends to the right. Inside the bubble, the text "But why all this trouble?" is written in a bold, black, sans-serif font.

**But why all  
this trouble?**

key goal of continuous deployment is to  
**reduce the risk**  
of releasing software

if it hurts, do it more frequently,  
and bring the pain forward

Jez Humble





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

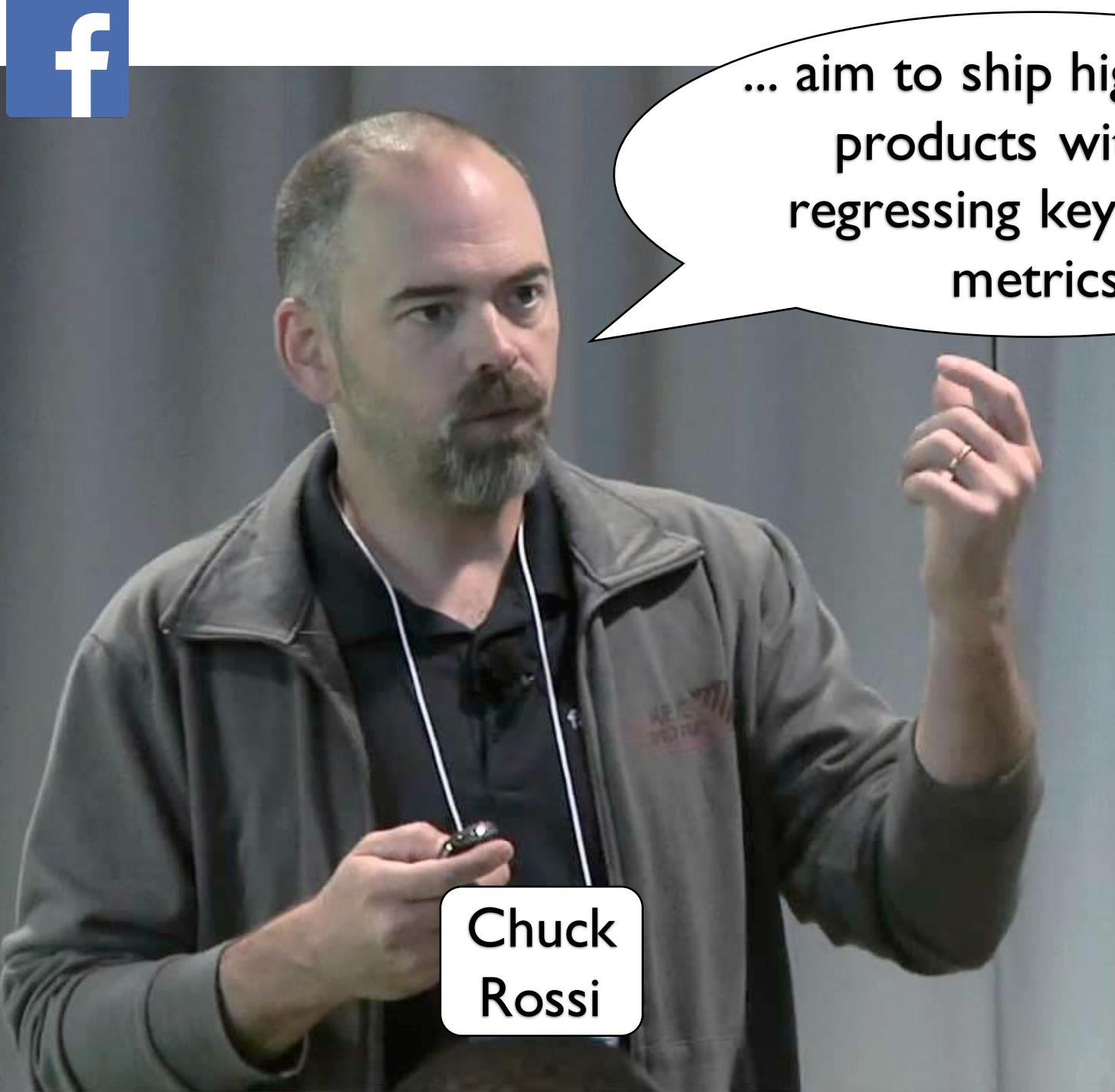
James Whittaker



# Before & After

- How quickly can we ship a chemspill release?
  - ~~4-6 weeks~~ 11 hours
- How long to ship a “new feature” release?
  - ~~12-18 months~~ 6 weeks
- How many active code lines?
  - ~~1 1/2~~ 42
- How many checkins per day?
  - ~~~15 per day~~ 325 per day





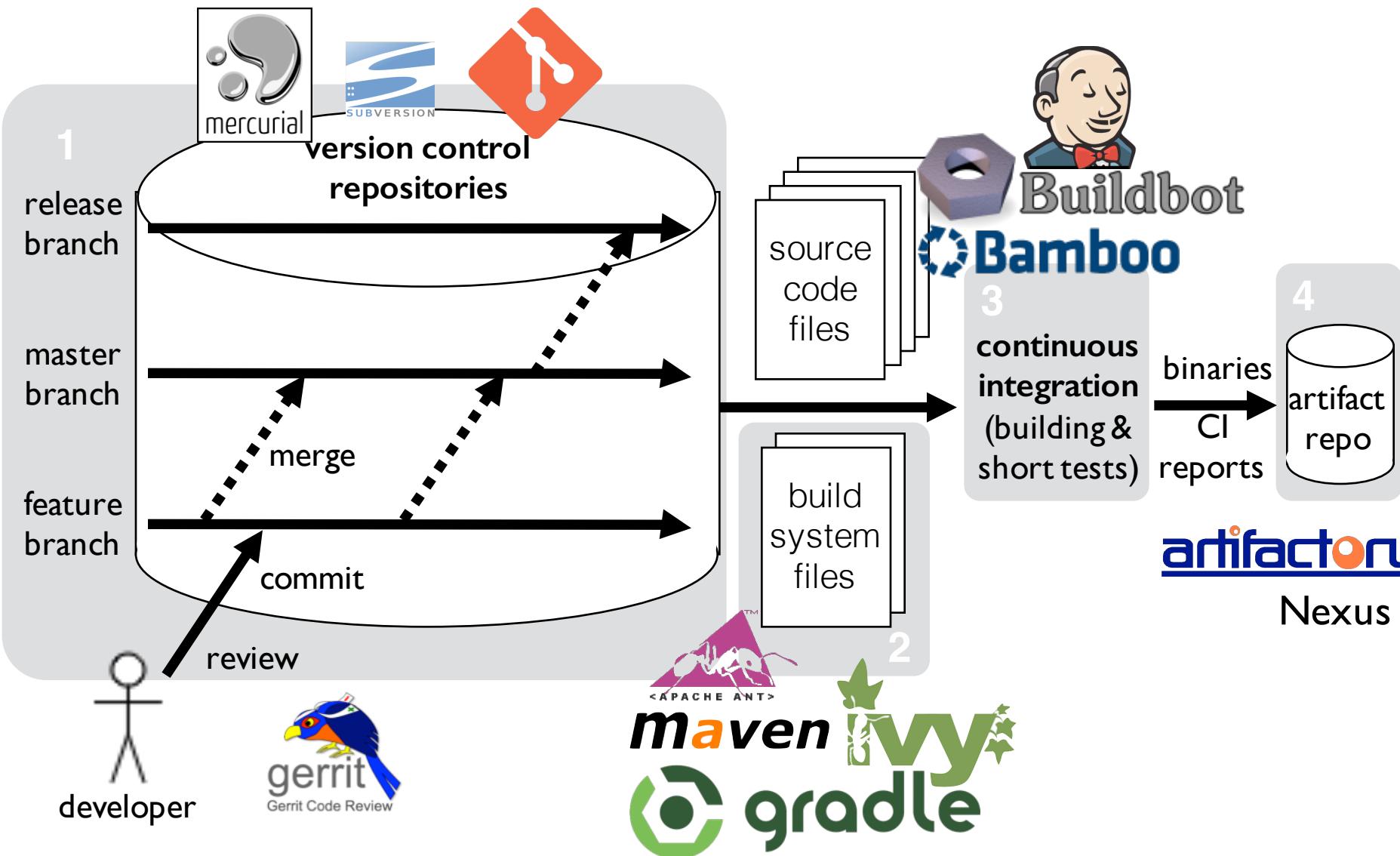
... aim to ship high quality  
products without  
regressing key quality  
metrics

Chuck  
Rossi

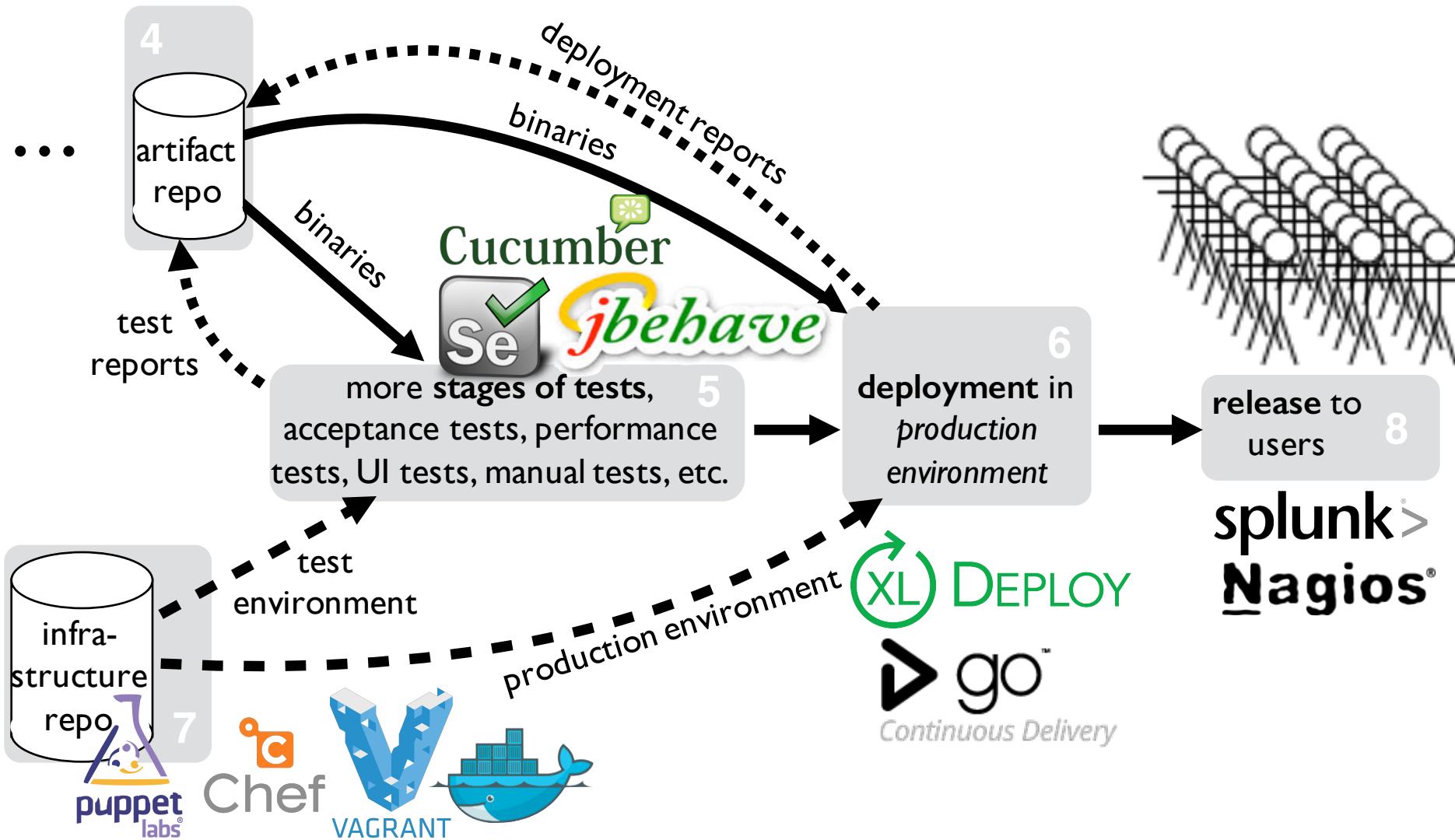
A man with dark hair and large black-rimmed glasses is shown from the chest up. He is wearing a dark suit jacket over a white shirt. His right hand is resting against his chin, with his fingers partially hidden in his pocket, suggesting a thoughtful or confused state. A large, hollow speech bubble originates from his head and extends to the right, containing the text.

OK, what do I  
need for this?

# Release Engineering Pipeline (I)



# Release Engineering Pipeline (2)





# Infrastructure-as-Code

```
# Install PostgreSQL server and client
include_recipe "postgresql::server"
include_recipe "postgresql::client"

# Make postgresql_database resource available
include_recipe "database::postgresql"

# Create database for Rails app
db = node["practicingruby"]["database"]
postgresql_database db["name"] do
  connection(
    :host      => db["host"],
    :port      => node["postgresql"]["config"]["port"],
    :username  => db["username"],
    :password  => db["password"],
  )
end
```

# PERIODIC TABLE OF DEVOPS TOOLS (V1)

XebiaLabs  
Deliver Faster

1	En	PERIODIC TABLE OF DEVOPS TOOLS (V1)												2	Frm																						
O		XebiaLabs Deliver Faster												Aws																							
12c		Open Source			Database			SCM			Build																										
3	Os	4	Os	Fr Free			CI			Repo Mgmt			Testing			Amazon Web																					
My		Gt		Fm Freemium			Deployment			Config / Provisioning			Containerization			Amazon Web																					
MySQL		Git		Pd Paid			Release Mgrnt			Cloud / IaaS / Paas						Amazon Web																					
11	En	12	Os	En Enterprise			Collaboration			Bl / Monitoring						Amazon Web																					
Mq		Sv		Logging			Security									Amazon Web																					
MSSQL		Subversion														Amazon Web																					
19	Os	20	Fm	21	Os	22	Os	23	En	24	Os	25	Pd	26	Os	27	Fr	28	Os	29	Fr	30	Os	31	Pd	32	Os	33	Fr	34	Os	35	Os	36	Er		
Pq		Gh		Mv		Gr		Mr		Jn		Ba		Tr		Ar		Fn		Se		Gn			Gd		Sf		Cb		Bc		Kb		Rs		Rackspace
PostgreSQL		Github		Maven		Gradle		Meister		Jenkins		Bamboo		Travis CI		Archiva		FitNesse		Selenium		Gatling			Deployer Manager		SmartFrog		Cobbler		Bcfg2		Kubernetes		Rackspace		
Mg		Bb		Br		At		Bm		Cs		Sn		Cr		Nx		Cu		Cj		Qu			Cp		Ju		Rd		Cf		Pk		Bx		
MongoDB		Bitbucket		Buildr		ANT		BuildMaster		Codeship		Snap CI		CircleCI		Nexus		Cucumber		Cucumber		Qunit			Capistrano		JuJu		Rundeck		CFEngine		Packer		Bluemix		
Db		Hg		Qb		Ub		Ta		Tc		Sh		Cc		Ay		Jt		Jm		Tn			Ry		Cy		Oc		No		Eb		Ad		
DB2		Mercurial		QuickBuild		UrbanCode Build		Visual		TeamCity		Shipable		CruiseCont		Artifactory		JUnit		JMeter		TestNG			RapidDeploy		CodeDeploy		Octopus Deploy		CA Nolio		ElasticBox		Apprenda		
73	Fr	74	En	75	Os	76	Os	77	Os	78	Os	79	Fm	80	Os	81	Os	82	Os	83	En	84	En	85	Os	86	En	87	En	88	En	89	Os	90	Os	91	
Cs		Hx		Msb		Rk		Lb		Co		Ca		Gu		Ng		Ap		Xltv		Tc			Go		Ef		Xld		Ud		Mo		Cf		Cloud
Cassandra		Helix		MSBuild		Rake		LuntBuild		Continuum		Continua		Gump		NuGet		Appium		XL		TestComple			ElectricFlow		XL Deploy		UrbanCode Deploy		Mesos		Cloud				

Share



## Embed



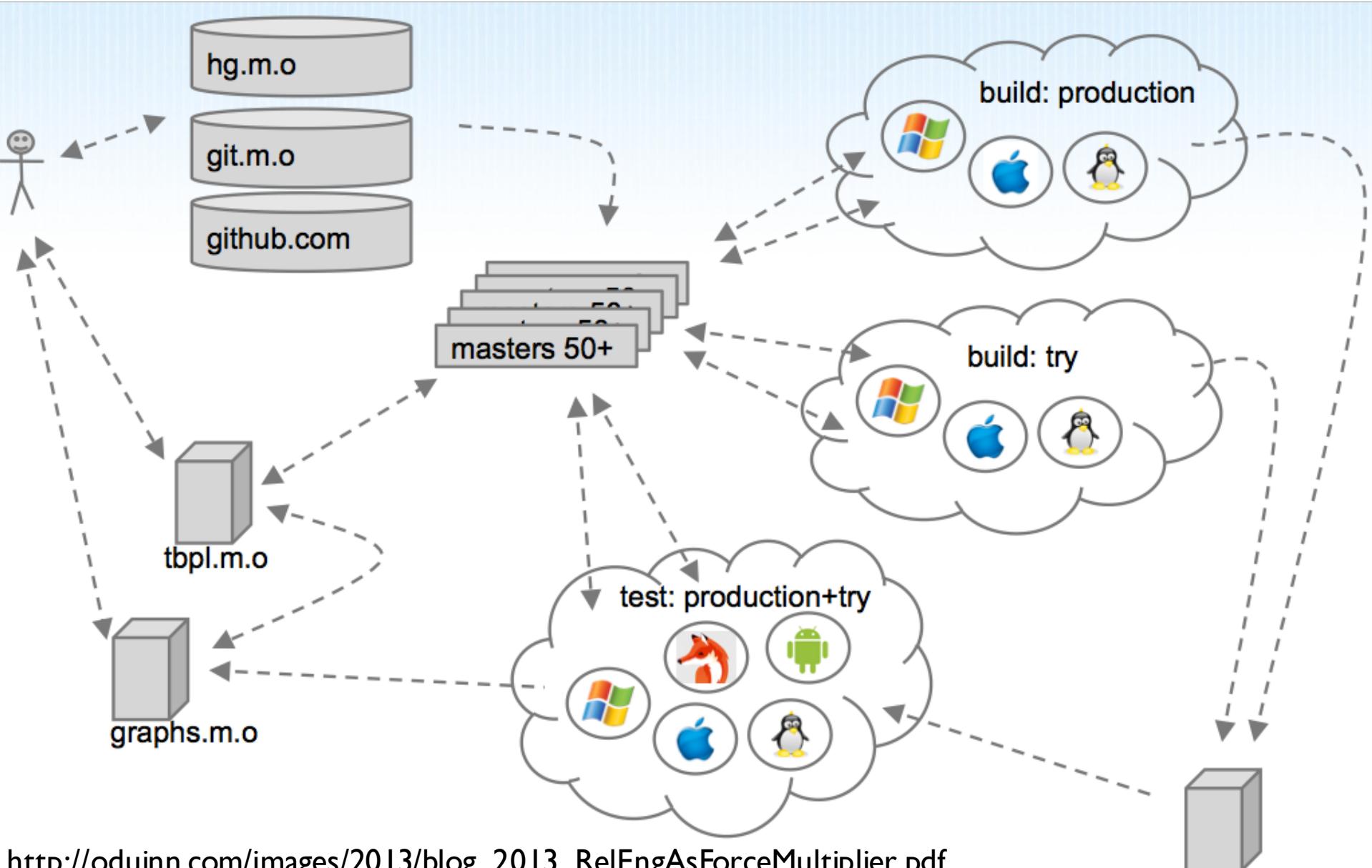
## Become Excellent!

 [Subscribe here!](#)

91	En	92	Ur	En	93	Ls	En	94	Bm	En	95	Hp	En	96	Pd	97	Pl	En	98	Sr	En	99	Fm	100	Pd	101	Fm	102	Fm	103	Fm	104	Pv	Pd	105	Er
Xlr		UrbanCode		CA		Service		BMC		Release		HP	Cedar		Excel		Plutora		Serena		Trello		Jira		Rf		Sl		Fd		Pivotal		Sn		ServiceNa	
XL		Release		Service				Release								Release		Release																		
106	Os	107	Fm	108	Os	109	Os	110	Os	111	Os	112	Fm	113	En	114	Fm	115	Os	116	Fm	117	Os	118	Os	119	Os	120	Er							
Ki		Nr		Ni		Gg		Ct		Gr		Le		Sp		Si		Ls		Lg		Gr		Sn		Tr		Cy								
Kibana		New Relic		Nagios		Ganglia		Cacti		Graphite		Logentries		Splunk		Sumo		Logstash		Loggly		Graylog		Snort		Tripwire		CyberArk								

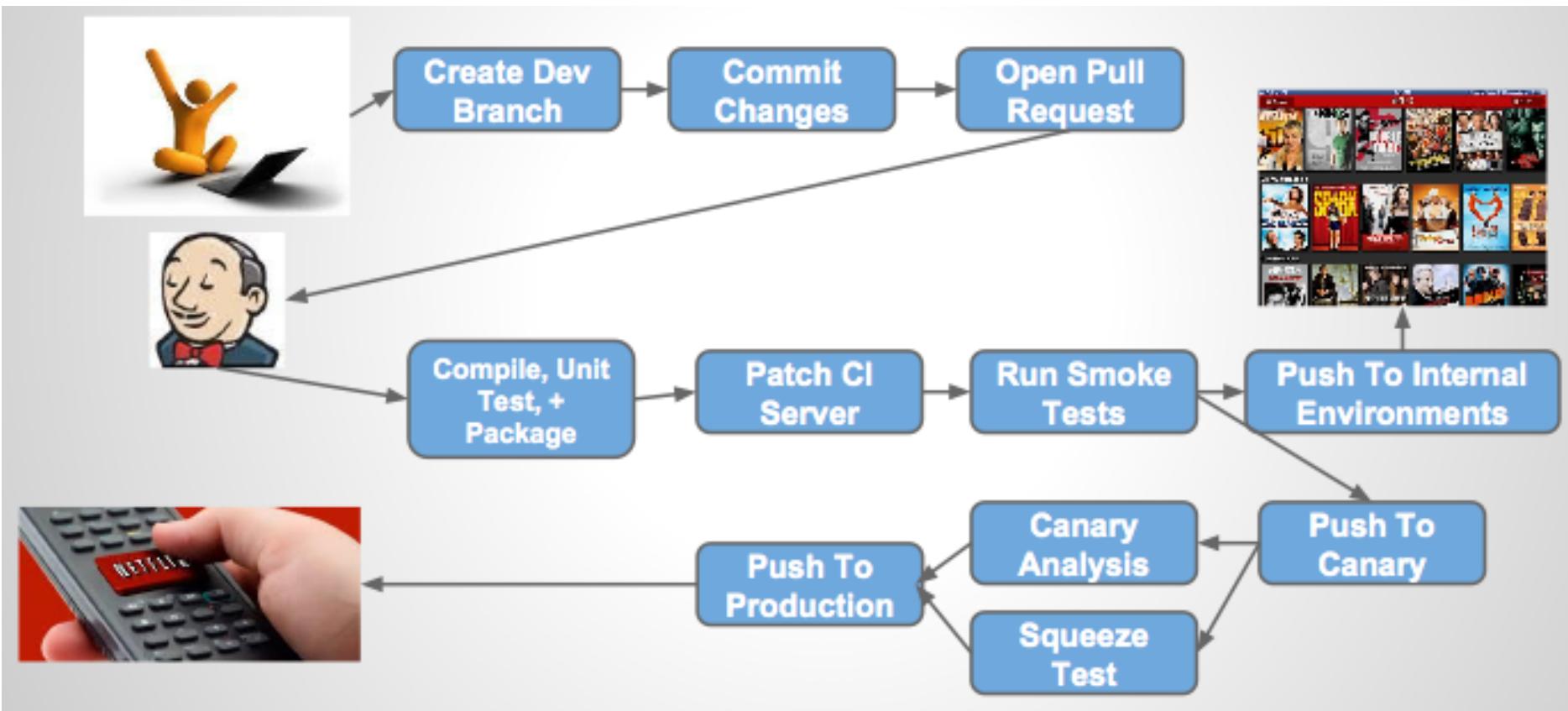


# Example Pipeline



# Another Example Pipeline

NETFLIX



A cartoon illustration of a man with dark hair and large black-rimmed glasses. He is wearing a dark suit jacket over a white shirt. His right hand is resting against his chin, with his fingers partially hidden in his pocket, suggesting a thoughtful or skeptical pose. A large, hollow speech bubble originates from his mouth and extends to the right. Inside the bubble, the words "Sounds easy, right?" are written in a bold, black, sans-serif font.

Sounds easy,  
right?

# Rapid-release Firefox meets corporate backlash

New Firefox versions now will emerge every six weeks as Mozilla tries to be more competitive. That's too rapid for some, but Mozilla is willing to leave them behind.

by [Stephen Shankland](#)  @stshank / June 27, 2011 3:33 AM PDT



Last week, Mozilla programmers and executives were jubilant when the [release of Firefox 5](#) marked the successful transition to a more competitive rapid-release development cycle.

Now, with a backlash from corporations and others who aren't equipped to handle that pace of change, things aren't quite so sunny. The organization and its community of supporters have begun some soul-searching about how to reconcile the conflicting priorities--developing software quickly but not leaving users behind.

Mozilla has concluded that [Firefox](#) isn't for corporations whose Web use doesn't move at the speed of today's Web, though. That decision frees



# TECHNOLOGY LAB / INFORMATION TECHNOLOGY

## Latest Windows 10 update shows how rapid releases work in practice

Problems revealed by the fast track were fixed before the rollout to the slow track.

by Peter Bright - Nov 26, 2014 2:05am CET



Share



Tweet



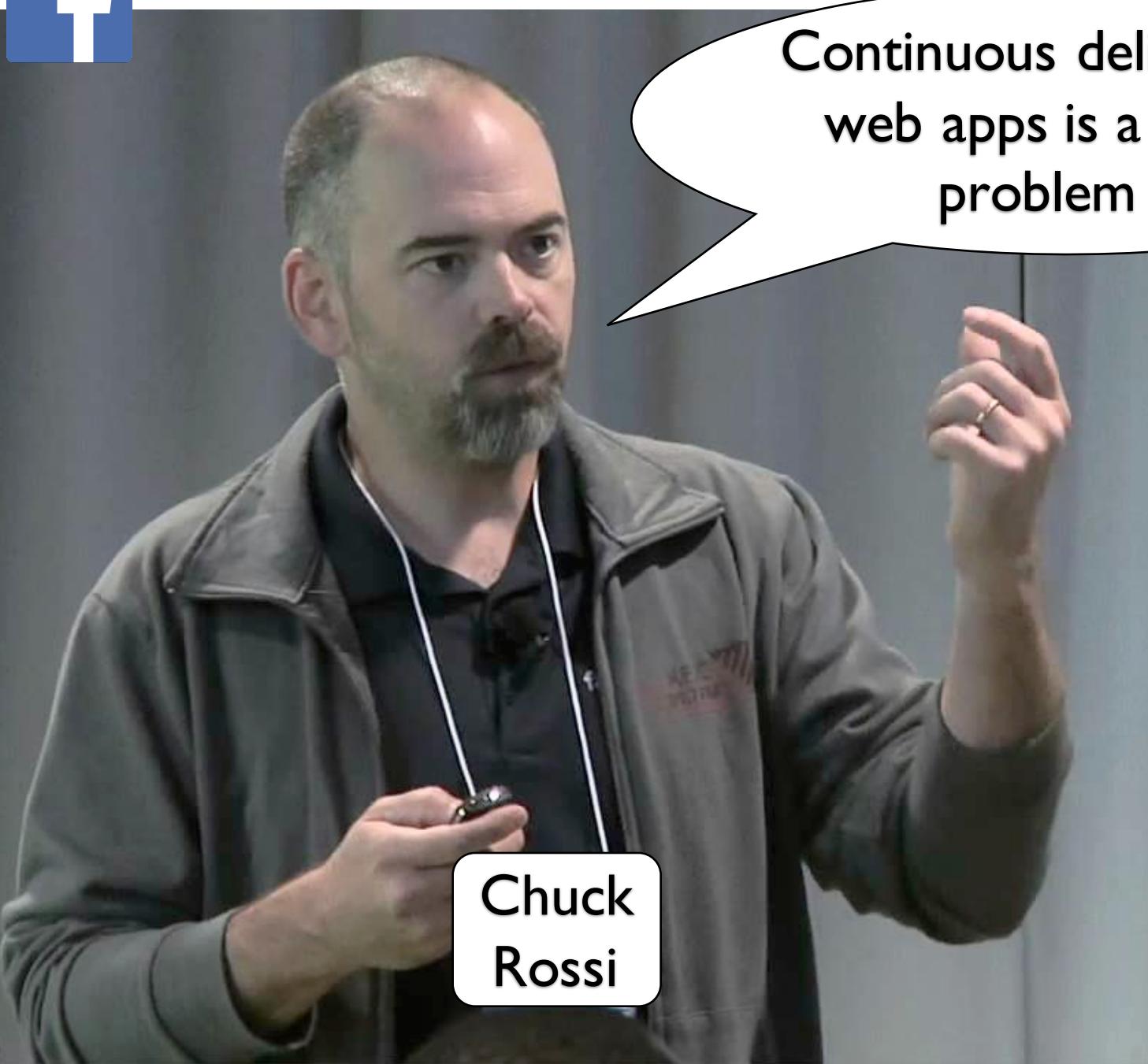
Windows 10's [updates and maintenance](#) are following a different, better path to all prior Windows releases: one with more regular updates and quicker access to new features for those who want it, while still offering enterprises a slower pace of delivery. With the [first update to the Windows 10 Technical Preview](#) a month ago, Microsoft also enabled a slow update track for the million or so

With the **three announced stable release cadences** (a fast consumer-oriented one, a medium business-oriented one, and a slow critical system-oriented one), and **two-speed insider program**, this will leave Windows users on up to five different versions of Windows 10.

ers have put  
st two weeks

were the  
en of death. A  
cond coming

ed out to slow



Chuck  
Rossi

SUN

DEC  
29TH

# Continuous Delivery is Mainstream



Noah  
Sussman

**Google, Amazon and Facebook all are using very aggressive Continuous Delivery workflows and have been doing so for years.**

**UPDATED January 21, 2014**

I meant the adjective “mainstream” in the sense of “not dangerous.” For example: “The Ramones are so mainstream, I only listen to Norwegian Death Metal.” Based on the large amount of feedback I have received, this is not everyone’s default definition of “mainstream.” Hopefully adding this paragraph to the post will clear up such ambiguity for future readers.

Once again, I did not mean to imply that everyone is doing CD. Everyone is not doing CD! But, **CD is no longer the risky experiment it was in 2010** when Chad Dickerson hired me to help [scale the CI system at Etsy](#) (which is how I got involved in this whole discussion in the first place). Today CD is a mature option and I think it is the best option available. But there are certainly other ways to build software and lots of people use

▼ DevOps Community



## Continuous Delivery is Mainstream—So What's the Problem?

For every successful organization engaging in continuous delivery, there are many, many others that are struggling. **Difficulties include buy-in from management or other teams who do not understand the value, inability to work across silos and lack of tooling (lack of support).** These are standard problems in the area of leading change.

Status updates

is continuous delivery possible, but surveys are demonstrating that there is significant value in embracing continuous delivery. The [2013 DevOps Survey](#) showed that high-performing organizations are delivering 30x more frequent releases than their peers and with much greater reliability—12x faster Mean Time to Recover (MTTR).

Members

Forums

▪ **Blog**

Bookmarks

Events

Wiki

Related Communities

Subcommunities

DevOps Community

Leadership

### So What's the Problem?

For every successful organization engaging in continuous delivery, there are many, many others that are struggling. Difficulties include buy-in from management or other teams who do not understand the value, inability to work across silos and lack of tooling (lack of support). These are standard problems in the area of [leading change](#).

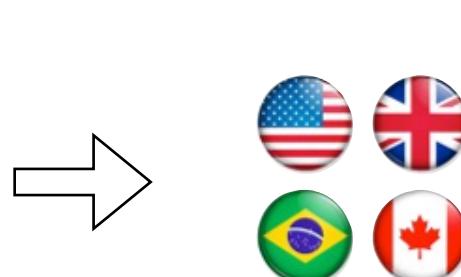
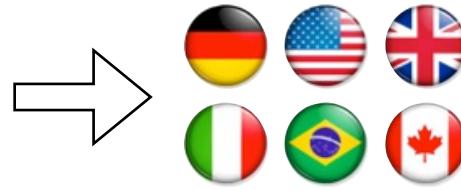
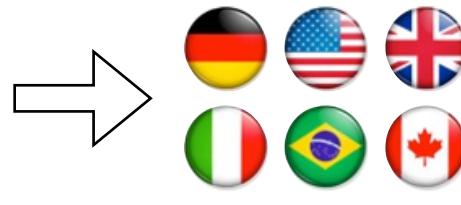
### One Size Does not fit all

A photograph of Chuck Rossi, a man with a beard and mustache, wearing a grey jacket over a black shirt. He is gesturing with his right hand while holding a small device in his left hand. A speech bubble originates from his mouth, containing the text below.

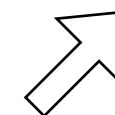
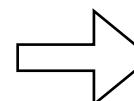
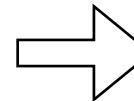
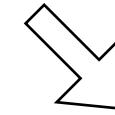
..., yet continuous  
delivery for mobile apps  
is a serious challenge

Chuck  
Rossi

# Heterogeneous Hardware, Oses, ...



i18n



**generate  
incremental  
updates for each  
supported old  
version**



# How Poor Software Release Management Creates Security Nightmares

Posted by [Alan Pearson](#) on Mar 17, 2015

With aggressive software release schedules, developers may find themselves **pressured into taking development shortcuts**, simply to meet a looming deadline.

response to software-hungry consumers, competitor strategies, or the increasing pace of hardware development. Whilst some of these



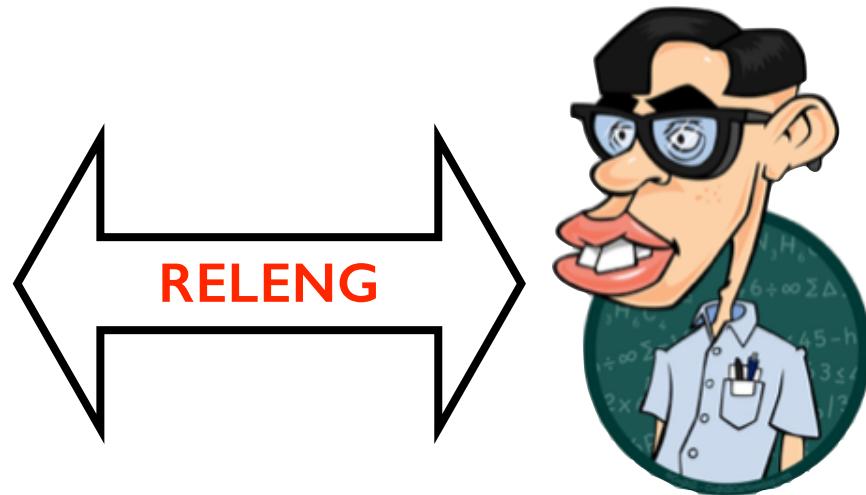
As a result, **aggressive software releases can have a serious and unintended side-effect: a build-up of technical debt**. With every software and update release, there's the potential for new bugs and technical vulnerabilities to be uncovered.

tool, it can also pose some real security issues.

# Are These Concerns Justified?



practitioners



researchers

© <http://www.flickr.com/photos/fernando/35759033/>

# 1st International Workshop on Release Engineering



6 practitioner  
talks

- [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**  
by [Roman Scheiter](#)  
(Director of Engineering Services at LinkedIn)



# 2nd International Workshop on Release Engineering

April 11, 2014, Mountain View,  
USA



7 practitioner  
talks



2 keynotes



Home/CFP

Venue

Program

IEEE Special Issue

PC Members

Organizers

Registration

History of RELENG

## 2nd International Workshop on Release Engineering 2014

**Release engineering** deals with all activities in between regular development and delivery of a software product to the end user, i.e., integration, build, test execution, packaging and delivery of software. Although research on this topic goes back for decades, the increasing heterogeneity and variability of software products along with the recent trend to reduce the release cycle to days or even hours starts to question some of the common beliefs and practices of the field.

The RELENG workshop series aims to provide a highly interactive forum for researchers and practitioners in release engineering to: (1)

News **NEW!**

July 10: **21 days left to submit a paper to the IEEE Software Special Issue on Release Engineering, getting closer!**

April 04: Detailed information about **RELENG 2014** venue is available

... and 6 academic talks

<http://releng.polymtl.ca/RELENG2014/html/index.html>

# 3rd International Workshop on Release Engineering

6 practitioner  
talks

May 19, 2015, Florence, Italy



1 keynote

## 3rd International Workshop on Release Engineering 2015

Home/CFP

Venue

Program

IEEE Special  
Issue

PC Members

Organizers

Registration

History of RELENG

**Release engineering** deals with all activities in between regular development and delivery of a software product to the end user, i.e., integration, build, test execution, packaging and delivery of software. Although research on this topic goes back for decades, the increasing heterogeneity and variability of software products along with the recent trend to reduce the release cycle to days or even hours starts to question some of the common beliefs and practices of the field.

The RELENG workshop series aims to provide a highly interactive forum for researchers and practitioners in release engineering to:

<http://releng.polymtl.ca/RELENG2015/html/index.html>

News NEW!

February 24: the first ever  
**IEEE Software Special  
Issue on Release  
Engineering is now  
available online!**

February 24: Davide Falessi  
speaks at the  
Guest Lecture Series

... and 6 academic  
talks

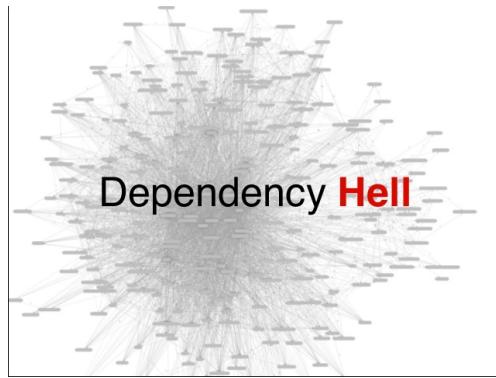
February 19: 6 talks, 6 papers

7 papers



... and 1  
roundtable  
article!

# PART 2: Insights from the Workshop



dependency hell



pipeline security



education



success & other stories



business value



# Dependency **Hell**

Causing unpredictable  
release schedules ...

a multi-branching development approach



trunk-only development

- code owners responsible for their own subsystem:
  - \* only backward-compatible changes are integrated
  - \* control features using feature flags
- no fixing of bugs after merging, instead roll back



LinkedIn

## To Branch or not to Branch

trunk-only development



a multi-branching development approach

- branches provide physical isolation
- different channels/streams of code changes and releases possible
- hard to roll-back, needs disciplined changes and merging



# Consolidating All Dependencies



- no shared libraries, everything rebuilt from code
- maintaining a full dependency graph of things to build
- metrics on every action for audit and analysis:
  - \* ratio passing tests
  - \* #changes per release
  - \* time between releases
  - \* #cherry-picks per release
  - \* time from commit to release
- features can be guarded by feature flags

# Roll Your Own Release (self-service!)

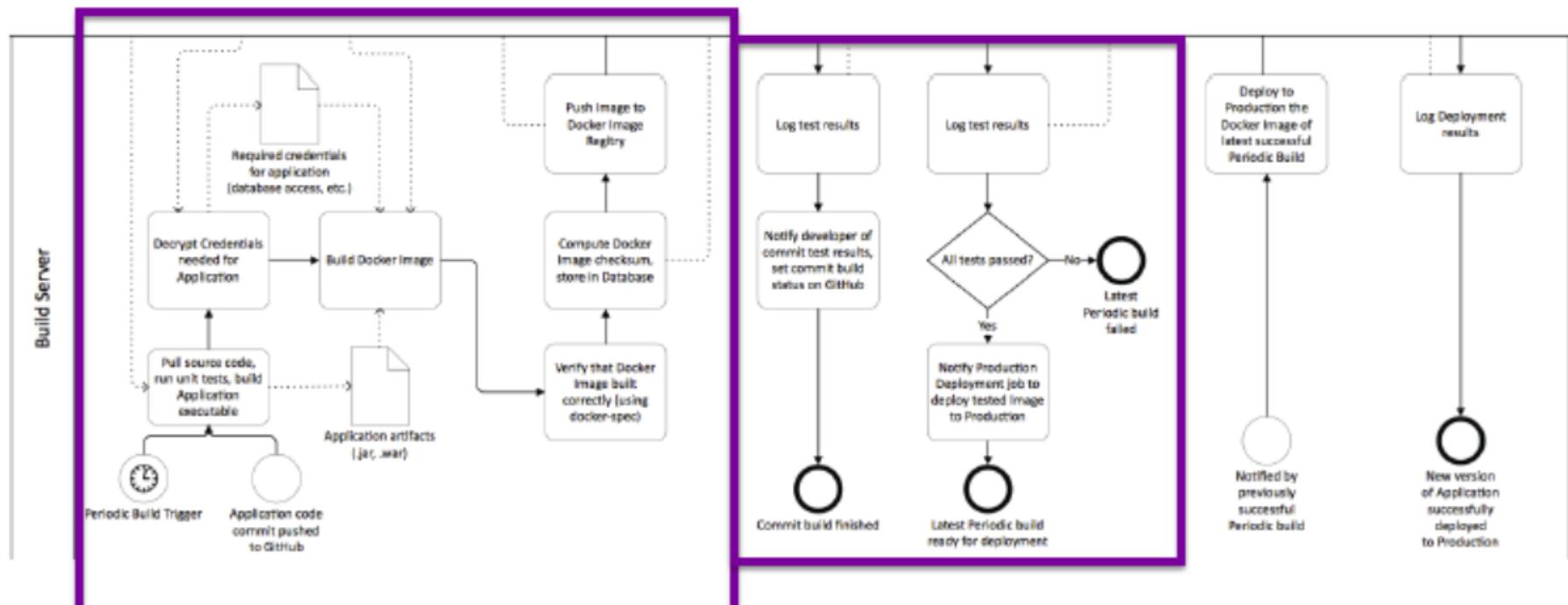


- releng team develops tools to automate the release process ...
- ... then developers use them to roll their own releases!
- the tools are also used to channel/promote best practices

# Is our Pipeline Secure?



# Security of a Release Engineering Pipeline



# Is the Image Deployed into Operation a Valid Image? How can we secure a release pipeline?

- Analyse a model of the pipeline to detect vulnerabilities (from design perspective)
- Restructure and remodel pipeline to remove vulnerabilities
- In practice, it's hard to remove all vulnerabilities



Can you Learn to be a Release  
Engineer? If, so How?

# The 10 Commandments of Release Engineering



Google

- 1- Thou shalt use a source code control system
- 2- Thou shalt use the right tool(s) for the job
- 3- Thou shalt write portable and low maintenance build files
- ...
- 10- Thou shalt apply these commandments to thyself

Since 2013, Courses on Release  
Engineering are Being Given at:

Polytechnique Montreal  
Carnegie Mellon University  
NC State University  
TU Delft  
RWTH Aachen University

...



Sharing **Successes and Failures** is Important!

# Dozens of Success Stories, Failures, Lessons Learned, ...

- Focus on solving company-threatening problems instead of on technology
- Talk with, and listen to, all your customers
- Show progress (measure what customers care about)
- Take baby steps, relentless baby steps  
(keep it simple)
- Make updates as easy as possible

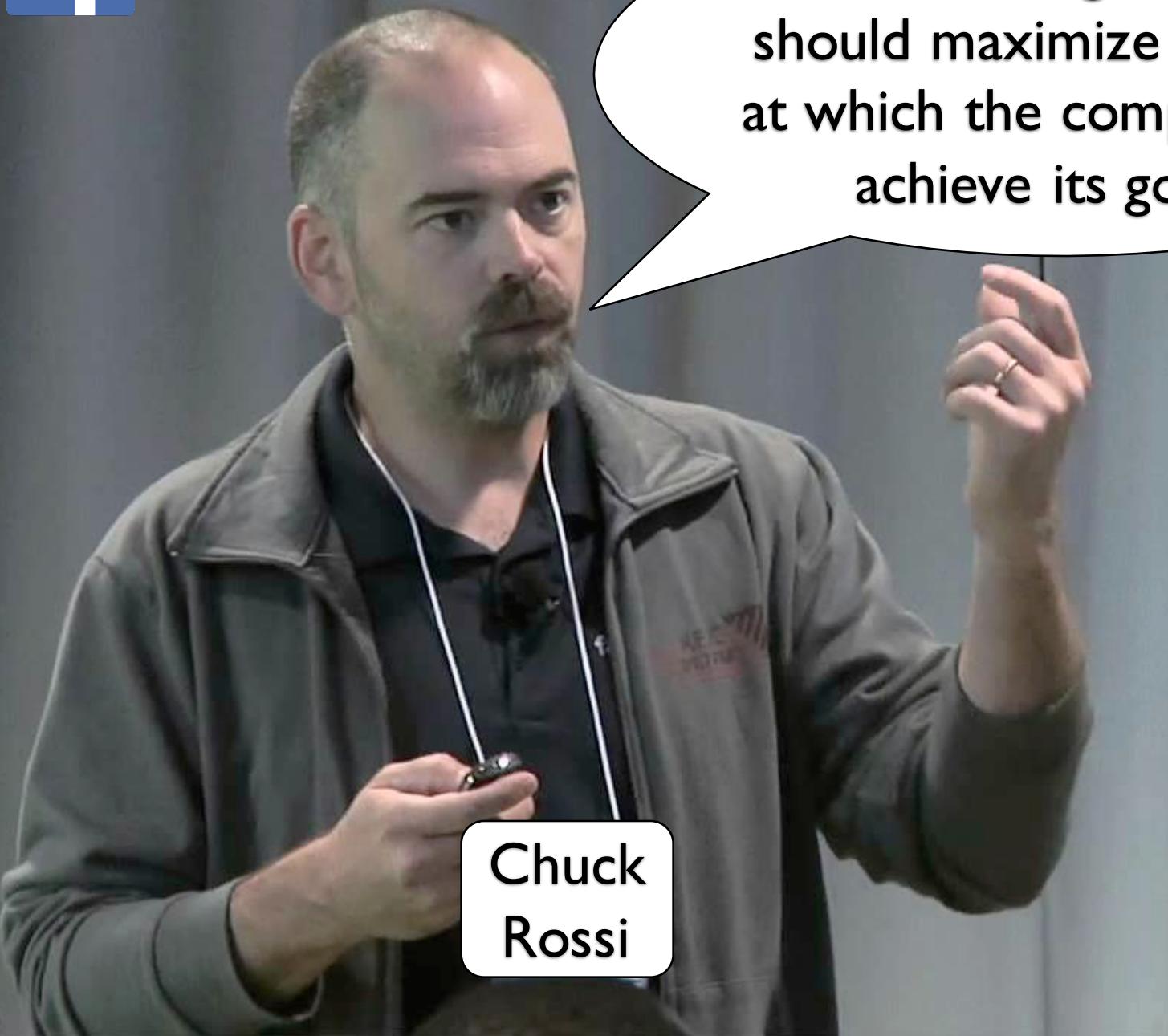




# Release Engineering **versus** and Business



Release engineering  
should maximize the rate  
at which the company can  
achieve its goals!



Chuck  
Rossi

# Costs/Benefits of Release Engineering ...

... can depend on:

- degree of automation
- build/test performance: \$/hour (cloud bill, bonuses, equity, salary, ...)
- context switches
- (in general) time spent by test, release, software, ... engineers waiting for build, test, certification, ... agility!
- ...

# PART 3: Insights from the Special Issue

# RELENG workshops maintain a community with lots of insights..., why not share it via an IEEE Special Issue on Release Engineering?

RELENG 2013

May 20, 2013, San Francisco, CA, USA

In conjunction with ICSE 2013



RELENG 2014

April 11, 2014, Mountain View, CA, USA

Google



RELENG 2015

May 19, 2015, Florence, Italy

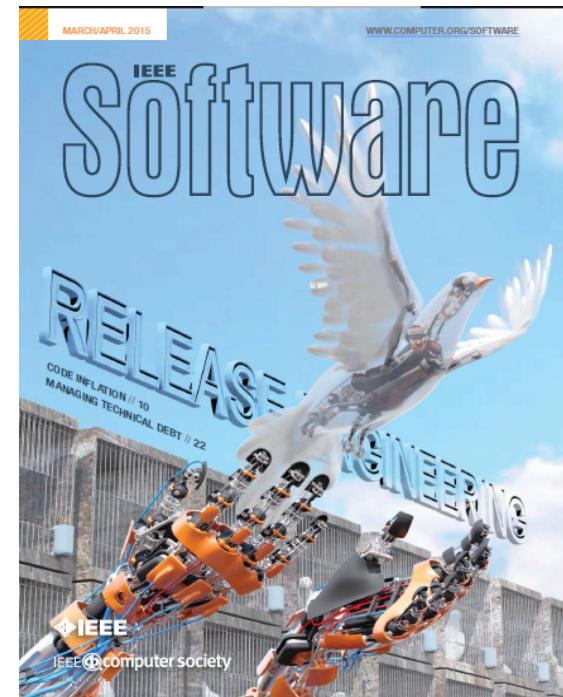


Mar/Apr 2015

# IEEE Special Issue on Release Engineering- I

## Table of Contents

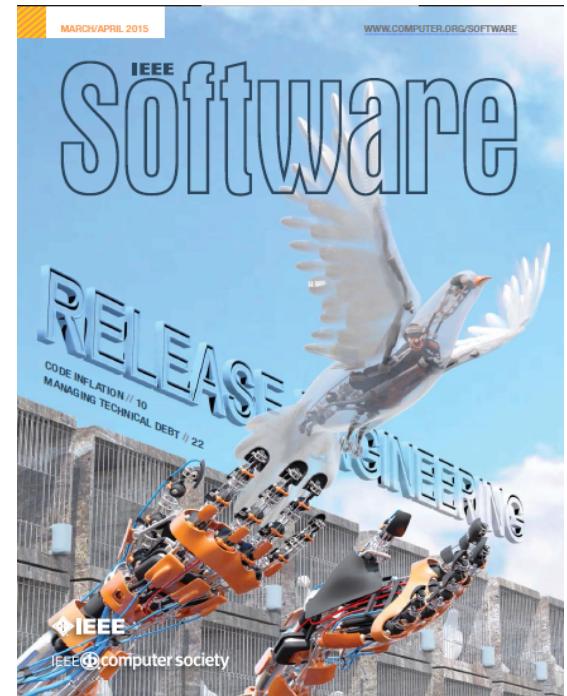
- Guest Editors Introduction Roundtable
- Continuous Delivery: Huge Benefits, but Challenges Too
- Why and How Should Open Source Projects Adopt Time-Based Releases?
- The Highways and Country Roads to Continuous Deployment



# IEEE Special Issue on Release Engineering-2

## Table of Contents Cont.

- Achieving Reliable High-Frequency Releases in Cloud Environments
- Release Stabilization on Linux and Chrome
- Rapid Releases and Patch Backouts: A Software Analytics Approach
- Vroom: Faster Build Processes for Java



# A Roundtable with Three Release Engineers

## Participants:

- Mozilla (Kim Moir)
- Google (Boris Debic)
- Facebook (Chuck Rossi)



## Question topics:

- What metrics are most valuable to release engineers?
- How do you maintain quality and stability?
- Are there limitations for continuous delivery?
- How can we educate others about the value of release engineering?

# My Favorite Roundtable Quotes

“On a given day we might let 5% of the browser population get a new release. We have automatic crash reporting in the browser.”

Moir, Mozilla

“Mobile deployments are more challenging than Web deployments because we don’t own the ecosystem.”

Rossi, Facebook

“I tell people that release engineering is the difference between manufacturing software in a startup compared to a repeatable and predictable approach.” Debic, Google



# The 38<sup>th</sup> International Conference on Software Engineering, Austin, TX, May 14-22, 2016

## International Workshop on Continuous Software Evolution and Delivery (CSED)

### Important Dates

- Abstract: January 15, 2016 AoE (mandatory)
- Submission: January 22, 2016 AoE
- Notification: February 19, 2016
- Camera Ready: February 26, 2016
- Workshop: May 14-15, 2016

Stay tuned for  
RELENG 2016!



"Thank You"  
CALL AGAIN  
and bring  
a Friend!

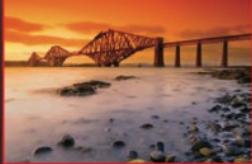
The Addison-Wesley Signature Series

A MARTIN FOWLER SIGNATURE Book  
Martin Fowler

# CONTINUOUS DELIVERY

RELIABLE SOFTWARE RELEASES THROUGH BUILD, TEST, AND DEPLOYMENT AUTOMATION

JEZ HUMBLE,  
DAVID FARLEY



The Addison-Wesley Signature Series

A MARTIN FOWLER SIGNATURE Book  
Martin Fowler

# CONTINUOUS INTEGRATION

IMPROVING SOFTWARE QUALITY  
AND REDUCING RISK

PAUL M. DUVALL  
WITH  
STEVE MATYAS  
ANDREW GLOVER



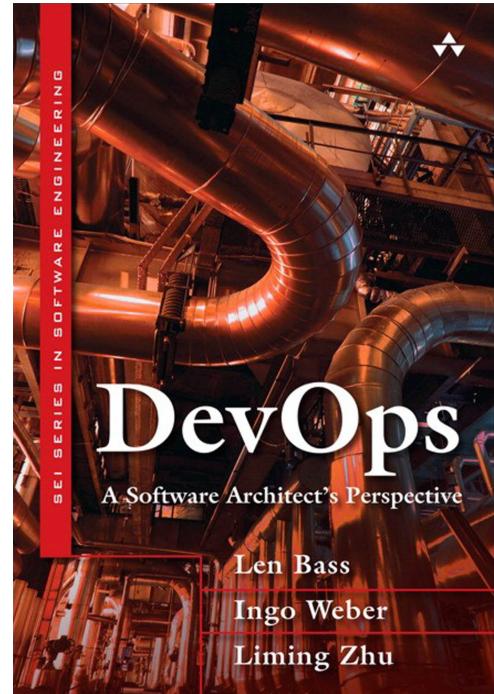
Forewords by Martin Fowler and Paul Julius

SEI SERIES IN SOFTWARE ENGINEERING

# DevOps

A Software Architect's Perspective

Len Bass  
Ingo Weber  
Liming Zhu



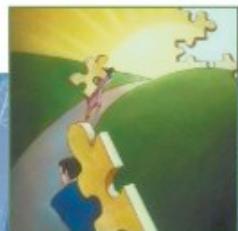
Help me test like Google

Life of a TE  
Life of an SDET  
Interviews with Googlers  
and more

James Whittaker • Jason Arbon • Jeff Carollo

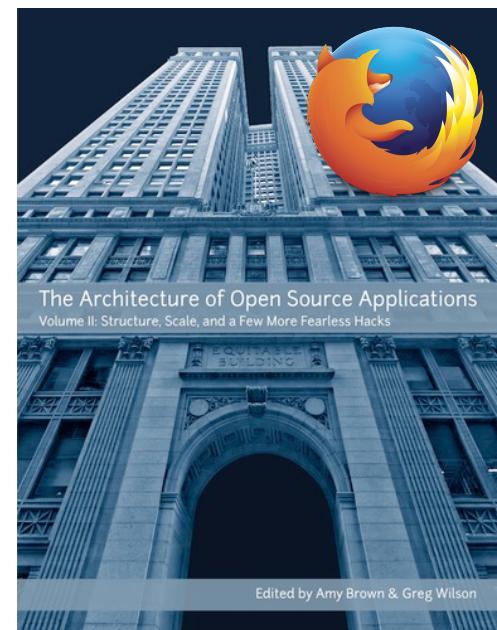
# SOFTWARE CONFIGURATION MANAGEMENT PATTERNS

Effective Teamwork, Practical Integration



STEPHEN P. BERCUZUK  
WITH BRAD APPLETON  
Foreword by Kyle Brown

SOFTWARE PATTERNS SERIES



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

# feature-based releases (i.e., when it's done)



vs.

# regular, time-based releases



6 months



6 weeks



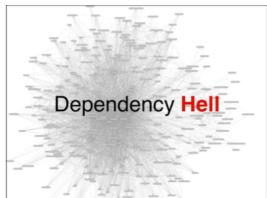
2 weeks (mobile)



twice/day  
(web)



dozens of  
times/day



dependency hell



pipeline security



education

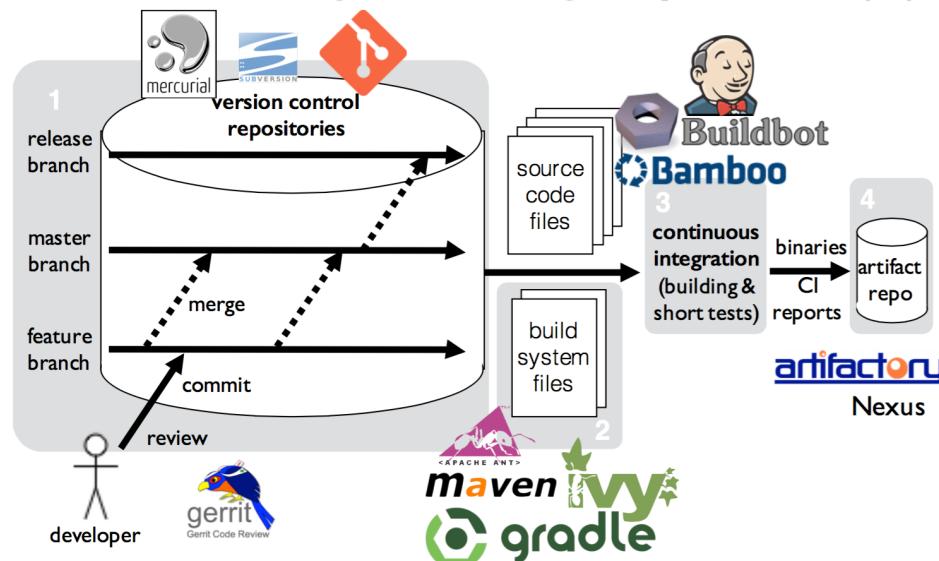


success & other stories



business value

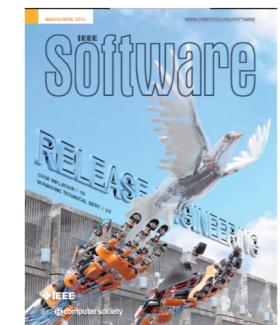
# Release Engineering Pipeline (I)



## A Roundtable with Three Release Engineers

### Participants:

- Mozilla (Kim Moir)
- Google (Boris Debic)
- Facebook (Chuck Rossi)



### Question topics:

- What metrics are most valuable to release engineers?
- How do you maintain quality and stability?
- Are there limitations for continuous delivery?
- How can we educate others about the value of release engineering?