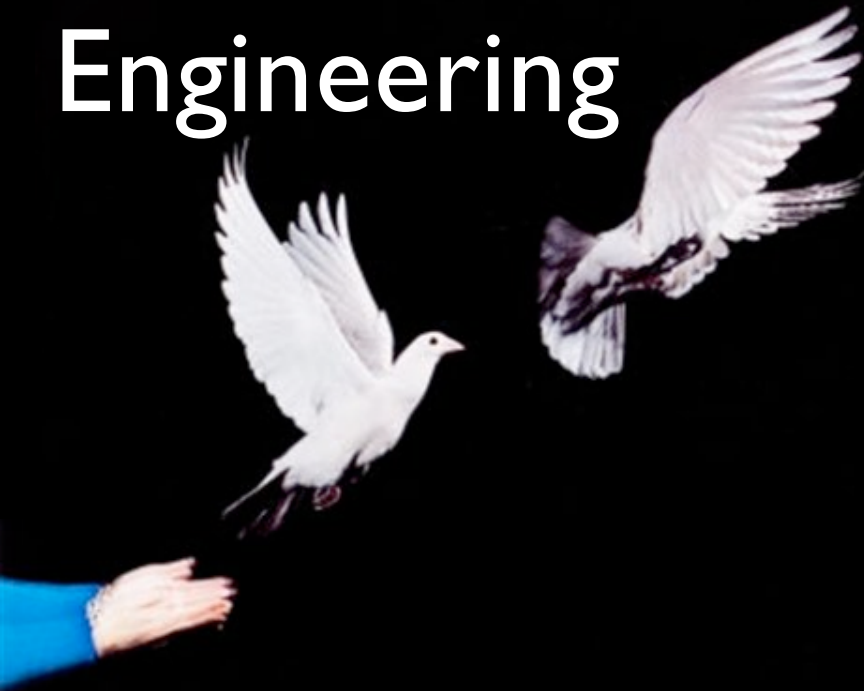


Lessons Learned from the
**International
Workshop**
on
**Release
Engineering**



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



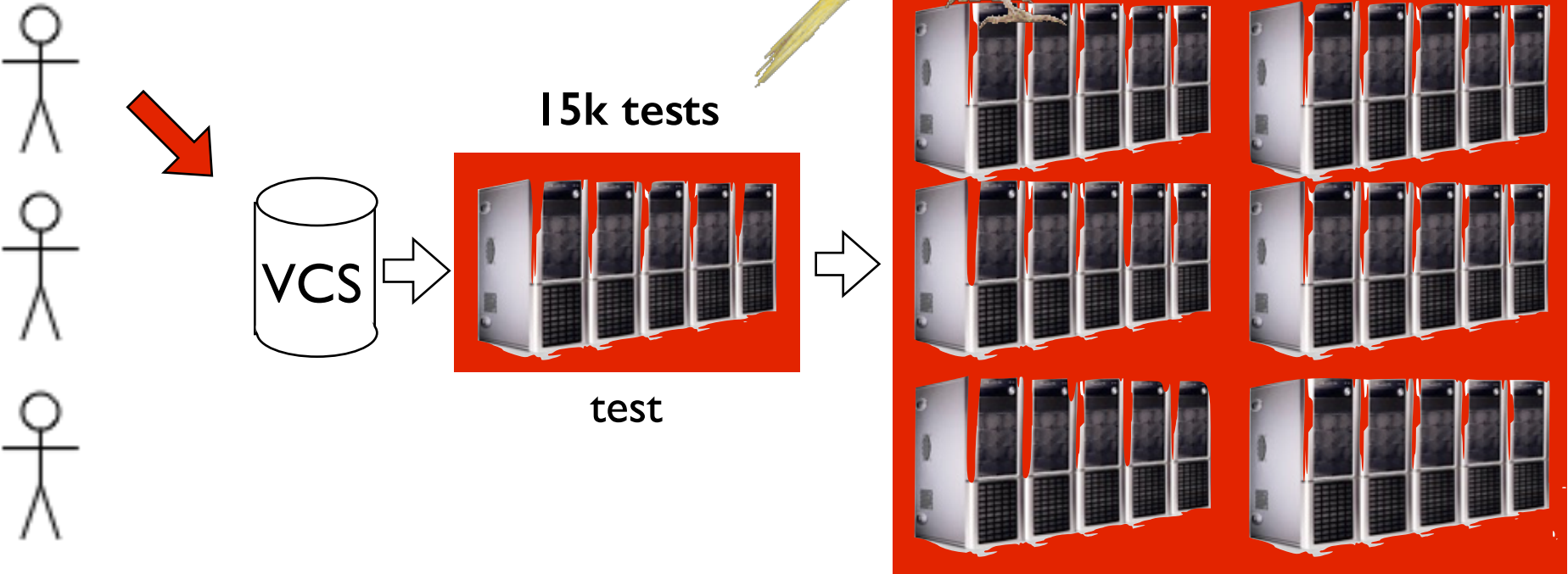
Over 2 Million people
like IMVU on Facebook!

M
C·S
I

FREE



Continuous Delivery the Early Days



continuous
integration

9 min.

6 min.

Nowadays ...

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

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

Firefox
Release!

Cycle Time !=
Development Time

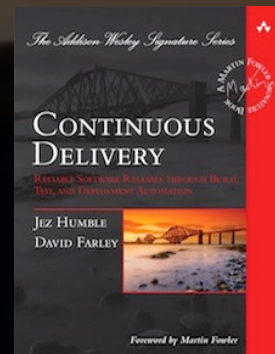
But why all
this trouble?



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.

Google

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





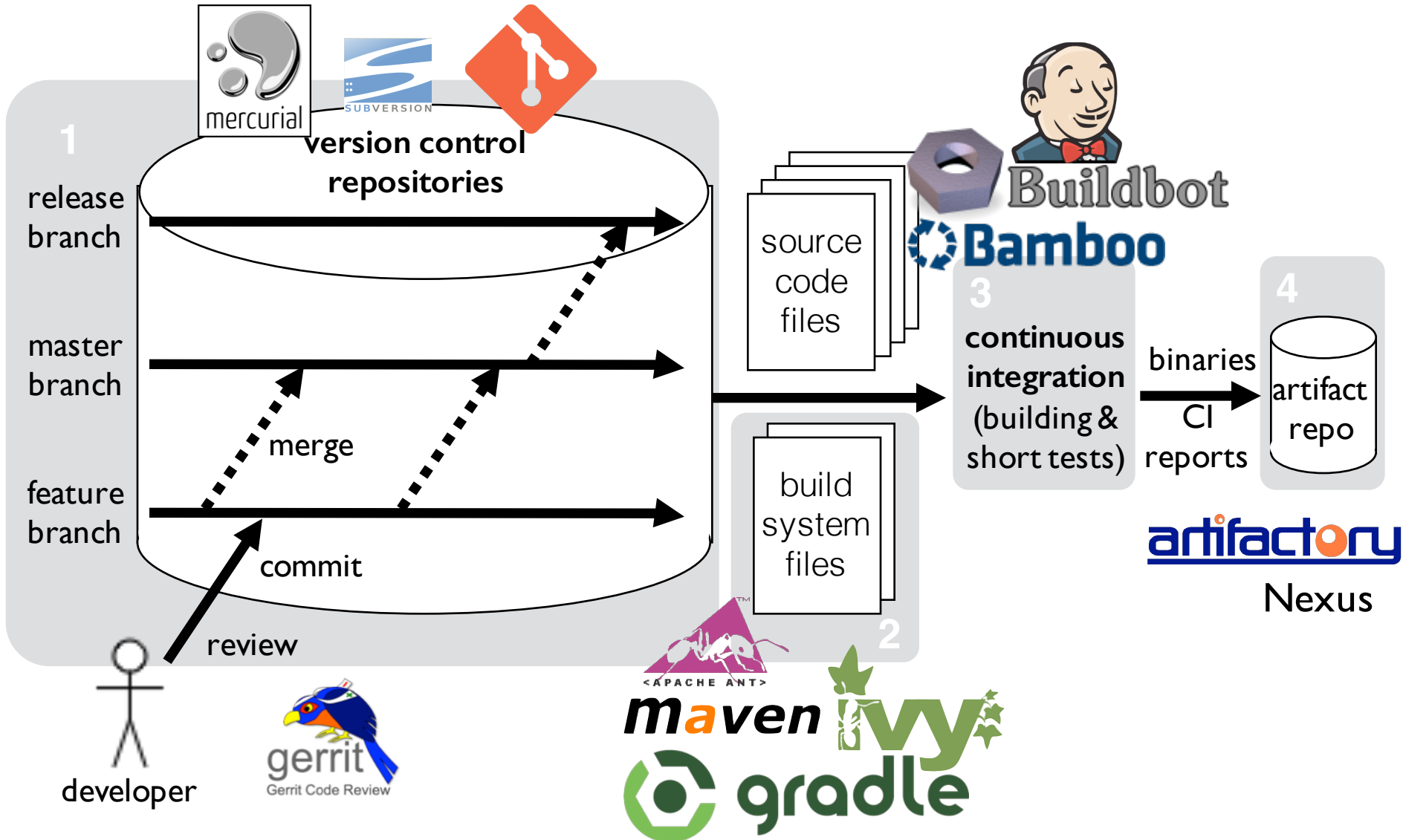
... aim to ship high quality
products without
regressing key quality
metrics

Chuck
Rossi

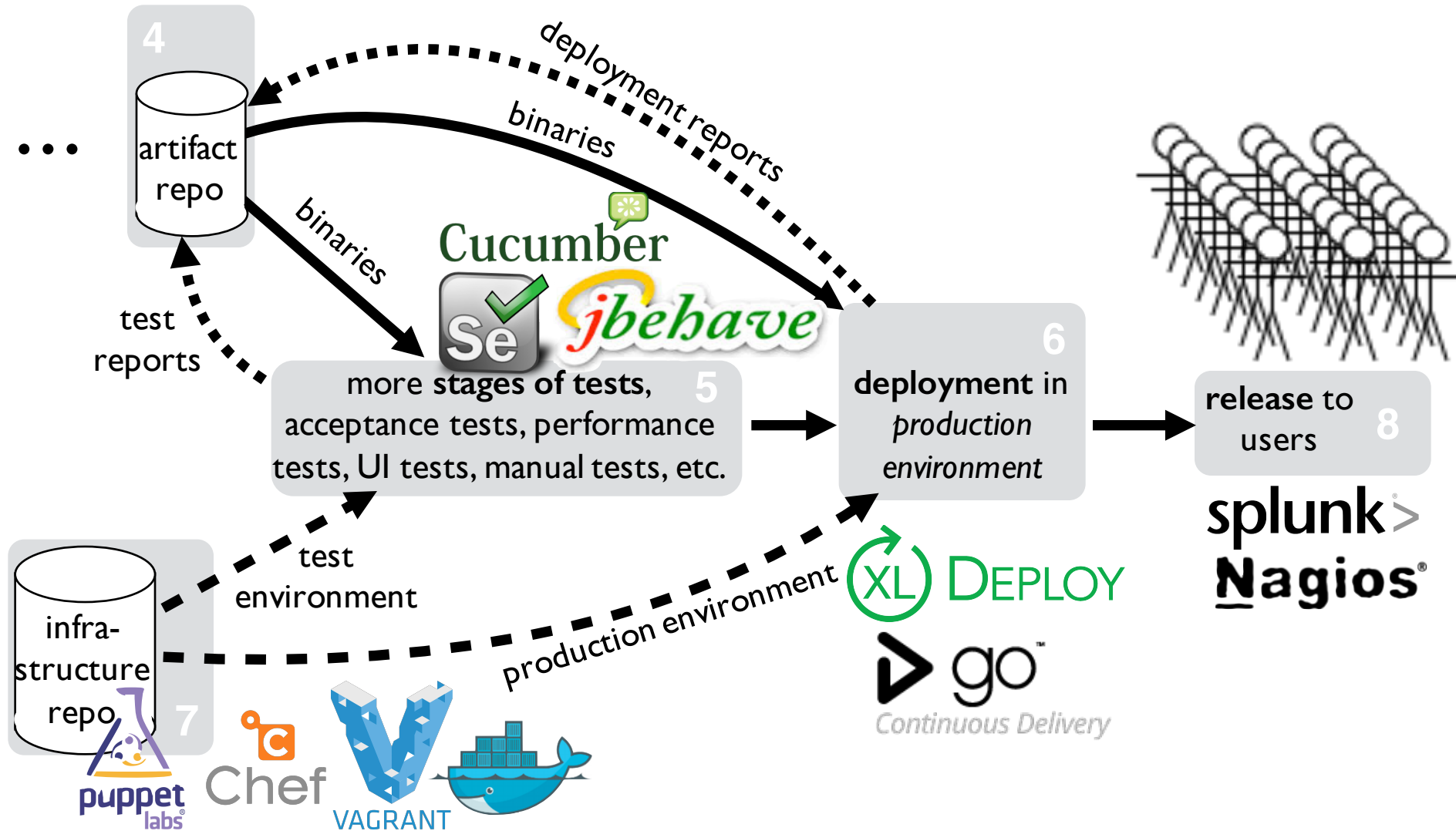


OK, what do I
need for this?

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

Os Open Source
Fr Free
Fm Freemium
Pd Paid
En Enterprise

Database



SCM



Build



CI



Repo Mgmt



Testing



Deployment



Config / Provisioning

Cloud / IaaS / PaaS

Release Mgmt

Collaboration

BI / Monitoring

Logging

Security

1 12c O En

3 3 Os My MySQL

4 4 Os Gt Git

11 11 En Mq MSSQL

12 12 Os Sv Subversion

19 19 Os Pq PostgreSQL

20 20 Fm Gh Github

21 21 Os Mv Maven

22 22 Os Gr Gradle

23 23 En Mr Meister

24 24 Os Jn Jenkins

25 25 Pd Ba Bamboo

26 26 Os Tr Travis CI

27 27 Fr Ar Archiva

28 28 Os Fn FitNesse

29 29 Fr Se Selenium

30 30 Os Gn Gatling

31 31 Pd Gd Deployman Manager

32 32 Os Sf SmartFrog

33 33 Fr Cb Cobbler

34 34 Os Bc Bcfg2

35 35 Os Kb Kubernetes

36 36 En Rs Rackspace

37 37 Os Mg MongoDB

38 38 Fm Bb Bitbucket

39 39 Os Br Buildr

40 40 Os At ANT

41 41 Fm Bm BuildMaster

42 42 Fm Cs Codeship

43 43 Fm Sn Snap CI

44 44 Fm Cr CircleCI

45 45 Os Nx Nexus

46 46 Fr Cu Cucumber

47 47 Os Cj Cucumber

48 48 Fr Qu Qunit

49 49 Fr Cp Capistrano

50 50 Fr Ju JuJu

51 51 Os Rd Rundeck

52 52 Os Cf CFEngine

53 53 Fr Pk Packer

54 54 Fm Bx Bluemix

55 55 En Db DB2

56 56 Os Hg Mercurial

57 57 Fm Qb QuickBuild

58 58 En Ub UrbanCode Build

59 59 Pd Ta Visual

60 60 Fm Tc TeamCity

61 61 Fm Sh Shippable

62 62 Os Cc CruiseCont

63 63 Os Ay Artifactory

64 64 Fr Jt JUnit

65 65 Fr Jm JMeter

66 66 Fr Tn TestNG

67 67 En Ry RapidDepl

68 68 Fm Cy CodeDepl

69 69 En Oc Octopus Deploy

70 70 Os No CA Nolio

71 71 En Eb ElasticBox

72 72 En Ad Apprenda

73 73 Fr Cs Cassandra

74 74 En Hx Helix

75 75 Os Msb MSBuild

76 76 Os Rk Rake

77 77 Os Lb LuntBuild

78 78 Os Co Continuum

79 79 Fm Ca Continua

80 80 Os Gu Gump

81 81 Os Ng NuGet

82 82 Os Ap Appium

83 83 En Xltv XL

84 84 En Tc TestCompl

85 85 Os Go Go

86 86 En Ef ElectricFlow

87 87 En Xld XL Deploy

88 88 En Ud UrbanCode Deploy

89 89 Os Mo Mesos

90 90 Os Cf Cloud

91 91 En Xlr XL

92 92 En Ur UrbanCode Release

93 93 En Ls CA Service

94 94 En Bm BMC Release

95 95 En Hp HP Codar

96 96 Pd Ex Excel

97 97 En Pl Plutora Release

98 98 En Sr Serena Release

99 99 Fm Tr Trello

100 100 Pd Jr Jira

101 101 Fm Rf HipChat

102 102 Fm Sl Slack

103 103 Fm Fd Flowdock

104 104 Pd Pv Pivotal Tracker

105 105 En Sn ServiceNow

106 106 Os Ki Kibana

107 107 Fm Nr New Relic

108 108 Os Ni Nagios

109 109 Os Gg Ganglia

110 110 Os Ct Cacti

111 111 Os Gr Graphite

112 112 Fm Le Logentries

113 113 En Sp Splunk

114 114 Fm Sl Sumo

115 115 Os Ls Logstash

116 116 Fm Lg Loggly

117 117 Os Gr Graylog

118 118 Os Sn Snort

119 119 Os Tr Tripwire

120 120 En Cy CyberArk

Share



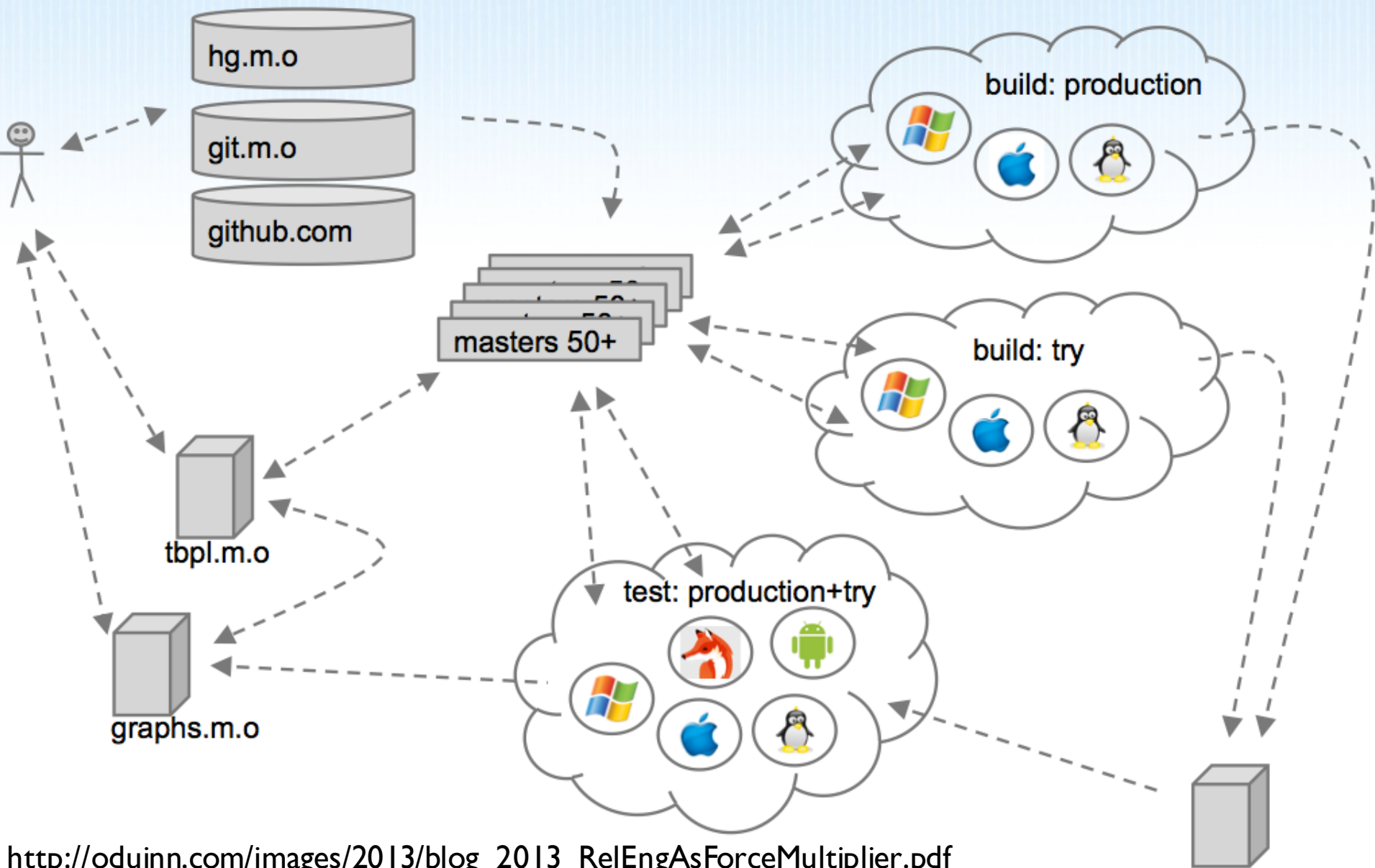
Embed



Become Excellent!

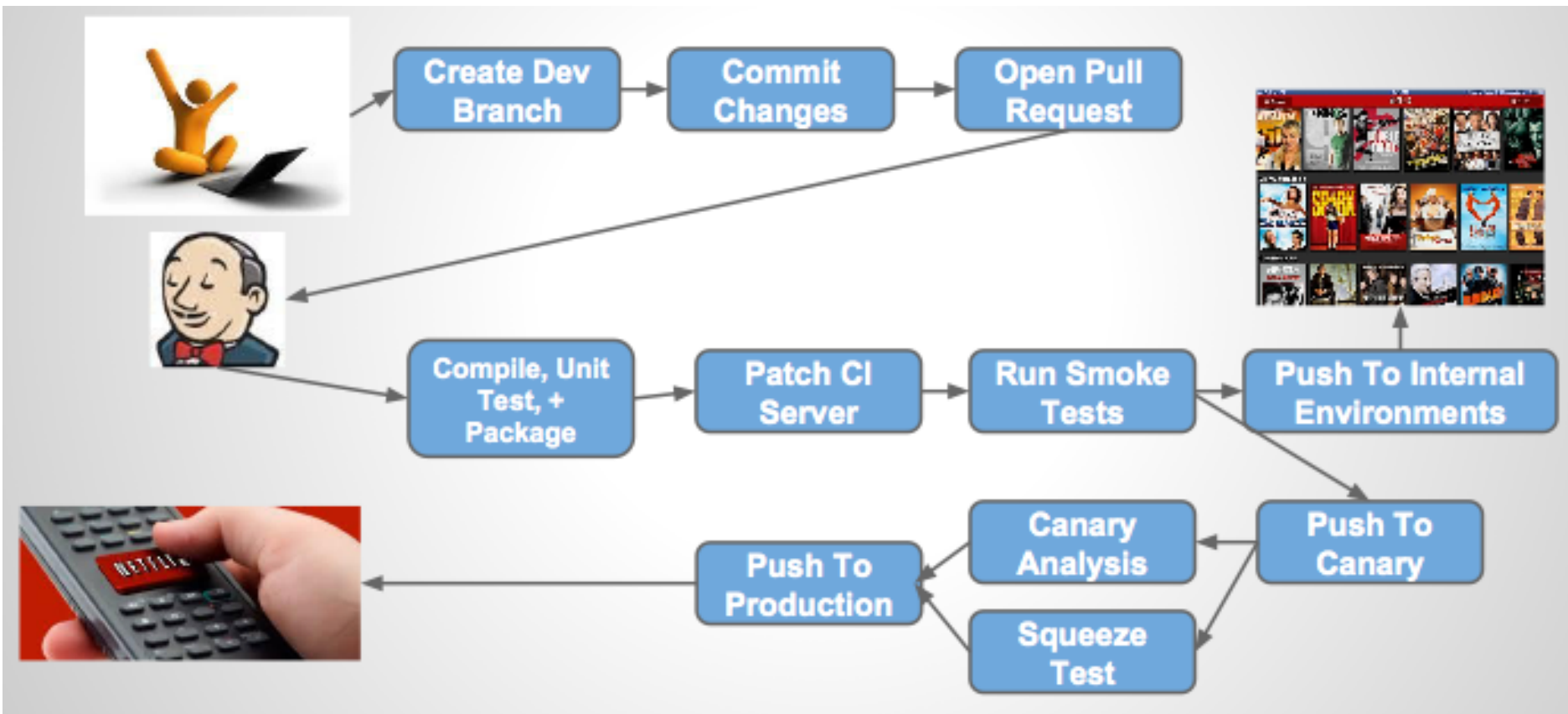
[Subscribe here!](#)

Example Pipeline



Another Example Pipeline

NETFLIX



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



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

81

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 fast 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.

s have put
st two weeks

were the
en of death. A
cond coming

d out to slow



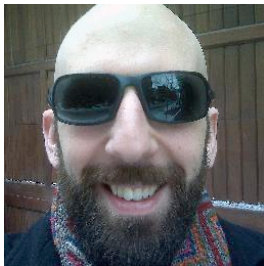
Continuous delivery for
web apps is a solved
problem ...

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

Members

Forums

■ Blog

Bookmarks

Events

Wiki

Related Communities

▼ Subcommunities

DevOps Community

Leadership

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).

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](#).

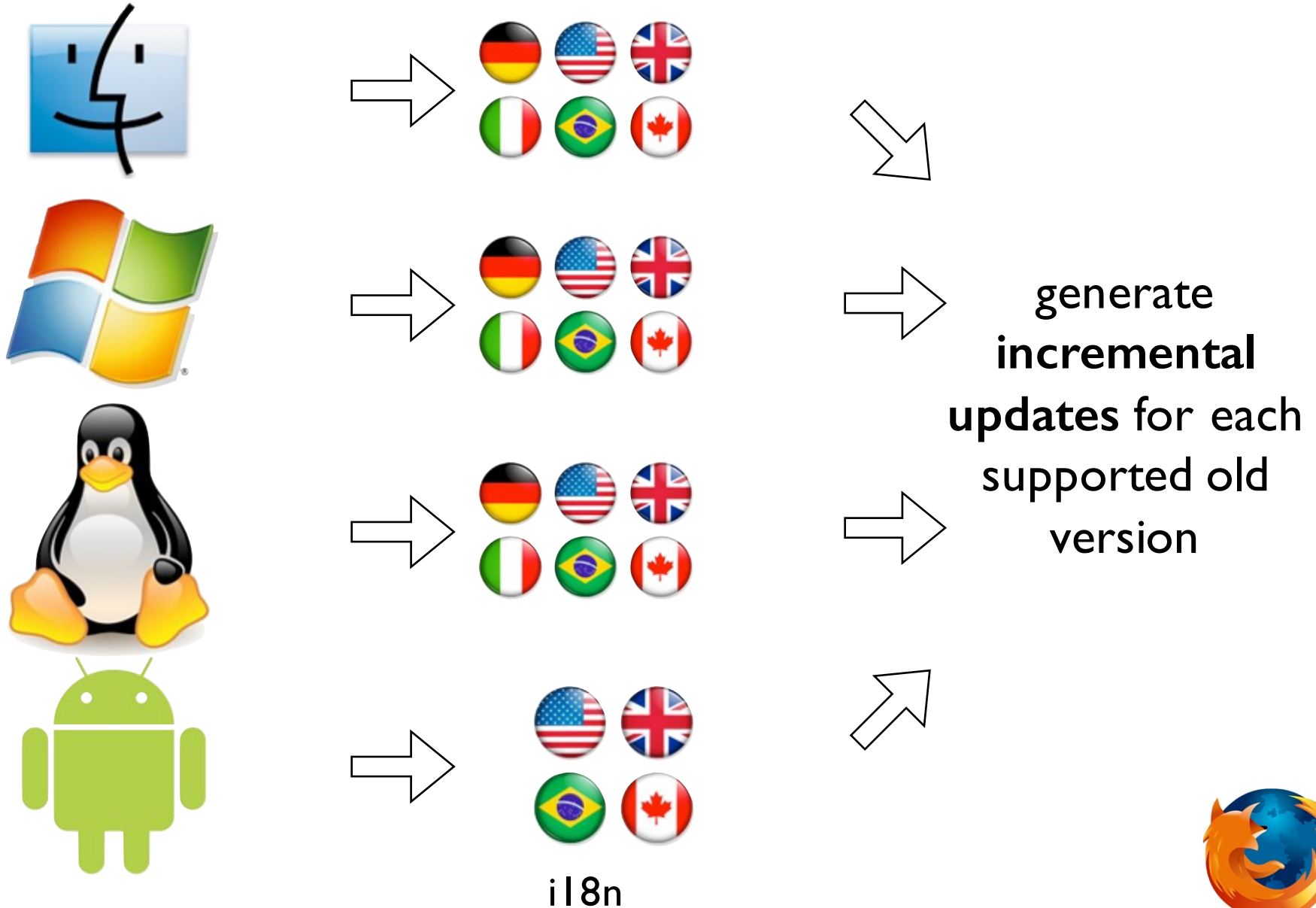
One Size Does not fit all



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

Chuck
Rossi

Heterogeneous Hardware, Oses, ...



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.



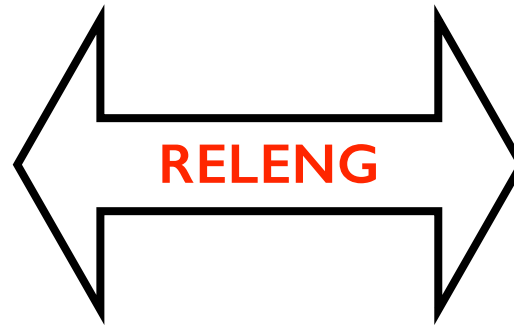
As a result, **aggressive software releases can have a serious and unintended side-effect: a build-up of technical debt**. With every software and update release, there's the potential for new bugs and technical vulnerabilities to be uncovered.

tool, it can also pose some real security issues.

Are These Concerns Justified?



practitioners



researchers

1st International Workshop on Release Engineering



6 practitioner
talks

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

2 keynotes



Keynote: **Release Engineering as a "force-multiplier"**
by **John O'Duinn**
(Director of Release Engineering at Mozilla Corporation)

... and 10 academic
talks

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

(Director of Engineering Services at LinkedIn)

LinkedIn



<http://releng.polymtl.ca/RELENG2013/html/index.html>

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** is **available**

... and 6 academic
talks

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

3rd International Workshop on Release Engineering

May 19, 2015, Florence, Italy



6 practitioner
talks



I keynote

Home/CFP

Venue

Program

IEEE Special
Issue

PC Members

Organizers

Registration

History of RELENG

3rd International Workshop on Release Engineering 2015

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:

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

... and 6 academic
talks

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

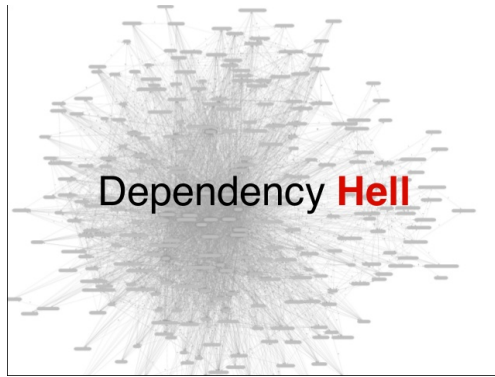
February 19: 6 talks, 6 papers

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!)

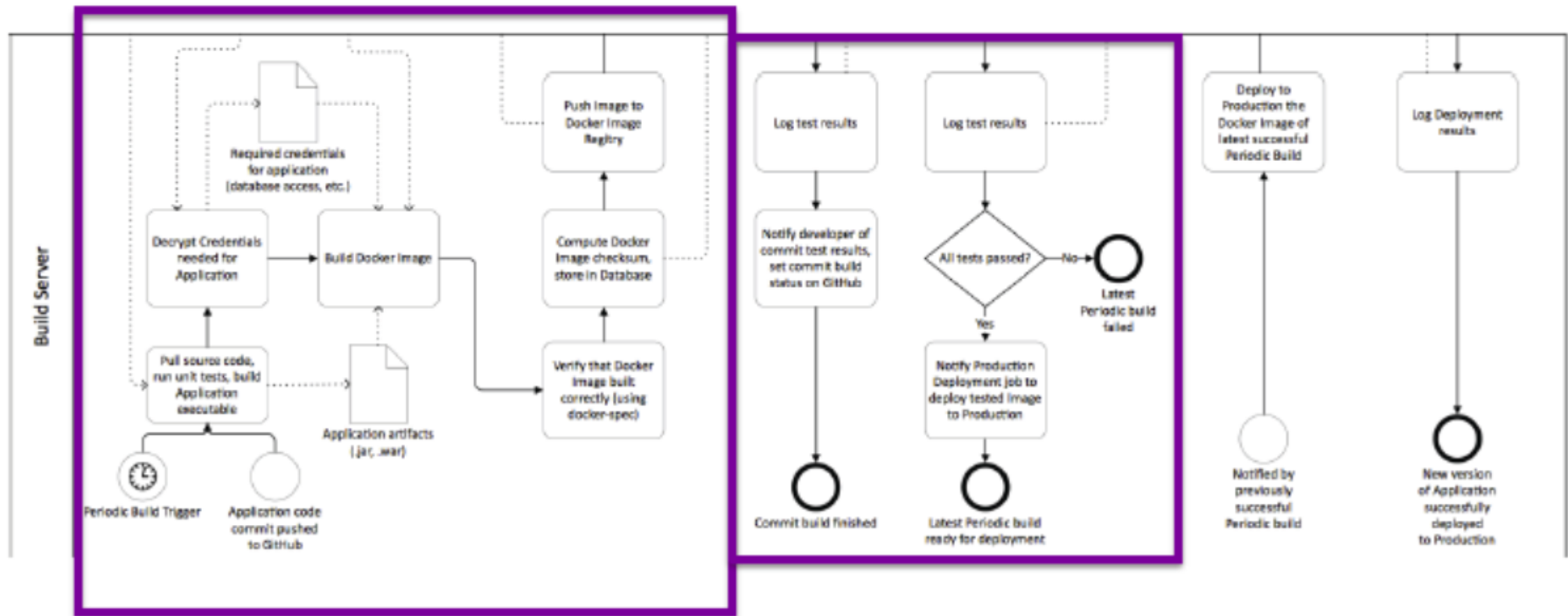
The Netflix logo, consisting of the word "NETFLIX" in a bold, red, sans-serif font.

- 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



- 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



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 38th International Conference on Software Engineering, Austin, TX, May 14-22, 2016

International Workshop on Continuous Software Evolution and Delivery (CSED)

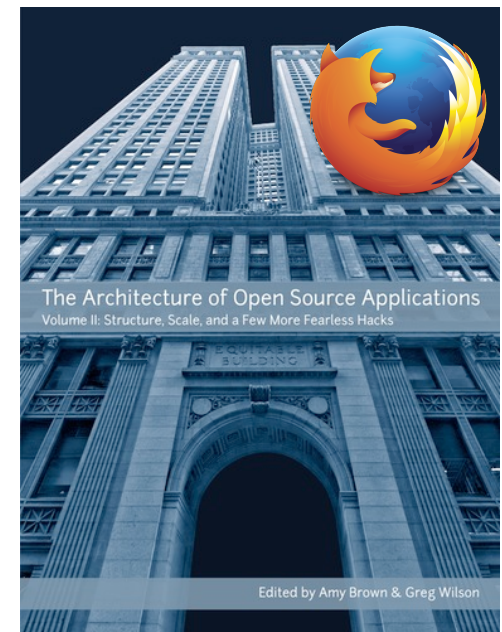
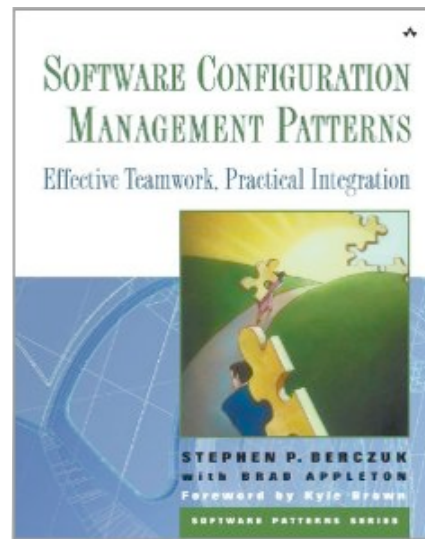
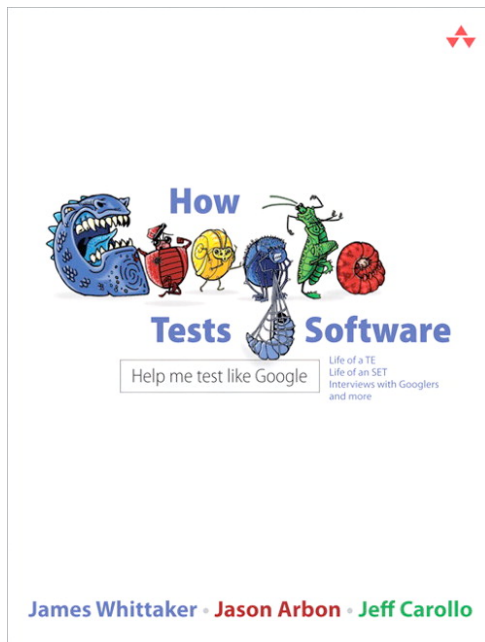
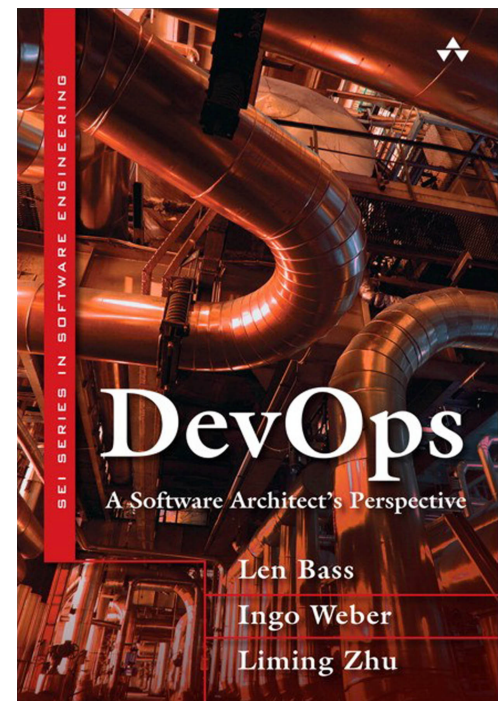
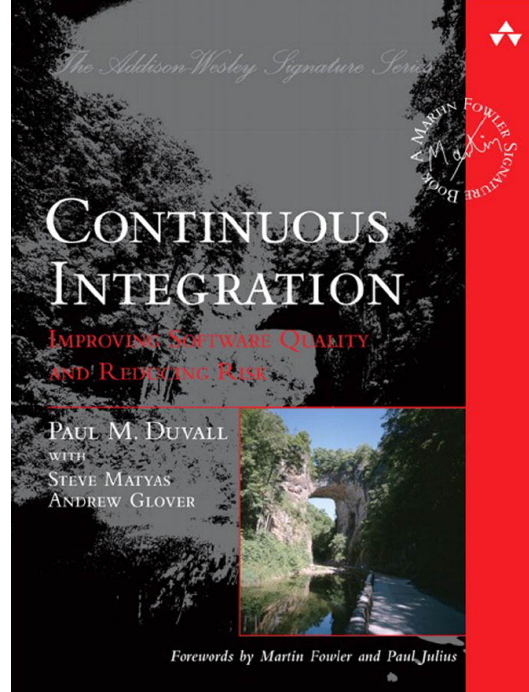
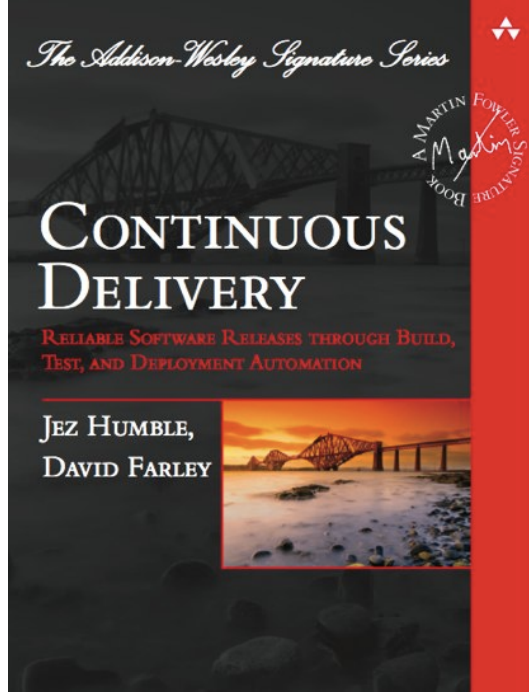
Important Dates

- 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!



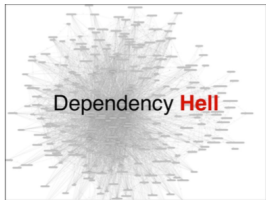
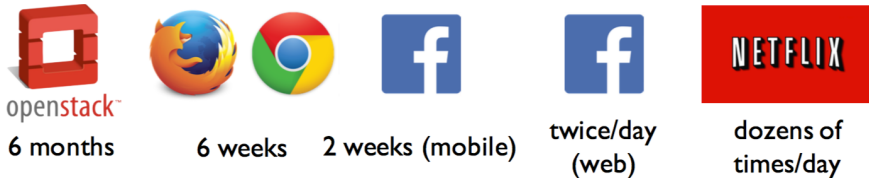
<http://releng.polymtl.ca/>

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



vs.

regular, time-based releases



dependency hell



pipeline security



education

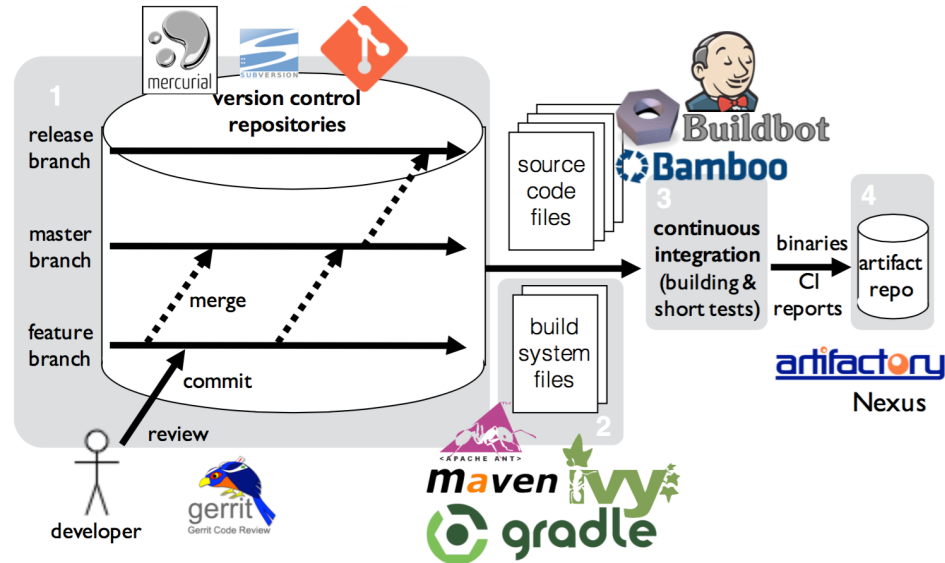


success & other stories



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?