

# NAVIGATING THE SOFTWARE DELIVERY MINEFIELD: DEVOPS & THE ART OF RELEASE ENGINEERING

J. PAUL REED  
RELEASE ENGINEERING APPROACHES  
DEVOPS ENTERPRISE SUMMIT / SAN FRANCISCO / 2017

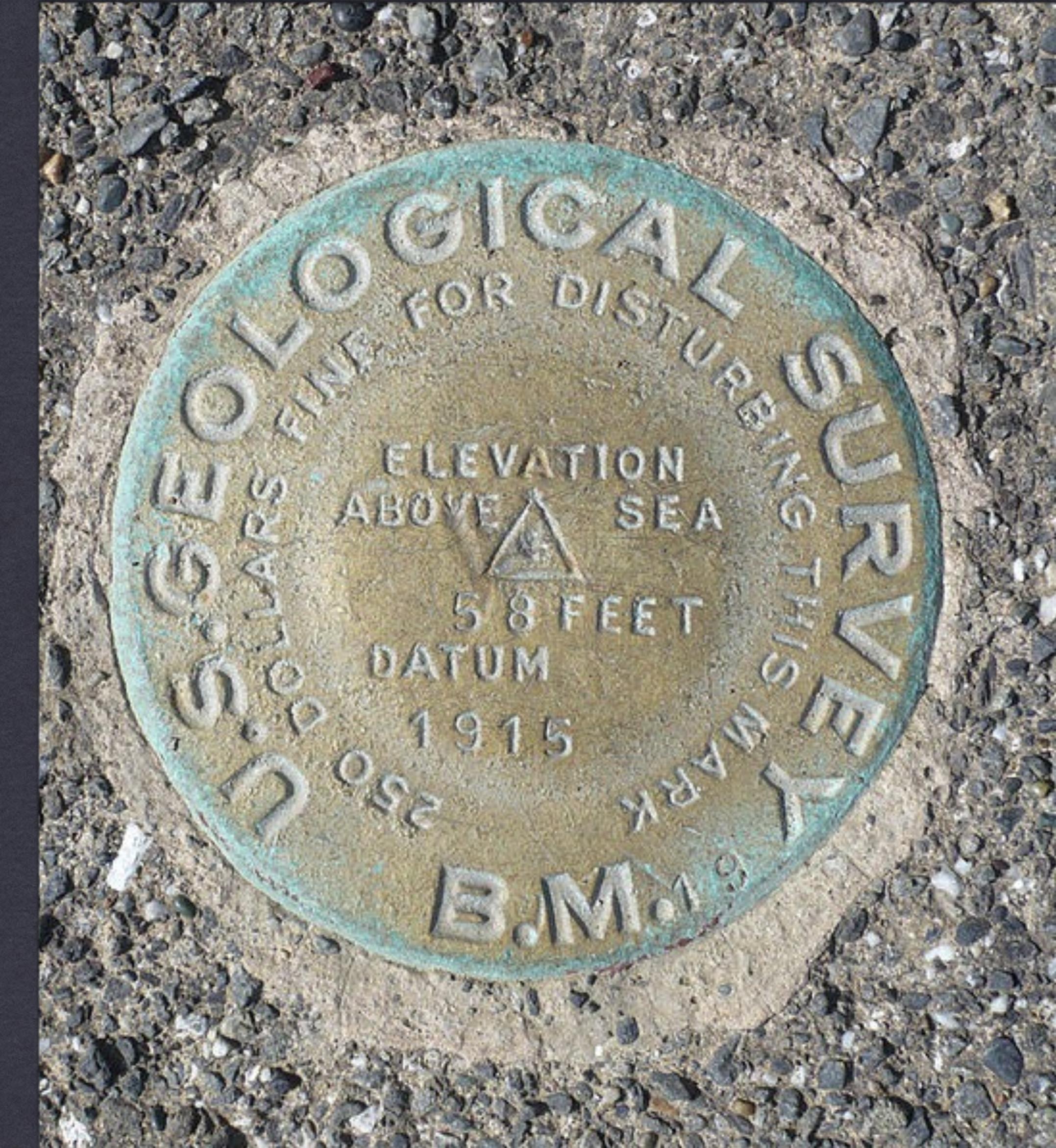


# J. PAUL REED

- [@JPAULREED ON !\[\]\(467d80e979964f7f8c752fb22248b5b7\_img.jpg\)](https://twitter.com/jpaulreed)
- [@SHIPSHOWPODCAST ALUM](https://twitter.com/SHIPSHOWPODCAST)
- **18+ YEARS IN BUILD/RELEASE  
ENGINEERING**
- **NOW, A DEVOPS CONSULTANT™**
- **MASTERS OF SCIENCE  
CANDIDATE IN HUMAN  
FACTORS AND  
SYSTEMS SAFETY**



LET'S START  
WITH A SURVEY



# A Long Time Ago, On A Release Engineering Team Far, Far Away...



@jpaulreed

#DOES17

# A Long Time Ago, On A Release Engineering Team Far, Far Away...



# Release Engineering: Not a Game



<http://jpaulreed.com/the-elevator-storyteller/>

@jpaulreed

#DOES17

# My Boss' Reaction



@jpaulreed

#DOES17

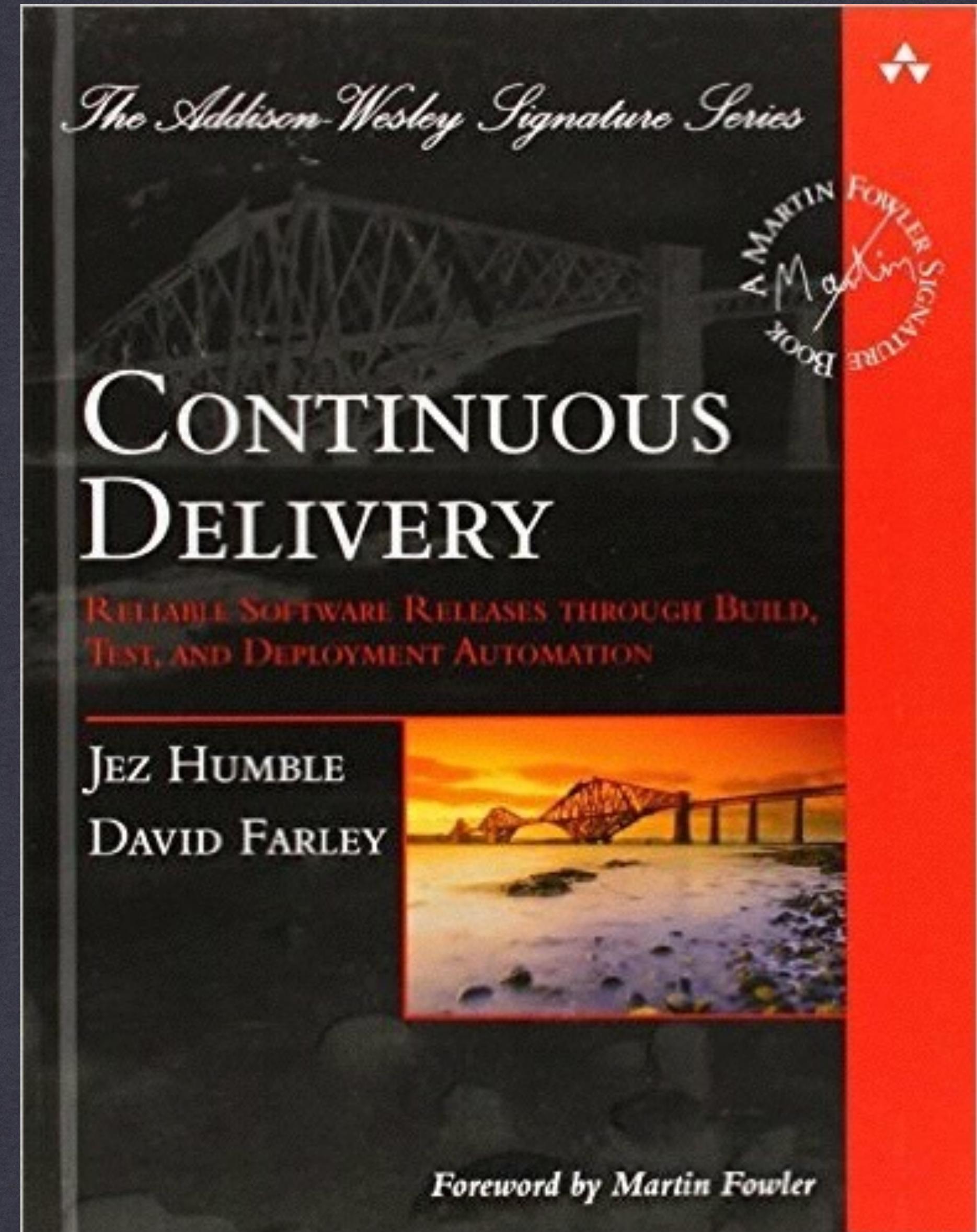
# Our Flight for Today

---

- “Do We Have a Release Engineering Problem?”
- The *Art* of Release Engineering: Human Factors

# CONTINUOUS DELIVERY

(WHO'S HEARD OF IT?)



*Continuous Delivery  
means minimizing  
lead time from idea  
to production and  
then feeding back to  
idea again.*

*– Rolf Andrew Russell*



*Continuous Delivery  
is a set of principles  
and practices to  
reduce the cost,  
time, and  
risk of delivering  
incremental changes  
to users.*

— Jez Humble



*Continuous Delivery Is Your  
Organization – Your Entire  
Organization – Caring  
About Release Engineering  
And Quality Assurance In A  
Way It Has Never Cared  
About Them Before. Ever.*

— Me

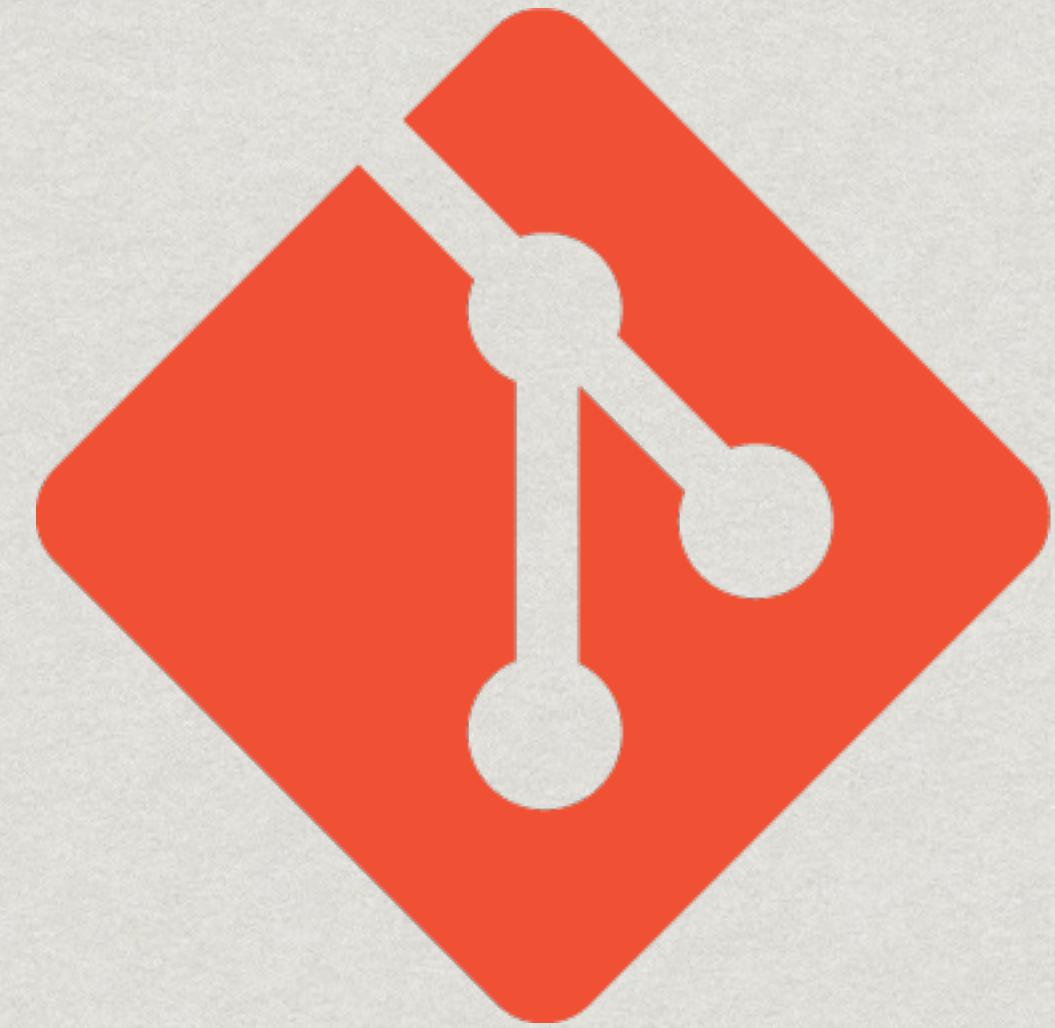


SAY IT WITH ME:  
RELEASE MANAGEMENT  
IS NOT  
RELEASE ENGINEERING

# “Release Engineering Smells”



# REL ENG SMELL #1: MODERN VERSION CONTROL CHAOS

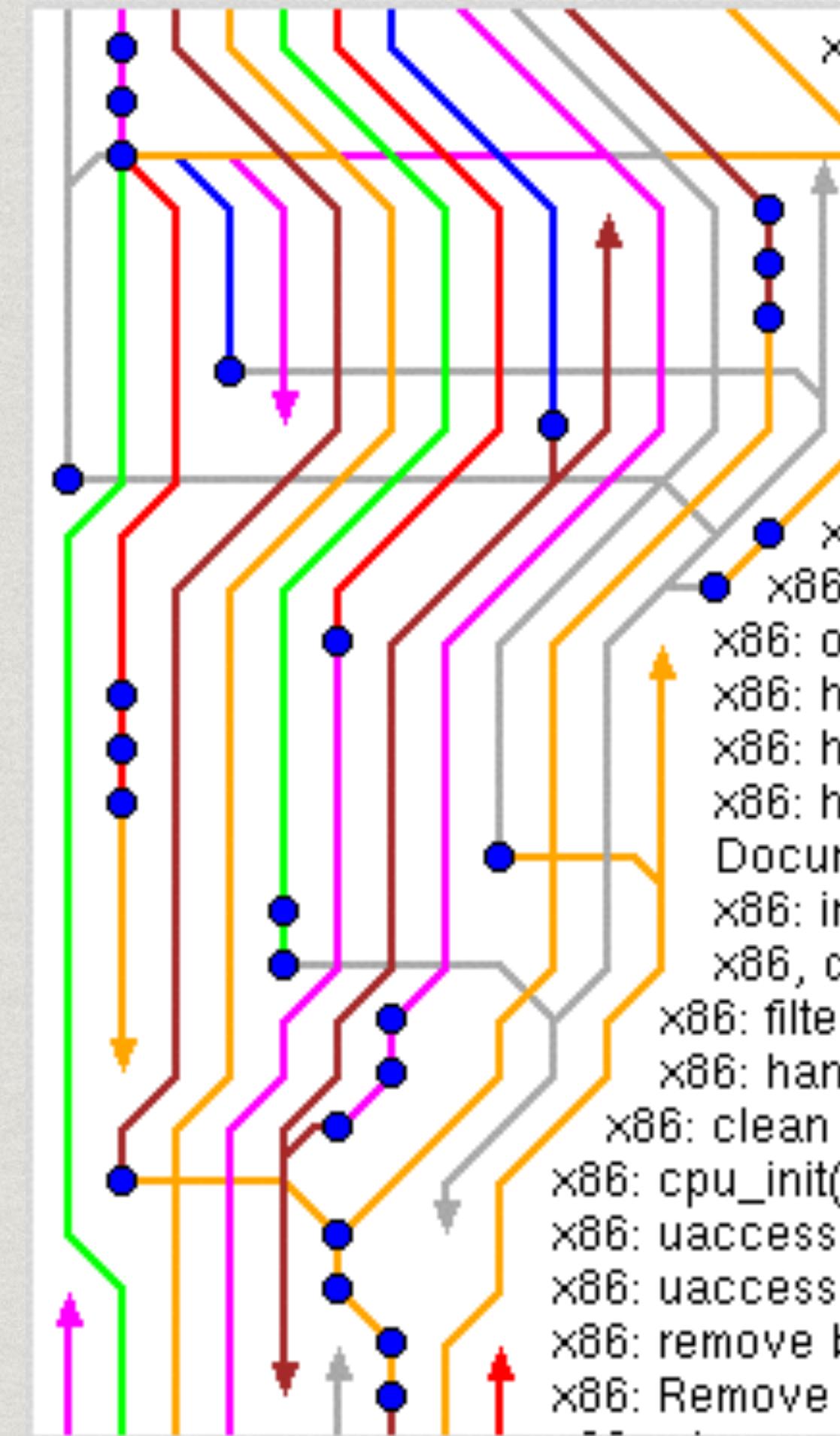


git



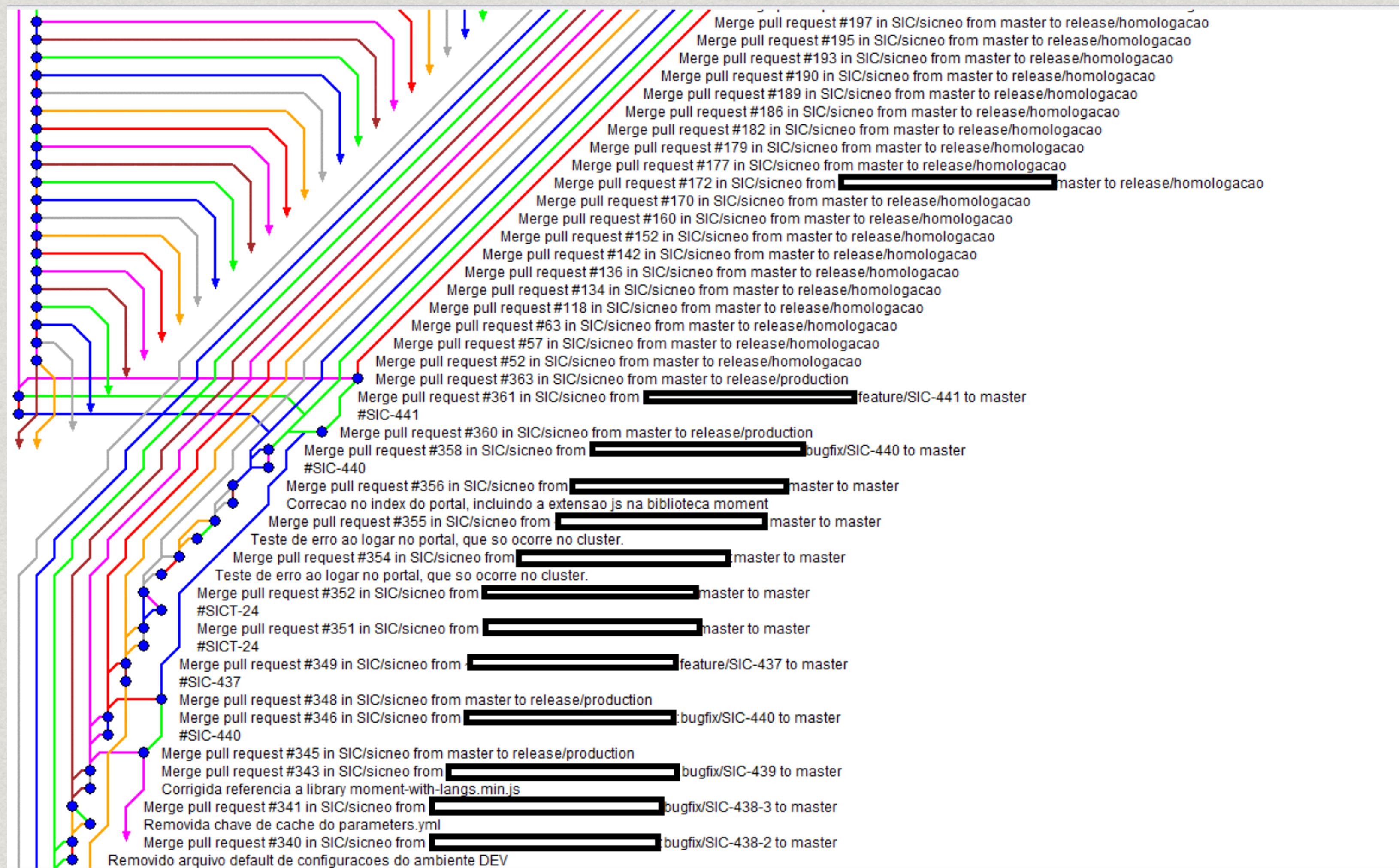
GitHub

# “What’s Our Commit History Look Like?”

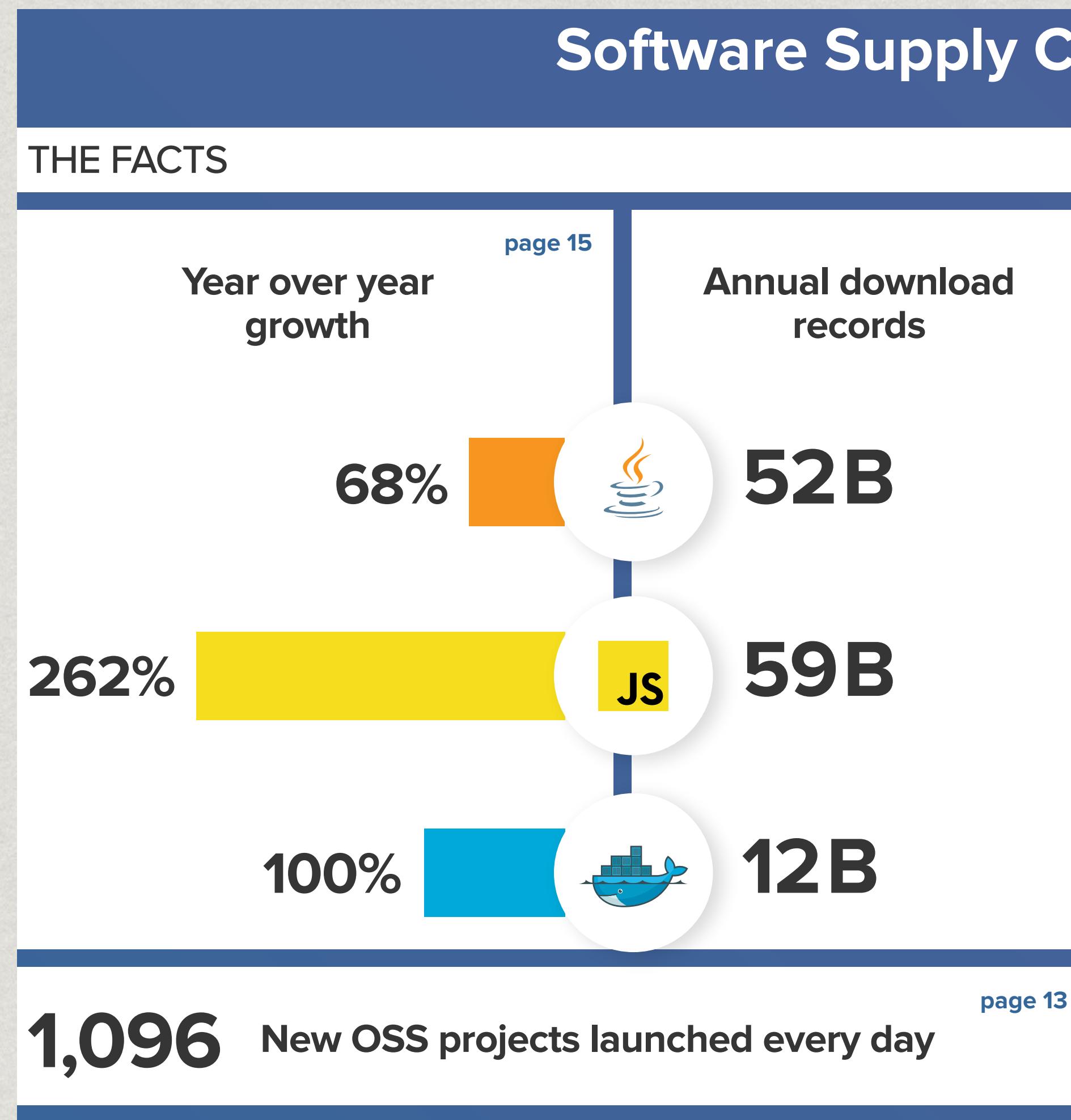


x86, genapic: refactor genapi	In	2009-01-27 14:07:42
x86, genapic: refactor gena	In	2009-01-27 14:07:42
Merge branches 'x86/as	In	2009-01-28 14:13:53
x86: ia32_signal: use {ge	Hi	2009-01-23 15:50:38
x86: signal: use {get put}	Hi	2009-01-23 15:50:10
x86: uaccess: introduce	Hi	2009-01-23 15:49:41
x86, rdc321x: remove/mo	In	2009-01-18 10:37:21
x86 setup: fix asm constr	Ai	2009-01-27 12:45:57
x86 PAT: ioremap_wc shot	ve	2009-01-09 16:13:13
x86: add pte_set_flags/clear_	Je	2009-01-22 14:24:22
x86/pvops: remove pte_flags pv	Je	2009-01-22 14:24:16
x86: optimise page fault entry, clea	Jc	2009-01-21 11:45:41
x86: headers cleanup - setup.h	Ci	2009-01-14 12:37:50
x86: headers cleanup - sigcontext	Ci	2009-01-14 12:37:49
x86: headers cleanup - prctl.h	Ci	2009-01-14 12:37:47
Documentation/x86/boot.txt: modify	Ba	2009-01-08 03:24:29
x86: improve early fault/irq printout	In	2009-01-25 21:09:00
x86, debug: remove early_printk()	In	2009-01-26 05:18:05
x86: filter CPU features dependent on	H.	2009-01-23 17:20:50
x86: handle PAT more like other CPU	H.	2009-01-22 16:17:05
x86: clean up stray space in <asm/proc	H.	2009-01-23 17:18:52
x86: cpu_init(): remove ugly #ifdef constru	Pe	2009-01-27 12:41:34
x86: uaccess: rename __put_user_u64() to	Hi	2009-01-19 16:37:41
x86: uaccess: fix style problems	Hi	2009-01-19 16:34:26
x86: remove byte locks	Ji	2009-01-12 03:53:45
x86: Remove never-called arch_setup_ms	M	2009-01-18 22:06:42

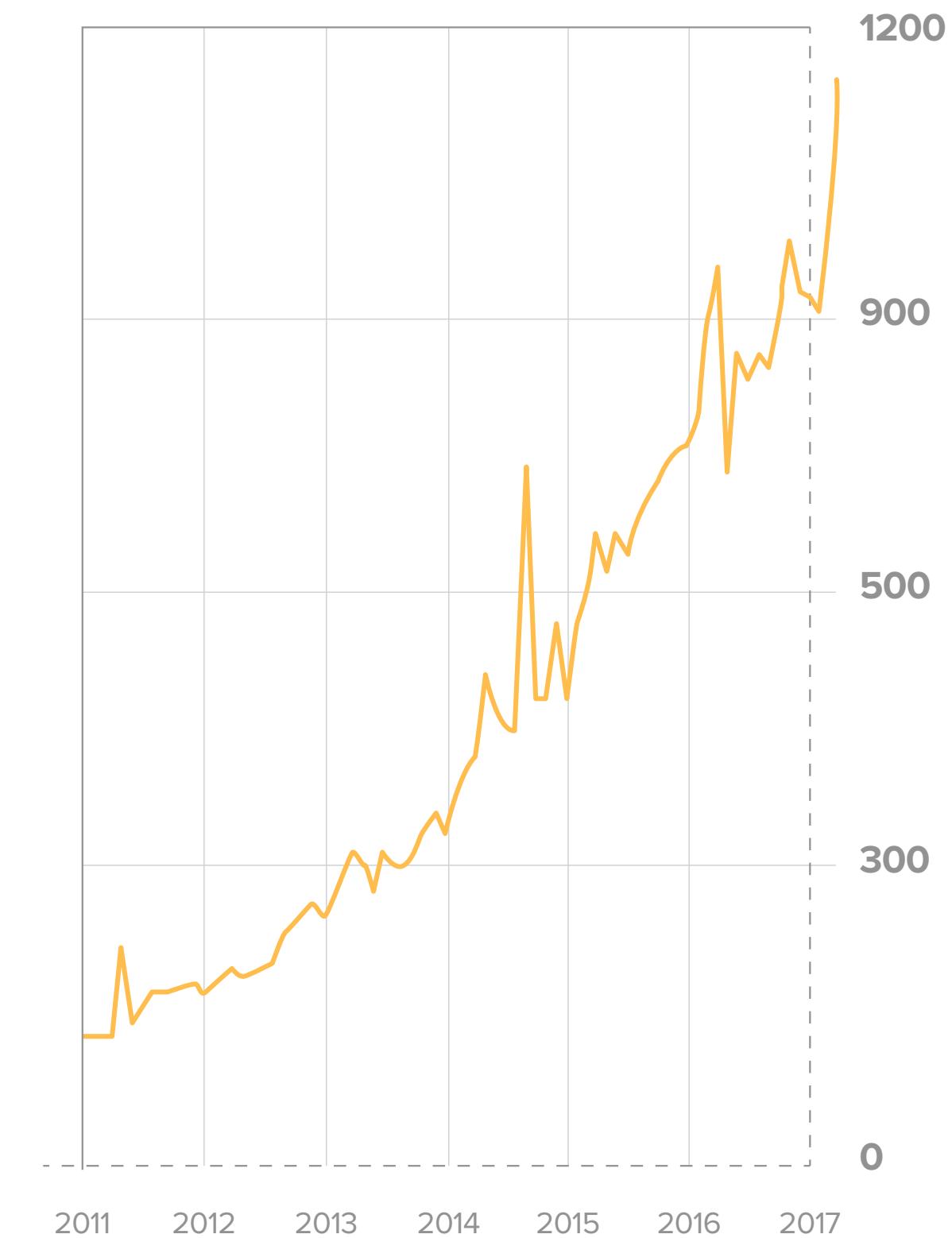
# “I’m sorry I asked...”



# An Explosion of Repositories



Average number of new OSS Projects coming to market per day



**REL ENG SMELL #2:  
“CAN SOMEONE JUST TELL  
ME: WHAT. IS. RUNNING.  
IN. PRODUCTION?!!”**

# The Answer Was Easy Once



# A Lil' Config Management Survey

- Raise your hand if, for the app you're responsible for:
  - You can tell me what artifacts are running in production right now
  - ... using a simple, human-parseable identifier
  - ... that is unique to the entire company
  - ... and is traceable back to the commit/action that generated the artifacts...
  - ... in a way that I can identify the *exact* person who wrote any arbitrary line of code in any part of the product
  - ... for all the open source dependencies, too
- And this is just the software artifacts: we haven't even talked operations yet!

# REL ENG SMELL #3: REPRODUCIBILITY IS A PROBLEM. STILL.

# “But, Containers Will Save Us!”

---



# Containers Will Not Save You



At least, not by themselves.

# Who Owns My Software Supply Chain?



YOU

Recommended reading from selected authors:

◀ [Rugged DevOps: Going Even Faster with Software Supply Chains](#) ▶

This reminder is brought to you by [@jpaulreed](#). Inspired by [WhoOwnsMyAvailability.com](#)

[whoownsyoursoftwaresupplychain.com](#)

<http://WhoOwnsMySoftwareSupplyChain.com>

@jpaulreed

#DOES17



# 2017 STATE OF THE SOFTWARE SUPPLY CHAIN

Sonatype's 3rd annual report on managing  
open source components to accelerate innovation.



<https://www.sonatype.com/2017-state-of-the-software-supply-chain-report>

@jpaulreed

#DOES17

DEVOPS,  
CONTINUOUS DELIVERY, LEAN,  
AND THE REST  
DO NOT ADDRESS THESE  
PROBLEMS

# THE ART OF RELEASE ENGINEERING?

# A Different View of the System

---



“HUMAN FACTORS?”

# An Example

---

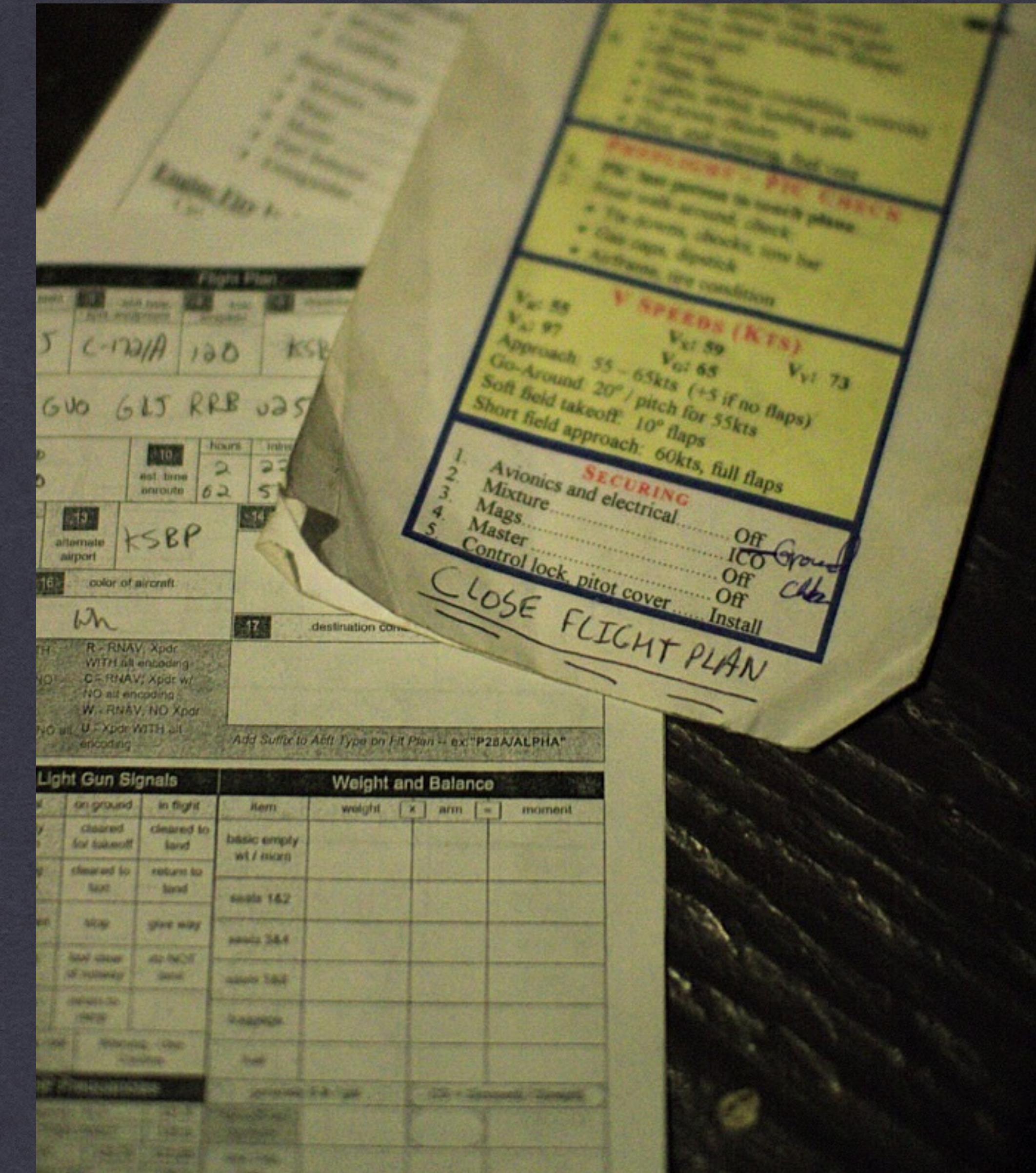
**ourapp-1.0.3-34.x86\_64.rpm**

# An Example

---

`ourapp-1.0.3-20171108.x86_64.rpm`

AUTOMATION  
RELEASE ENGINEERS  
DON'T JUST IMPLEMENT  
AUTOMATION, THEY  
DESIGN AUTOMATION  
FOR OTHERS IN  
COMPLEX SOCIO-  
TECHNICAL SYSTEMS.



# Another Example: Versioning

---

Which is easier for two people to discuss?

- “I have revision alpha-charlie-delta-three-bravo-niner checked out in my Git clone. Which do you have?”
- “I have revision 28492 synced in my repository clone.”
- (A fun question: what revision precedes these two?)

# Versioning: A “Solved Problem,” right?!

auroraeosrose  
@auroraeosrose

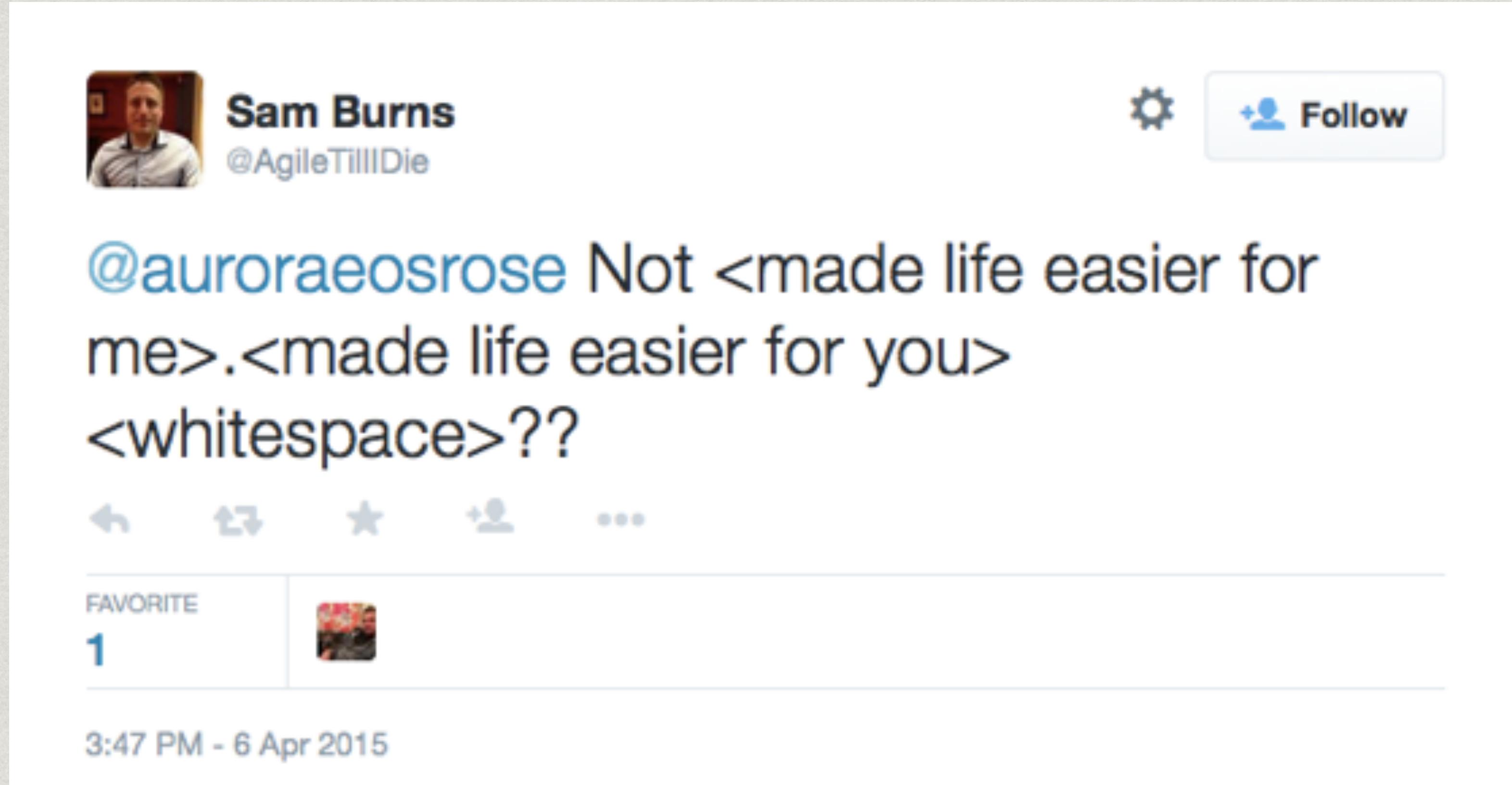
Follow

As an author semver is <I WARNED YOU>. <This is shiny>. <oops my bad>

RETWEETS 260 FAVORITES 166

3:36 PM - 6 Apr 2015

# Versioning: A “Solved Problem,” right?!



Sam Burns  
@AgileTillIDie

@auroraeosrose Not <made life easier for me>.<made life easier for you>  
<whitespace>??

FAVORITE  
1

3:47 PM - 6 Apr 2015

# Versioning: A “Solved Problem,” right?!

**Kurtis R.G.**  
@kraintboltgreene

@auroraeosrose Yeah but usually it's  
<ohgodnot1>.<lol>.<whatever>

FAVORITE  
1

1:00 AM - 13 Apr 2015

# Versioning: A “Solved Problem,” right?!

 **Bryan HorstmannAllen**  
@bdha

Follow

OpenSSL release today was broken. Instead of bumping the version they're refreshing the tarballs to include missing files. OMFG OPENSSL WTF

---

RETWEETS LIKES  
**266** **150**



11:28 AM - 3 Dec 2015

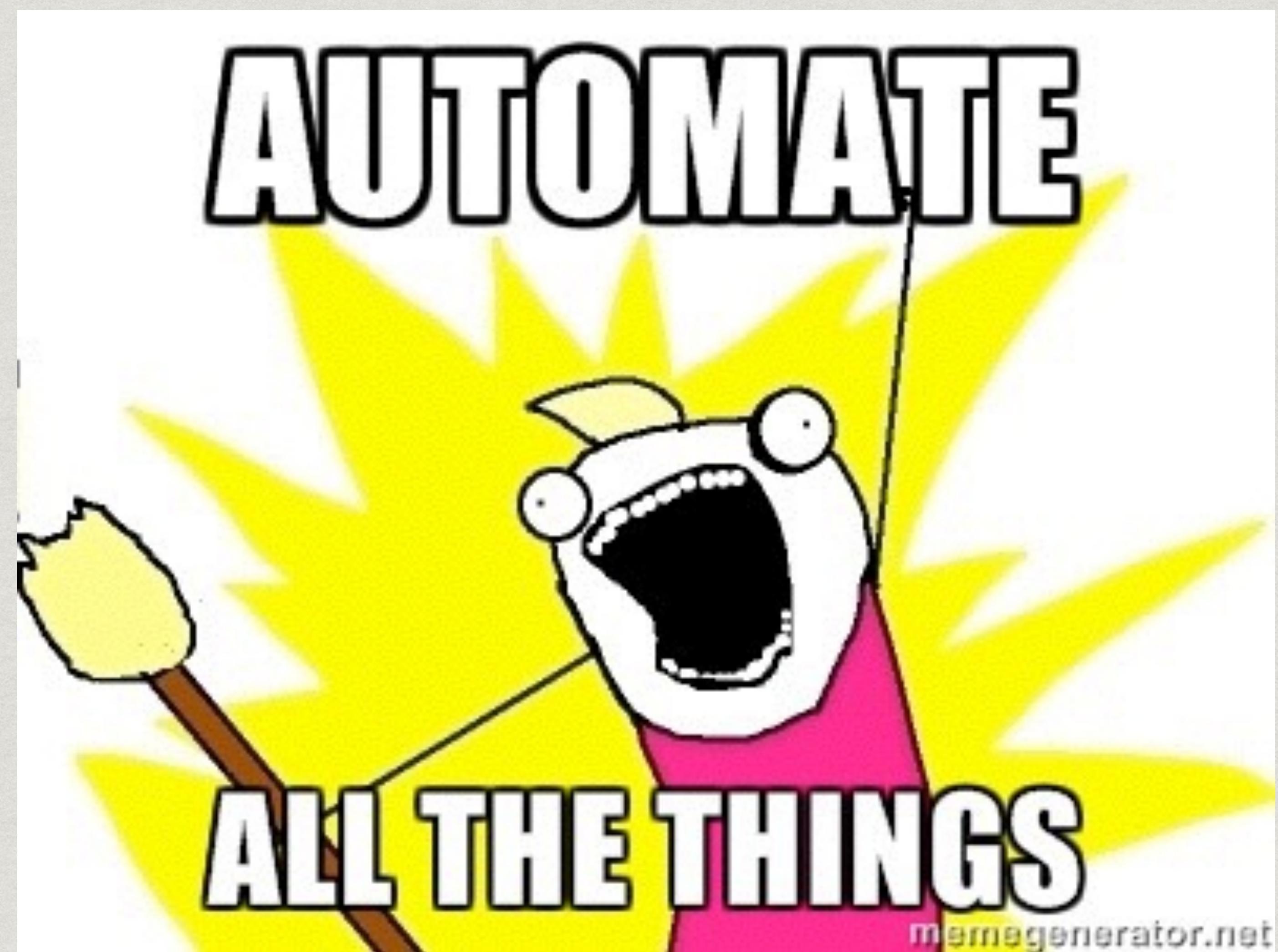
• • •

# Other Human Factors Yellow Flags

---

- Opaque stages of the Continuous Delivery Pipeline
- Direct subversion of the Continuous Delivery pipeline (and its semantics)

# Our Human Factors Elephant



# Learning from Aviation

---

- Air Inter Flight 148





@jpaulreed

#DOES17



@jpaulreed

#DOES17



# Learning from Aviation

---

- Air Inter Flight 148
- Air France Flight 447

# Air France Flight 447 and the Safety Paradox of Automated Cockpits

By *99% Invisible*



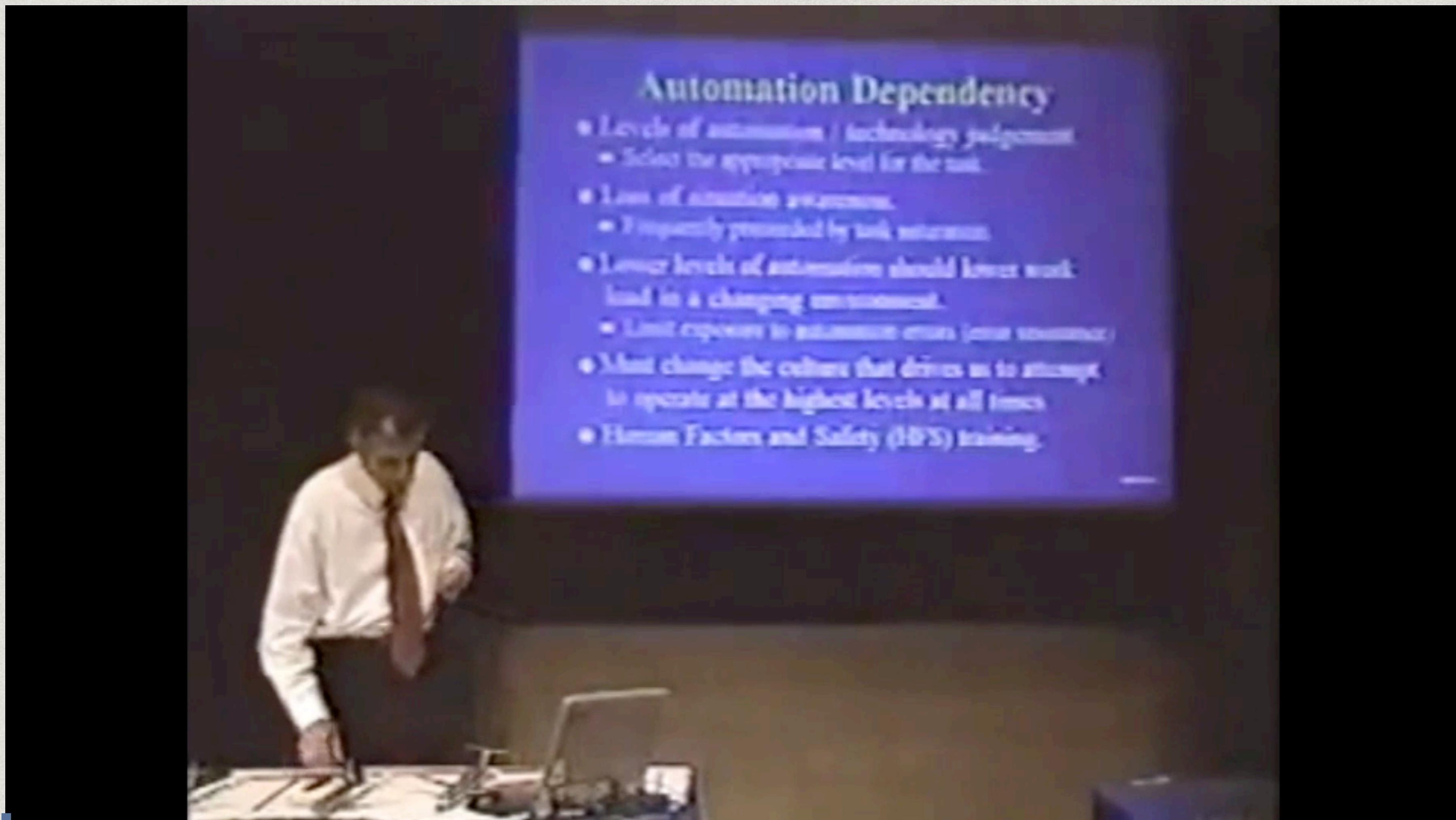
# Learning from Aviation

---

- Air Inter Flight 148
- Air France Flight 447
- Air France Flight 296, aka “Habsheim”



# “Children of the Magenta”



# More Resources (HERE AT DOES!)



**Dr. Sidney Dekker**

The Human Factor: Inspiring the Pursuit of Success and Averting Drift Into Failure

Tuesday, 5:00 pm

@jpaulreed

**John Allspaw**

Coping With Complexity:  
Resilience Engineering Research In DevOps  
Wednesday, 4:25 pm



#DOES17

# TAKEAWAY #1: DELIVERY IS A FOCAL POINT OF OUR SOFTWARE SYSTEMS

# TAKEAWAY #2: DEVOPS IS NOT A SUBSTITUTE FOR RELEASE ENGINEERING

# TAKEAWAY #3: YOUR RELEASE ENGINEERING PRACTICES MUST BE INFORMED BY HUMAN FACTORS



J. Paul Pumpkin Spice  
@jpaulreed

“Release engineering doesn’t matter anymore.”

Me:



J. Paul Reed  
[preed@release-approaches.com](mailto:preed@release-approaches.com)  
[@jpaulreed](https://jpaulreed.com)  
<http://jpaulreed.com>

Release Engineering  
Approaches  
Simply Ship. Every Time.

<http://release-approaches.com>

Anonymous Feedback  
<http://sayat.me/jpaulreed>