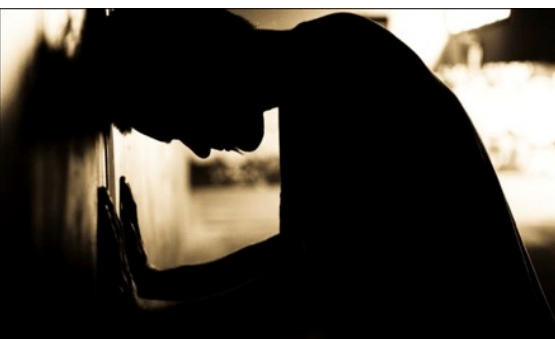


Top Enterprise DevOps Adoption Patterns:

My Eighteen Year Journey Studying High Performing IT Organizations

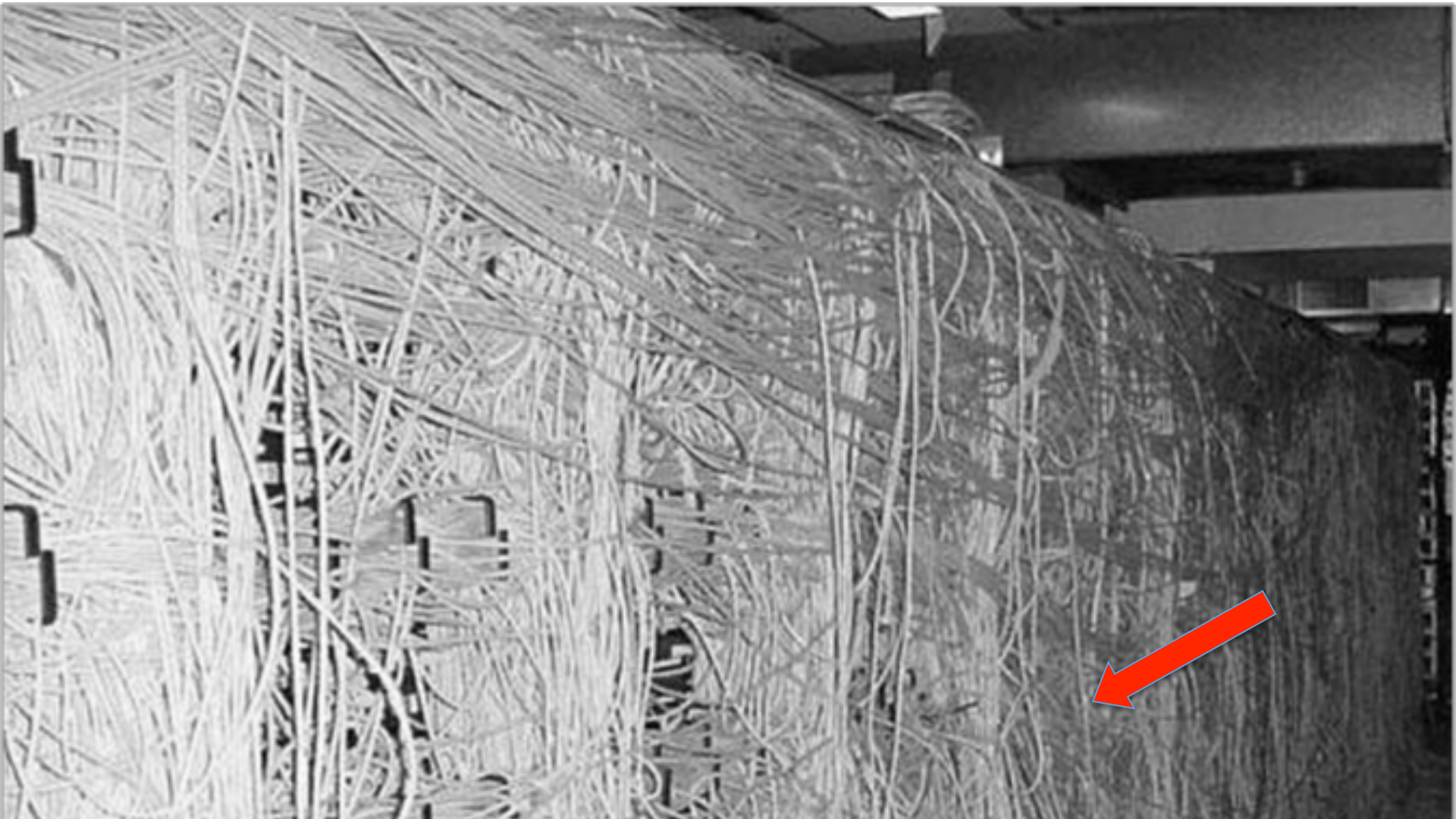
Gene Kim



The Downward Spiral...











IT Operations





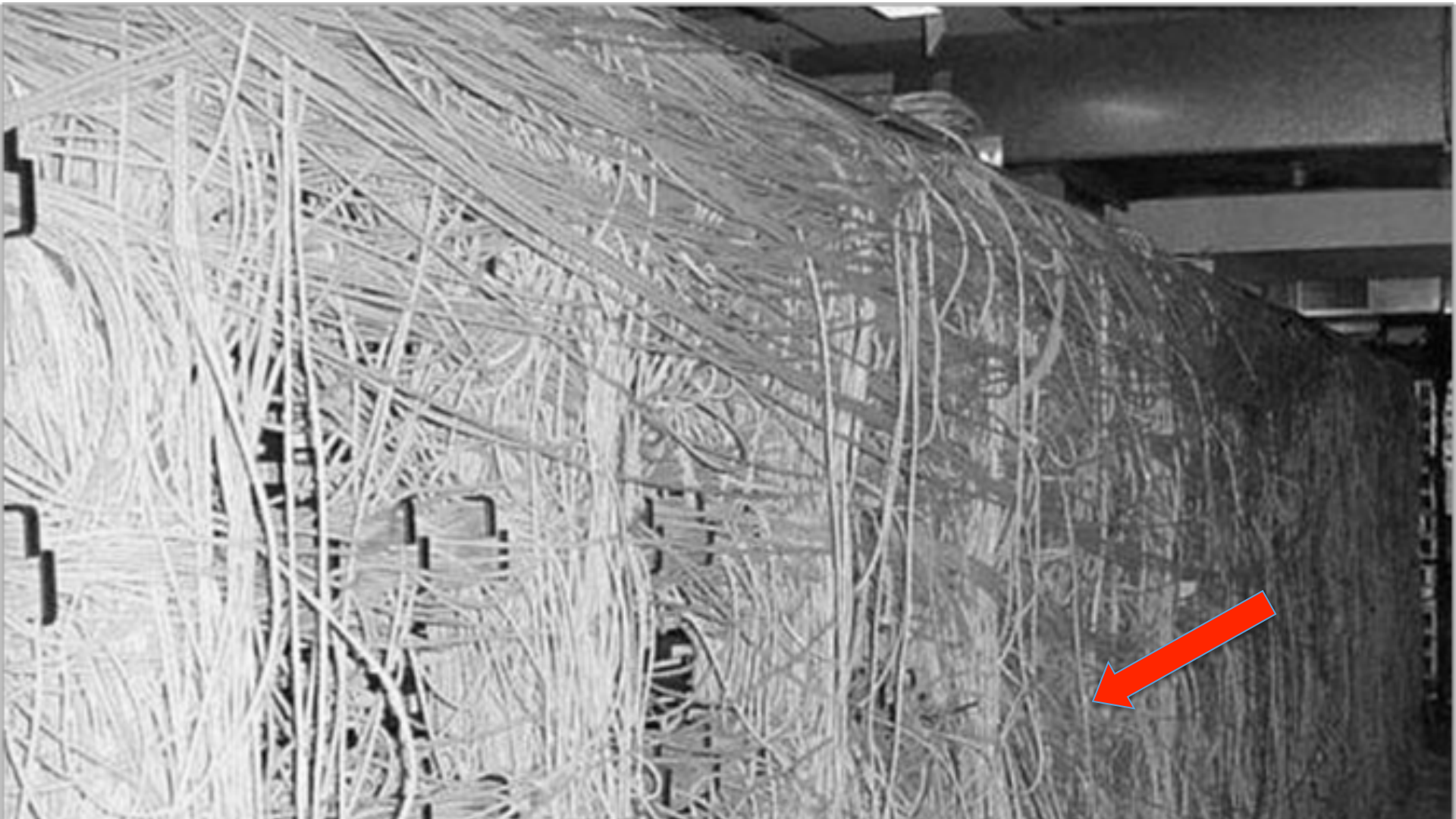
The Product Managers



The Developers









BIG COPS. SMALL TOWN. MODERATE VIOLENCE.

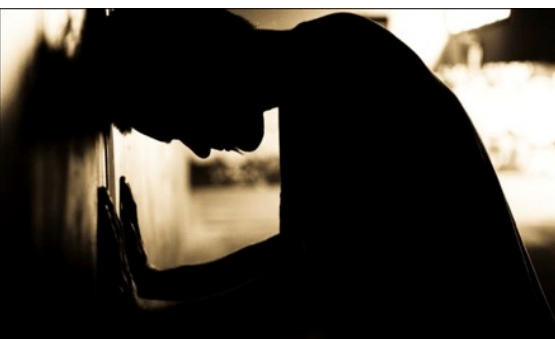
SHANNY PEEVE MEX FROST

HOT FUZZ

FROM THE DIRECTOR OF "STRAIN OF THE HEART"







The Downward Spiral...



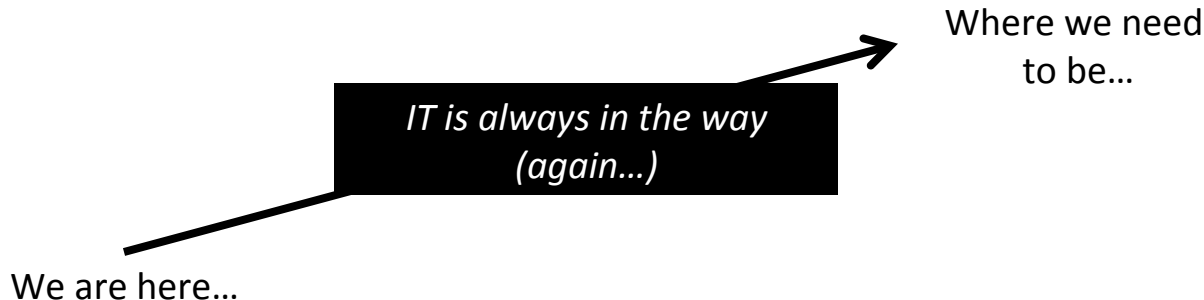
The IT Core Chronic Conflict

- Every IT organization is pressured to simultaneously:
 - Respond more quickly to urgent business needs
 - Provide stable, secure and predictable IT service

Source: The authors acknowledge Dr. Eliyahu Goldratt, creator of the Theory of Constraints and author of The Goal, has written extensively on the theory and practice of identifying and resolving core, chronic conflicts.

Every Company Is An IT Company...

- 95% of all capital projects have an IT component...
- 50% of all capital spending is technology-related



There Is A Better Way...

**Google, Amazon, Netflix,
Spotify, Etsy, Spotify, Twitter,
Facebook...**

**Walmart, Verizon, Raytheon,
Target, Nordstrom,
U.S. Dept of Homeland
Security...**

10 deploys per day
Dev & ops cooperation at Flickr

John Allspaw & Paul Hammond
Velocity 2009





**Little bit weird
Sits closer to the boss
Thinks too hard**



**Pulls levers & turns knobs
Easily excited
Yells a lot in emergencies**



Ops who think like devs
Devs who think like ops



Dev and Ops

**DevOps
is incomplete,
is interpreted wrong,
and is too isolated**

. *Ops

^(?<dept>.+)Ops\$

Amazon 2015: 136K deploys/day

AWS Summit

**Thousands of teams +
Microservices architecture +
Multiple environments +
Continuous delivery**

= 50 million deployments a year



Making Changes When It Matters Most

“By installing a rampant innovation culture, we performed 165 experiments in the peak three months of tax season.”

“Our business result? Conversion rate of the website is up 50 percent. Employee result? Everyone loves it, because now their ideas can make it to market.”

—Scott Cook, Intuit Founder

Who Is Doing DevOps?

- Google, Amazon, Netflix, Etsy, Spotify, Twitter, Facebook ...
- Microsoft, IBM, CA, SAP, HP, Red Hat, ...
- GE Capital, Capital One, Nationwide, BNP Paribas, Amex, BNY Mellon, MasterCard, Paychex, Intuit, ...
- The Gap, Nordstrom, Macy's, Williams-Sonoma, Target ...
- General Motors, Raytheon, LEGO, Bosch ...
- UK Government, US Department of Homeland Security ...
- Kansas State University...

Who else?

High Performers Are More Agile

200x

more frequent
deployments

2,555x

faster lead times
than their peers

High Performers Are More Reliable

3x

lower change
failure rate

24x

faster mean time
to recover (MTTR)

High Performers Are More Secure And Controlled

2x

less time spent
remediating
security issues

29%

more time spent
on new work

High Performers Win In The Marketplace

2x

more likely to
exceed profitability,
market share &
productivity goals

50%

higher market
capitalization growth
over 3 years*

High Performers Win In The Marketplace

2.2x

higher employee
Net Promoter Score

Continuous Delivery: From dinosaur to spaceship in 2 years

Darren Hague / SAP Global IT
February 5, 2013

Public



Recap: Impact of Continuous Delivery in SAP Global IT

- **Before:** Production releases ~monthly
- **Now:** Production release ~twice a week
- **Before:** Pre-release QA cycle 1-2 weeks
- **Now:** QA cycle < 1 day
- **Before:** Error in Prod? Shitstorm & late night
- **Now:** Switch to Blue in <1 minute, fix next day
- **Before:** Project idea to go-live in 6-12 months
- **Now:** New project can be in Production in 1 week
- **Before:** Business stakeholders frustrated
- **Now:** Business stakeholders happy

Technology supports all this, but the team still has to deliver working code.



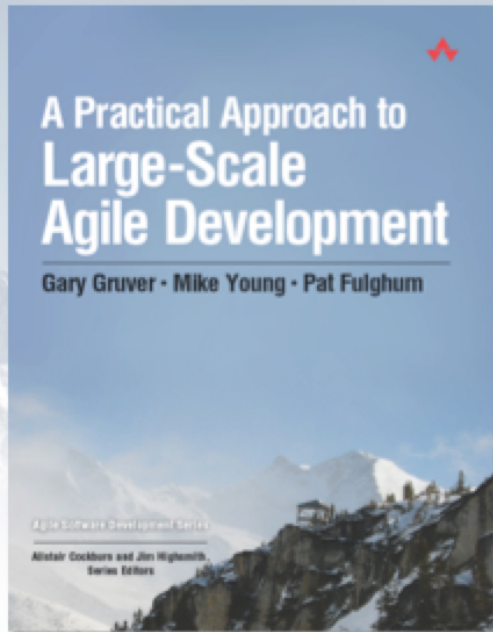
Result: Cloud & DevOps culture throughout SAP

As of late 2015:

- **100s of internal apps and external cloud services running on the Monsoon platform:**
SAP Anywhere, SAP Business One, Multiposting, SuccessFactors, Afaria MobileSecure, SeeWhy, Ariba TradeWorld, Hybris, FieldGlass, SAP Cloud for Customer, SAP Cloud Identity, ...
- **1000s of developers using the Monsoon platform**
- **10000s of VMs**
- **10000s of storage volumes**
- **Deployed in 6 regions & 12 availability zones**



HP LaserJet Firmware

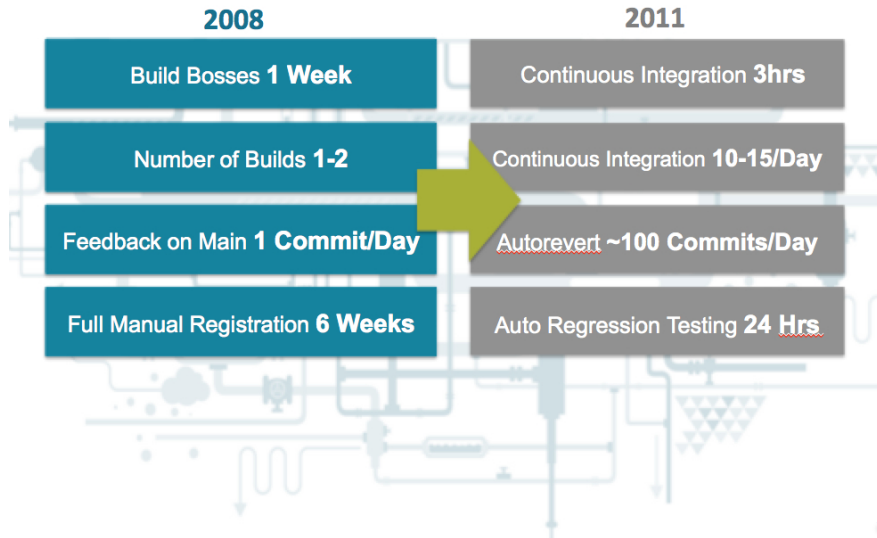


Gary Gruver:

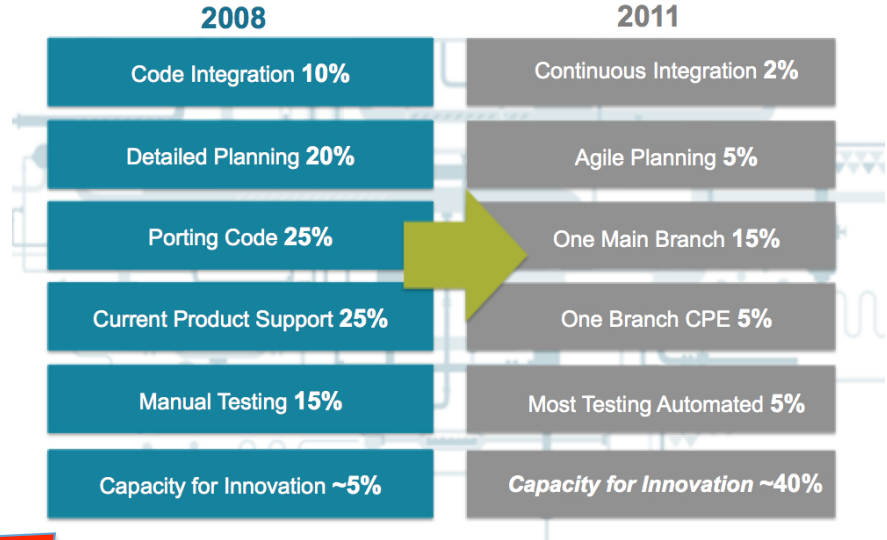
- 24 year veteran in high tech
- Led HP's 400+ developers on a journey from waterfall to agile development
- Currently VP of QE, Release, and Operations at Macys.com

HP LaserJet Firmware CI/CD Pipeline

Cycletime Driver Improvements



Development Cost Driver Improvements



Now includes
Eli Goldratt's
Personal Story
"My Saga"

Eliyahu M. Goldratt and Jeff Cox

THE GOAL

A PROCESS OF ONGOING IMPROVEMENT

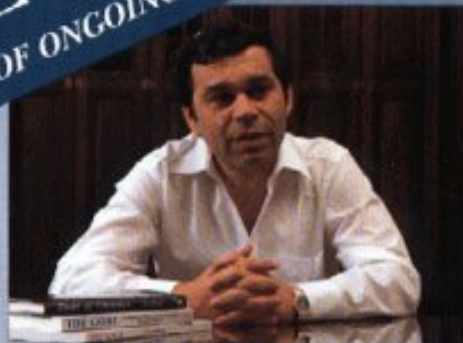
Eli Goldratt has been described by Fortune as a "guru to industry" and by Business Week as a "genius". His book, *The Goal*, is a gripping fast-paced business novel.

"Goal readers are now doing the best work of their lives."

Success Magazine

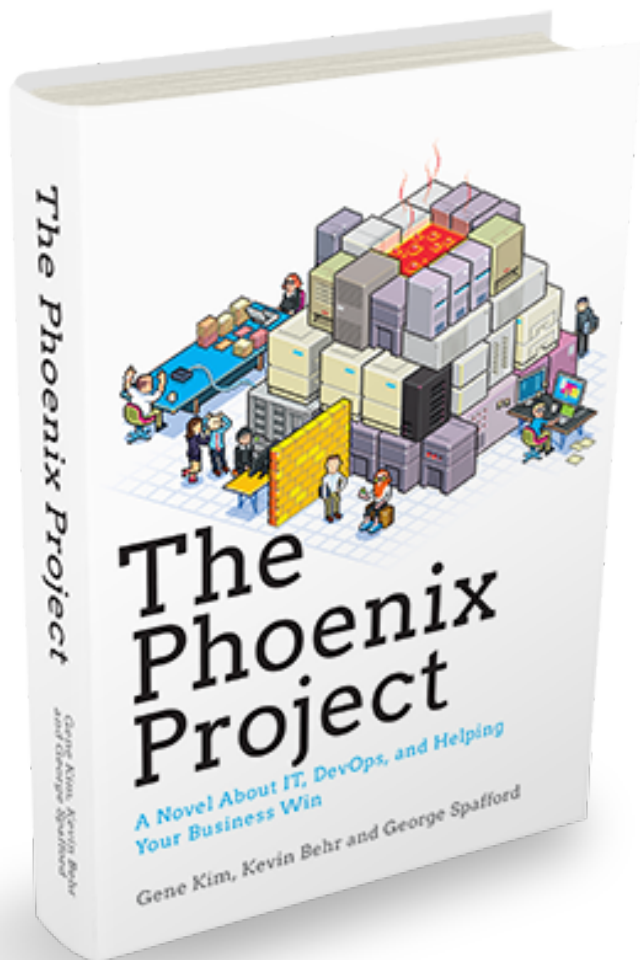
"A factory may be an unlikely setting for a novel, but the book has been wildly effective..."

Tom Peters



THE BEST-SELLING BUSINESS
NOVEL THAT INTRODUCED THE
**THEORY OF
CONSTRAINTS**
AND CHANGED HOW
AMERICA DOES BUSINESS.

**OVER 2 MILLION
COPIES SOLD!**
SECOND REVISED EDITION



“This book will have a profound effect on IT, just as *The Goal* did for manufacturing.”

—**Jez Humble,**
co-author *Continuous Delivery*

“This is the IT swamp draining manual for anyone who is neck deep in alligators.”

—**Adrian Cockcroft,**
Cloud Architect at Netflix

“This is *The Goal* for our decade, and is for any IT professional who wants their life back.”

—**Charles Betz,** IT architect, author
“*Architecture and Patterns for IT*”

The First Way: Flow



**“deploys per day”
vs.
“lead time”**

**“What is your lead time
for changes?”**

**“How long does it take to go from
*code committed to code successfully
running in production?*”**

Design and Development (“the Product”)	Testing and Operations (“the Process”)
Create new products and services that improve the lives of customers by using hypothesis-driven development and fast feedback from users.	Enable fast flow from Development to Production and reliable service to customers by standardizing work, reducing variability and batch sizes.
Feature design and implementation may require work that has never been performed before. Estimates are highly uncertain. Outcomes are highly variable.	Integration, Test and Deployment operations must be performed frequently (i.e., whenever an engineer checks in code into version control, multiple times per day) and as quickly as possible. Cycle times should be well-known and predictable. Outcomes should have low variability.

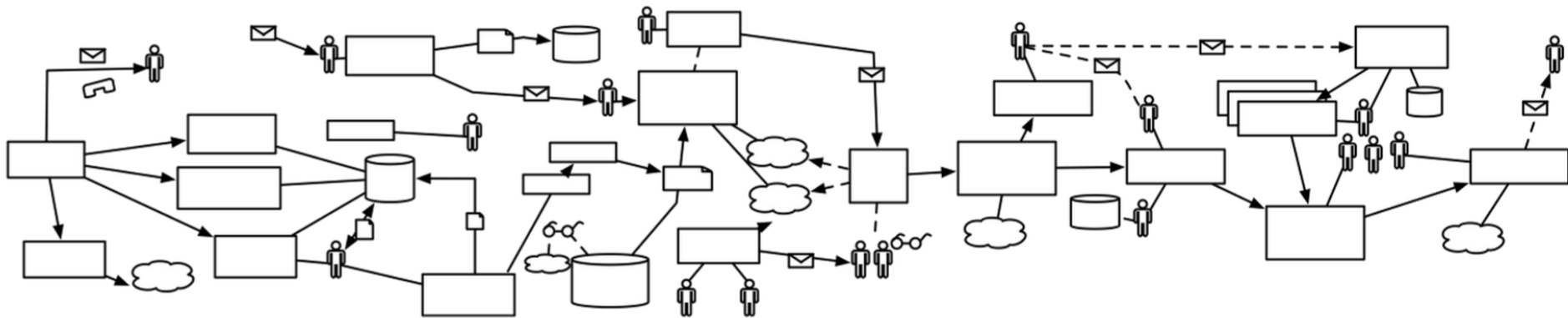
Design and Development (“the Product”)	Testing and Operations (“the Process”)
Create new products and services that improve the lives of customers by using hypothesis-driven development and fast feedback from users.	Enable fast flow from Development to Production and reliable service to customers by standardizing work, reducing variability and batch sizes.
Feature design and implementation may require work that has never been performed before. Estimates are highly uncertain. Outcomes are highly variable.	Integration, Test and Deployment operations must be performed frequently (i.e., whenever an engineer checks in code into version control, multiple times per day) and as quickly as possible. Cycle times should be well-known and predictable. Outcomes should have low variability.

Design and Development (“the Product”)	Testing and Operations (“the Process”)
Create new products and services that improve the lives of customers by using hypothesis-driven development and fast feedback from users.	Enable fast flow from Development to Production and reliable service to customers by standardizing work, reducing variability and batch sizes.
Feature design and implementation may require work that has never been performed before. Estimates are highly uncertain. Outcomes are highly variable.	Integration, Test and Deployment operations must be performed frequently (i.e., whenever an engineer checks in code into version control, multiple times per day) and as quickly as possible. Cycle times should be well-known and predictable. Outcomes should have low variability.

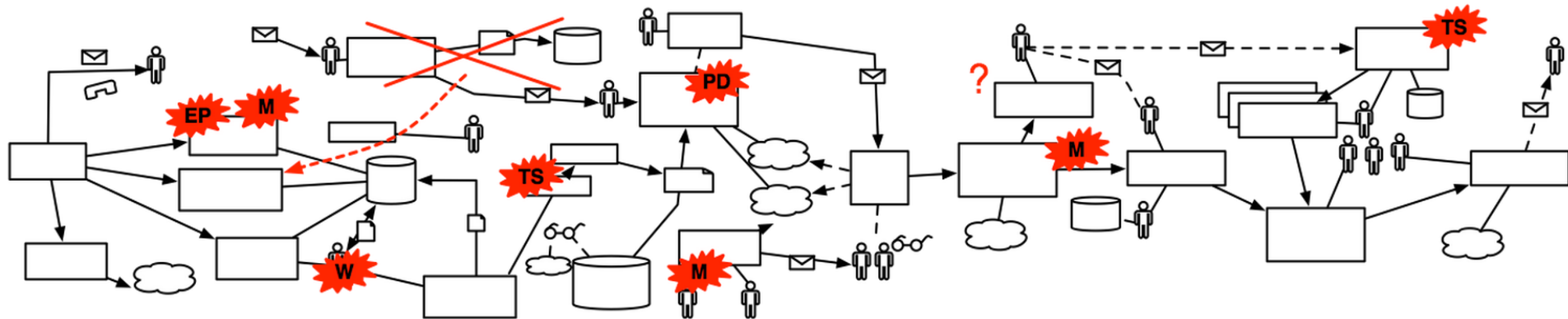
“The DevOps journey began when Nordstrom stopped optimizing for cost, and instead started optimizing for delivery speed.”

– Courtney Kissler
VP eCommerce & Store Technologies
Nordstrom, Inc.

Lead Time = 9 months



Lead Time = 9 months





IT'S A TRAP



Create One Step Environment Creation Process

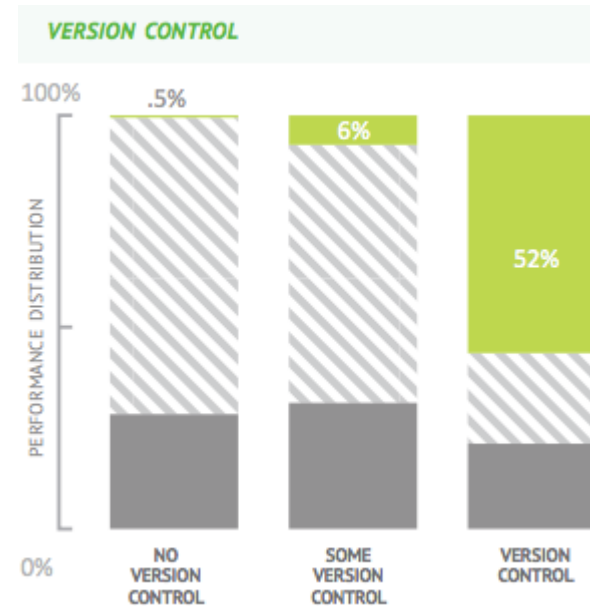
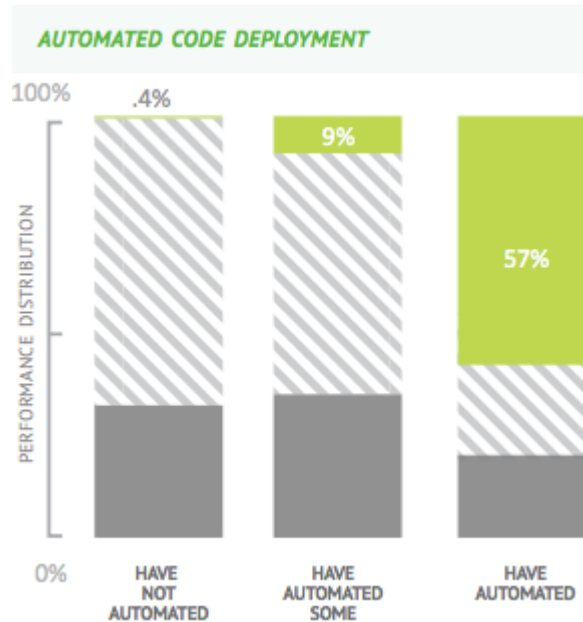
- Make environments available early in the Development process
- Make sure Dev builds the code and environment at the same time
- Create a common Dev, QA and Production environment creation process

**If I had a magic wand,
I'd change the Agile sprints and
definition of “done”:**

**“At the end of each sprint, we must
have working and shippable code...
demonstrated in an environment
that resembles production.”**

How organizations achieve high performance

- 89% are using infrastructure version control
- 82% are using automated code deployments



Deploy Smaller Changes, More Frequently *

facebook [Sign Up](#)

Facebook Chat

By Eugene Letuchy on Tuesday, May 13, 2008 at 10:56pm

One of the things I like most about working at Facebook is the ability to launch products that are (almost) immediately used by millions of people. Unlike a three-guys-in-a-garage startup, we don't have the luxury of scaling out infrastructure to keep pace with user growth; when your feature's userbase will go from 0 to 70 million practically overnight, scalability has to be baked in from the start. The project I'm currently working on, Facebook Chat, offered a nice set of software engineering challenges:

Real-time presence notification:

The most resource-intensive operation performed in a chat system is not sending messages. It is rather keeping each online user aware of the online-idle-offline states of their friends, so that conversations can begin.

 **Facebook Engineering**

Notes by Facebook Engineering
[All Notes](#)

[Get Notes via RSS](#)
[Embed Post](#)

Deploy Smaller Changes, More Frequently *

- Decouple feature releases from code deployments
- Deploy features in a disabled state, using feature flags
- Require all developers check code into trunk daily (at least)
- Practice deploying smaller changes, which dramatically reduces risk and improves MTTR

“As a lifelong Ops practitioner, I know we need DevOps to make our work humane.

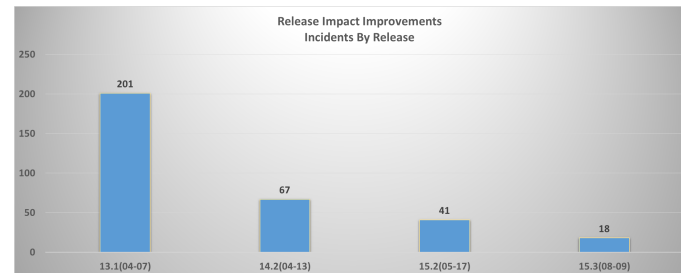
In the past, I’ve worked every holiday, on my birthday, my spouse’s birthday, and even on the day my son was born.”

**Nathan Shimek
Engineering Manager, New Context**



CSG: Reducing Batch Size By 50%

- Apps supporting bill printing and customer care for 50MM customer, 6B transactions per month
- 20 technology platforms, including mainframe VSAM and DB2, Java, desktop client
- Moved from 2 to 4 releases per year
- Shared Operations Team performed daily deployments to UAT

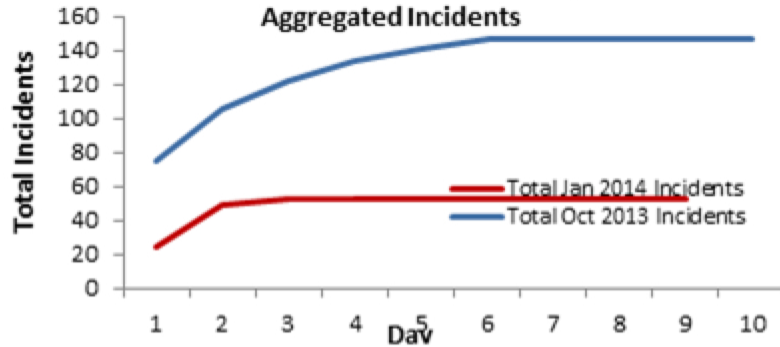


*And the customer got the feature in
half the time!*

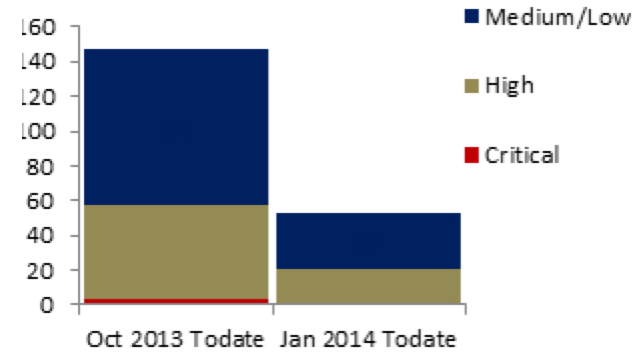
Source: Scott Prugh, Chief Architect, CSG, Inc.

CSG: Reducing Batch Size By 50%

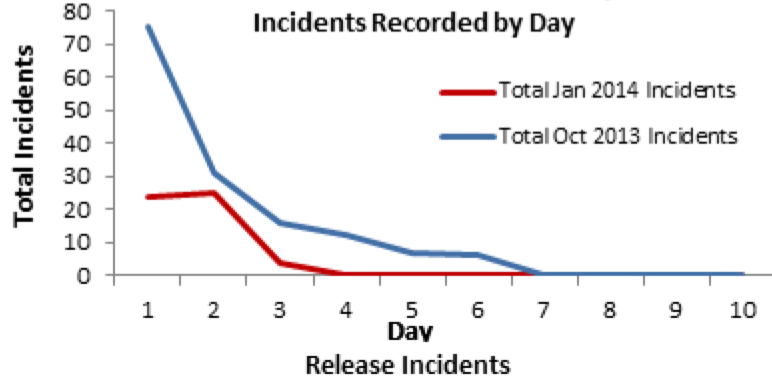
Oct'13 vs Jan '14 Release Comparison



Release Incidents

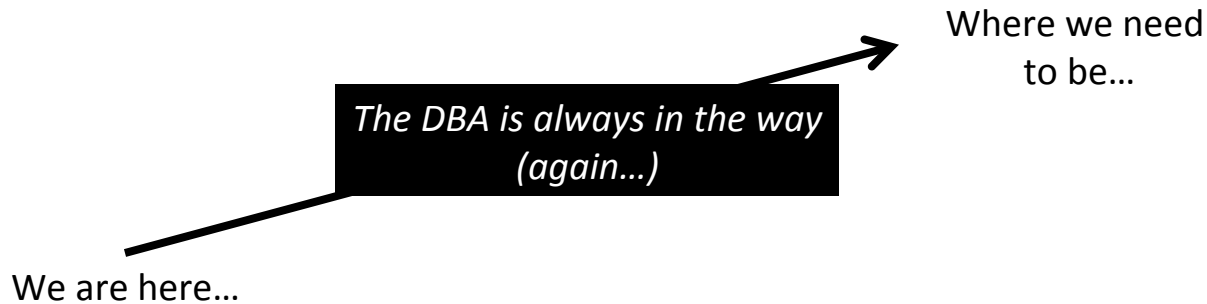


Oct '13 vs. Jan '14 Release Comparison



And the customer got the feature in half the time!

How It Feels To Be A DBA...



Amazon 2010: ~15K deploys/day

Amazon May Deployment Stats (production hosts & environments only)

11.6 seconds

Mean time between deployments (weekday)

1,079

Max # of deployments in a single hour

10,000

Mean # of hosts simultaneously receiving a deployment

30,000

Max # of hosts simultaneously receiving a deployment 

Amazon 2015: 136K deploys/day

AWS Summit

**Thousands of teams +
Microservices architecture +
Multiple environments +
Continuous delivery**

= 50 million deployments a year



This is now...

Etsy Code Deployment

What once required **6-14 hours** and an “Army”

...Now takes **15 minutes** and **1 person**

30+

Deploys
per day
2013

50

Deploys per day
March 2014
QCon London

80-90

Deploys per day
April 2014
Chef Conf

2013 Mike Brittain, Continuous Deployment: The Dirty Details

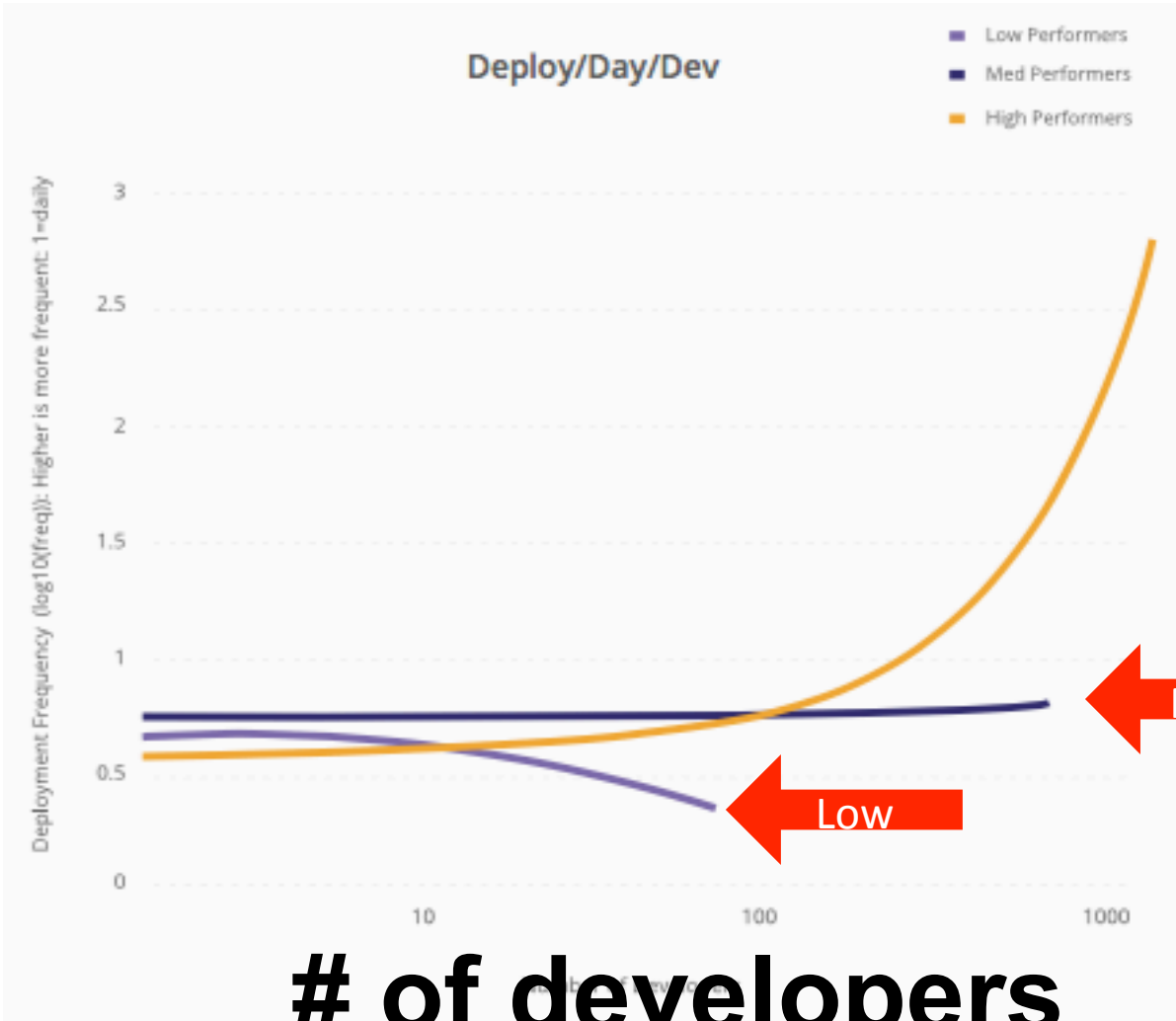
3/2014 Daniel Schauenberg, Qcon London

4/2014 tweet @philkates

“deploys / day”

“deploys / day / dev”

deploys / day



High (linear)

Med

Low

of developers

Breaking The Bottlenecks In The Flow

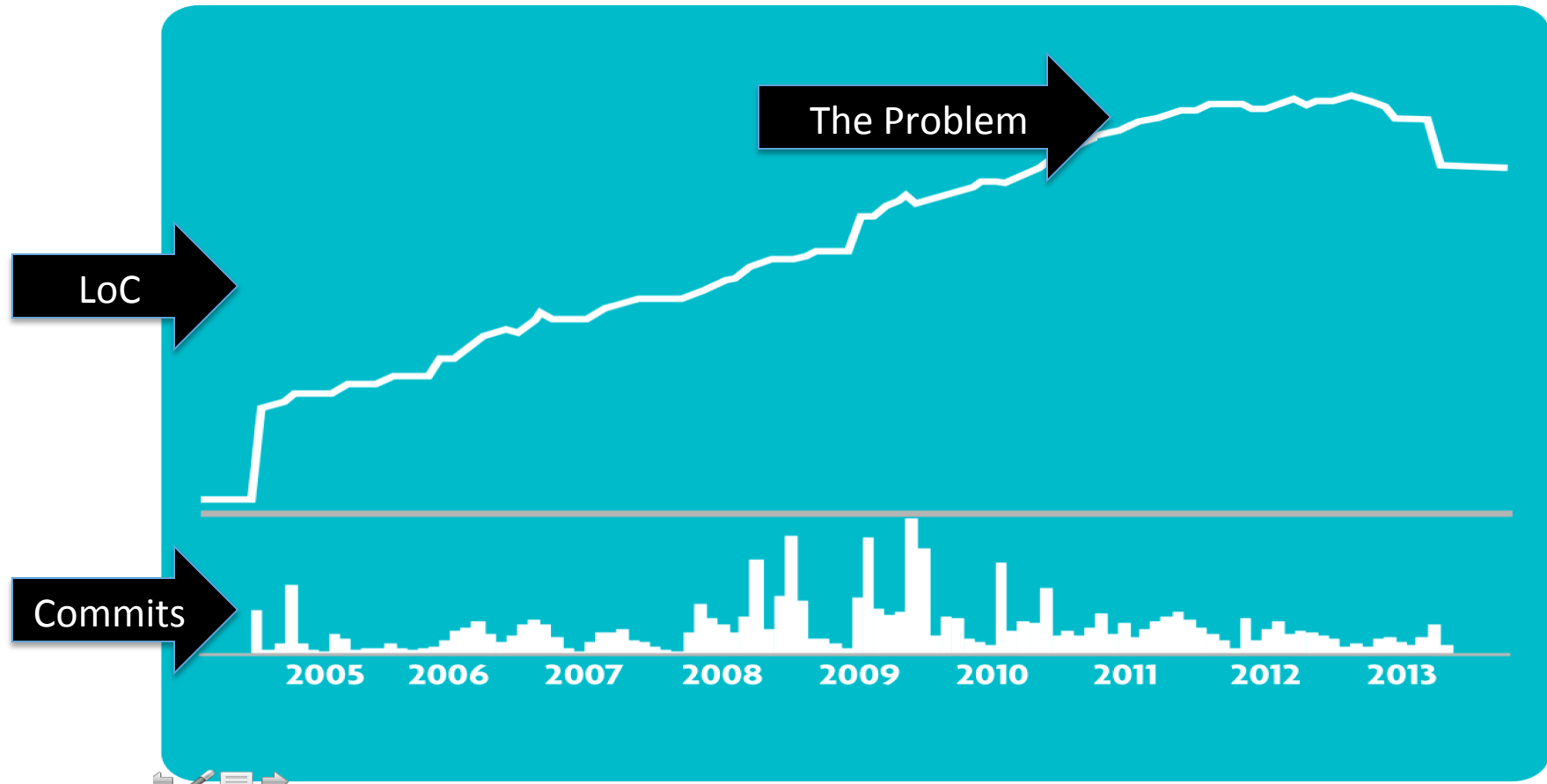
- Environment creation
- Code deployment
- Test setup and run (mention @rohansingh)
- Overly tight architecture
- Development
- Product management

“In November 2011, running even the most minimal test for CloudFoundry required deploying to 45 virtual machines, which took a half hour. This was way too long, and also prevented developers from testing on their own workstations.

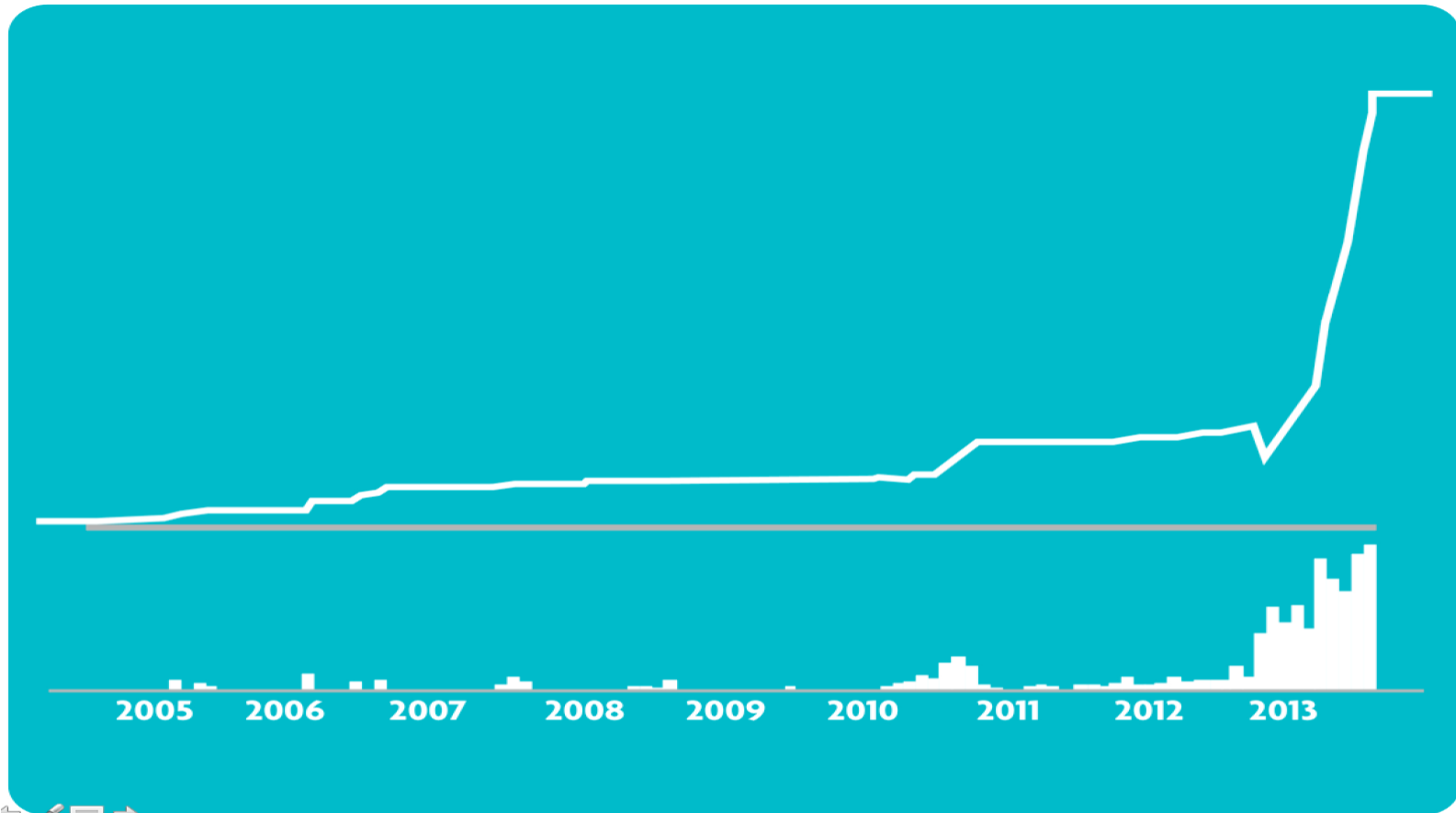
By using containers, within months, we got it down to 18 virtual machines so that any developer can deploy the entire system to single VM in six minutes.”

— Elisabeth Hendrickson, Director of Quality Engineering, Pivotal Labs

Blackboard Learn: 2005-Present



Blackboard Learn Building Blocks



The First Way: Outcomes

- Creating single repository for code and environments
- All Ops artifacts in version control
- Determinism in the release process
- Consistent Dev, Test and Production environments, all properly built before deployment begins
- Developers checking in code daily, being productive
- Automated regression testing
- Features being deployed daily without catastrophic failures
- Decreased lead time
- Faster cycle time and release cadence

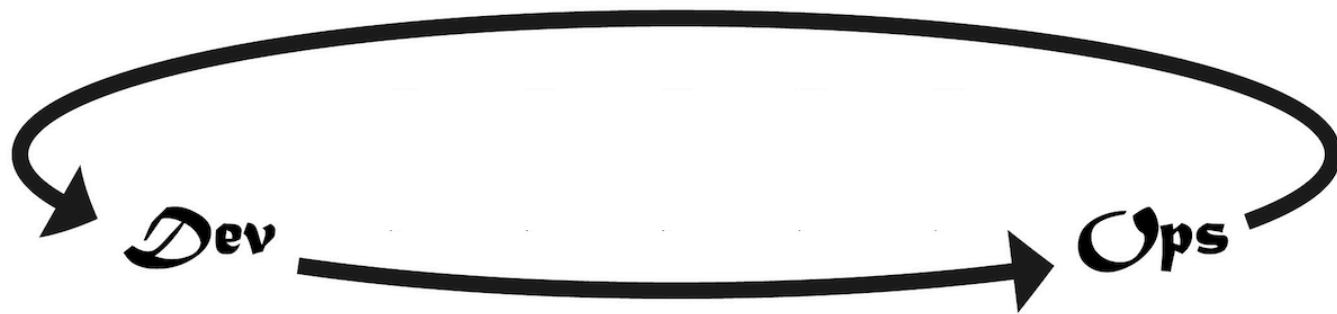
Top Predictors Of IT Performance (2014)

- Version control of all production artifacts
- Continuous integration and deployment
- Automated acceptance testing
- Peer-review of production changes (vs. external change approval)
- High trust culture
- Proactive monitoring of the production environment
- Win-win relationship between Dev and Ops

The First Way: Outcomes

- Creating single repository for code and environments
- Determinism in the release process
- Consistent Dev, Test and Production environments, all properly built before deployment begins
- Features being deployed daily without catastrophic failures
- Decreased lead time
- Faster cycle time and release cadence

The Second Way: Feedback





Your Toyota is My Toyota

"If I suspect a problem, I pull the Andon cord and the whole production line stops immediately, and it doesn't start again until the issue has been resolved. I've pulled it thousands of times and it's good because it makes everyone personally responsible for producing the highest quality car."

Bridie Tucker
Team Member, Trim Assembly, Toyota Burnaston, UK

Today
Tomorrow
Toyota

Visit Bridie Tucker at VTIMT.com/bridie

How many times per day is the andon cord pulled in a typical day at a Toyota manufacturing plant?

3,500 times per day

Why would Toyota do something so disruptive as stopping production thousands of times per day?

“It’s the only way we can build 2,000 vehicles per day – that’s one completed vehicle every 55 seconds.”

Google Dev And Ops (2013)

- 15,000 engineers, working on 4,000+ projects
- All code is checked into one source tree (billions of files!)
- 5,500 code commits/day
- 75 million test cases are run daily

*"Automated tests transform fear into boredom."
-- Eran Messeri, Google*

Developers Carry Pagers

“We found that when we woke up developers at 2am, defects got fixed faster than ever”

– Patrick Lightbody,
VP Prod Mgmt, New Relic

“You build it, you run it.”

– Werner Vogels
CTO, Amazon

Developers Carry Pagers

“As a developer, there has never been a more satisfying point in my career than when I wrote the code, I pushed the button to deploy it, I watched the metrics to see if it actually worked in production, and fixed it if it broke.”

– Tim Tischler
Director of Operations Engr,
Nike, Inc.

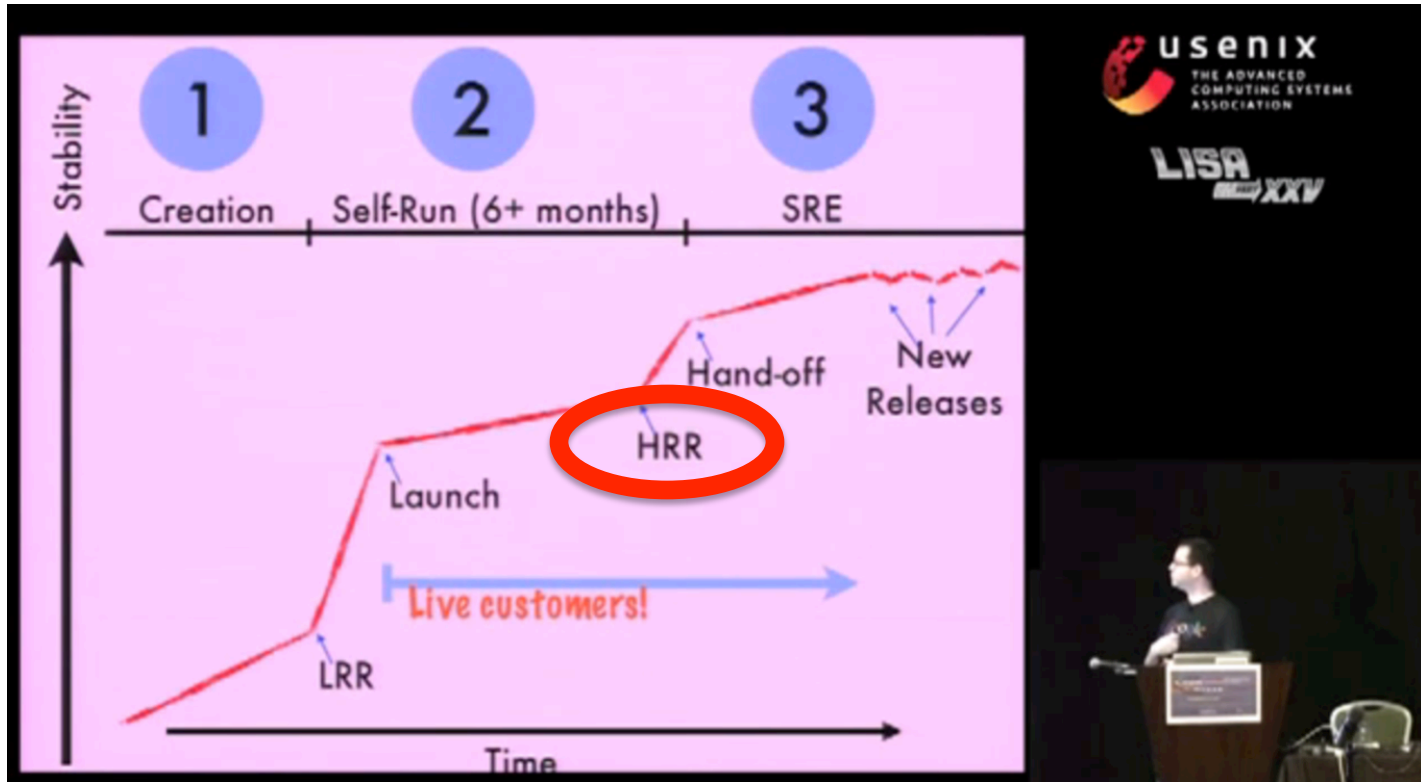
“As a developer, the most satisfying points in my career?”

“It’s when I wrote the code, pushed the button to deploy it, watched the metrics to see if it actually worked in production, and fixed it if it broke.”

Tim Tischler
Director of Operations Engineering
Nike, Inc.



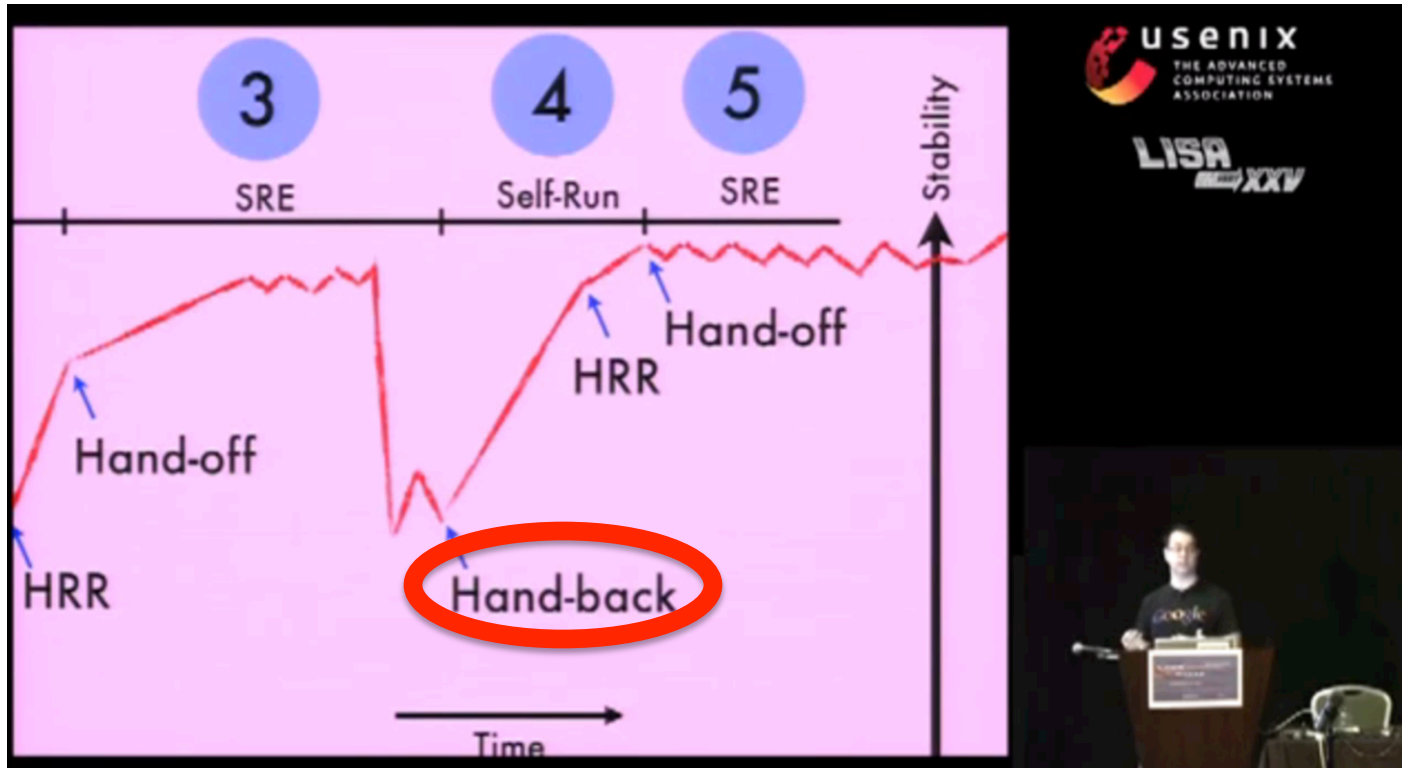
Devs Initially Self-Manage Their Own Code



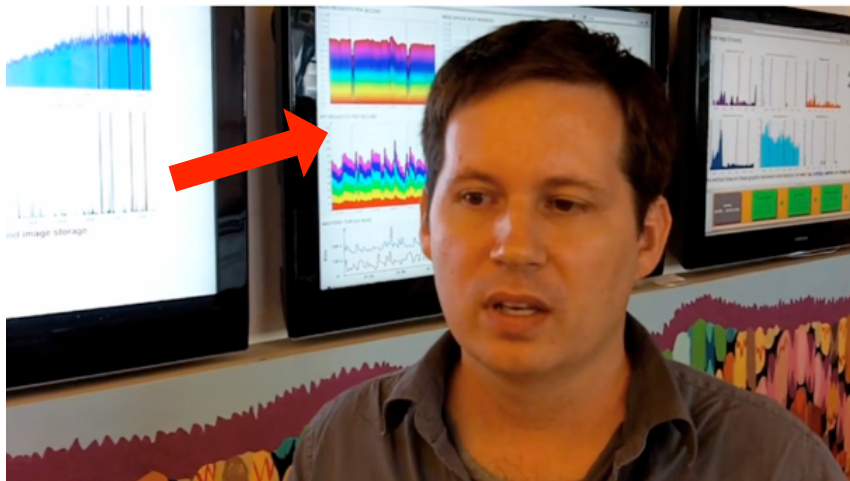
Test Whether Developers Qualify For IT Operations Resources

- Types/frequency of pager alerts
- Maturity of monitoring
- System architecture review
- Release process
- Defect counts and severity
- Production hygiene

Return Fragile Services Back To Dev



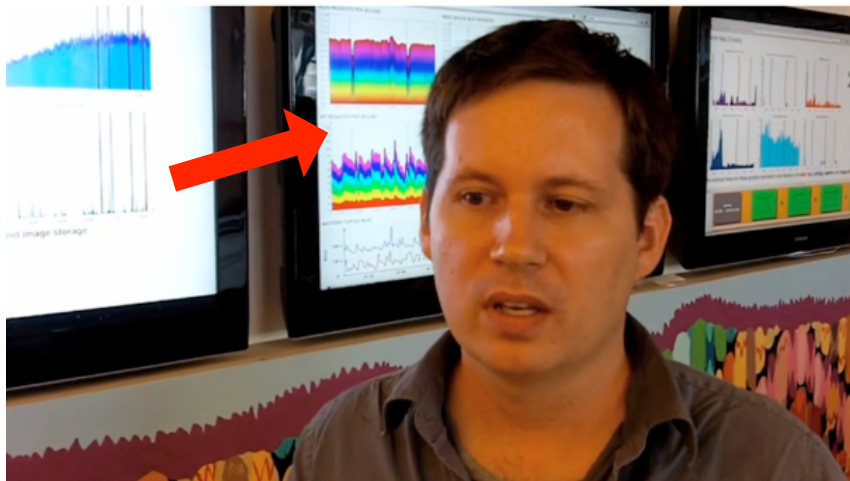
Pervasive Production Telemetry



“Having a developer add a monitoring metric shouldn’t feel like a schema change.”

– John Allspaw,
SVP Tech Ops,
Etsy

Pervasive Production Telemetry



“Having a developer add a monitoring metric shouldn’t feel like a schema change.”

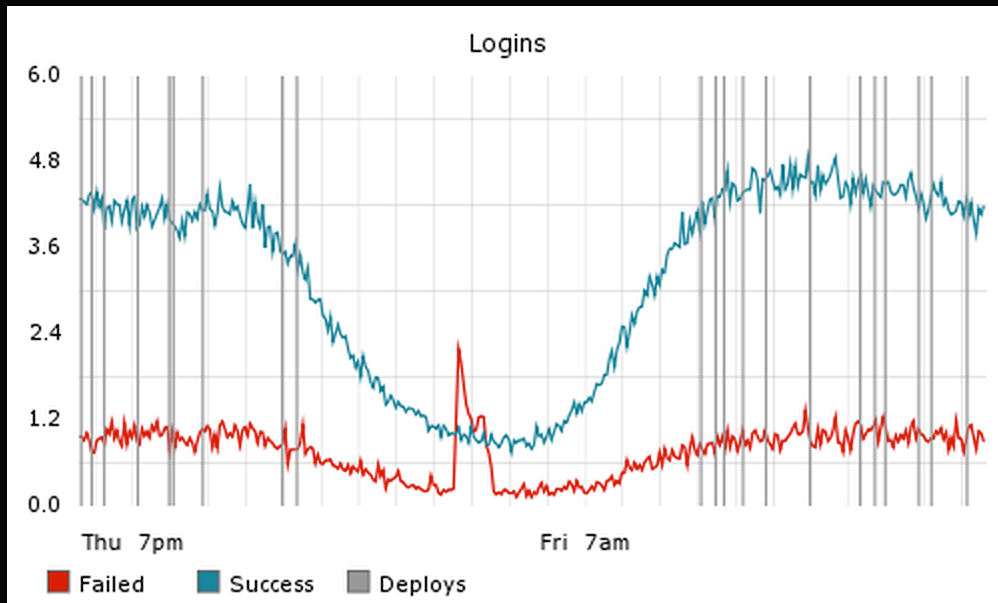
– John Allspaw,
SVP Tech Ops,
Etsy

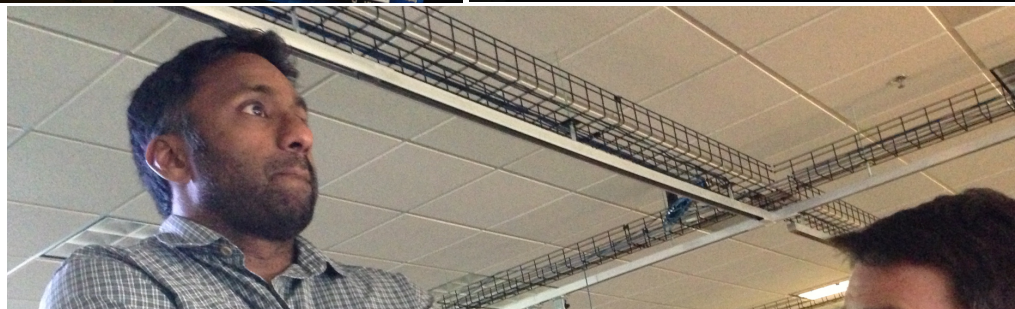
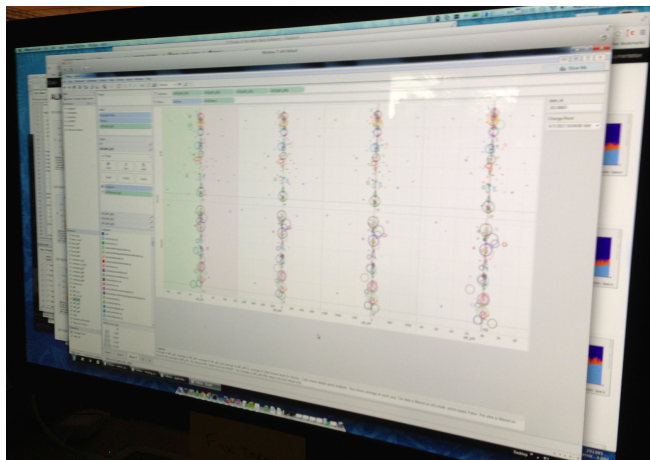
Measure Anything

Here's how we do it using our PHP StatsD library:

```
StatsD::increment("grue.dinners");
```

That's it. That line of code will create a new counter on the fly and increment it every time it's executed. You can then go look at your graph and bask in the awesomeness, or for that matter, spot someone up to no good in the middle of the night:





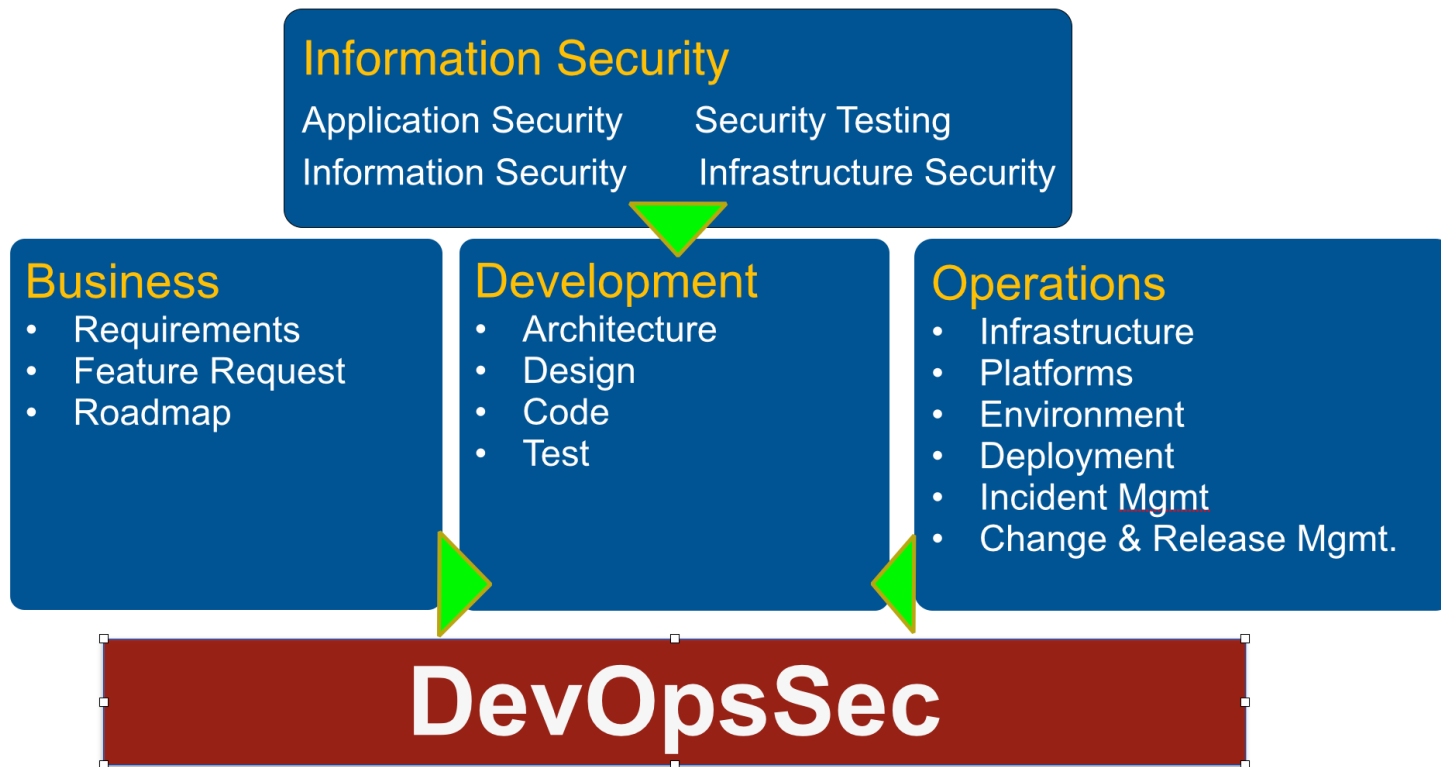
*People actually look at the logs!
(Mention Verizon PCI Data Breach Study)*

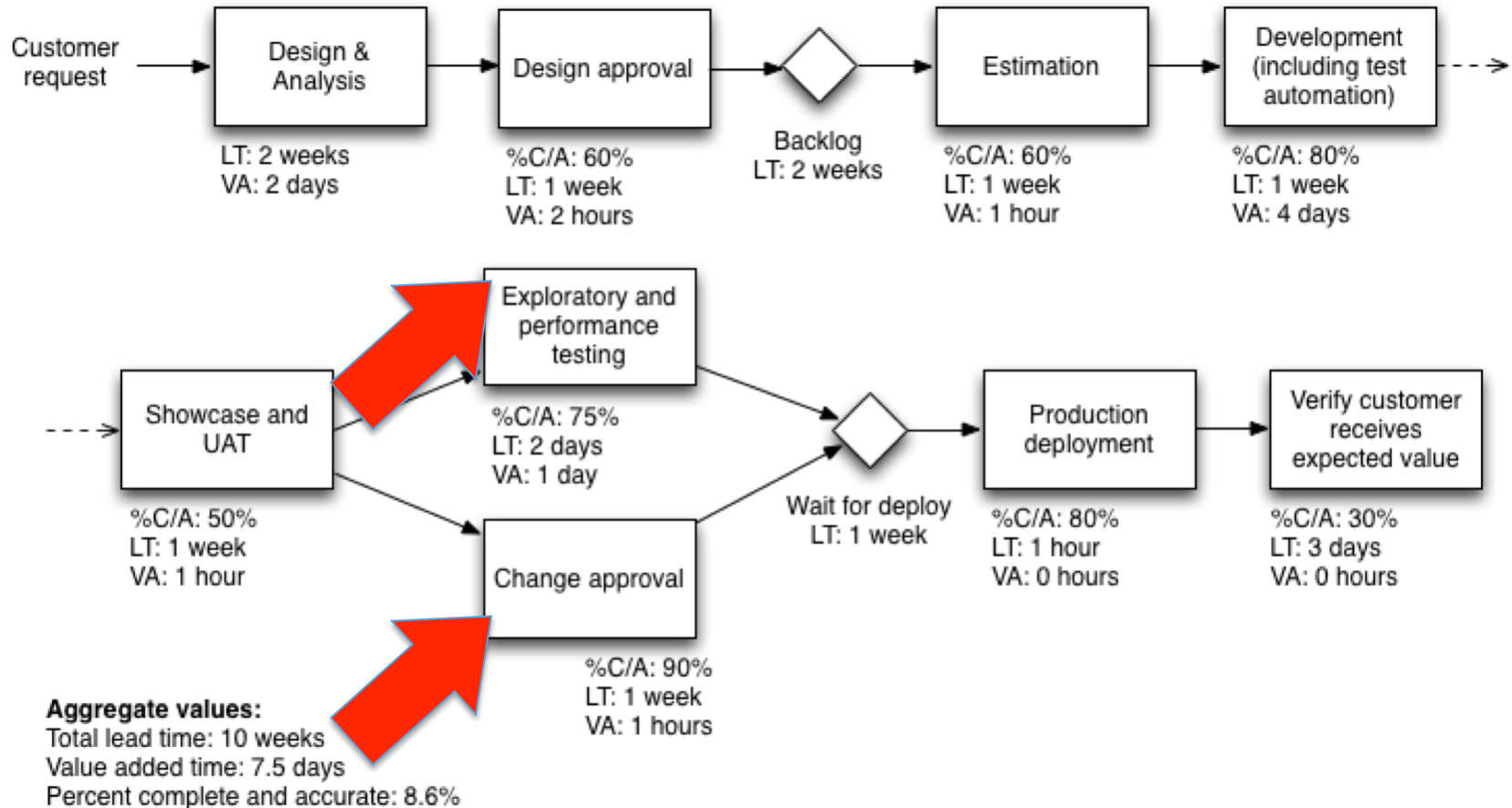


* **Justin Collins, Neil Matatall & Alex Smolen
from Twitter**



Capital One: DevOpsSec





One Of The Highest Predictors Of Performance

Scenario A (Pathological)	Scenario B (Bureaucratic)	Scenario C (Generative)
Information is hidden	Information may be ignored	Information is actively sought
Messengers are “shot”	Messengers are tolerated	Messengers are trained
Responsibilities are shirked	Responsibility is compartmented	Responsibilities are shared
Bridging between teams and functional groups is discouraged	Bridging between teams and functional groups is allowed but discouraged	Bridging between teams and functional groups is rewarded
Failure is covered up	Organization is just and merciful	Failure causes enquiry
New ideas are crushed	New ideas create problems	New ideas are welcomed

One Of The Highest Predictors Of Performance

Scenario A (Pathological)	Scenario B (Bureaucratic)	Scenario C (Generative)
Information is hidden	Information may be ignored	Information is actively sought
Messengers are “shot”	Messengers are tolerated	Messengers are trained
Responsibilities are shirked	Responsibility is compartmented	Responsibilities are shared
Bridging between teams and functional groups is discouraged	Bridging between teams and functional groups is allowed but discouraged	Bridging between teams and functional groups is rewarded
Failure is covered up	Organization is just and merciful	Failure causes enquiry
New ideas are crushed	New ideas create problems	New ideas are welcomed

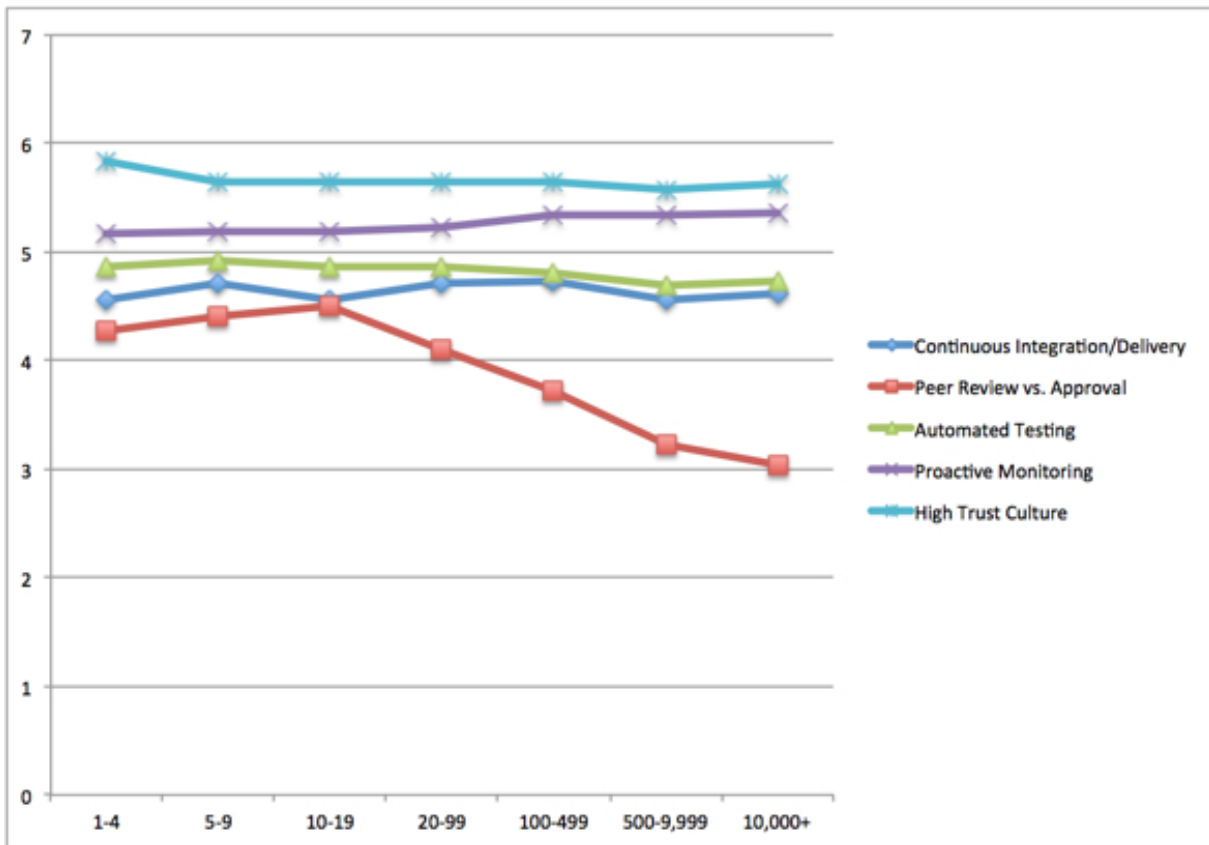
When Countermeasures Backfire

- What is our standard command-and-control countermeasure when the following occurs:
 - Change control failures?
 - Testing failures?

New engineer to John Allspaw:
“Is it okay for me to make this change?”

John Allspaw:
“I don’t know. Is it?”

Can Large Orgs Adopt These Practices?



Yes!
*(Automated testing,
Continuous integration,
proactive monitoring, even
high trust cultures!)*

*The only practice not being
adopted is Peer Review vs.
Change Approval!*

LEGAL/REGULATORY | AUGUST 2, 2012, 9:07 AM | 357 Comments

Knight Capital Says Trading Glitch Cost It \$440 Million

BY NATHANIEL POPPER



Brendan McDermid/Reuters

@RealGeneKim

The Second Way: Outcomes

- Peer review of code and environment changes
- Disciplined automated testing enabling many simultaneous small, agile teams to work productively
- Proactive monitoring of the production environment
- Defects and security issues getting fixed faster than ever
- High trust culture
- All groups communicating and coordinating better
- Everybody is getting more work done

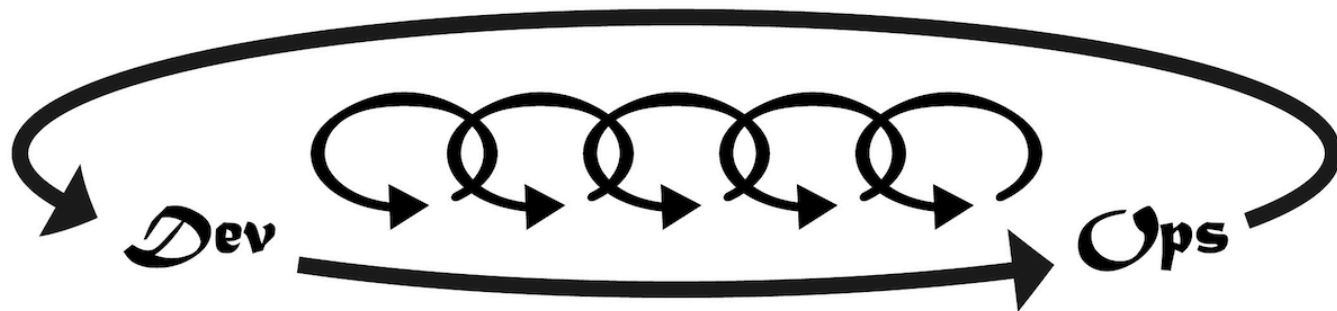
Top Predictors Of IT Performance (2014)

- Version control of all production artifacts
- Continuous integration and deployment
- Automated acceptance testing
- Peer-review of production changes (vs. external change approval)
- High trust culture
- Proactive monitoring of the production environment
- Win-win relationship between Dev and Ops

The Second Way: Outcomes

- Defects and security issues getting fixed faster than ever
- Disciplined automated testing enabling many simultaneous small, agile teams to work productively
- All groups communicating and coordinating better
- Everybody is getting more work done

The Third Way: Continual Experimentation And Learning



Break Things Early And Often

“Do painful things more frequently, so you can make it less painful... We don’t get pushback from Dev, because they know it makes rollouts smoother.”

– Adrian Cockcroft,
Former Architect, Netflix
(Now Technology Fellow,
Battery Ventures)

Amazon EC2 outage downs Reddit, Quora



The sky is falling! Amazon's cloud seems to be down (raining?) so we're experiencing some issues too. Be back soon!

5 hours ago via web

Retweeted by [RealAmandaStone](#) and others



SCVNGR and other sites took to Twitter after a rare and major outage of Amazon's cloud-based Web service.



990 people recommend this.

Inject Failures Often

The Netflix Tech Blog

5 Lessons We've Learned Using AWS

We've sometimes referred to the Netflix software architecture in AWS as our Rambo Architecture. Each system has to be able to succeed, no matter what, even all on its own. We're designing each distributed system to **expect and tolerate failure** from other systems on which it depends.

One of the first systems our engineers built in AWS is called the **Chaos Monkey**. The Chaos Monkey's job is to randomly kill instances and services within our architecture. If we aren't constantly testing our ability to succeed despite failure, then it isn't likely to work when it matters most – in the event of an unexpected outage.

**You Don't Choose Chaos Monkey...
Chaos Monkey Chooses You**



The 2014 AWS Reboot

“When we got the news about the emergency EC2 reboots, our jaws dropped. When we got the list of how many Cassandra nodes would be affected, I felt ill.

“Then I remembered all the Chaos Monkey exercises we’ve gone through. My reaction was, ‘Bring it on!’”

– Christos Kalantzis
Netflix Cloud DB Engineering

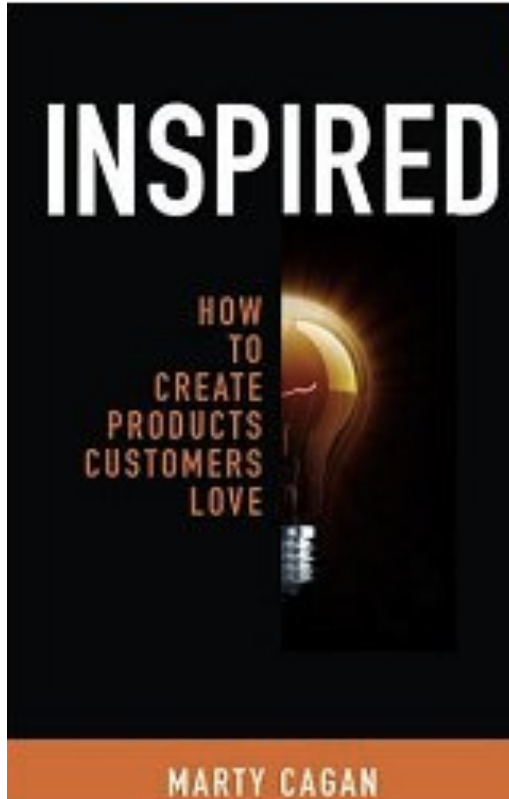
The 2014 AWS Reboot

“Out of our 2700+ production Cassandra nodes, 218 were rebooted. 22 Cassandra nodes did not reboot successfully.

“Netflix customers experienced no downtime that weekend.”

– Bruce Wong
Netflix Chaos Engineering

Allocate 20% Of Cycles To Technical Debt Reduction



[CNET](#) › [News](#) › [E-Business](#)

August 6, 1999 3:50 PM PDT

eBay online again after 14-hour outage

By [Tim Clark](#)
Staff Writer, CNET News



Entreprise Tech

Inside Operation InVersion, the Code Freeze That Saved LinkedIn

By Ashlee Vance  | April 10, 2013



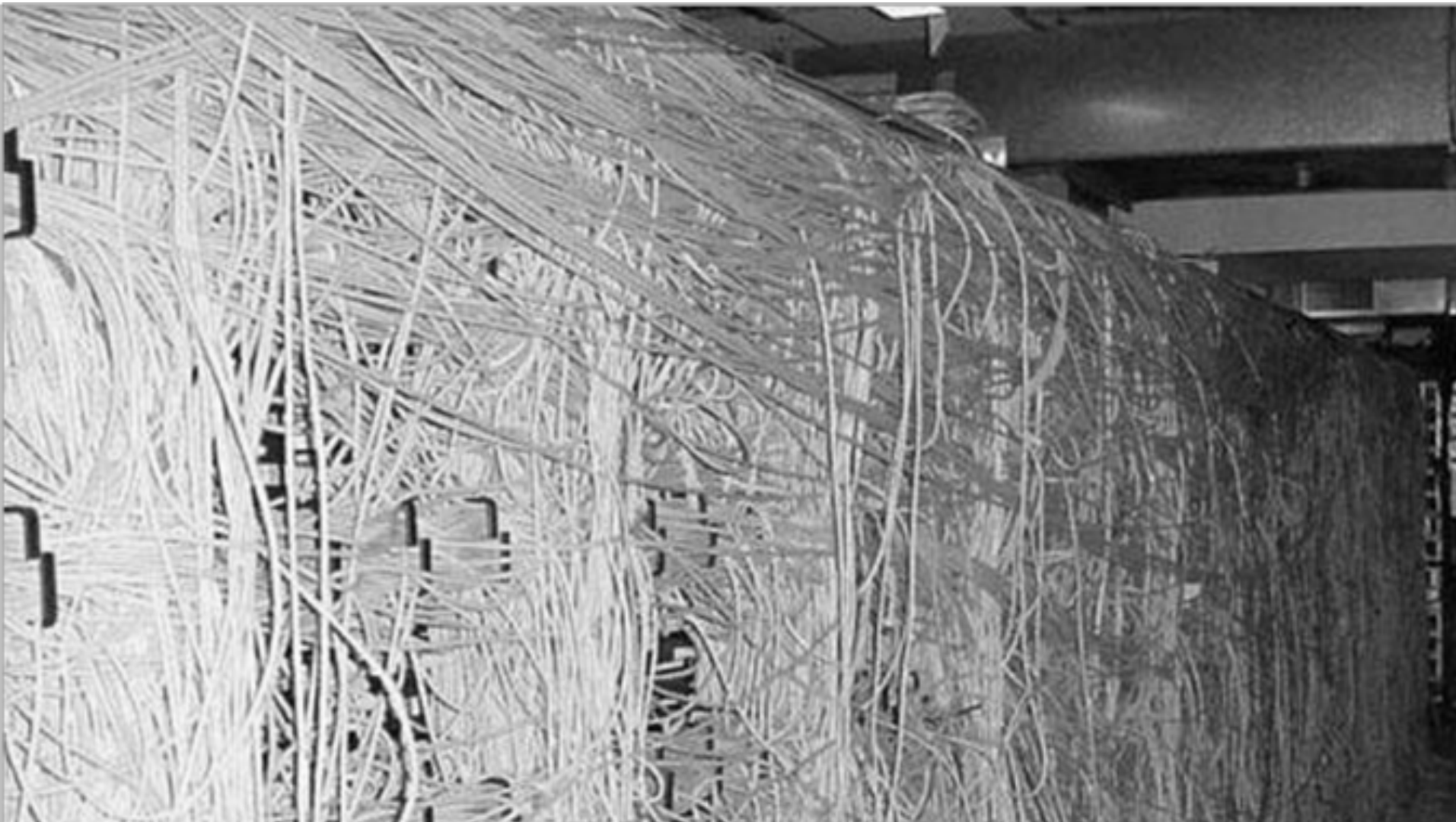
“By November 2011, Kevin Scott, LinkedIn’s top engineer, had had enough. The system was taxed as LinkedIn attracted more users, and engineers were burnt out.

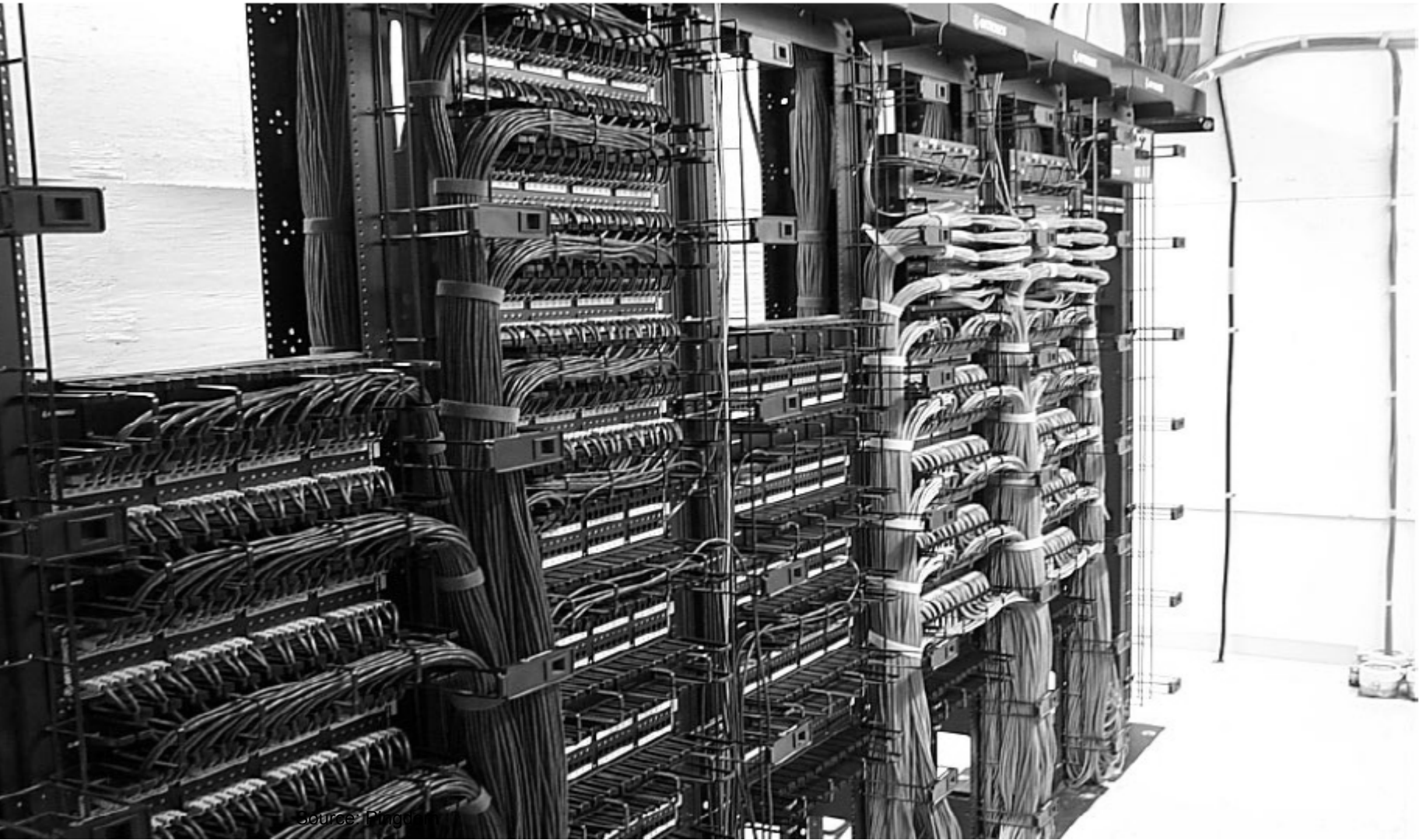
“To fix the problems, Scott, who’d arrived from Google that February, launched Operation InVersion.

“He froze development on new features so engineers could overhaul the computing architecture.

“‘We had to tell management we’re not going to deliver anything new while all of engineering works on this project for the next two months,’ Scott says. ‘It was a scary thing.’”







DevOps Enterprise Summit Learnings

Planning and Organizing Initiatives DevOps

DevOps Enterprise: Lessons Learned

- On Nov 7-9, we will hold the third DevOps Enterprise Summit, a conference for horses, by horses
- Speakers included fifty leaders from:
 - Macy's, Disney, Target, GE Capital, Western Union, Sherwin Williams, Blackboard, Nordstrom, Telstra, US Department of Homeland Security, CSG, Raytheon, IBM, Ticketmaster, MITRE, Marks and Spencer, Barclays Capital, Microsoft, Nationwide Insurance, Capital One, Gov.UK, Fidelity, Rally Software, Neustar, Walmart, PNC, ADP, ...

Observations

- They were using the same technical practices and getting the same sort of metrics as the unicorns
 - Target: 100+ deploys per week, < 10 incidents per month, enabled 53 business initiatives
 - Capital One: 100s of deploys per day, lead time of minutes
 - Macy's: 1,500 manual tests every 10 days, now 100Ks automated tests run daily
 - Disney: Has embedded nearly 100 Ops engineers into LOB teams across the enterprise
 - Nationwide Insurance: Retirement Plans app (COBOL on mainframe)
 - Raytheon: testing and certification from months to a day
 - US CIS: security and compliance testing run every code commit

Observations

- The transformation stories are among the most courageous I've ever heard –
 - Often the transformation leader was putting themselves in personal jeopardy
 - Why? Absolute clarity and conviction that it was the right thing for the organization

Chivas Nambiar, Verizon

Measuring Success

As we onboard hundreds of apps onto the DevOps process and toolchain, every team is asked to measure and set goals across standard parameters



verizon[✓]

Cycle Time



Months → Days/Hours

Deploy Frequency



1 / Month → Daily/Weekly

Defects per Feature



20%-50% Reduction

Financial Impact



↑ % Benefit



Before	After	Improvement
2-week releases (schedule)	24-hour releases (release when ready)	93%
9-day regression time	3-hour regression time	99%
10k tests per month - manual	40k tests per month (85% automation)	4x increase
2500 open production defects	500 open production defects	5x decrease



C
We

- Productive Di
- Continuous In
- Discipline ove
- Automation o
- Quality over T
- "We can if..."
- Collaboration
- Learning vers
- People versus



- Launch
- Single c
- Multipl
- Moved to Agile development process around 4 years ago
- Operational environment has multiple MSPs, vendors, and stakeholders

- Teams need to be treated like teams regardless of location
- Time must be invested in the team, regardless of location
- The improvements were real
 - 88% reduction in release time
 - 94% reduction in regression time
 - 4x increase in tests run
 - Reduction in defects from 2500 to 500
- Partnership is a key part of the process, as silos must be removed
- Executive sponsorship is required

- The music never changes

What made us successful

- Teams need to be treated like teams regardless of location
- Time must be invested in the team, regardless of location
- The improvements were real
 - 88% reduction in release time
 - 94% reduction in regression time
 - 4x increase in tests run
 - Reduction in defects from 2500 to 500
- Partnership is a key part of the process, as silos must be removed
- Executive sponsorship is required

Michael Hrenko, Blue Shield of CA

Solving for the database

Current Process: 3 DAYS

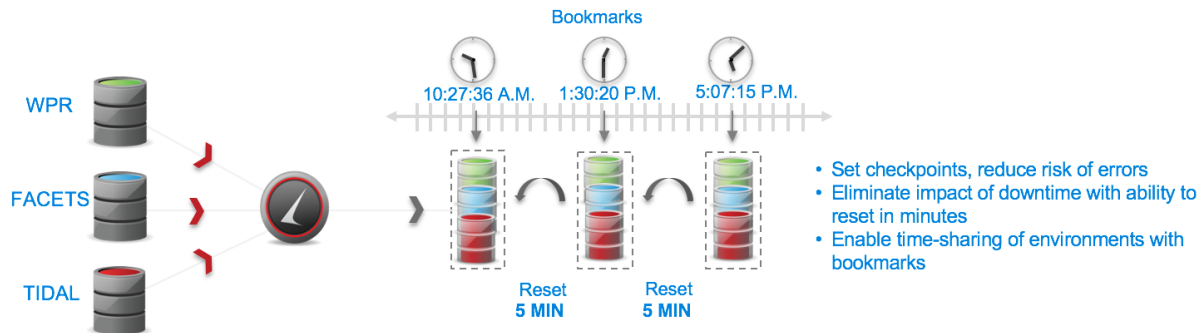
*Multiple admins,
manual processes.*

Mon	Tues	Wed
Request submitted	Restore blue copy backup set, apply archived redo log	Execute post-refresh script

Database as a Service: 5 MIN

*Self-service,
automated.*

Mon	Tues	Wed
Reset		



Heather Mickman, Target, Inc.

- Abolished the TEP-LARB process
- As a result, she won the Lifetime Achievement Award from her grateful team

An ATM Story...

Capability 3

- Spreading new knowledge throughout the organization
 - The new discovery of local knowledge and improvements are turned into global improvements, shared throughout the organization
 - Learning is fed back into the system to prevent future failures

High trust culture, blameless post-mortems when things go wrong, single source code repositories enterprise-wide, ...

Top Challenges

- Creating automated tests for legacy applications
- Culture and leadership issues
- Roles and responsibilities
- Information security and compliance
- Metrics

Why DevOps Is So Breathtaking

Functional-Orientation (Optimize for cost)	Market-Orientation (Optimize for speed)
Enables expertise-based division of labor	Enable speed and meeting customer demand
Mechanistic, role-based, hierarchical	Organic, high cross-unit interdependence, flat organization
Enables skill development and within-function knowledge transfer	Inter-function coordination Market responsiveness

Organizational Adoption

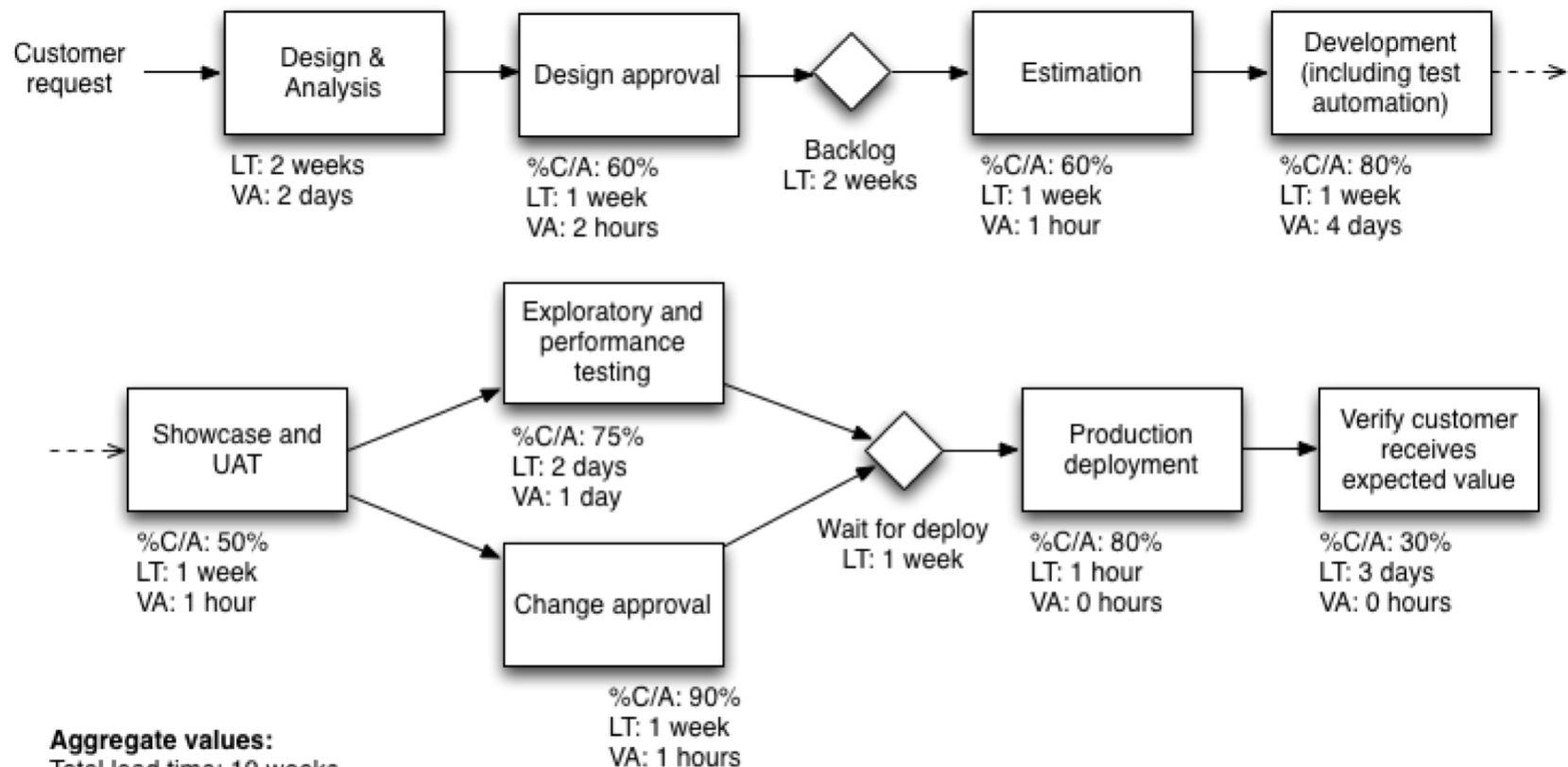
- Create a dedicated team
- Pick an initial value stream
 - Green vs. brown field
 - System of record vs. system of engagement
 - Sufficiently large win potential
- Organize the team by value stream, not by areas of functional expertise
 - Re-imagining the next-generation IT Operations organization
 - Shared services around platforms, testing, deployment, monitoring

Organizational Adoption

- Keep planning horizons short
- Create Hack Days and other delivery forums
- Stay focused on the constraint
- Think through ideal DevOps adoption

Integrating Continuous Delivery Into ITIL

- Process areas of Release and Configuration Management are now automated
- Problem Management now resides in Development backlog
- Failures must result in automated tests in the continuous deployment and testing pipeline
- Reliance on reviewing “80 word essays” in RFCs in Word docs is reduced





TARGET TECH

FlashBuilds: A new approach we leverage to solving complex business problems

TARGET FLASHBUILDS

*Crushing business problems with: One part flash
mob, one part scrum, one part hackathon ... all
to iterate to awesome*

10 Nov 2014 in DevOps © 15 minutes read

At a couple of recent conferences around DevOps ([MSP DevOps Days](#) and [DevOps Enterprise Summit](#)), [Heather Mickman](#) and [Ross Clanton](#) from Target explained

Security Automation at Risk I/O

Chef All the Things!

Open-Sourced Cookbooks

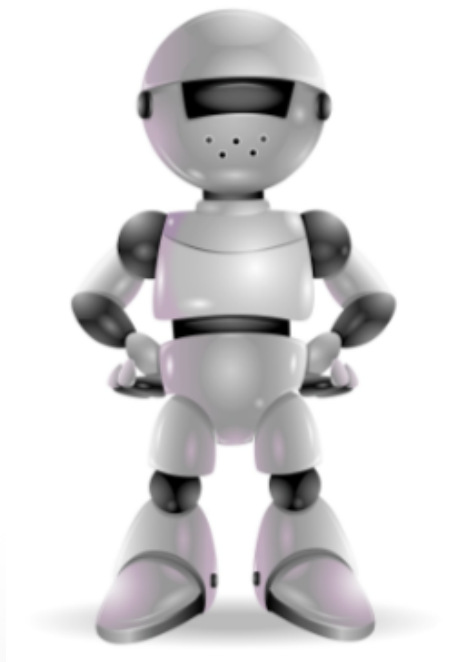
ModSecurity (airbag)	Nessus (air bag ctrl)	Nmap (brakes)	SSH
iptables (shoulder belt)	encrypted volumes	Duo 2FA	openVPN

Test All the Things! (including security)

Static + Dynamic Throughout

Continuous Integration via CircleCI

ChatOps = Slack + graphite + logstash + sensu + pagerduty



DevOps as a Compliance Enabler

Automation as Evidence & Doc

Cookbooks

Leveraging the ELK Stack

Elasticsearch

Logstash

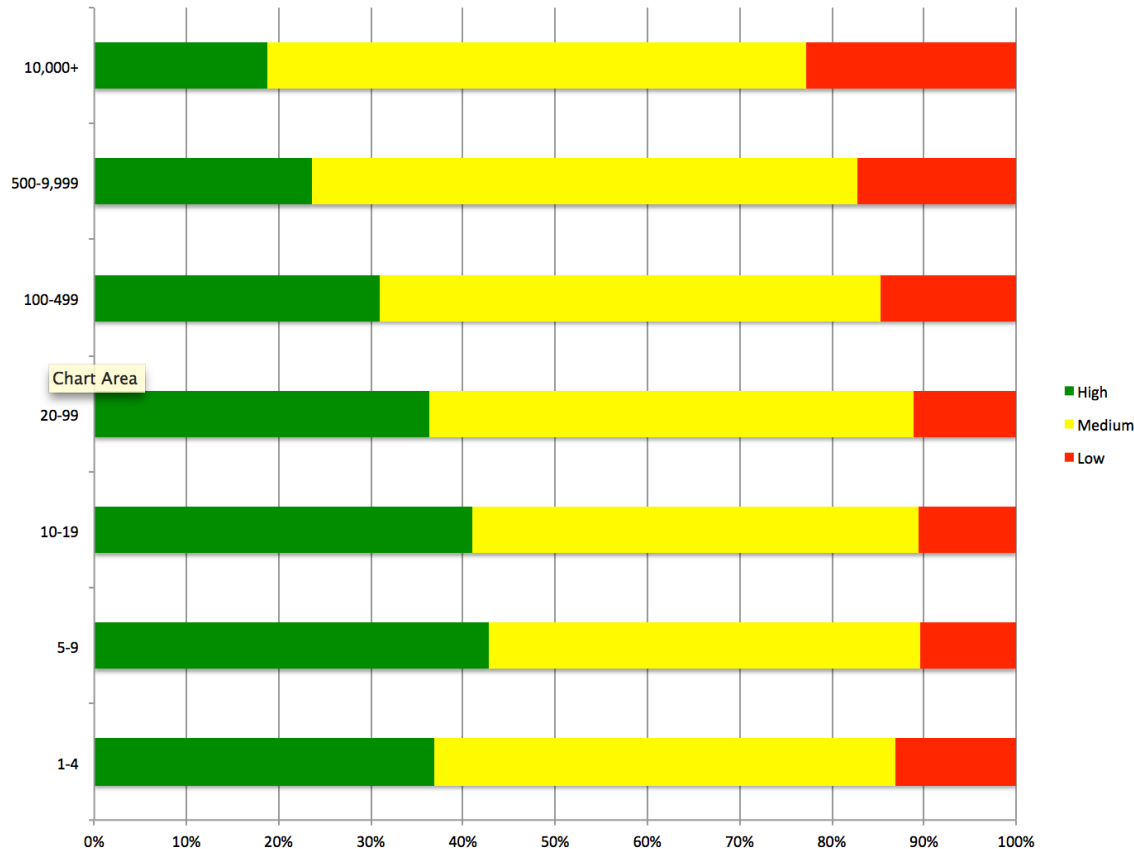
Kibana

Github + Code Climate + Risk I/O

Compliance Automation Extra Credit: <https://telekomlabs.github.io/>



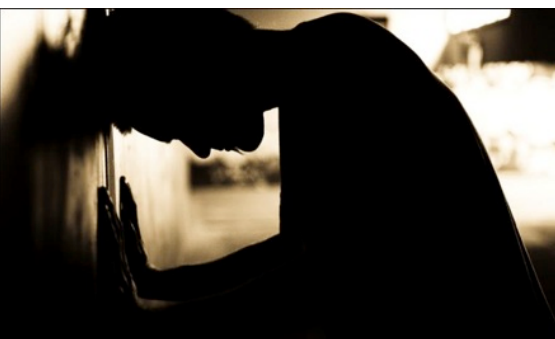
Can Large Orgs Be High Performers?



Yes.

But orgs with 10,000+ employees 40% less likely to be high performing vs. 500 employee orgs...

**Why Do I Think This Is
Important?**

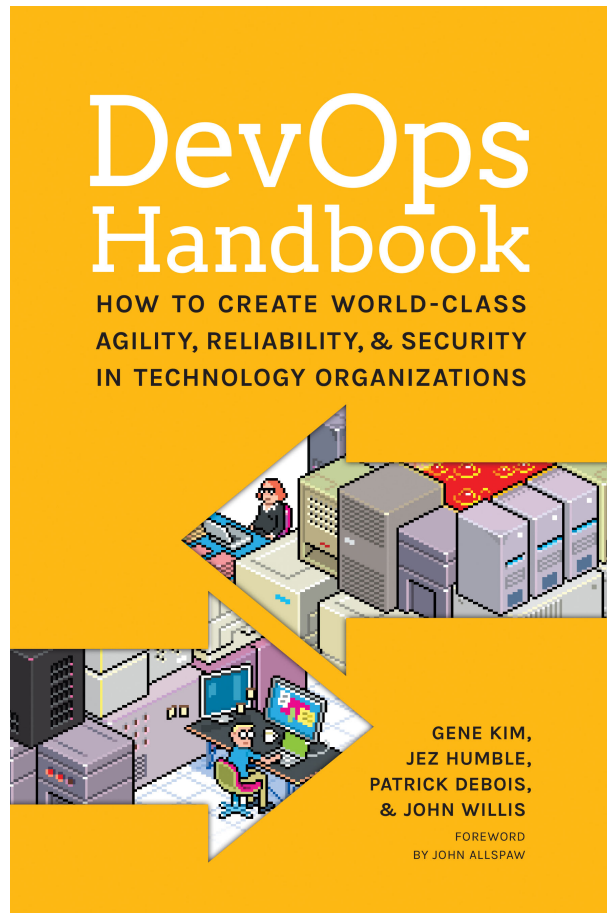


The Downward Spiral...



The DevOps Handbook Is Here!

- 5+ years in the making
- 23 chapters
- 48 case studies
- 98,124 words
- 48 images
- 503 endnotes
- 192 footnotes



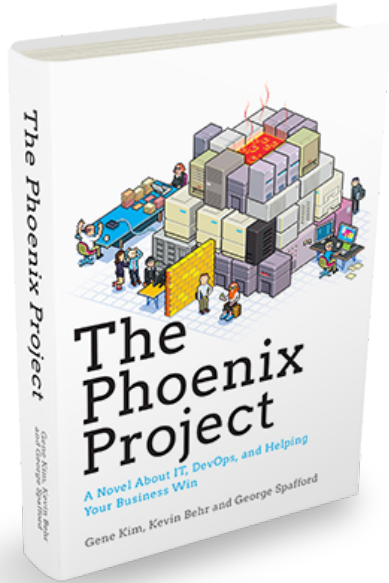


Opportunity Cost Of Wasted IT Spending?

**\$2,600,000,000,000.00 per year
(\$2.6 Trillion US)**

Our Mission

Positively influence the lives of one million IT professionals by 2017.



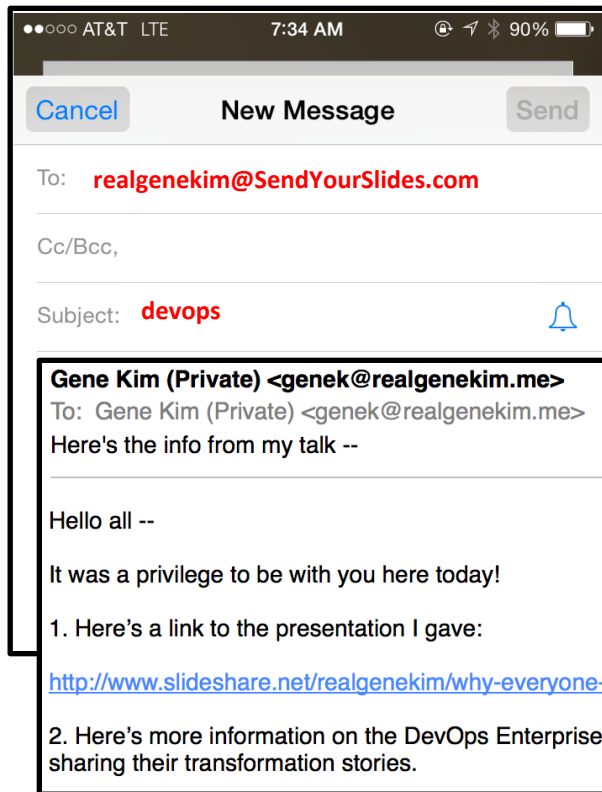
Want More Learn More?

To receive the following:

- A copy of this presentation
- Videos and slides from DevOps Enterprise 2014-2016
- The 140 page excerpt of *The DevOps Handbook*
- The 140 page excerpt of *The Phoenix Project*
- Link to the DevOps Audit Defense Toolkit
- One hour excerpt of *The Phoenix Project* audiobook

Just pick up your phone, and send an email:

To: **realgenekim@SendYourSlides.com**
Subject: **devops**



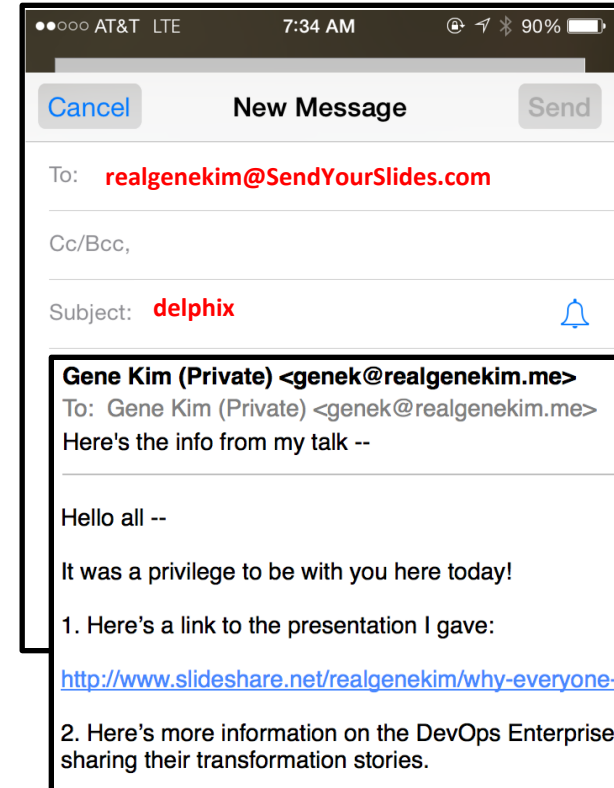
Want More Learn More?

To receive the following:

- A copy of this presentation
- The 140 page excerpt of *The Phoenix Project*
- Videos and slides from DevOps Enterprise 2014
- Information on DevOps Enterprise 2015
- Link to the DevOps Audit Defense Toolkit
- Announcement of *The Phoenix Project* audiobook
- See early drafts of our upcoming *DevOps Cookbook*

Just pick up your phone, and send an email:

To: **realgenekim@SendYourSlides.com**
Subject: **delphix**



Why Everyone Needs DevOps Now:

**My Fifteen Year Journey Studying
High Performing IT Organizations**

Gene Kim

What About Infosec?

- Ed Bellis
- Former CISO of Orbitz
- VP Information Security at Bank of America
- Currently CEO of Risk I/O



Risk I/O DevOps By the Numbers

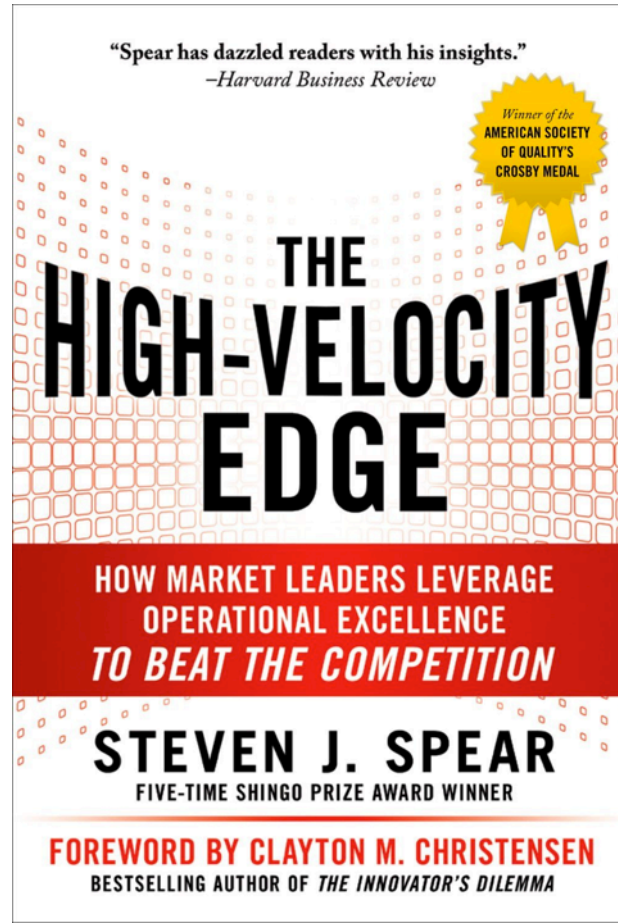
Small & Frequent Commits

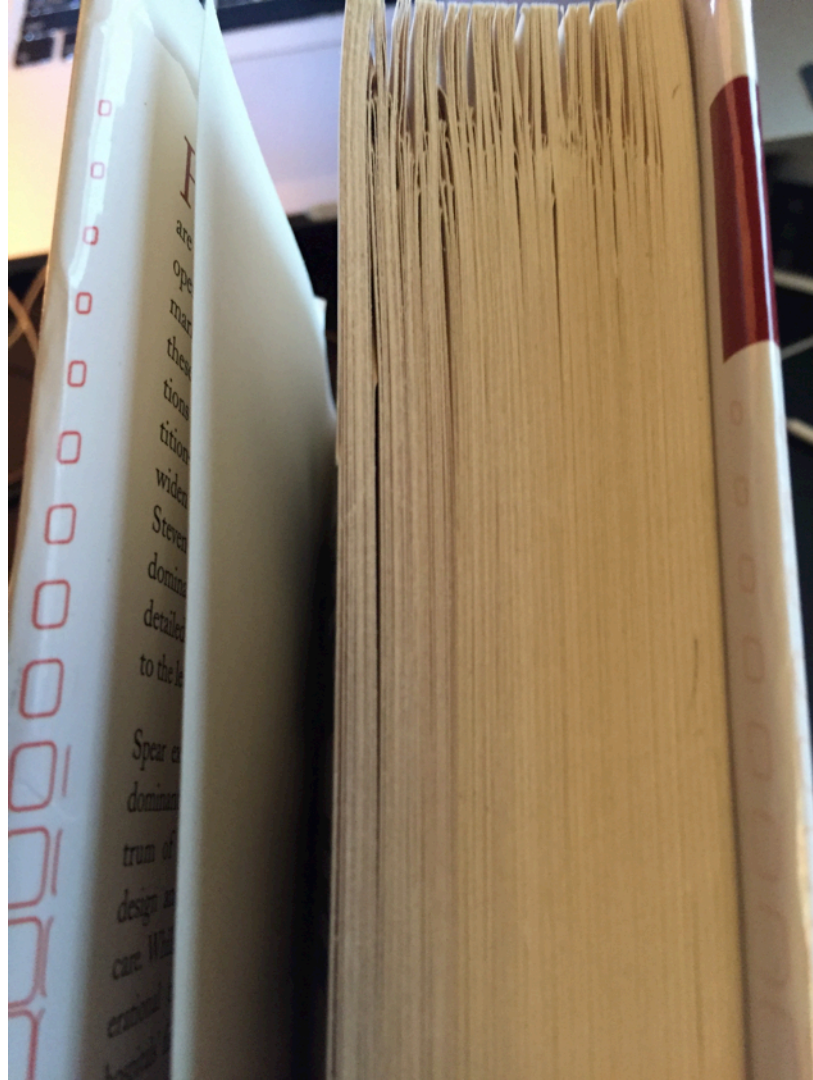
- Average between 75 & 125 commits to Master/week
- Simplicity is your friend



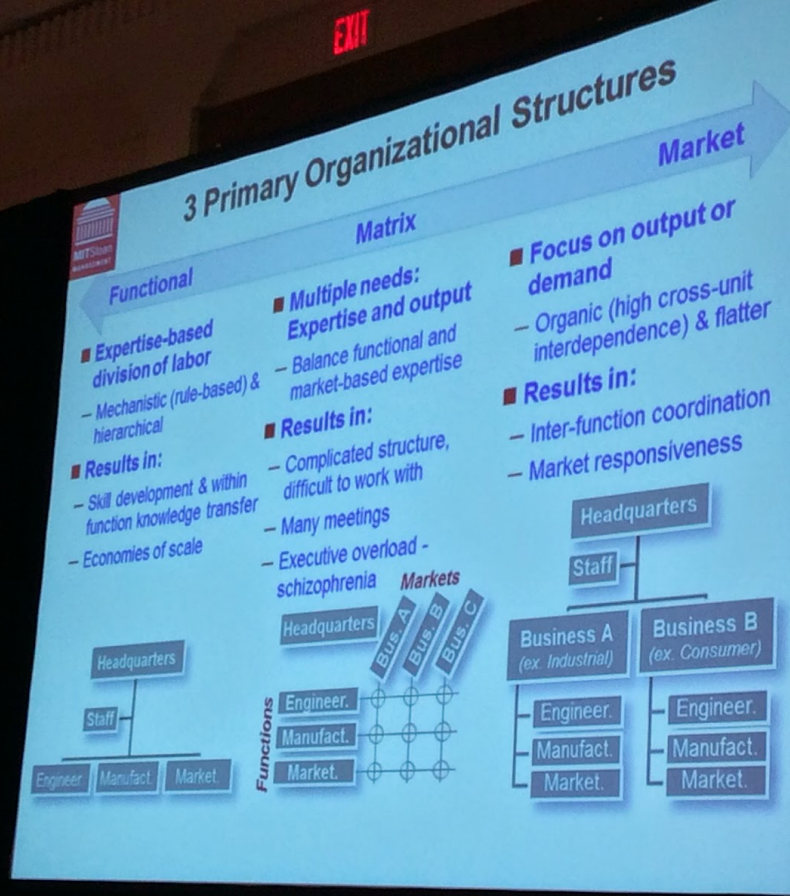
Dev/Test Ratio Story

Dr. Steve Spear





Dr.



Dr. Steven Spear

- “While designing perfectly safe systems is likely beyond our abilities, safe systems are close to achievable” when the following conditions are met...

Capability 1

- See problems as they occur:
 - Complex work is managed so that problems in design are revealed
 - They see problems as they occur, through relentless testing of assumptions

*Automated testing in the deployment pipeline,
proactive monitoring of the production environment, ...*

Capability 2

- Swarming and solving problems as they are seen to build new knowledge
 - Problems that are seen are solved so that new knowledge is built quickly
 - Improvement of daily work is prioritized above daily work

Stopping work when builds, tests, deployments and services break, enabling fast feedback loops, especially to Dev...

Capability 3

- Spreading new knowledge throughout the organization
 - The new discovery of local knowledge and improvements are turned into global improvements, shared throughout the organization
 - Learning is fed back into the system to prevent future failures

High trust culture, blameless post-mortems when things go wrong, single source code repositories enterprise-wide, ...

Capability 4

- Leading by developing
 - The job of leaders is not to command and control, but to create other capable leaders who can perpetuate this system of work

“Culture isn’t just touchy-feely kumbayah. Instead, it is the consistent response by a group of people to conditions. When we change culture, we fundamentally shift how people respond to a situation.

– Dr. Steven Spear

“The most effective way is for senior leaders to change the conversation from ‘did you carry your orders out?’ to ‘what did you learn today?’ ”

– Dr. Steven Spear