

# What is Rugged All About? Faster/Safer w/ Rugged DevOps

*Joshua Corman @joshcorman*  
*Rugged Software @RuggedSoftware*

**goto;**  
conference



**Click 'engage'  
to rate session.**

Rate **12** sessions to get the  
supercool GOTO reward

 Join the conversation #gotoldn

**HONEY BADGER**



**DON'T CARE**

HONEY BADGER



CARES





~ Marc Andreessen 2011

**SOFTWARE IS EATING THE WORLD**

*Multicast vvv. rainbow*



Thu Jul 19 00:00:00 2001 (UTC)

Victims: 159

<http://www.caida.org/>

Copyright (C) 2001 UC Regents, Jeff Brown for CAIDA/UCSD





# Trade Offs Costs & Benefits



# Beyond **Heartbleed**: OpenSSL in 2014

(31 in NIST's NVD thru December)

CVE-2014-3470	6/5/2014	CVSS Severity: 4.3 MEDIUM ← SIEMENS *
CVE-2014-0224	6/5/2014	CVSS Severity: 6.8 MEDIUM ← SIEMENS *
CVE-2014-0221	6/5/2014	CVSS Severity: 4.3 MEDIUM
CVE-2014-0195	6/5/2014	CVSS Severity: 6.8 MEDIUM
CVE-2014-0198	5/6/2014	CVSS Severity: 4.3 MEDIUM ← SIEMENS *
CVE-2013-7373	4/29/2014	CVSS Severity: 7.5 HIGH
CVE-2014-2734	4/24/2014	CVSS Severity: 5.8 MEDIUM ** DISPUTED **
CVE-2014-0139	4/15/2014	CVSS Severity: 5.8 MEDIUM
CVE-2010-5298	4/14/2014	CVSS Severity: 4.0 MEDIUM
<b>CVE-2014-0160</b>	<b>4/7/2014</b>	<b>CVSS Severity: 5.0 MEDIUM ← HeartBleed</b>
CVE-2014-0076	3/25/2014	CVSS Severity: 4.3 MEDIUM
CVE-2014-0016	3/24/2014	CVSS Severity: 4.3 MEDIUM
CVE-2014-0017	3/14/2014	CVSS Severity: 1.9 LOW
CVE-2014-2234	3/5/2014	CVSS Severity: 6.4 MEDIUM
CVE-2013-7295	1/17/2014	CVSS Severity: 4.0 MEDIUM
CVE-2013-4353	1/8/2014	CVSS Severity: 4.3 MEDIUM
CVE-2013-6450	1/1/2014	CVSS Severity: 5.8 MEDIUM
...		

As of today, internet scans by MassScan reveal 300,000 of original 600,000 remain unpatched or unpatchable



# Heartbleed + (UnPatchable) Internet of Things == \_\_\_\_ ?

## In Our Bodies



## In Our Homes



## In Our Cars



## In Our Infrastructure





**ShellShock**  
{bashbug}

## MODIFIED MERCALI INTENSITY SCALE

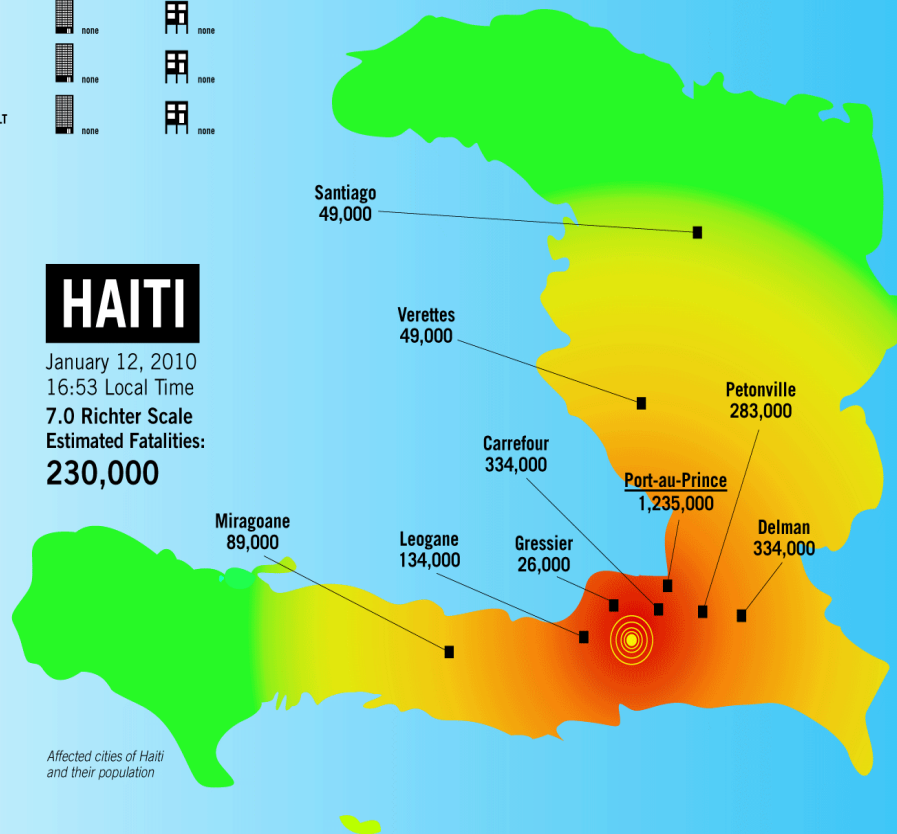
Shaking	Structural Damage to Resistant Buildings	Structural Damage to Vulnerable Buildings
<b>X</b> EXTREME	very heavy	very heavy
<b>IX</b> VIOLENT	heavy	heavy
<b>VIII</b> SEVERE	moderate/heavy	heavy
<b>VII</b> VERY STRONG	moderate	moderate/heavy
<b>VI</b> STRONG	light	moderate
<b>V</b> MODERATE	very light	light
<b>IV</b> LIGHT	none	none
<b>II-III</b> WEAK	none	none
<b>I</b> NOT FELT	none	none

# A TALE OF TWO QUAKES

In the span of two months, two massive earthquakes struck in Haiti and Chile. But while the temblor in Chile registered much higher on the Richter scale, the loss of life and damage in Haiti was far more severe. Why is that? Chile—which has experienced serious earthquakes in recent decades—has a robust building code to make sure buildings are earthquake resistant; Haiti has no code to speak of. And a look at both quake's scores on the Modified Mercalli Intensity Scale—which is used to measure how earthquakes affect those experiencing them—shows that while Chile's quake may have been stronger overall, Haiti had a larger population and more urban areas hit by more intense and damaging shaking.

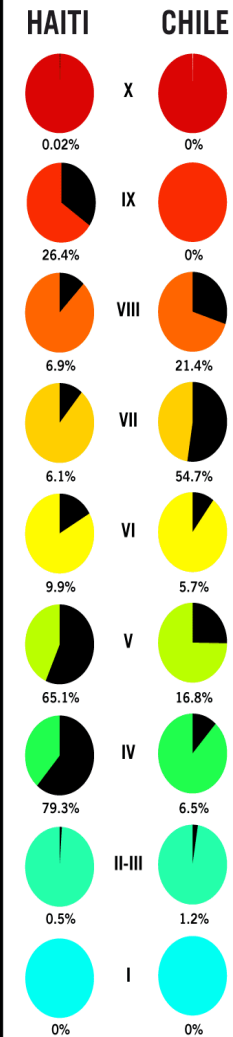
## HAITI

January 12, 2010  
16:53 Local Time  
7.0 Richter Scale  
Estimated Fatalities:  
**230,000**



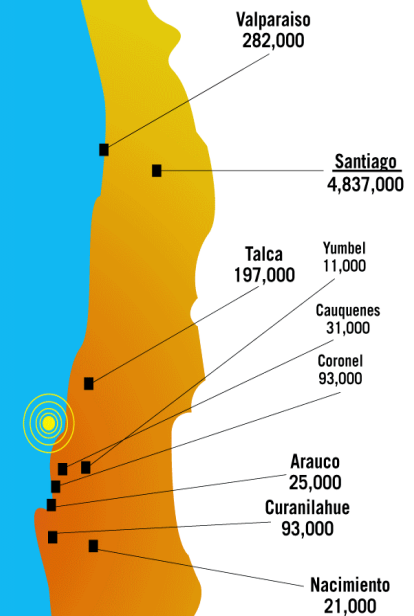
Affected cities of Haiti and their population

## POPULATION AFFECTED (percentage)



## CHILE

February 27, 2010  
03:34 Local Time  
8.8 Richter Scale  
Estimated Fatalities:  
**279**



Affected cities of Chile and their population

SOURCE: U.S. Geological Survey  
A collaboration between GOOD and kartsonnelker



# *The Rugged Manifesto*

*I am rugged... and more importantly, my code is rugged.*

*I recognize that software has become a foundation of our modern world.*

*I recognize the awesome responsibility that comes with this foundational role.*

*I recognize that my code will be used in ways I cannot anticipate, in ways it was not designed, and for longer than it was ever intended.*

*I recognize that my code will be attacked by talented and persistent adversaries who threaten our physical, economic, and national security.*

*I recognize these things - and I choose to be rugged.*

*I am rugged because I refuse to be a source of vulnerability or weakness.*

*I am rugged because I assure my code will support its mission.*

*I am rugged because my code can face these challenges and persist in spite of them.*

*I am rugged, not because it is easy, but because it is necessary... and I am up for the challenge.*



*I recognize that software has become a foundation of our modern world.*

*I recognize the awesome responsibility that comes with this foundational role.*

*I recognize that my code will be used in ways I cannot anticipate, in ways it was not designed, and for longer than it was ever intended.*

*I recognize that my code will be attacked by talented and persistent adversaries who threaten our physical, economic, and national security.*

*I recognize these things - and I choose to be rugged.*

*I am rugged because I refuse to be a source of vulnerability or weakness.*

*I am rugged because I assure my code will support its mission.*

# I Am The Cavalry

The Cavalry isn't coming... It falls to us:

## Problem Statement

Our society is adopting connected technology *faster than we are able to secure it.*

## Mission Statement

To ensure connected technologies with the potential to impact public safety and human life are *worthy of our trust.*



Medical



Automotive



Connected  
Home



Public  
Infrastructure

**Why** Trust, public safety, human life

**How** Education, outreach, research

**Who** Infosec research community

**Who** Global, grass roots initiative

**What** Long-term vision for cyber safety

**Collecting** existing research, researchers, and resources

**Connecting** researchers with each other, industry, media, policy, and legal

**Collaborating** across a broad range of backgrounds, interests, and skillsets

**Catalyzing** positive action sooner than it would have happened on its own



# Security is Dead. Long Live Rugged DevOps: IT at Ludicrous Speed...

*Josh Corman, Gene Kim*  
VERY ROUGH 1<sup>ST</sup> Draft



Session ID: CLD-106

Session Classification: Intermediate

**RSA**CONFERENCE2012



SOURCEfire



QUALYS<sup>®</sup>  
ON DEMAND SECURITY



CORETRACE

FORTINET



Akamai

FASTER FORWARD



CONFERENCE

Where The World  
Talks Security

SOURCEfire



QUA  
ON DEMAND

FORTINET



Akamai

FASTER FORWARD

ENCE

Where The World  
Talks Security





h/t @petecheslock DevOpsDays Austin 2015

# The New Lifecycle

Impact on  
Releases per Year  
(Cycle Time)

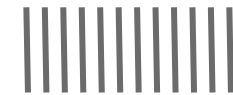
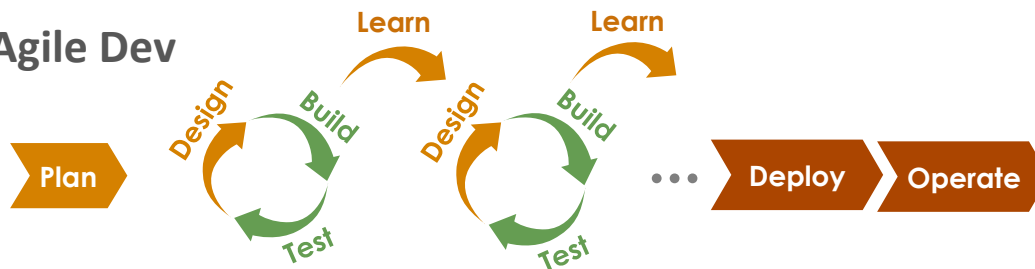
## Traditional Lifecycle (Waterfall)



1-2

*Cycle Time: Months-Years*

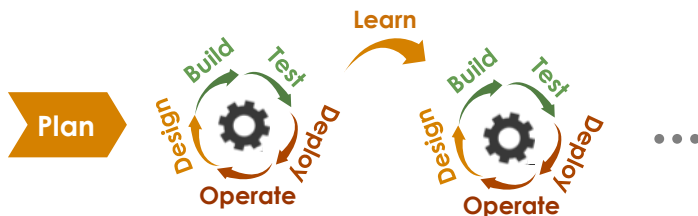
## Agile Dev



10-20

*Cycle Time: Days-Weeks*

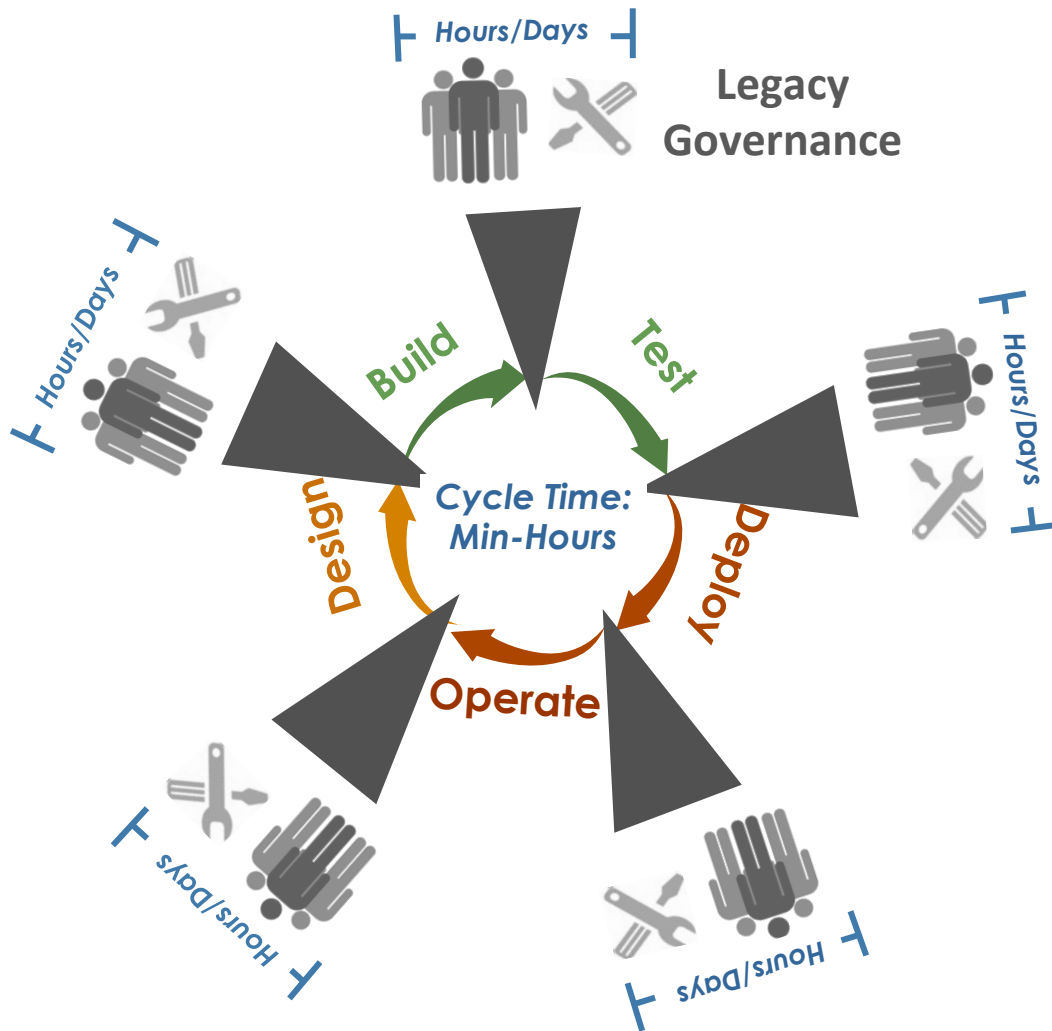
## Modern Lifecycle (+DevOps, Continuous \*)



100-200

*Cycle Time: Minutes-Hours*

# Cycle Time Squeeze



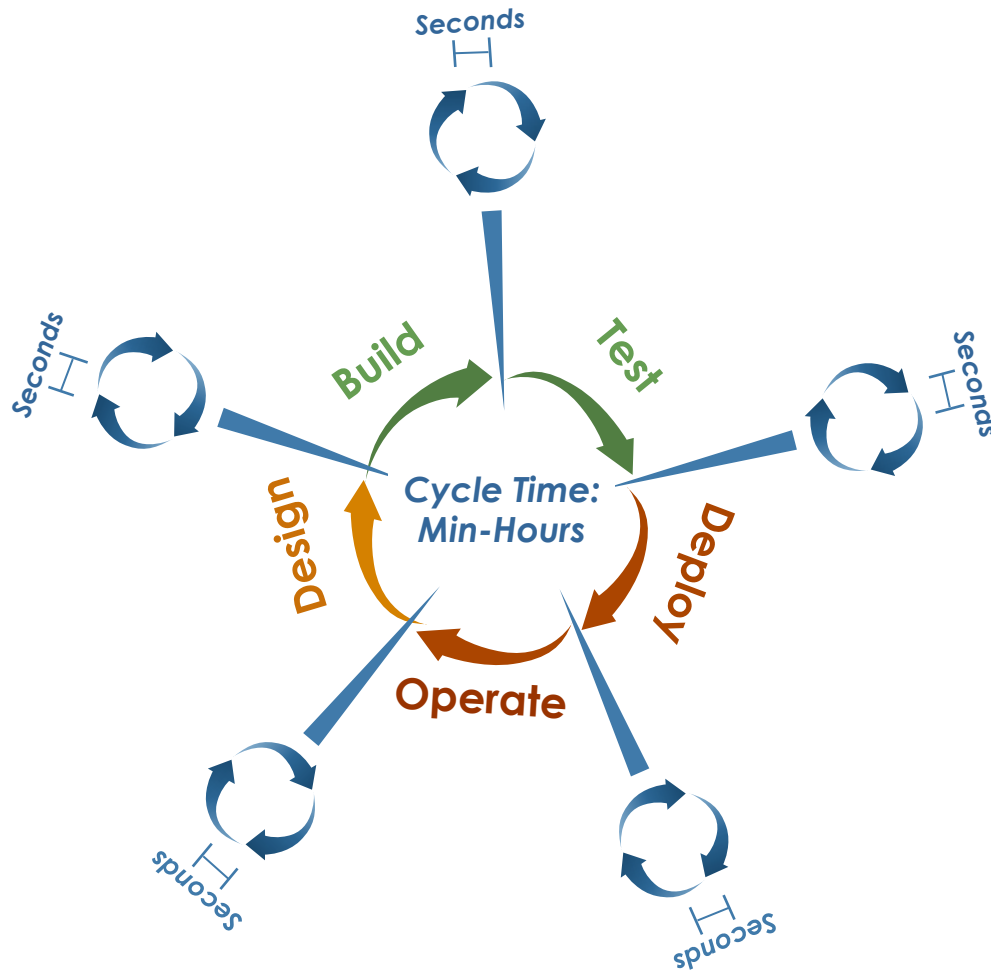
*If it does not fit,  
It does not get done.*

- Work Arounds
- Batch Scans
- Rework
- Exposure

Go Fast **OR** Sleep at Night



# Cycle Time Synergy



## *Continuous Security for Continuous Delivery*

- No Interruption
- Entire Lifecycle
- Solve Early
- Avoid Rework

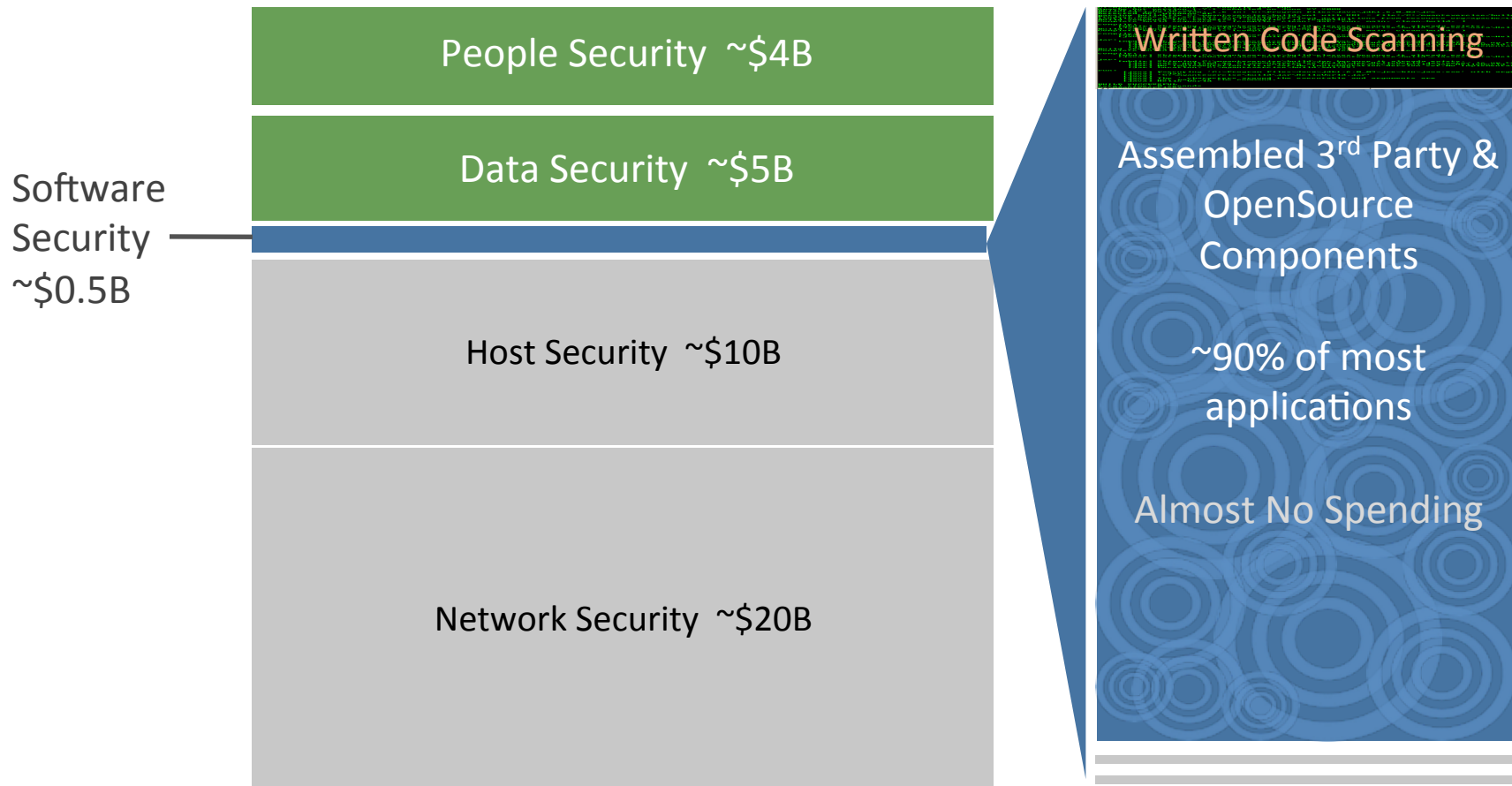
Go Fast **AND** Sleep at Night

# SW Status Quo: Most attacked; least spend

Worse, w/in Software, existing dollars go to the  $\leq 10\%$  written

spending

attack risk



Source: Normalized COBIT spending across IDC, Gartner, The 451 Group; since groupings vary





SOURCE: @joshcorman & @mortman RSAC2015

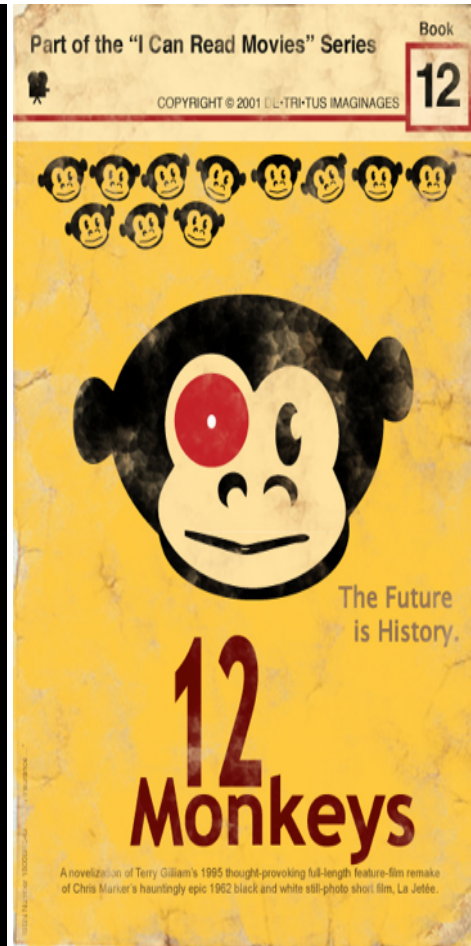


# 1) Instrumentation



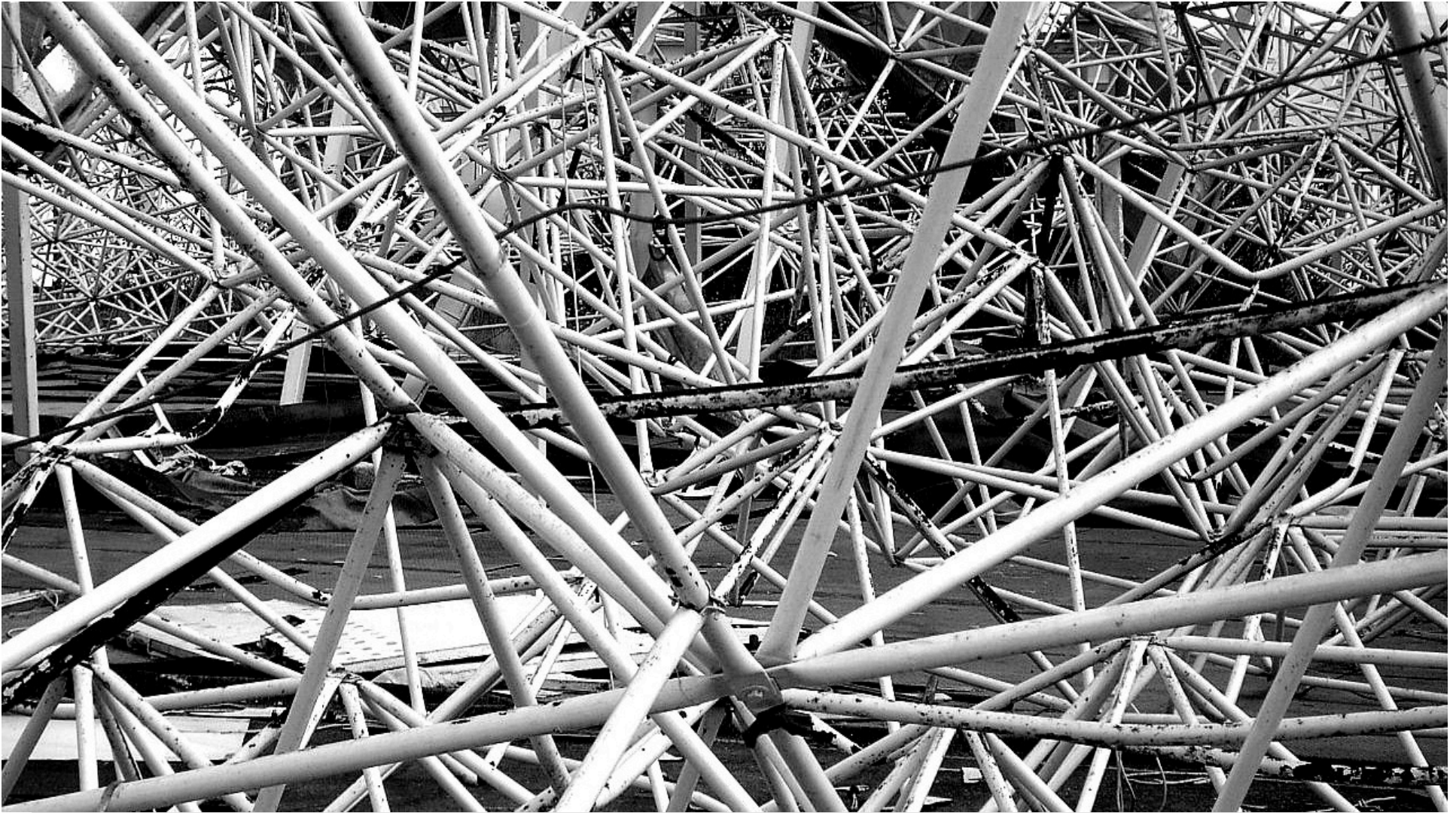
SOURCE: @joshcorman & @mortman RSAC2015

## 2) Be Mean To Your Code!





### 3) Complexity is the enemy of “all the things”



SOURCE: @joshcorman & @mortman RSAC2015



## 4) Change Management (finally)



SOURCE: @joshcorman & @mortman RSAC2015

## 5) Empathy tears down walls



SOURCE: @joshcorman & @mortman RSAC2015



# **Rugged DevOps**

## **Going Even Faster**

### **With Software Supply Chains**

**Gene Kim**

Researcher and Author  
IT Revolution Press  
@RealGeneKim

**Joshua Corman**

CTO  
Sonatype  
@joshcorman

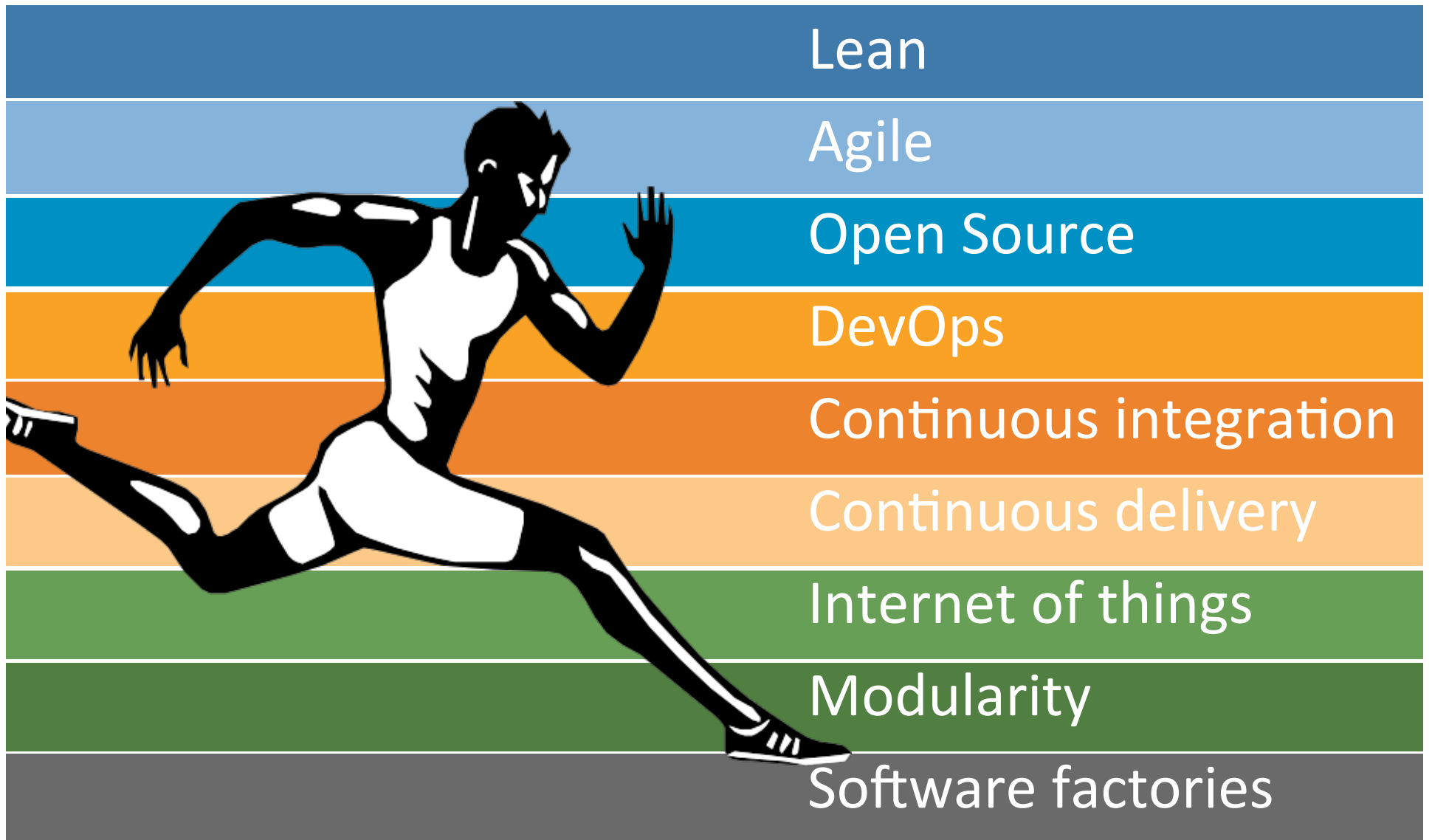






# Software is an innovation differentiator







## Raw innovation

Innovation at  
any cost

Quality?

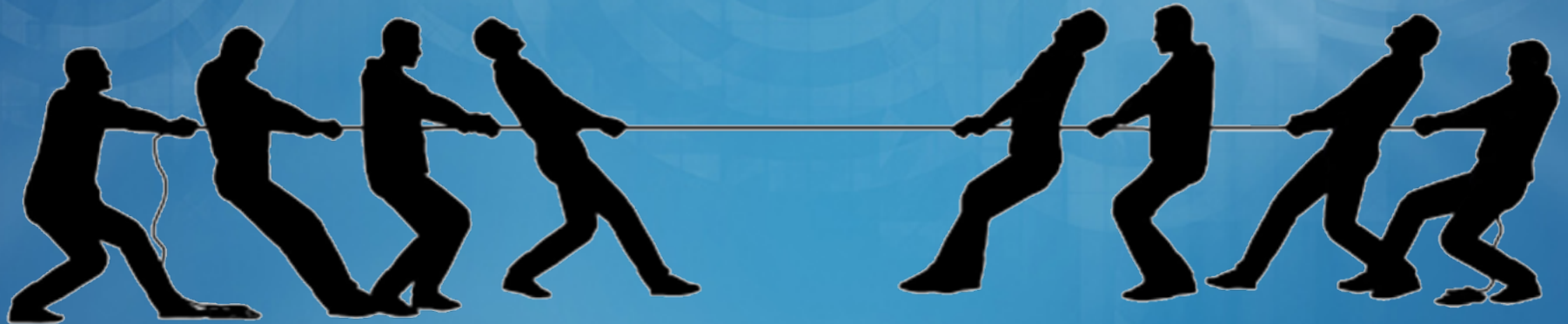
Security?

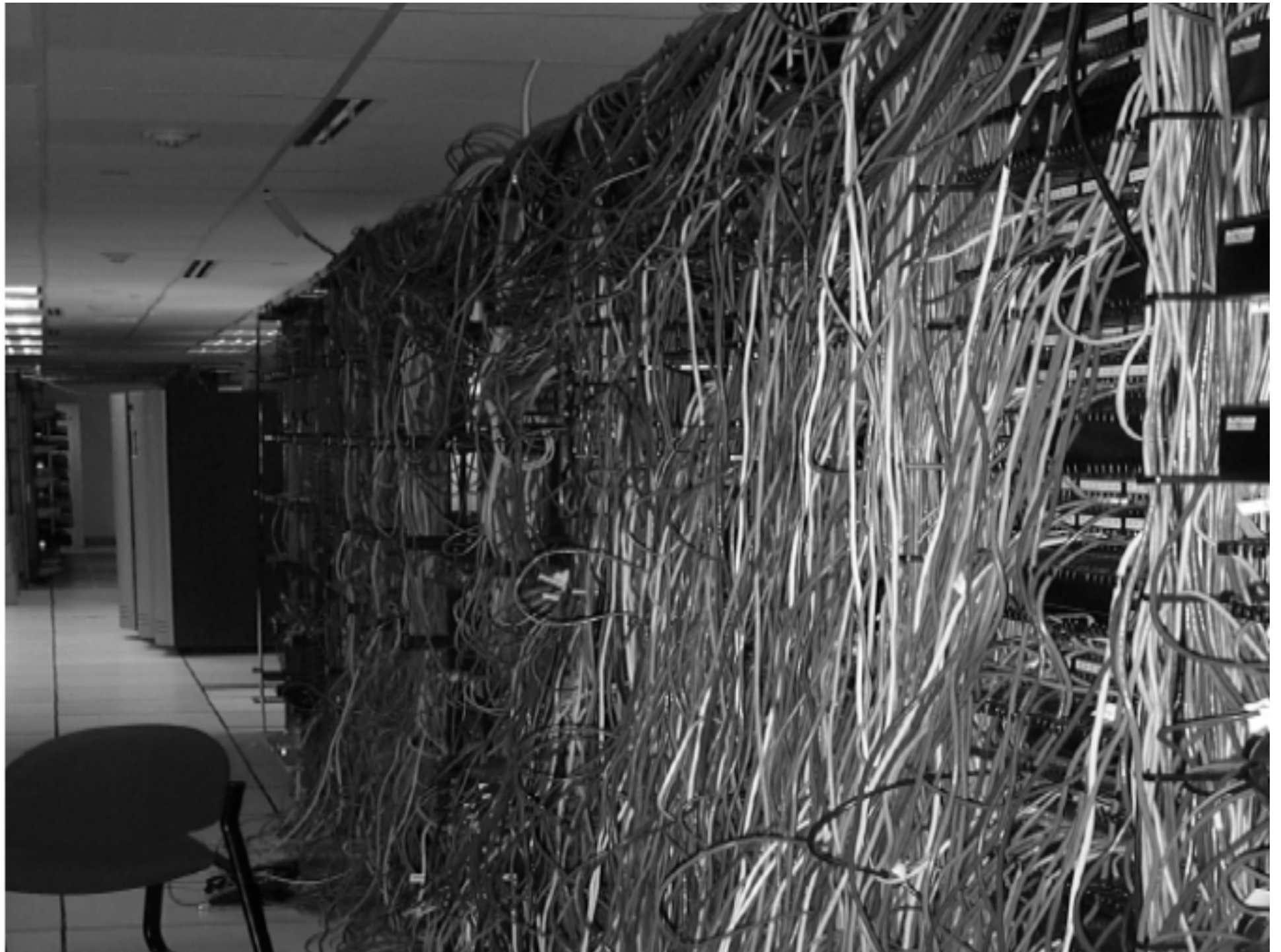
Maintainability?

Repeatability?

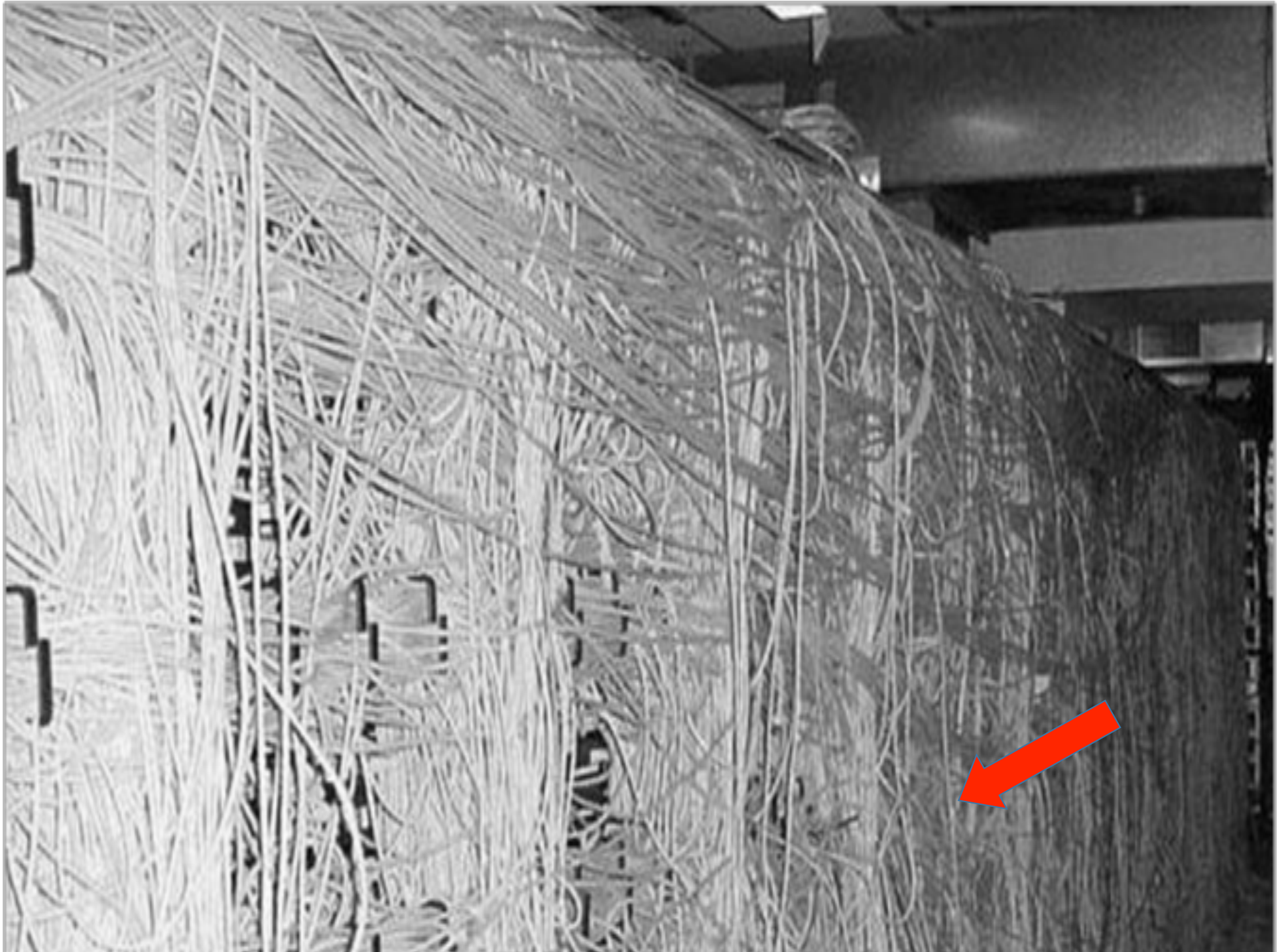
## Net innovation

Net value to the  
organization

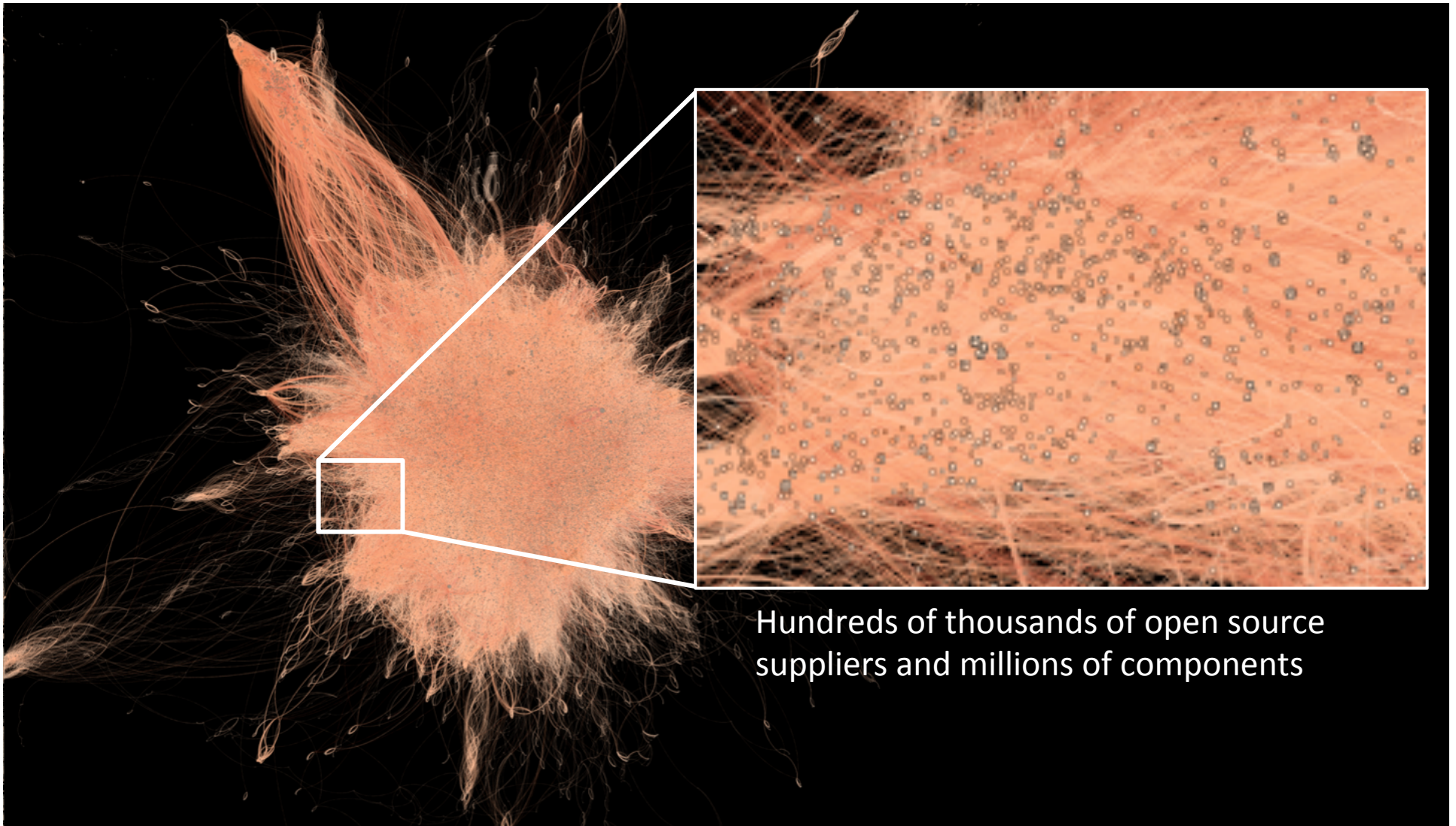








# Your software supply chain is complicated

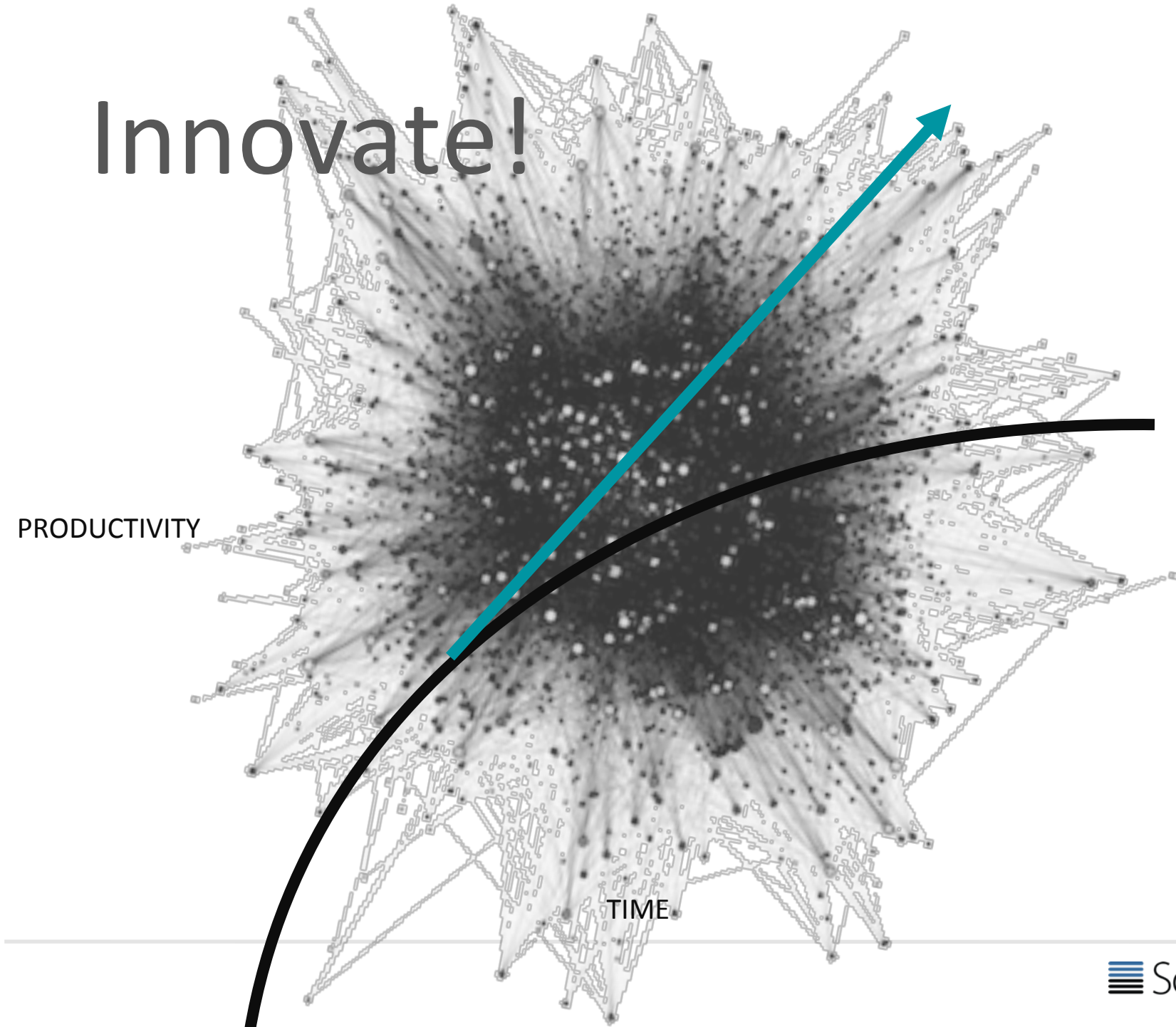




# Innovate!

PRODUCTIVITY

TIME





**“It is not enough to do your best;  
you must know what to do,  
and then do your best”**

- W. Edwards Deming

LeadershipQuote.org



ON TIME



ON BUDGET



ACCEPTABLE  
QUALITY/RISK












Agile goats seek the fruit of Morocco's argan trees. Herders and barriers of thorny branches help thwart the animals.

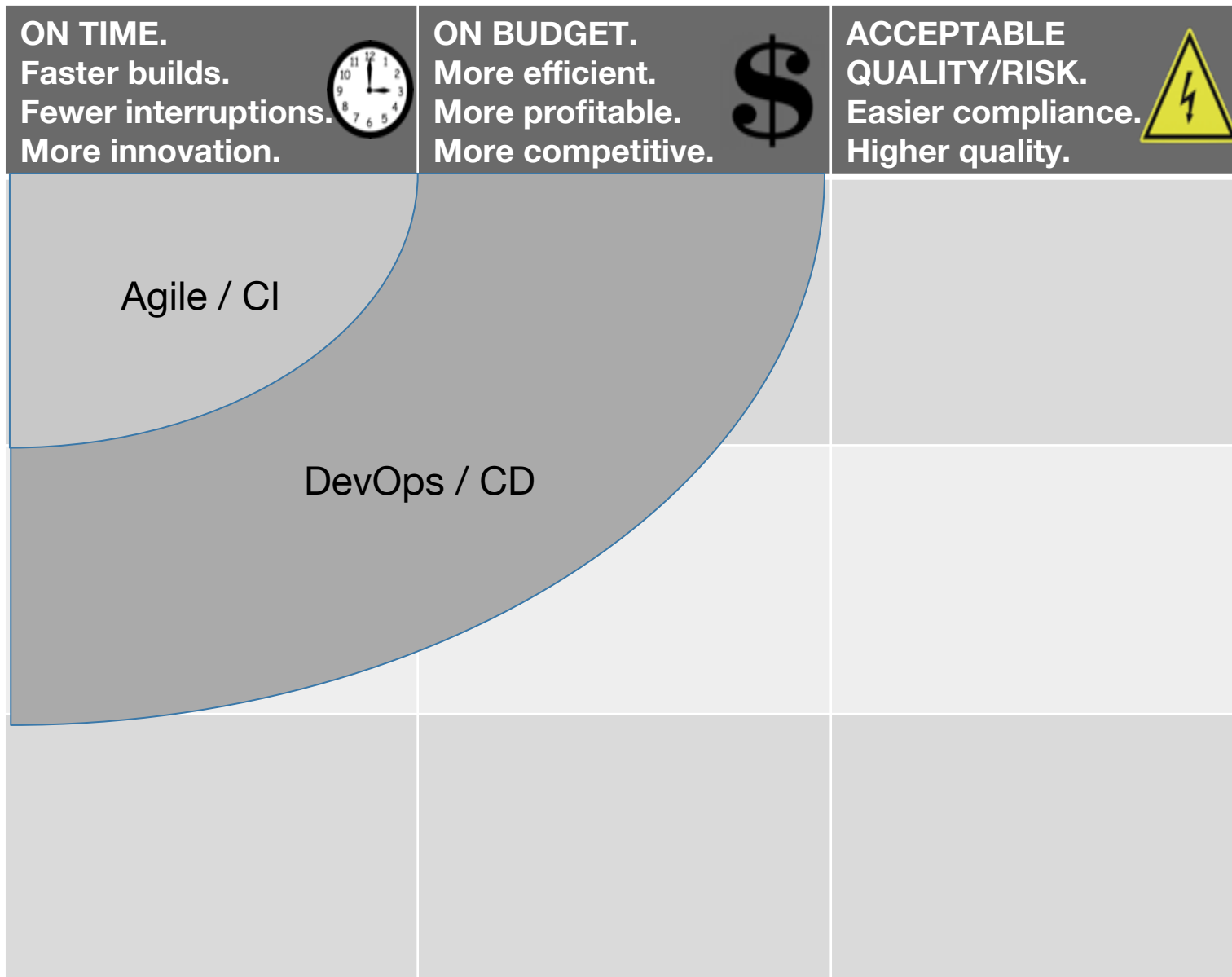


<b>ON TIME.</b> Faster builds. Fewer interruptions. More innovation. 	<b>ON BUDGET.</b> More efficient. More profitable. More competitive. 	<b>ACCEPTABLE QUALITY/RISK.</b> Easier compliance. Higher quality. 
Agile / CI		



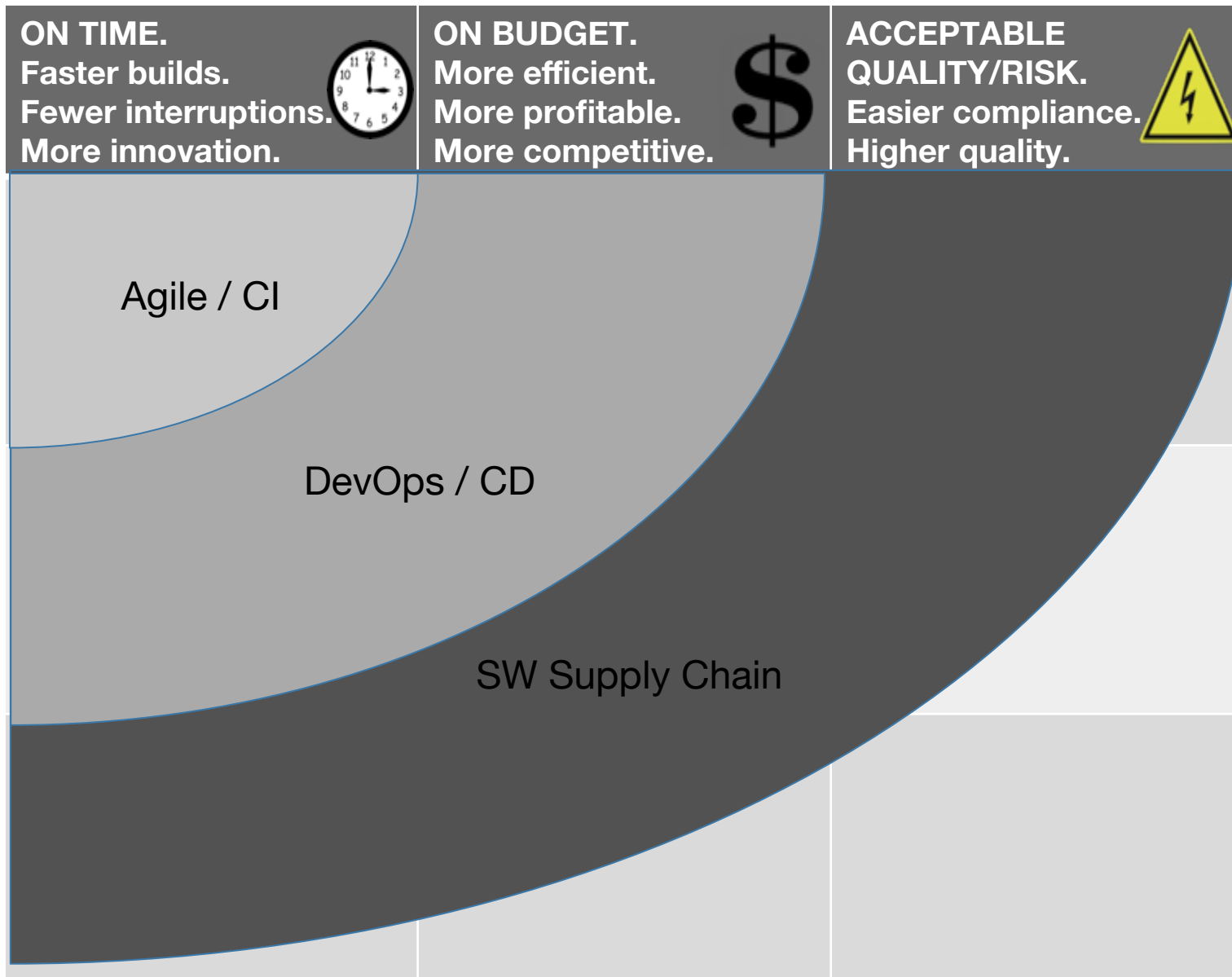


DEV OPS IN A BOX





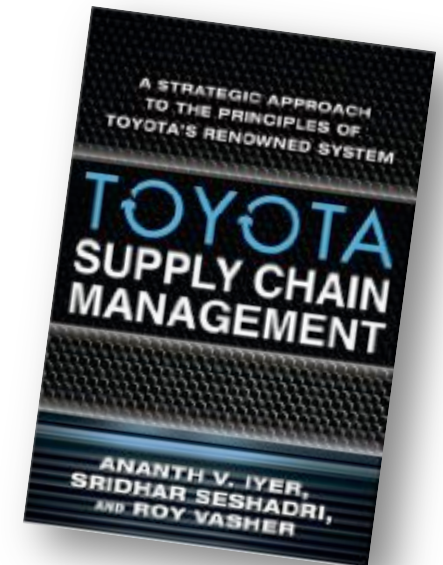






# Supply chain advantage

	Toyota Advantage	Toyota Prius	Chevy Volt
Unit Retail Price	<b>61%</b>	\$24,200	\$39,900
Units Sold/Month	<b>13x</b>	23,294	1,788
In-House Production	<b>50%</b>	27%	54%
Plant Suppliers	<b>16%</b>	125	800
<i>Firm-Wide Suppliers</i>	<b>4%</b>	224	5,500



Source: *Toyota Supply Chain Management: A Strategic Approach to Toyota's Renowned System*, by Ananth Iyer and Sridhar Seshadri

# A Fuller Embrace of Deming

**We can improve software supply chains with three proven principles:**

Use better & fewer  
suppliers

Use higher  
quality parts

Track what you use  
and where





ON TIME



ON BUDGET



ACCEPTABLE  
QUALITY/RISK

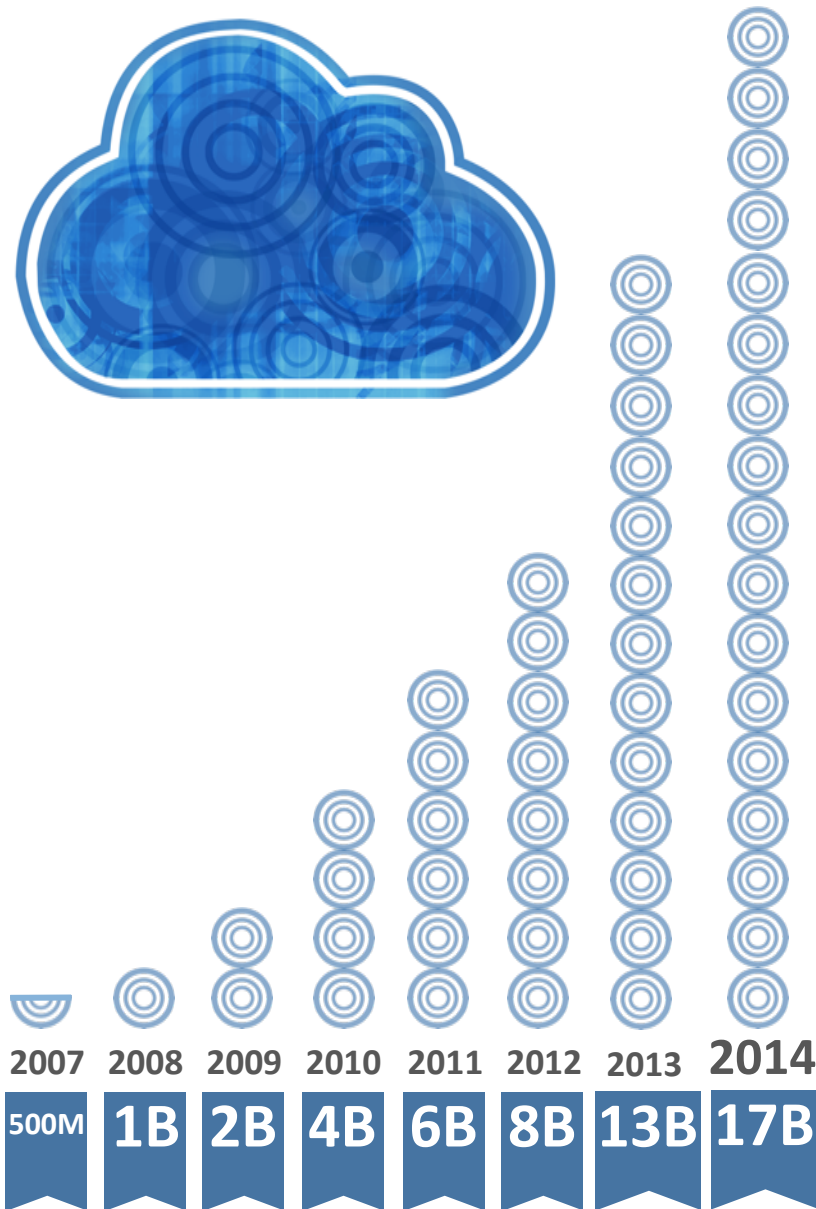


Use better & fewer  
suppliers

Use higher  
quality parts

Track what you use  
and where





Open source usage is  
**EXPLODING**

Yesterday's source  
code is now replaced with  
**OPEN SOURCE**  
components

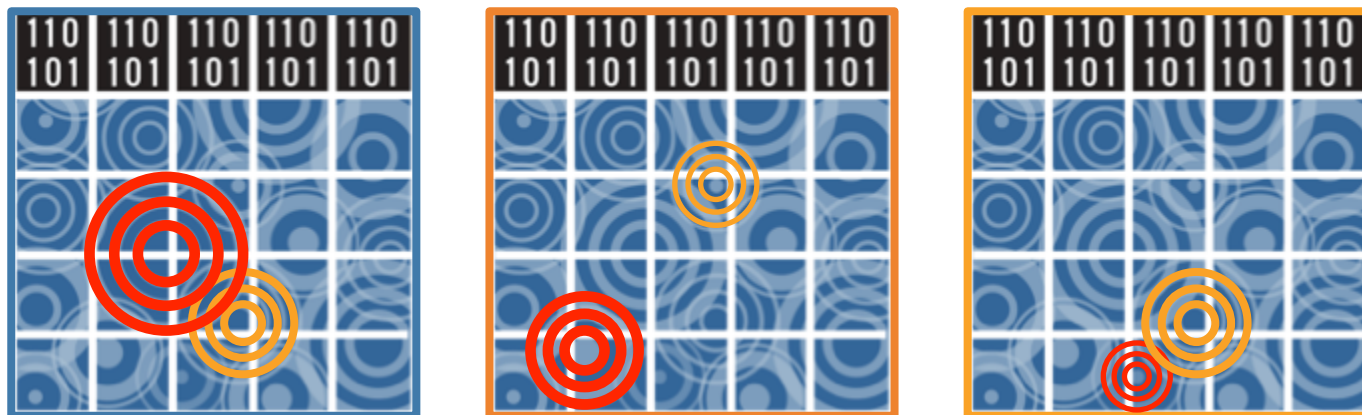


# THINK LIKE AN ATTACKER

Now that software is

## ASSEMBLED...

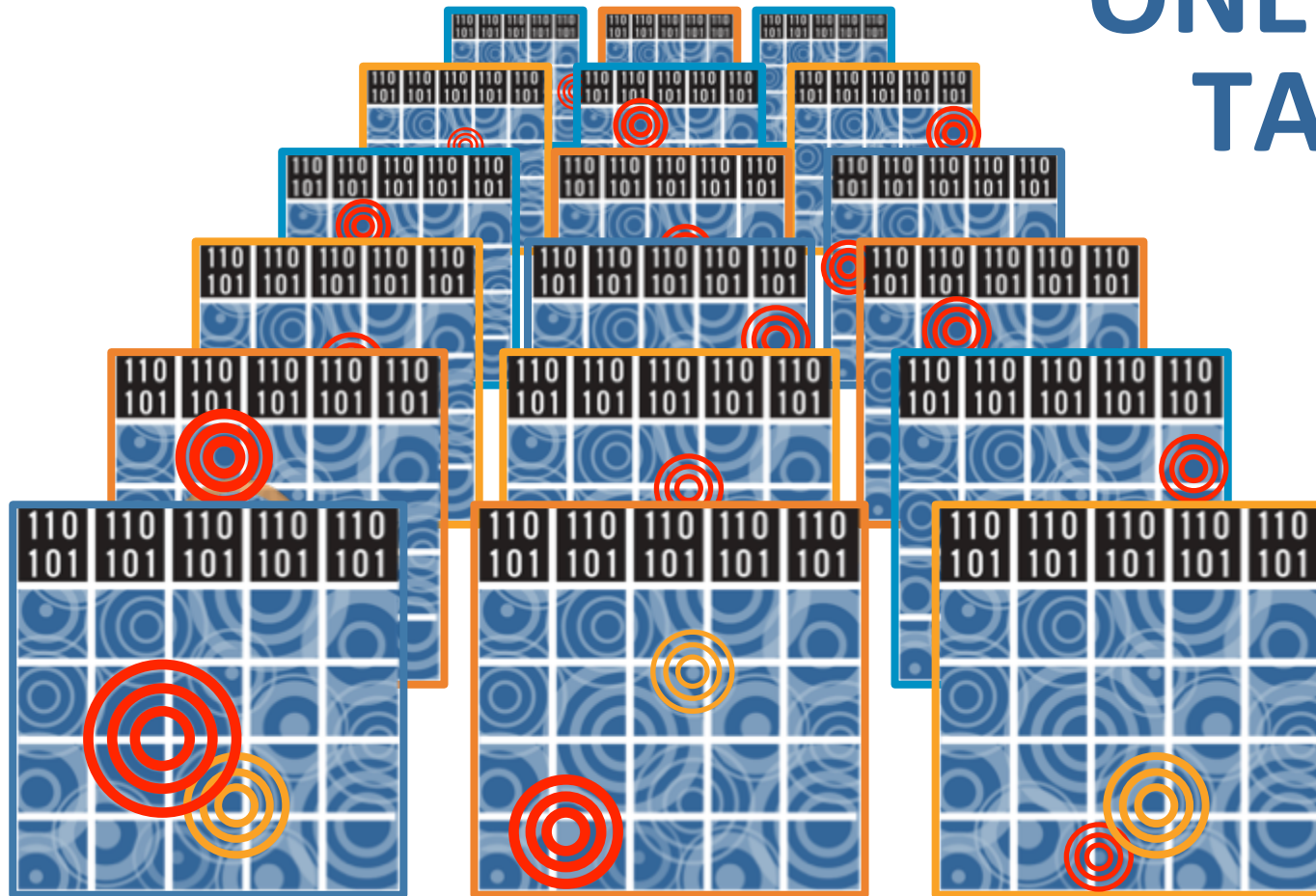
Our shared value becomes  
our shared attack surface



# THINK LIKE AN ATTACKER

One risky component,  
now affects thousands of victims

## ONE EASY TARGET





Global Bank

Software  
Provider

Software  
Provider's Customer

State University

Three-Letter  
Agency

Large Financial  
Exchange

Hundreds of Other  
Sites



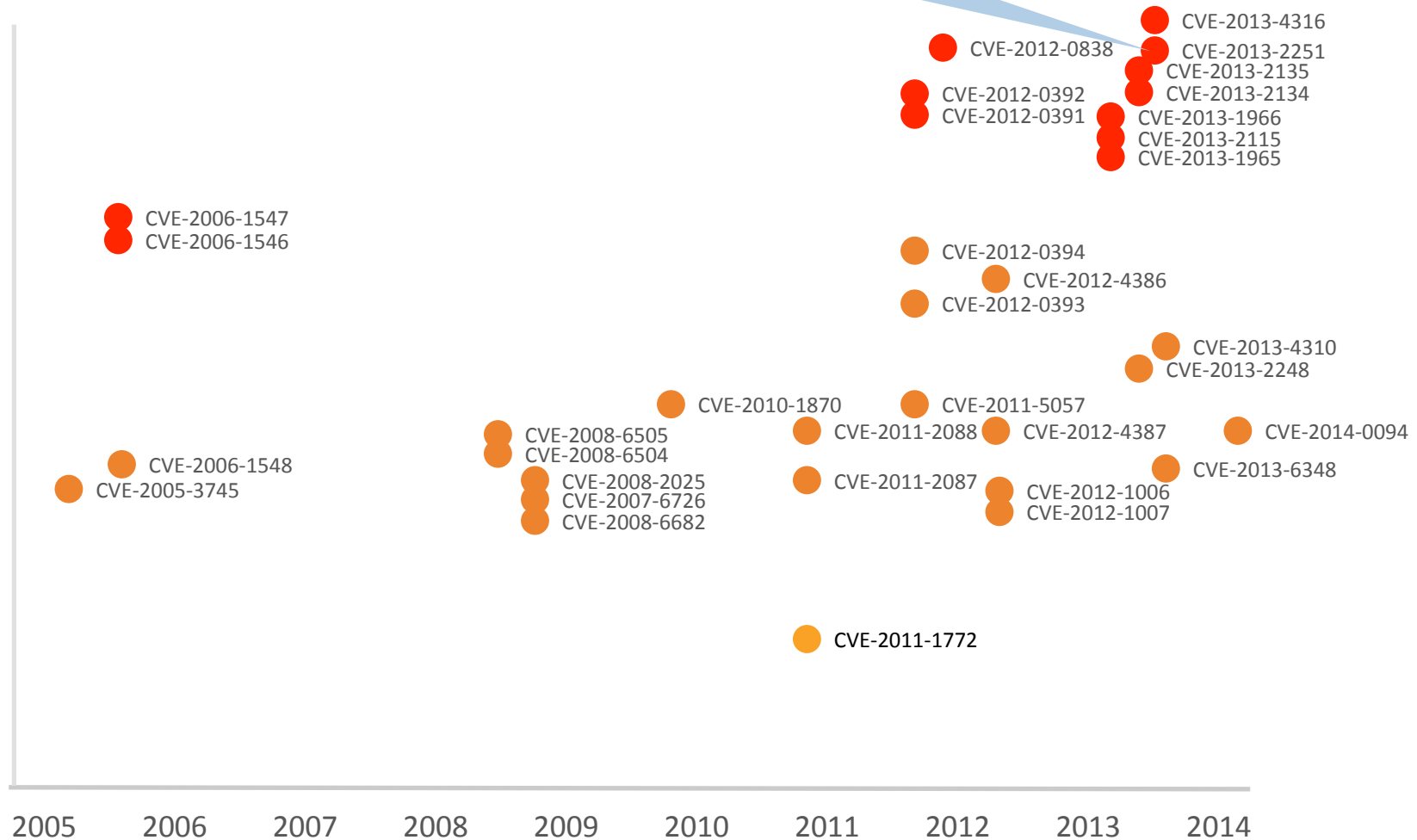
# w/many eyeballs, all bugs are??? Struts

Latent 7-11 yrs

CVSS

10.0  
9.0  
8.0  
7.0  
6.0  
5.0  
4.0  
3.0  
2.0  
1.0

2005 2006 2007 2008 2009 2010 2011 2012 2013 2014





# Houston, we have a problem

In 2014, organizations downloaded  
a version of Bouncy Castle  
with a level 10 vulnerability

**42,124**

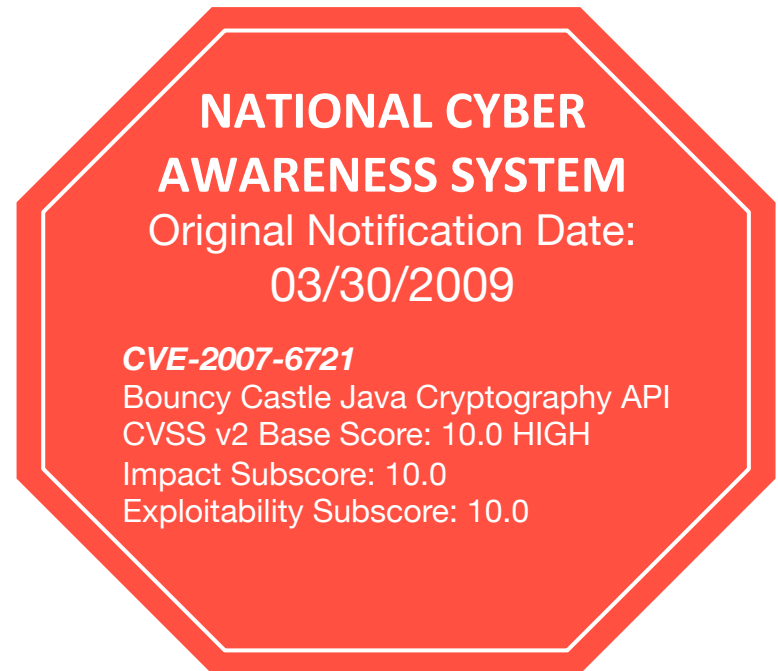
times into

**XXX,XXX**

applications...

**7+**

years after the  
vulnerability was fixed.



**BOUNCY CASTLE**

## HTTPCLIENT 3.X

### NATIONAL CYBER AWARENESS SYSTEM

Original Release Date:  
**11/04/2012**

#### **CVE-2012-5783**

Apache Commons HttpClient 3.x  
CVSS v2 Base Score: **5.8 MEDIUM**  
Impact Subscore: **4.9**  
Exploitability Subscore: **8.6**

In December 2013,

# 6,916 DIFFERENT

organizations downloaded  
a version of httpclient with broken  
ssl validation (cve-2012-5783)

# 66,824 TIMES ...

More than **ONE YEAR**  
**AFTER THE ALERT**

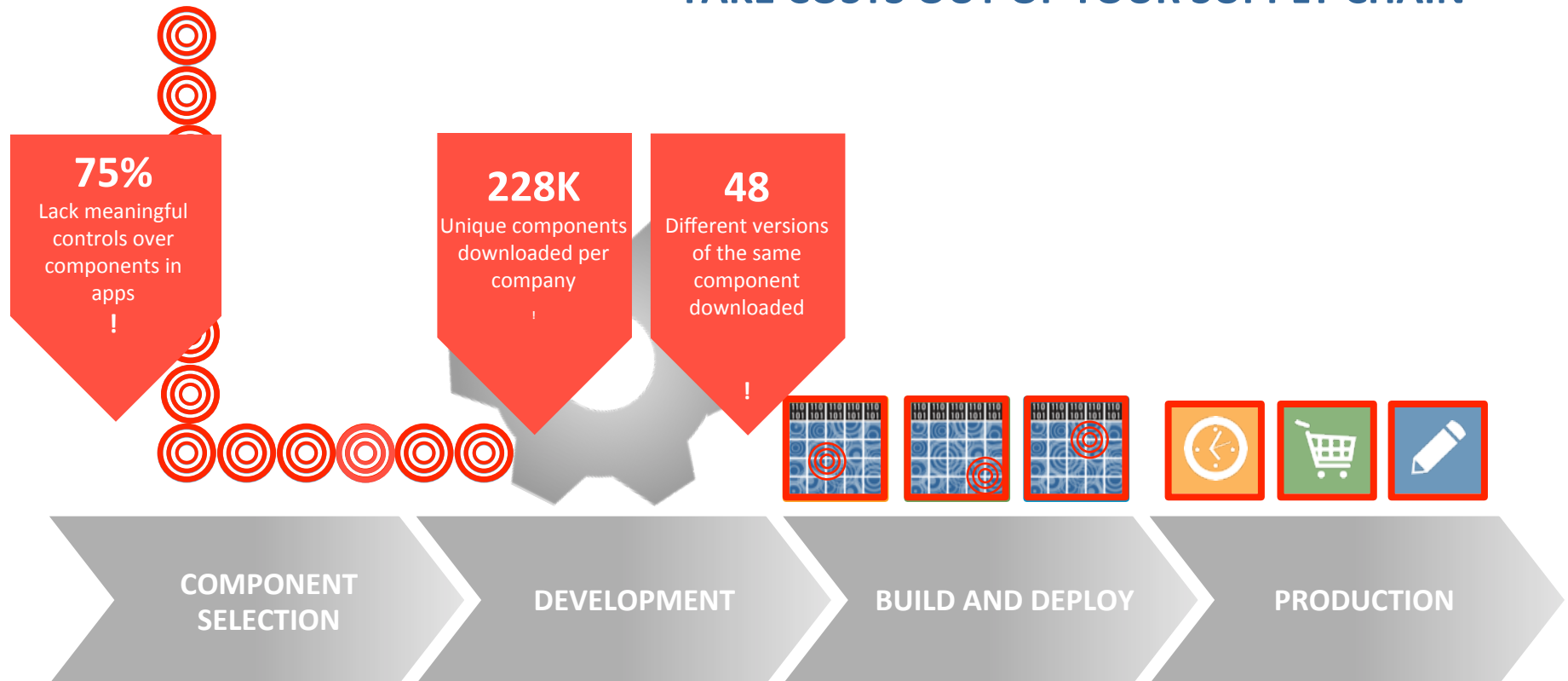




Current approaches

# AREN'T WORKING

TAKE COSTS OUT OF YOUR SUPPLY CHAIN



# COMMERCIAL RESPONSES TO OPENSLL



# OSS PROJECT INTEGRITY/HYGEINE VARIES (WILDLY)

## COLUMNS

### Almost Too Big to Fail

DAN GEER AND JOSHUA CORMAN



Dan Geer is the CISO for Is-Q-Tel and a security researcher with a quantitative bent. He has a long history with the USENIX Association, including officer positions, program committees, etc. [danjgeer.org](http://danjgeer.org)



Joshua Corman is the chief technology officer for Sonatype. Previously, Corman served as a security researcher and strategist at Akamai Technologies, The 451 Group, and IBM Internet Security Systems. A respected innovator, he co-founded Rugged Software and I Am the Cavalry to encourage new security approaches in response to the world's increasing dependence on digital infrastructure. He is also an adjunct faculty for Carnegie Mellon's Heikze College, UNO Research, and a Fellow at the Ponemon Institute. Josh received his bachelor's degree in philosophy, graduating *summa cum laude*, from the University of New Hampshire. [joshcorman@gmail.com](mailto:joshcorman@gmail.com)

Both dependence on open source and adversary activity around open source are widespread and growing, but the dynamic pattern of use requires new means to estimate if not bound the security implications. In April and May 2014, every security writer has talked about whether it is indeed true that with enough eyeballs, all bugs are shallow. We won't revisit that topic because there may be no minds left to change. Unarguably:

- Dependence on open source is growing in volume and variety.
- Adversary interest tracks installed base.
- Multiple levels of abstraction add noise to remediation needs.

We begin with two open source examples.

#### Apache Struts CVE-2013-2251, July 6, 2013 - CVSS v2 9.3

Apache Struts is one of the most popular and widely depended upon open source projects in the world. As such, when this highly exploitable vulnerability was discovered, it was promptly used to compromise large swaths of the financial services sector. While Heartbleed (see below) got full media frenzy, many affected by 2013-2251 learned of the problem from FBI victim notifications under 42 U.S.C. § 10607. The FBI-ISAC issued guidance [1] telling institutions (read, victims) to scrutinize the security of third-party and open source components throughout their life cycle of use. It is not noteworthy that an open source project could have a severe vulnerability; what is of note is that this flaw went undetected for at least seven years (if not a lot longer from WebWork 2/pre-Struts 2 code base)—an existence proof that well-vetted code still needs a backup plan.

#### OpenSSL (Heartbleed) CVE-2014-0160, April 7, 2014 - CVSS v2 5.0

The Heartbleed vulnerability in OpenSSL garnered tremendous media and attacker activity this past April. While only scored with a CVSS of 5.0, it is a % with the power of a 10<sup>2</sup> since sniffing usernames, passwords, and SSL Certificates provides stepping stones to far greater impact. In contrast to the Struts bug above, this flaw was introduced only two years prior, but it, too, went unnoticed by many eyeballs—it was found by bench analysis [2].

#### Dependence on Open Source Is Growing

Sonatype, home to author Corman, serves as custodian to Central Repository, the largest parts warehouse in the world for open source components. At the macro level, open source consumption is exploding in Web applications, mobility, cloud, etc., driven in part by increasingly favorable economics. Even (risk averse, highly regulated) government and financial sectors, which previously resisted "code of unknown origin/quality/security," have begun relaxing their resistance. According to both Gartner surveys and Sonatype application analysis, 90+% of modern applications are not so much written as assembled from third-party building blocks. It is the open source building blocks that are taking the field, and not just for commodity applications (see Figure 1).

41% ever get fixed  
390 days  
224 days (CVSS 10s)

[https://www.usenix.org/system/files/login/articles/15\\_geer\\_0.pdf](https://www.usenix.org/system/files/login/articles/15_geer_0.pdf)



7 Projects (Suppliers)

0 Avg Versions (Parts) .

1 Parts Warehouse(s) .

Debt/Risks

Struts2

Bouncy Castle

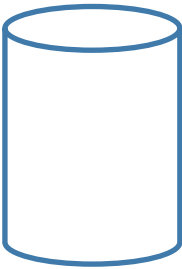
httpClient

Log4j

Logback

Tomcat

XStream



0 % Vulnerable CVSS 7+

7 Projects (Suppliers)

3 Avg Versions (Parts)

1 Parts Warehouse(s)

Debt/Risks

Struts2



Bouncy Castle



httpClient



Log4j



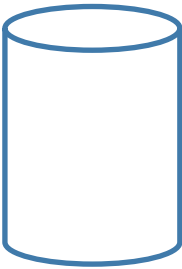
Logback



Tomcat



XStream



0 % Vulnerable CVSS 7+

7 Projects (Suppliers)

4 Avg Versions (Parts)

1 Parts Warehouse(s)

Debt/Risks

Struts2

Bouncy Castle

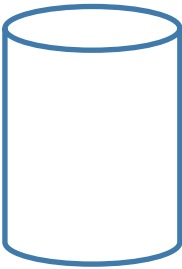
httpClient

Log4j

Logback

Tomcat

XStream



25 % Vulnerable CVSS 7+



7 Projects (Suppliers)

4 Avg Versions (Parts)

1 Parts Warehouse(s)

Debt/Risks

Struts2

Bouncy Castle

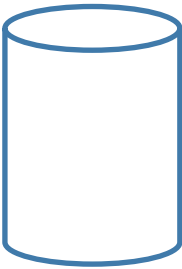
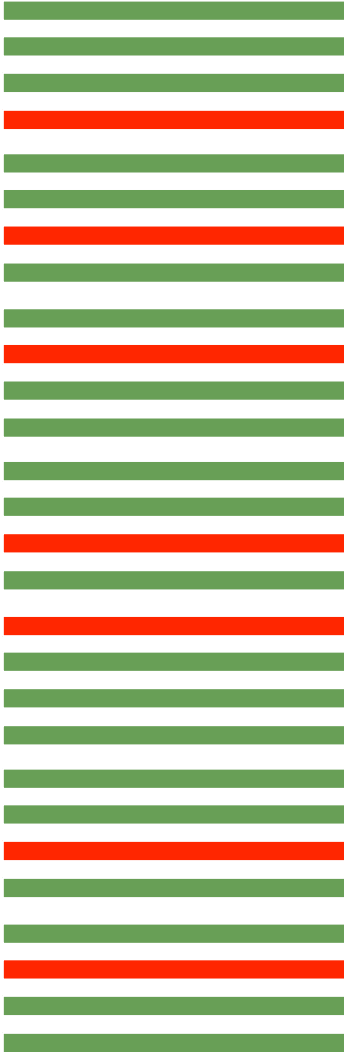
httpClient

Log4j

Logback

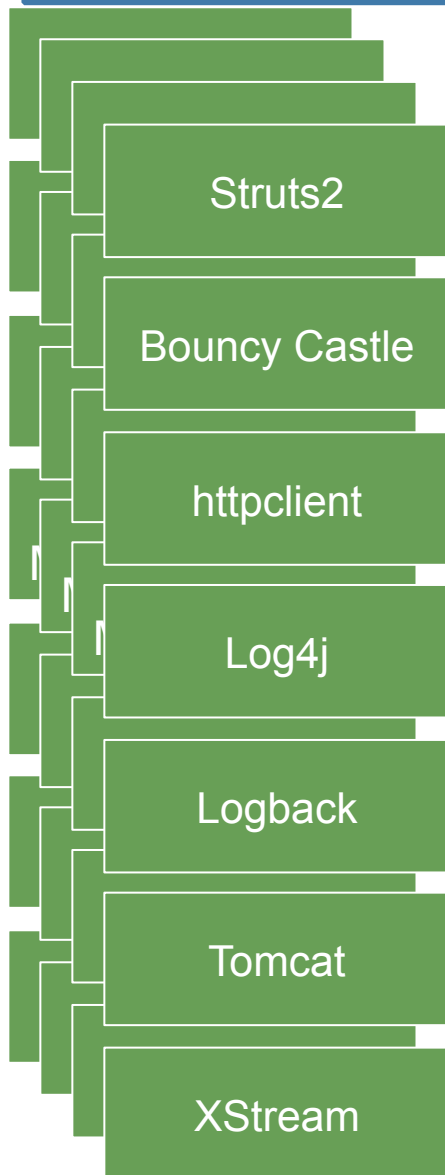
Tomcat

XStream

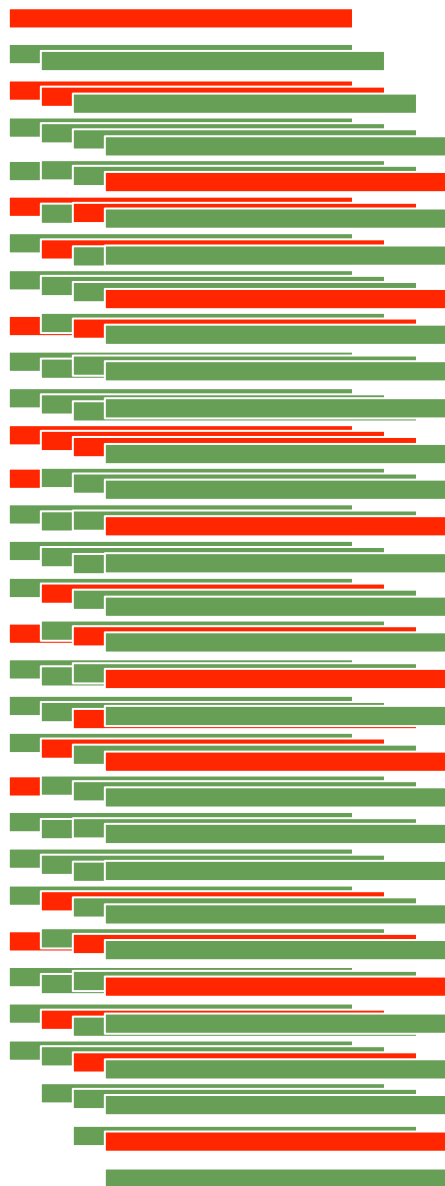


25 % Vulnerable CVSS 7+

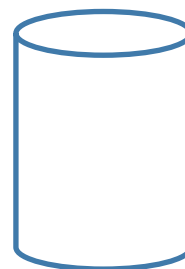
28 Projects (Suppliers)



4 Avg Versions (Parts)



1 Parts Warehouse(s)

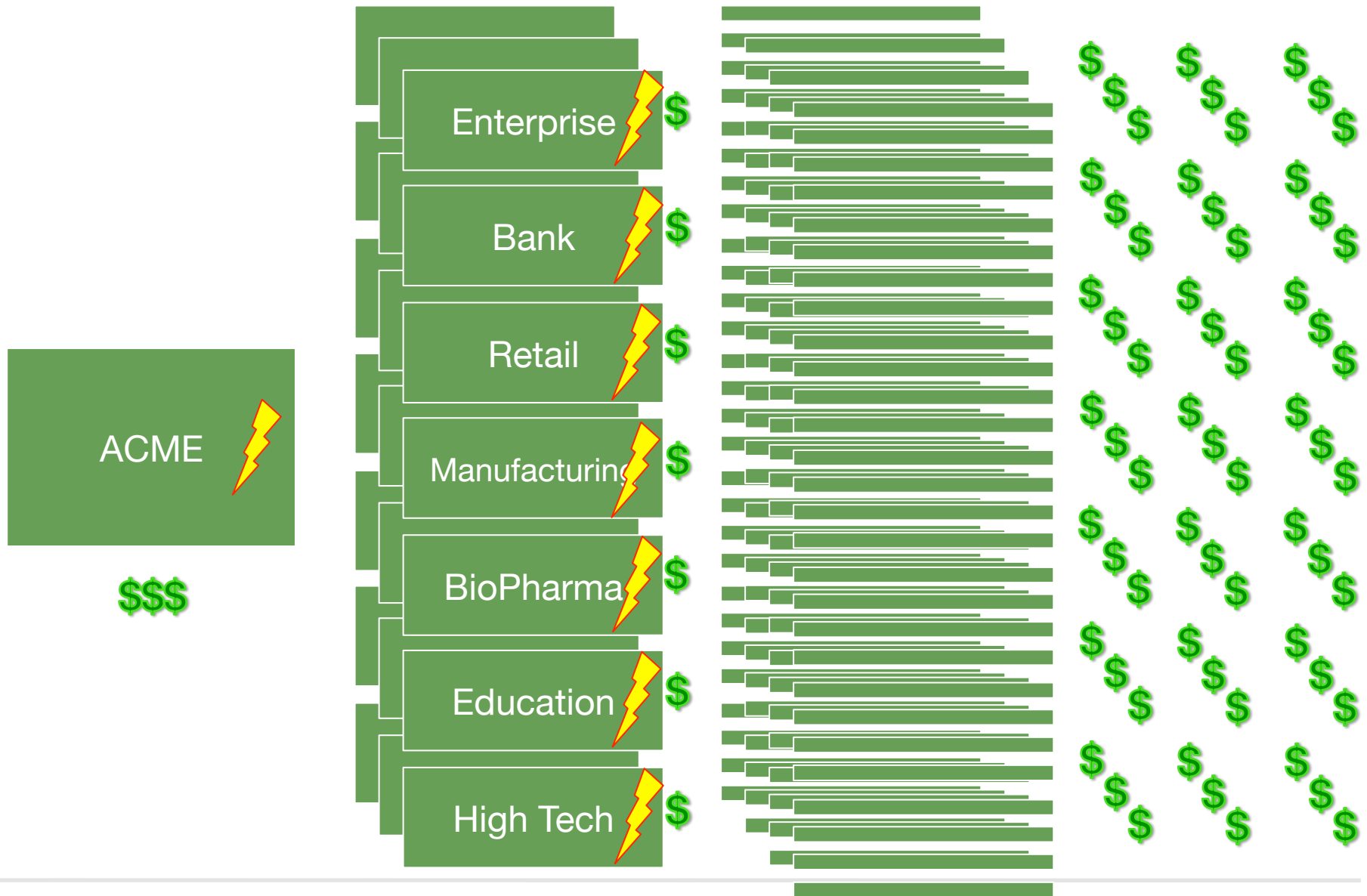


Debt/Risks



25 % Vulnerable CVSS 7+

# COTS: TRUE COSTS (& LEAST COST AVOIDERS)





SEARCH 

U.S. REPRESENTATIVE

**ED ROYCE**

39TH DISTRICT OF CALIFORNIA

HOME

ABOUT ED

SERVICES

NEWS

ISSUES

LEGISLATION

39TH DISTRICT

STUDENTS

CONTACT ED



E-NEWSLETTER SIGN UP

News

Columns and Opinions



Ed in the News



Speeches &amp; Statements



Press Releases



Multimedia



Photos



## PRESS RELEASES

### Reps. Royce, Jenkins to Shore Up Security of Government Used Software

Washington, Dec 4, 2014 | [Saat Alety](#) (202-225-4111) | [0 comments](#)

Today, U.S. Representatives Ed Royce (R-CA) and Lynn Jenkins (R-KS) introduced H.R. 5793, the "Cyber Supply Chain Management and Transparency Act of 2014." The legislation will ensure all contractors of software, firmware or products to the federal government provide the procuring agency with a bill of materials of all third party and open source components used, and demonstrate that those component versions have no known vulnerabilities.

"As a house is only as strong as its foundation, it's no wonder cyber attacks are on the rise with reports showing 71 percent of software contains components with critical vulnerabilities," said Rep. Royce. "This bill protects our nation's cyber infrastructure by ensuring the building blocks that make it up are secure and uncompromised."

"I have voiced concerns to the government agencies in charge of healthcare.gov that our nation's cyber infrastructure was vulnerable and not secure," said Rep. Jenkins. "But the problem is not limited to one website; the entire federal government lacks guidelines for website security. This vital legislation will put the appropriate checks and balances in place to ensure that the government has the tools it needs to create a more sound and secure system for taxpayers."

## **H.R. 5793 “Cyber Supply Chain Management and Transparency Act of 2014”**

### **Elegant Procurement Trio**

#### **1) Ingredients:**

**Anything sold to \$PROCURING\_ENTITY must provide a Bill of Materials of 3<sup>rd</sup> Party and Open Source Components (along with their Versions)**

#### **2) Hygiene & Avoidable Risk:**

**...and cannot use known vulnerable components for which a less vulnerable component is available (without a written and compelling justification accepted by \$PROCURING\_ENTITY)**

#### **3) Remediation:**

**...and must be patchable/updateable – as new vulnerabilities will inevitably be revealed**

## PROCUREMENT TRIO + BOUNCY CASTLE

### NATIONAL CYBER AWARENESS SYSTEM

Original Notification Date:  
**03/30/2009**

#### ***CVE-2007-6721***

Bouncy Castle Java Cryptography API  
CVSS v2 Base Score: **10.0 HIGH**  
Impact Subscore: **10.0**  
Exploitability Subscore: **10.0**

In 2013, **4,000**  
organizations downloaded  
a version of Bouncy Castle  
with a level 10 vulnerability  
**20,000 TIMES ...**  
Into **XXX,XXX** Applications...

**SEVEN YEARS**  
after the vulnerability was fixed



# Two little words

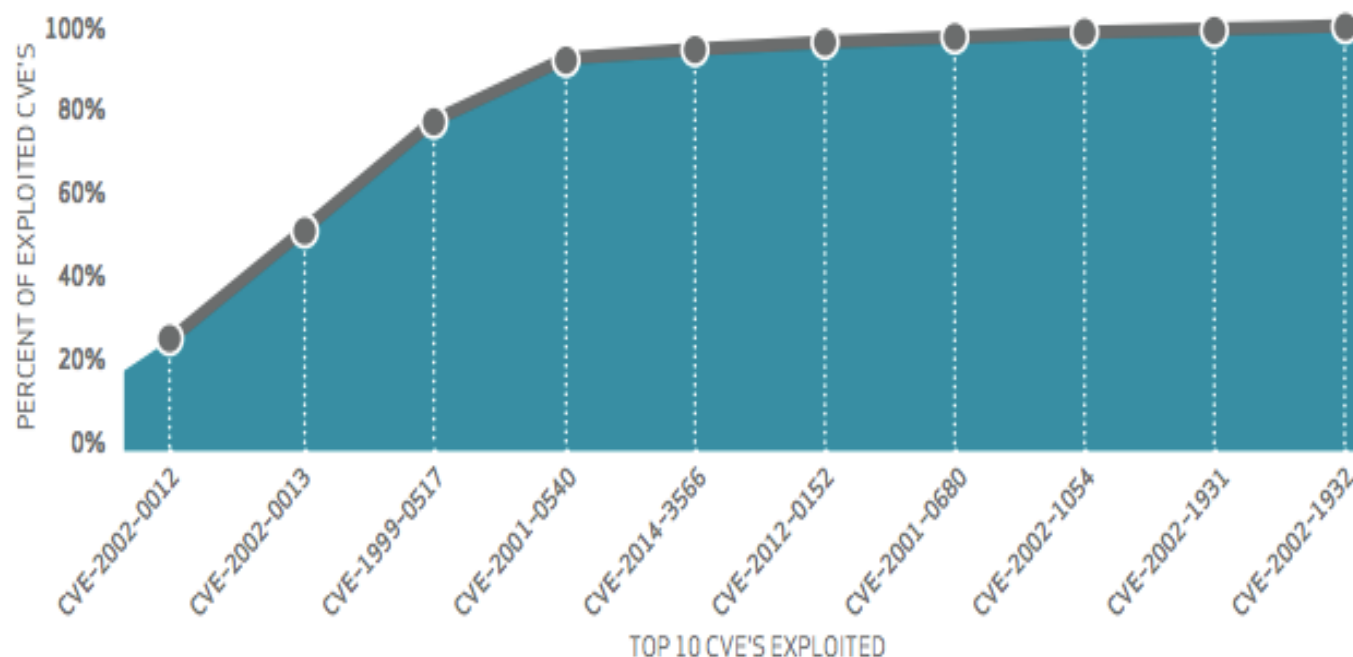
# Known Vulnerabilities

# Hot off the presses 2015 VZ DBIR

## NOT ALL CVEs ARE CREATED EQUAL.

If we look at the frequency of exploitation in Figure 11, we see a much different picture than what's shown by the raw vulnerability count of Figure 12. **Ten CVEs account for almost 97%** of the exploits observed in 2014. While that's a pretty amazing statistic, don't be lulled into thinking you've found an easy way out of the vulnerability remediation rodeo. Prioritization will definitely help from a risk-cutting perspective, but beyond the top 10 are 7 million other exploited vulnerabilities that may need to be ridden down. And therein, of course, lies the challenge; once the "mega-vulns" are roped in (assuming you could identify them ahead of time), how do you approach addressing the rest of the horde in an orderly, comprehensive, and continuous manner over time?

*About half of the CVEs exploited in 2014 went from publish to pwn in less than a month.*



**Figure 11.**

Cumulative percentage of exploited vulnerabilities by top 10 CVEs



- 1) Fewer/Better Suppliers**
- 2) Better Supply from High Quality Suppliers**
- 3) Traceability and Visibility throughout manufacturing**

**1) Less Unplanned /Unscheduled  
Work (and painful Context  
Switching)**

**2) Faster MTTI/MTTR when things  
do go wrong**

**> 30% Boost**



# **Rugged Line-Up**







# Continuous Acceleration

with Software Supply Chain Rigor

Joshua Corman  
@joshcorman





*Please*

**Remember to  
rate this session**

*Thank you!*



# Thanks !