RSAConference2019
San Francisco | March 4 – 8 | Moscone Center

SESSION ID: ASD-T09

Release Your Inner DevSecOp

Shannon Lietz
Director, Intuit

James Wickett
Head of Research, Signal Sciences

Got a good story? We're writing a book

I'm are writing a book along with James Wickett, Ernest Mueller and John Willis on DevSecOps.

We are looking for stories of DevSecOps transformations, journeys, successes and failures.

book@devsecops.org

shannon lietz <@devsecops />

MY pseudo JOURNEY LINE...

WHAT MAKES ME HUMAN...

HOW I SPEND MY DAYS...

James Wickett (@wickett)

Head of Research Signal Sciences

Instructor, LinkedIn Learning

- Six courses on DevOps, DevSecOps, CI/CD, Security Automation

DevOps Days Austin Organizer

- Come to Texas y'all!

Get the slides now:

james@signalsciences.com

Paying $1,500 to browse Twitter and hang out on Slack

Half-listening to Conference Talks

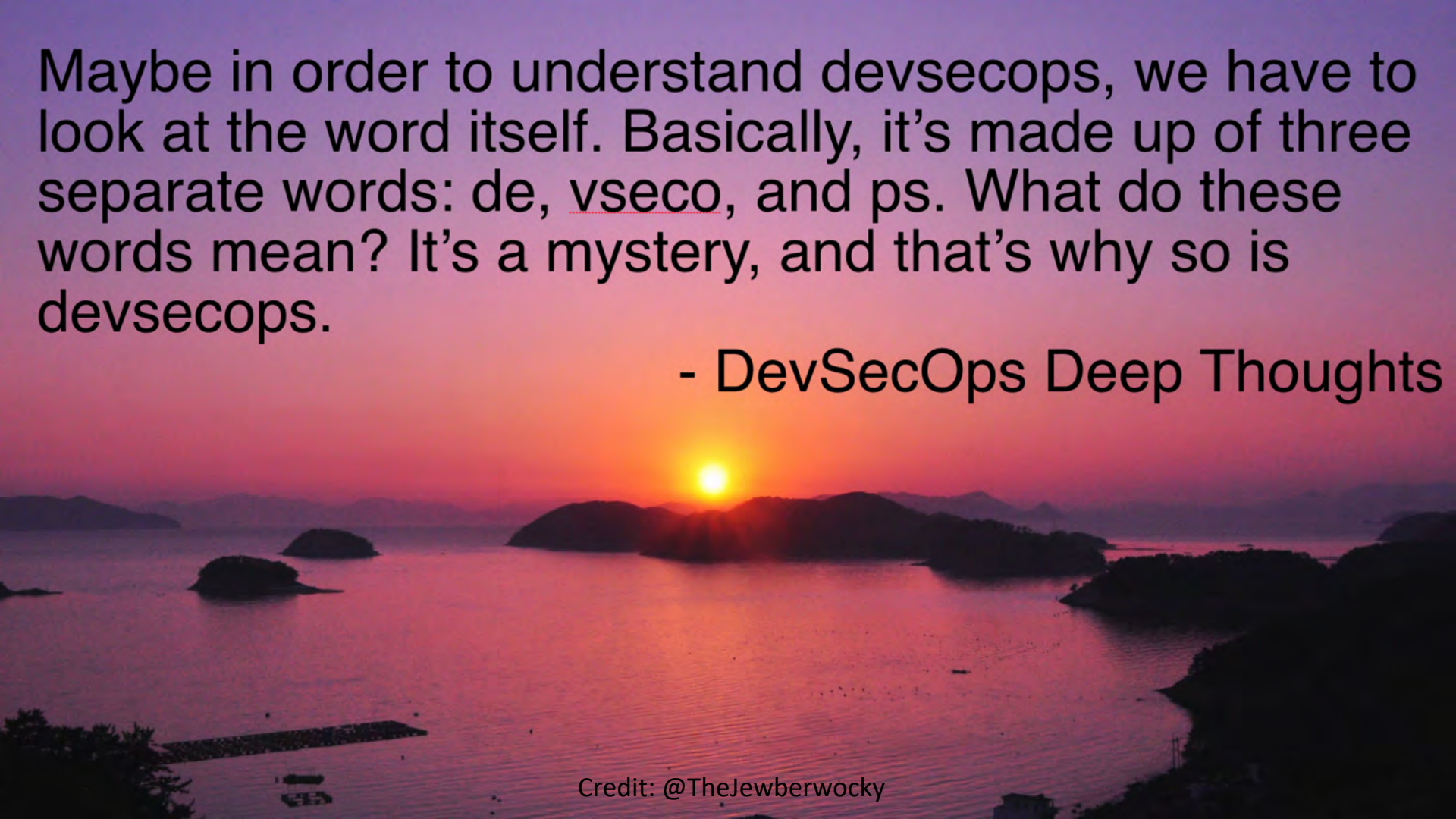
In Depth

O RLY?

@ThePracticalDev

DevSecOp

[divh-sek-op]


Maybe in order to understand devsecops, we have to look at the word itself. Basically, it's made up of three separate words: de, vseco, and ps. What do these words mean? It's a mystery, and that's why so is devsecops.
- DevSecOps Deep Thoughts
Credit: @TheJewberwocky

An inclusive person participating in the movement of security into devops.

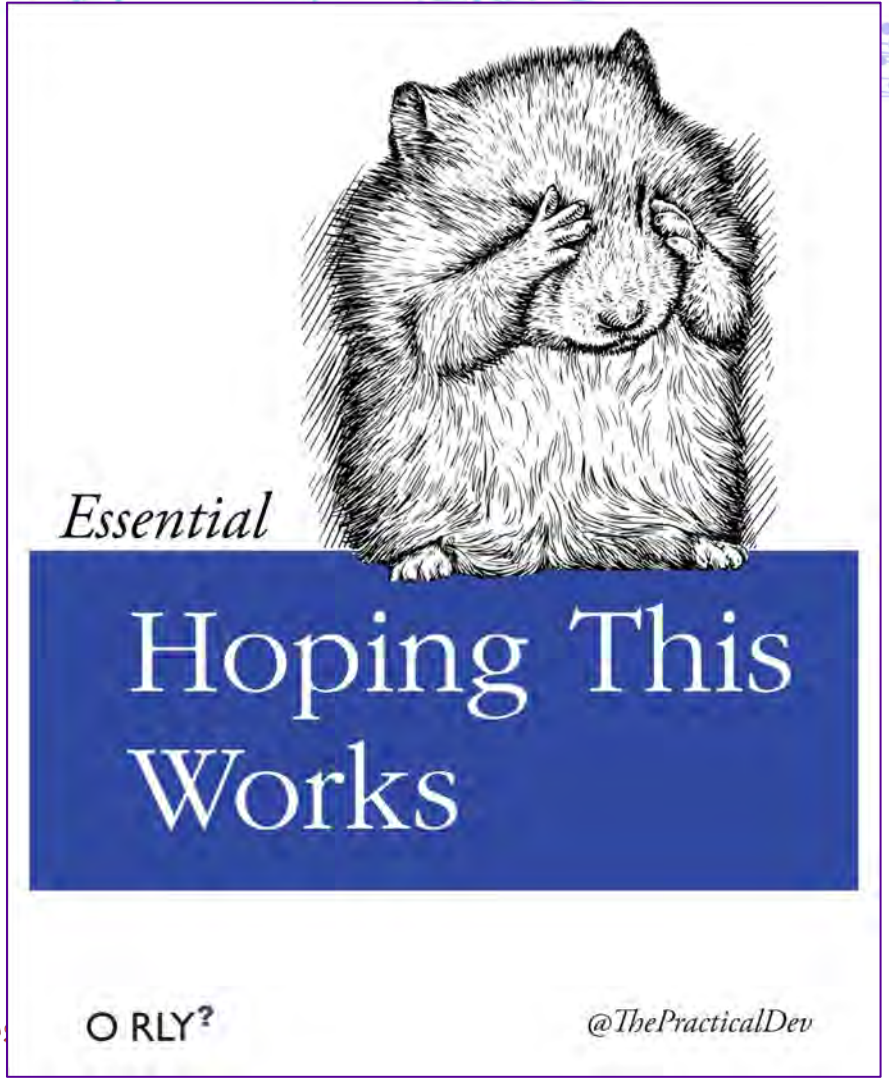
…not a tool
…not a security CI/CD pipeline
…not a CI/CD pipeline with security in it
…can't be bought on expo floor

It can be you.

The 10-fold Path of DevSecOps

The Journey

1. See the new world
2. Recognize place in value chain
3. Know Agile and DevOps
4. Live out Bi-directional Empathy
5. Do Security for Developers' Benefit
6. Operationalize DevSecOps
7. Make Security as Normal
8. Track Adversary Interest
9. Create Security Observability
10. Build the Future


Essential
Hoping This Works
O RLY?
@ThePracticalDev

1.See the New World


Justin Garrison
@rothgar
Following
The new OSI model is much easier to understand
Software
Software
Software
Software
Software
Software
Software
11:22 AM - 18 Jul 2017
2,754 Retweets 3,895 Likes
93
2.8K
3.9K

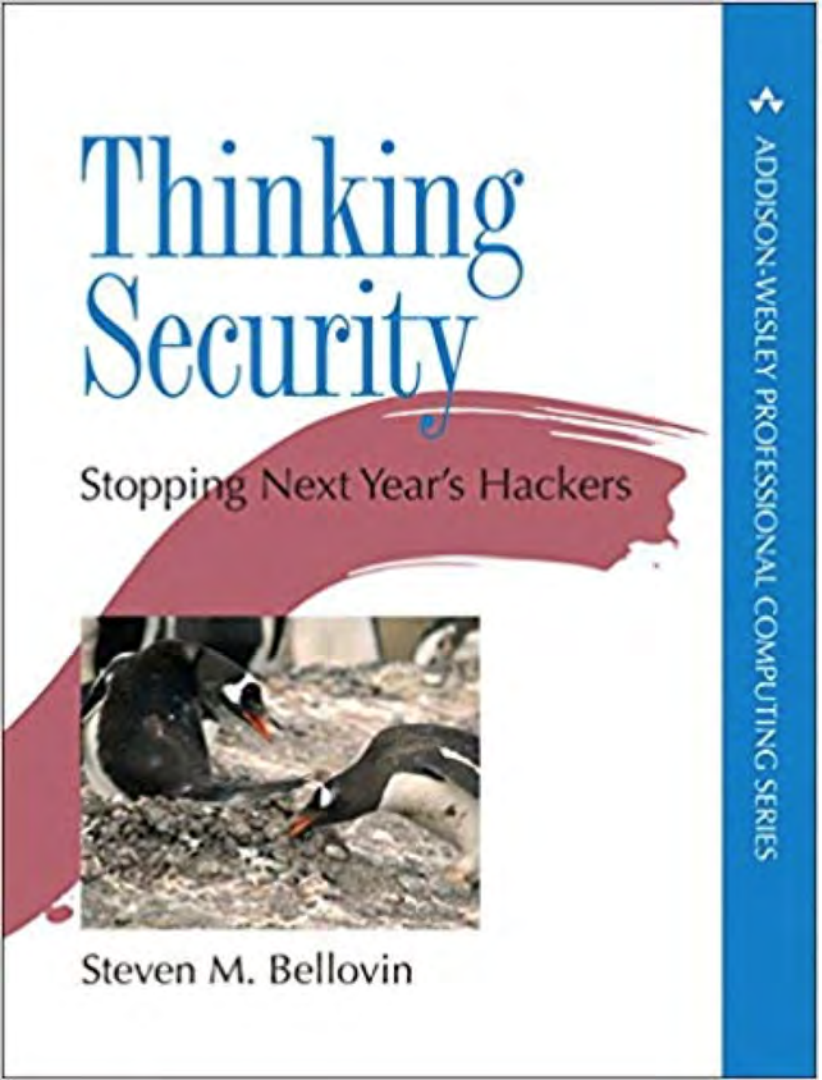
3 Major Movements

1. Waterfall -> *Agile* -> **DevOps**
2. *Monolith* -> **Microservices**
3. *Datacenter* -> **Cloud**

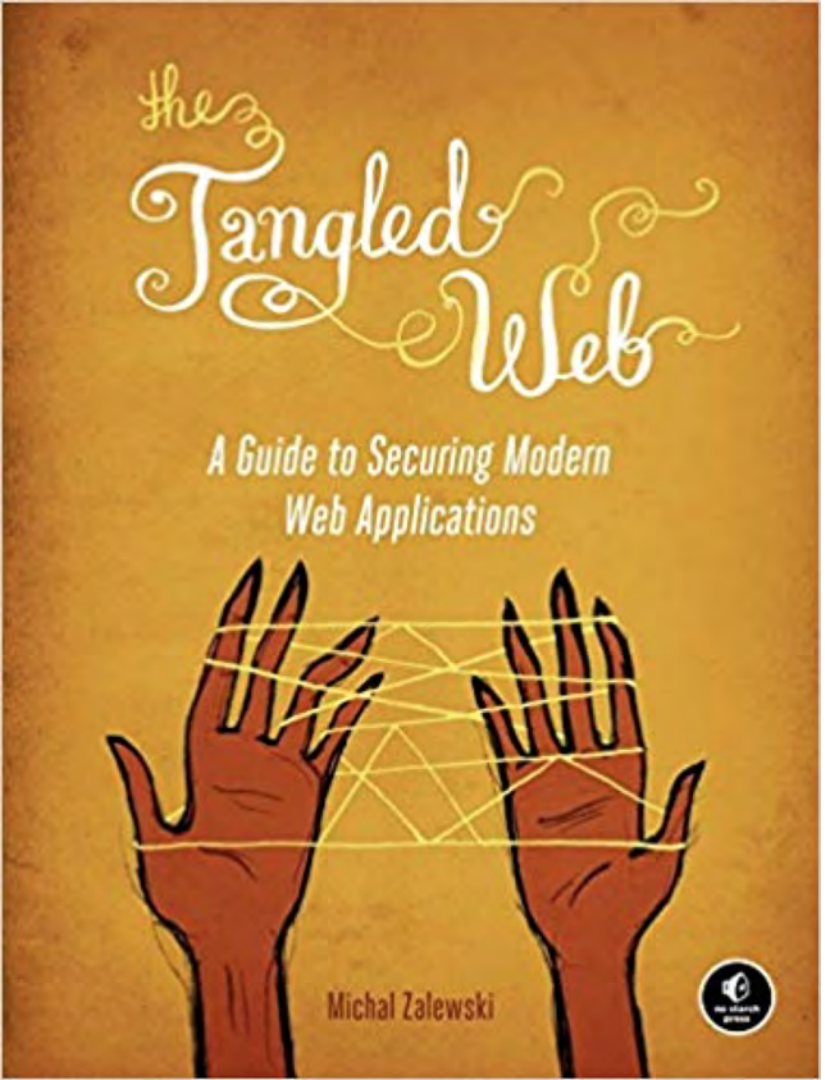
The Developer Revolt is Real

While engineering teams are busy deploying leading-edge technologies, security teams are still focused on fighting yesterday's battles.

- SANS 2018 DevSecOps Survey

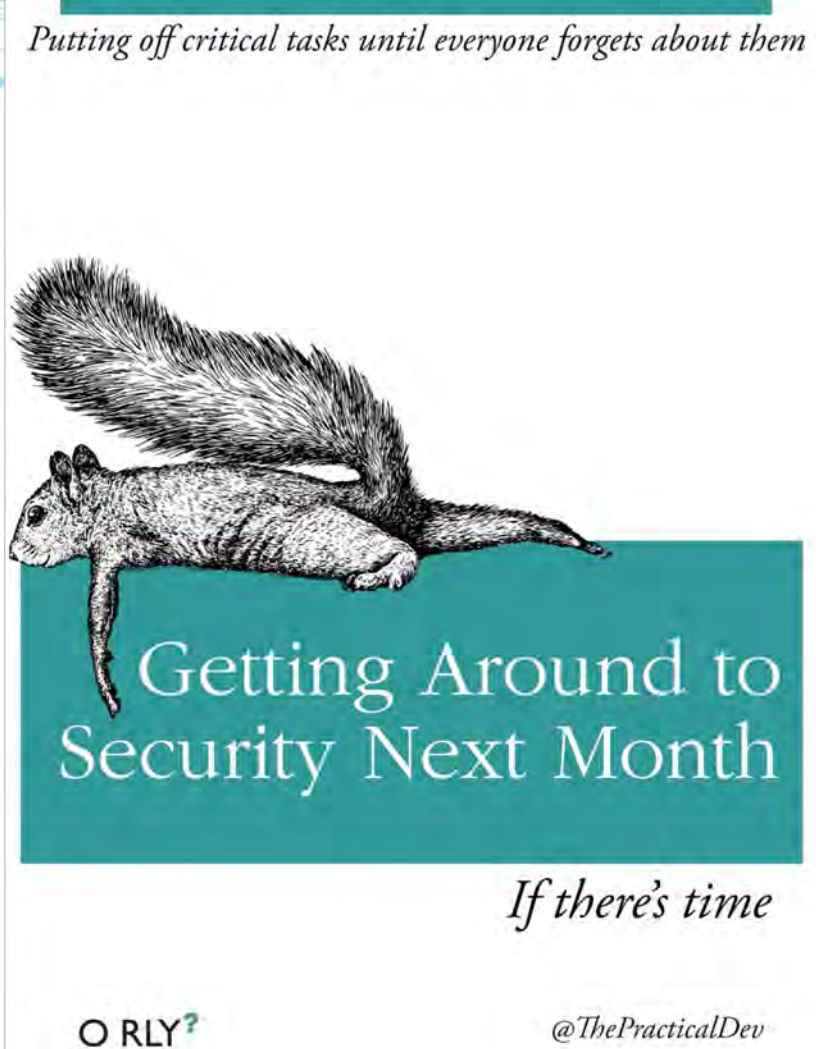


Companies are spending a great deal on security, but we read of massive computer-related attacks. Clearly something is wrong. The root of the problem is twofold: we're protecting the wrong things, and we're hurting productivity in the process.

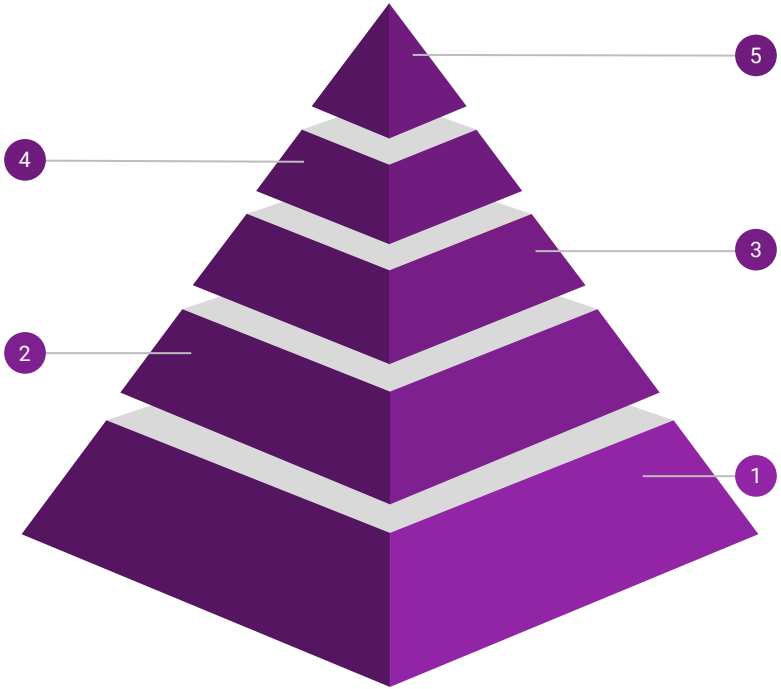


[Security by risk assessment] introduces a dangerous fallacy: that structured inadequacy is almost as good as adequacy and that underfunded security efforts plus risk management are about as good as properly funded security work

Meanwhile, Devs be like...



Security Hierarchy of Needs



Encryption
Prevent data loss by leveraging the benefits of encryption for selectively eliminating injection attacks and transparently protecting access to data

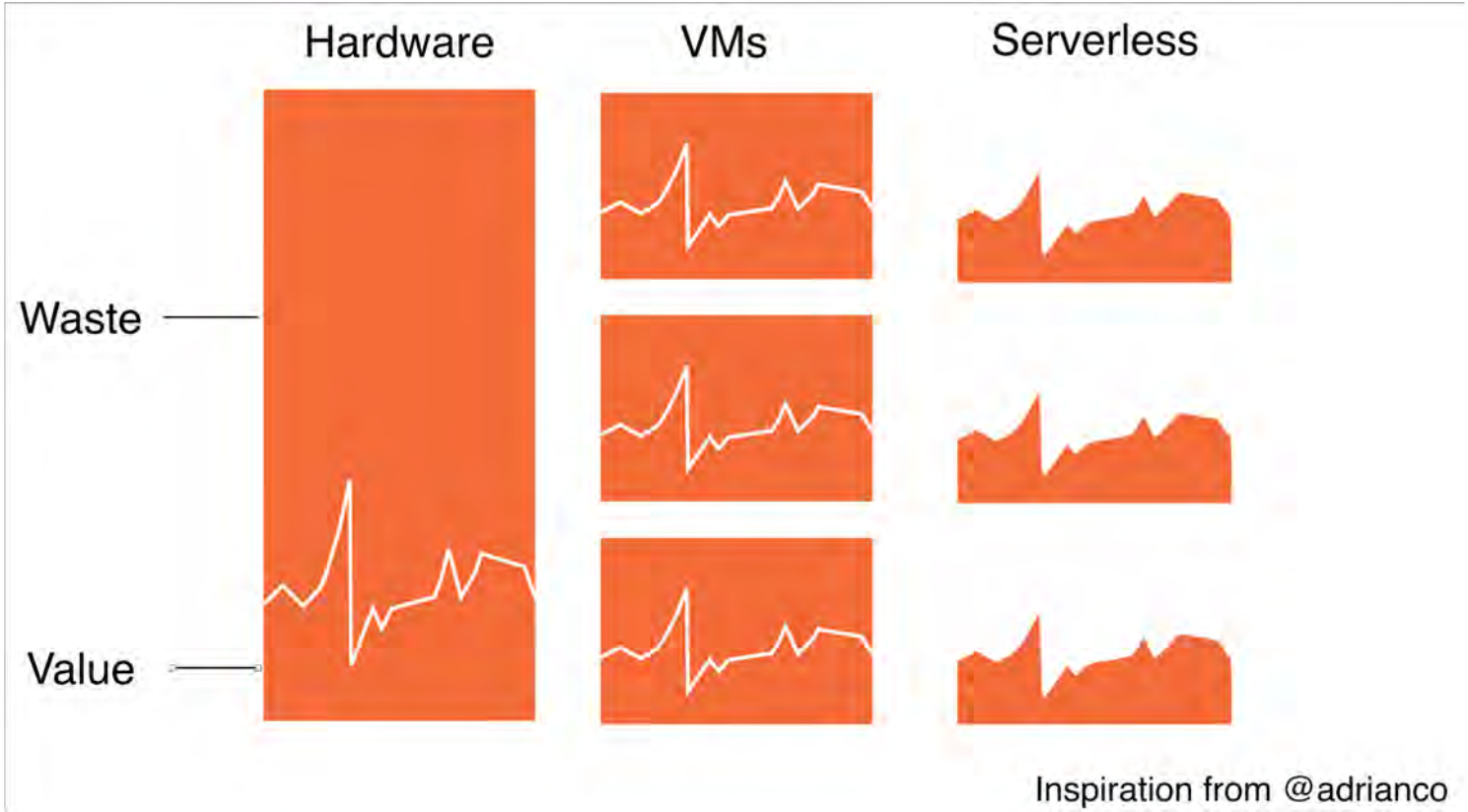
Logging
Implement event logging and audit trails to ensure visibility, detection and response for abuse cases

Zoning & Containment
Establish trust boundaries and control blast radius to ensure workload safety, viability and resiliency

Authentication
Focus on securing access through identity verification and two-factor authentication mechanisms to establish trusted usage

Asset Management
Maintain a manifest of all components, configurations and attribution data to ensure proper asset handling, lifecycle management and to speed up response events

2. Recognize Place in the Value Chain

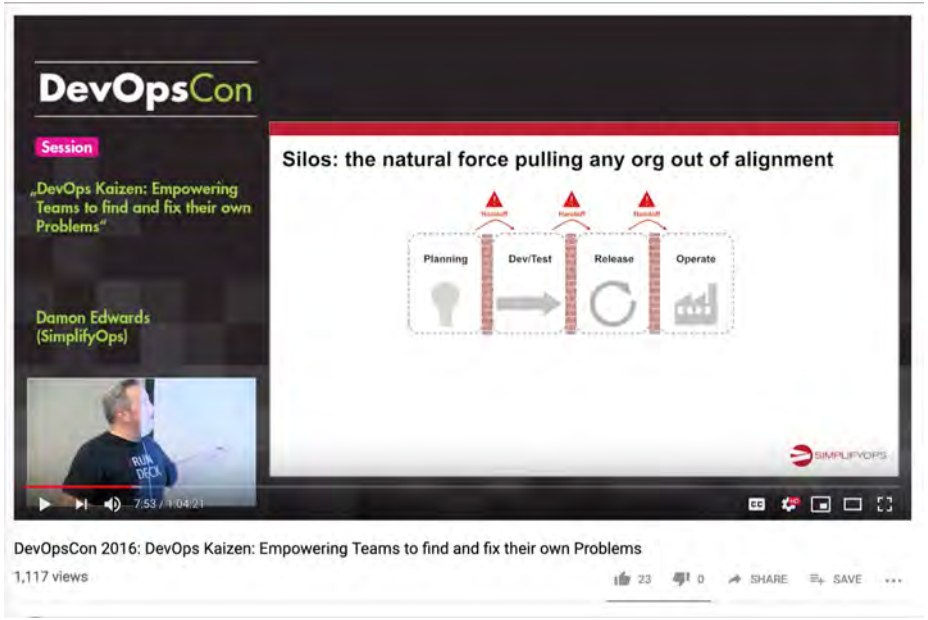

Hardware
VMs
Serverless
Waste
Value
Inspiration from @adrianco

The Inequitable Distribution of Labor

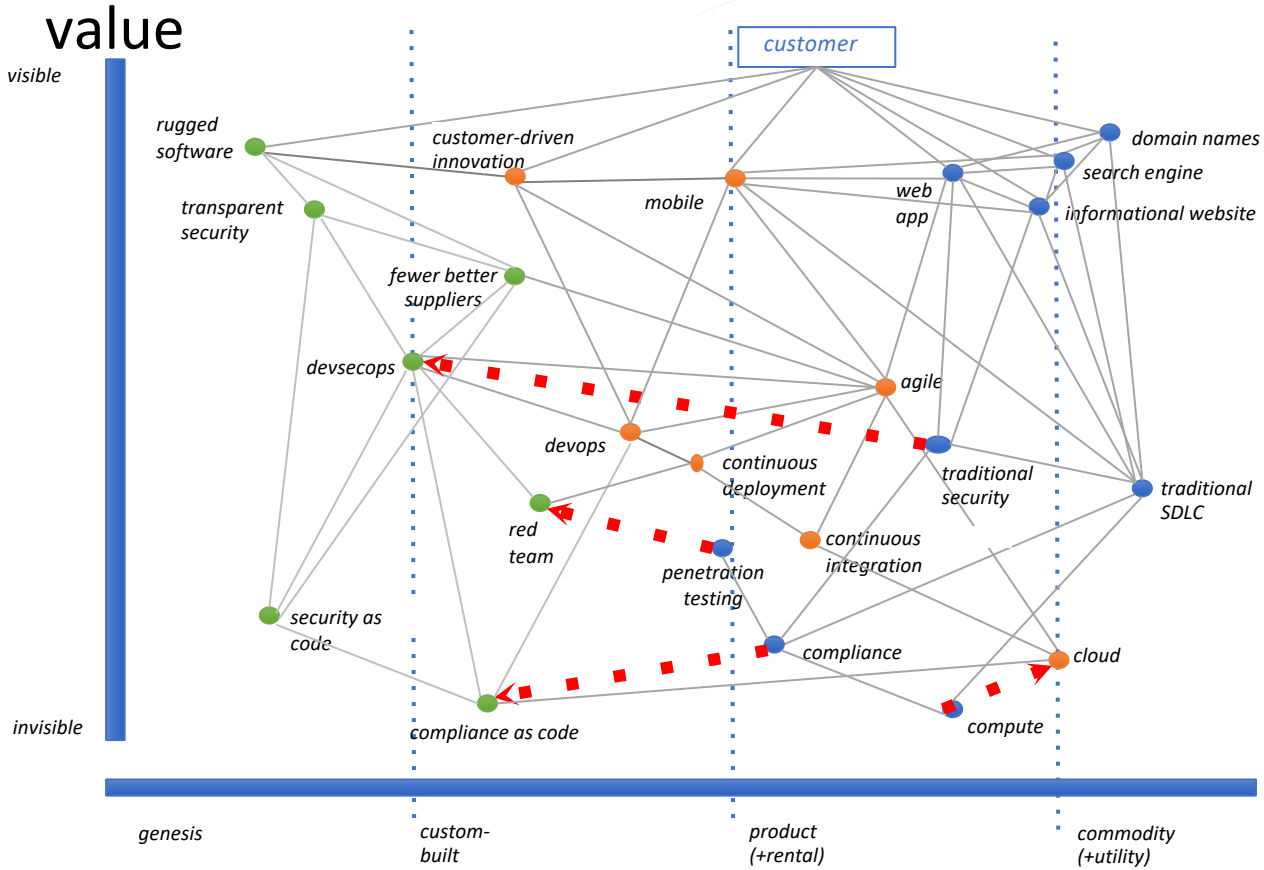
10:1
Dev:Ops

100:10:1
Dev:Ops:Sec

Value Stream Mapping



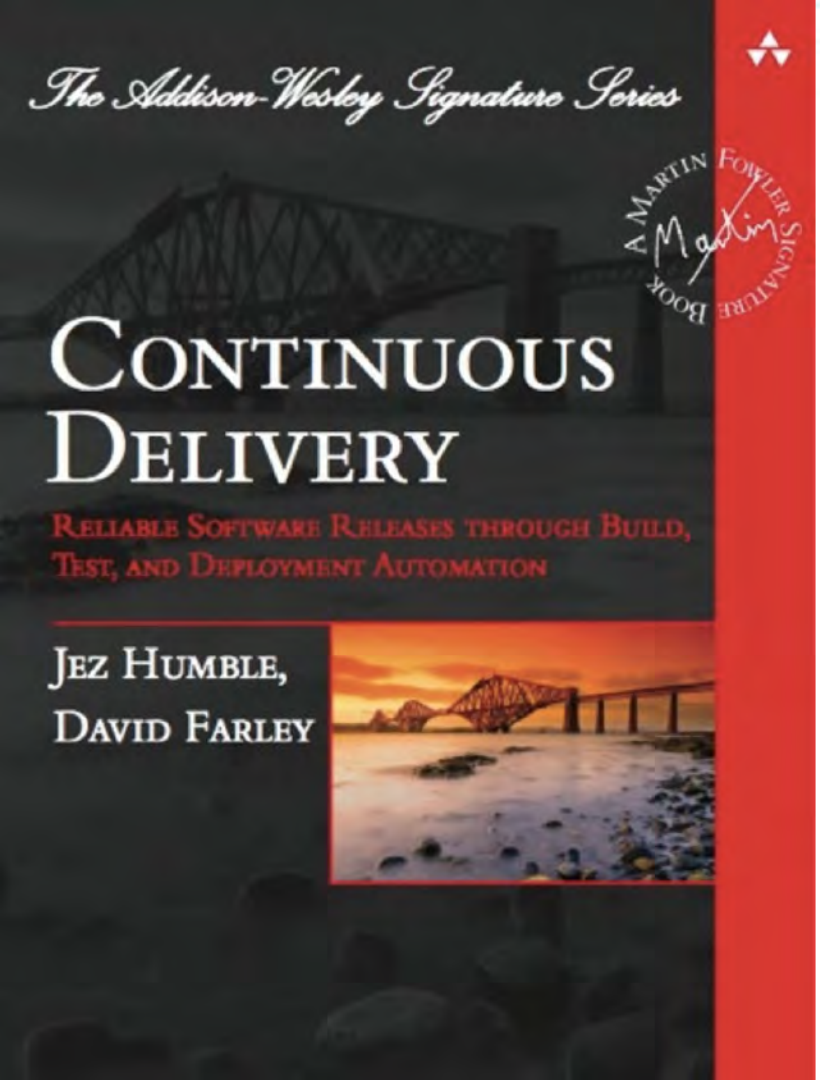
@DamonEdwards
https://www.youtube.com/watch?v=gutKcKjdwRQ



- commodity bound
- growth
- emerging

Catching up takes commitment

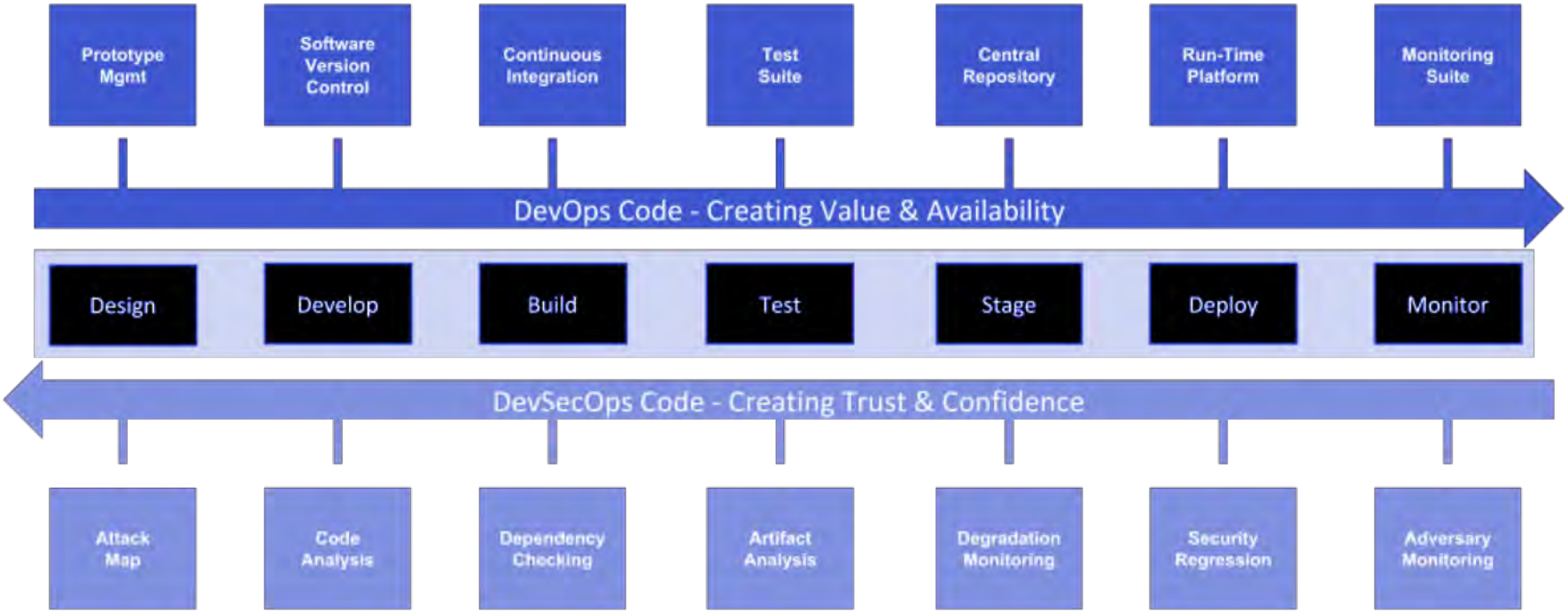
3. Know Agile and DevOps



Continuous Delivery is how little you can deploy at a time

We optimized for cycle time—the time from code commit to production

Roughly 15,000 deploys in the last 3.5 yrs


Prototype Mgmt
Software Version Control
Continuous Integration
Test Suite
Central Repository
Run-Time Platform
Monitoring Suite
DevOps Code - Creating Value & Availability
Design
Develop
Build
Test
Stage
Deploy
Monitor
DevSecOps Code - Creating Trust & Confidence
Attack Map
Code Analysis
Dependency Checking
Artifact Analysis
Degradation Monitoring
Security Regression
Adversary Monitoring

You might be, if...

- Security is writing code and identify with developers
- You have completed value stream mapping and/or wardley map
- You have started logging for security feedback within your CI/CD Pipeline
- Read The Phoenix Project

RSAConference2019

Are you ready to advance?

4. Live out Bi-directional Empathy

Culture is the most important aspect to devops succeeding in the enterprise

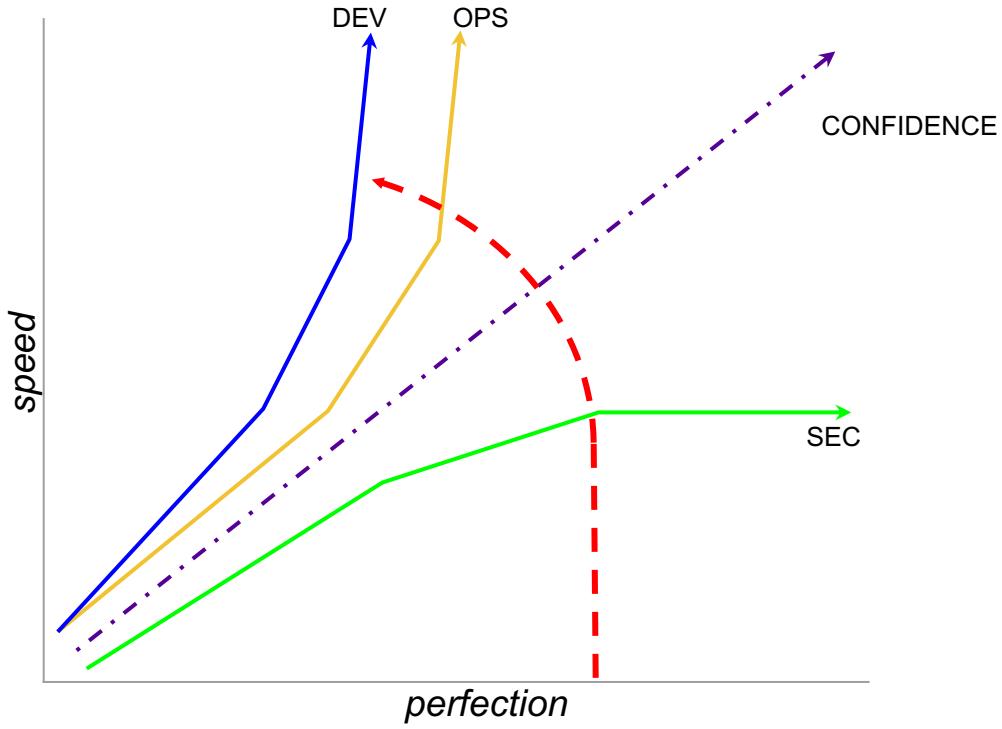
- Patrick DeBois

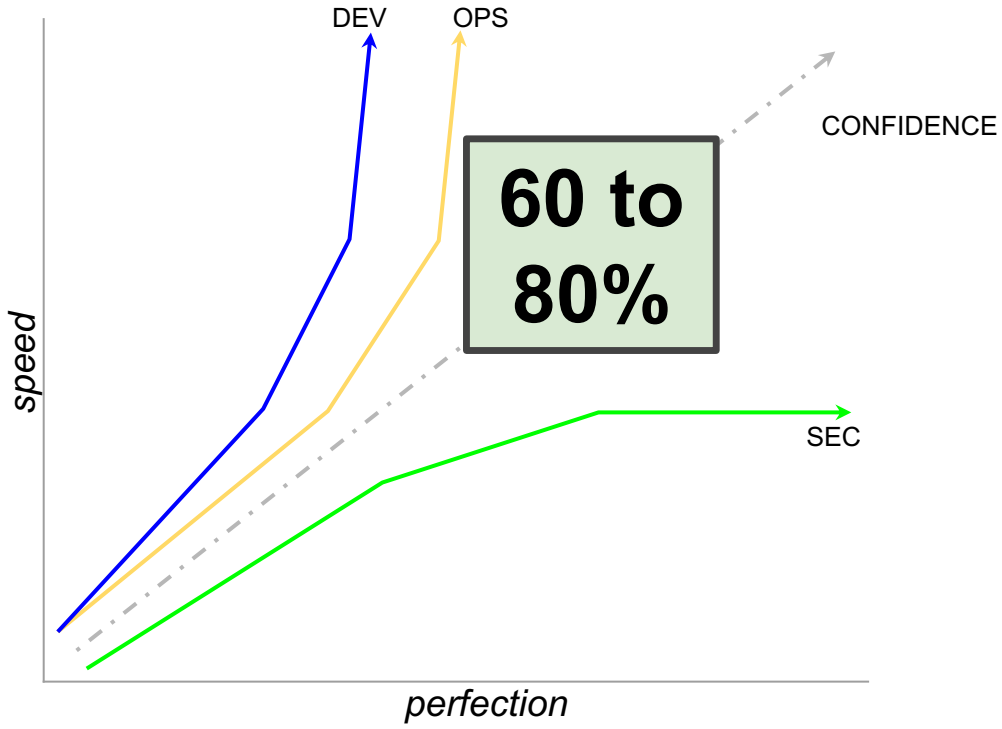
A security team who embraces openness about what it does and why, spreads understanding.

- Rich Smith

4 Keys to Culture

- Mutual Understanding
- Shared Language
- Shared Views
- Collaborative Tooling


DEV
OPS
CONFIDENCE
SEC
speed
perfection


DEV
OPS
CONFIDENCE
60 to
80%
SEC
speed
perfection

5. Do Security Testing for Developers' Benefit

You cannot train developers to write secure code.

A bug is a bug is a bug

Now with user-generated content!

Essential

O RLY?

Defect Density studies range from 0.5 to 10 per KLOC

No matter what,
it isn’t zero

But my app is only a few lines of code

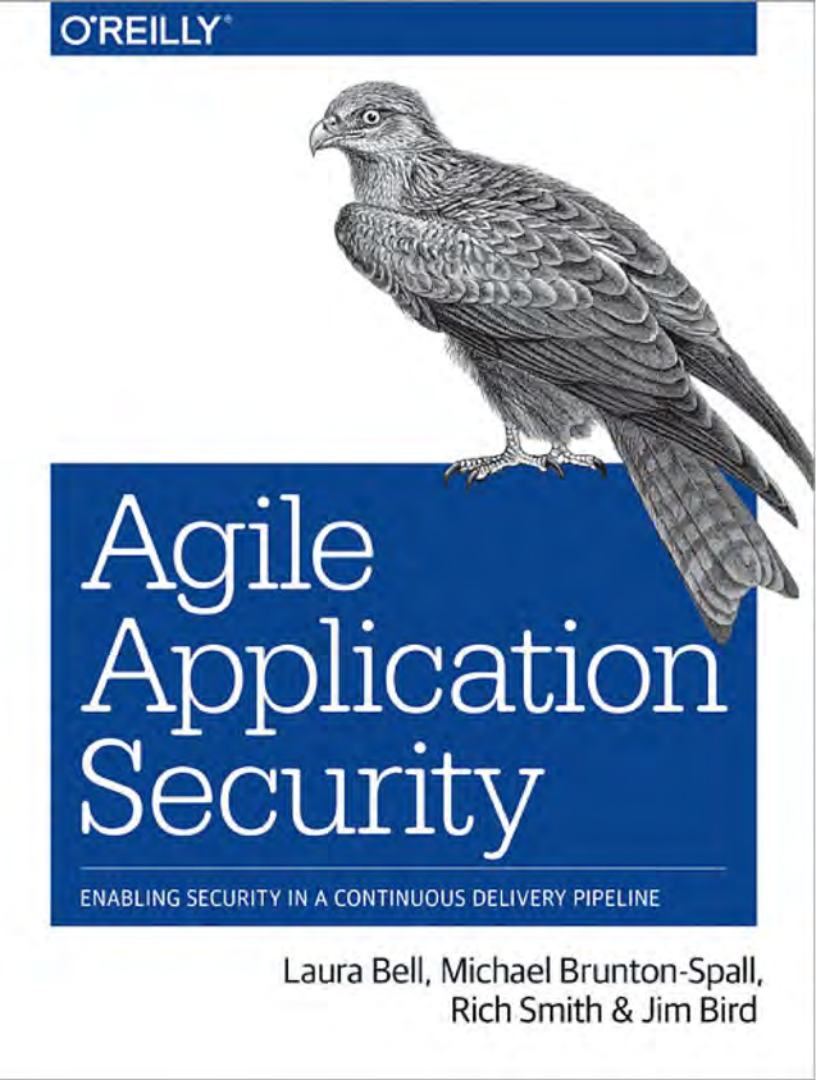
Is it?

222 Lines of Code
5 direct dependencies
54 total deps (incl. indirect)

(example thanks to snyk.io)

460,046

Lines of Code



The goal should be to come up with a set of automated tests that probe and check security configurations and runtime system behavior for security features that will execute every time the system is built and every time it is deployed.

@devsecops @wickett


Influence Organization Culture
Automate and Monitor
Across the Software Pipeline
Add Telemetry
and Feedback Loops
Identify Resource Misutilization
EFFECTIVENESS
INFLUENCE ACROSS TEAMS

6. Operationalize DevSecOps

Balance the *ilities

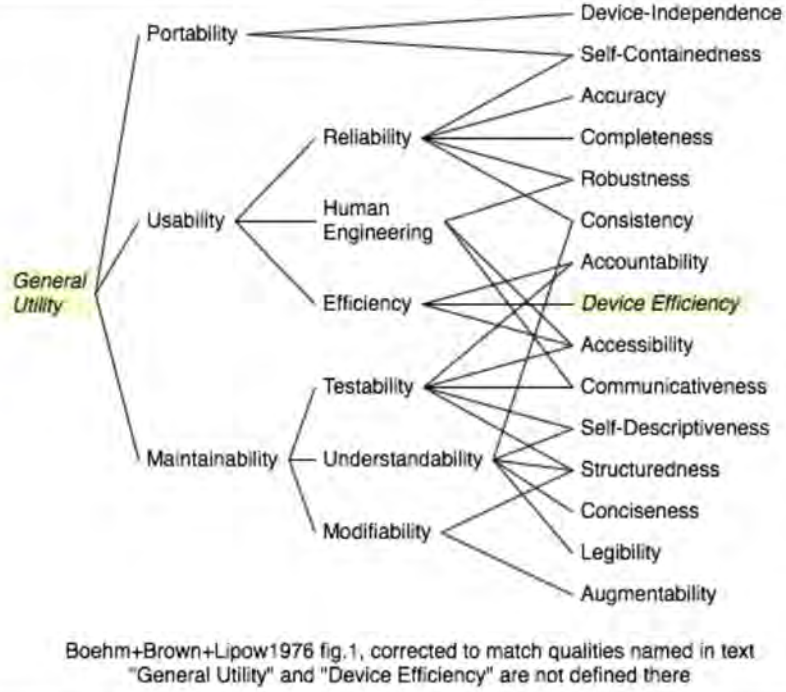


Figure 1. Boehm et al.'s Figure 1 (page 595), corrected to match the qualities named in the text. Note: "General Utility" and "Device Efficiency" appear in the figure but are not defined in the text. [Original figure]

Cavano and McCall's 11 Quality Factors (1978)

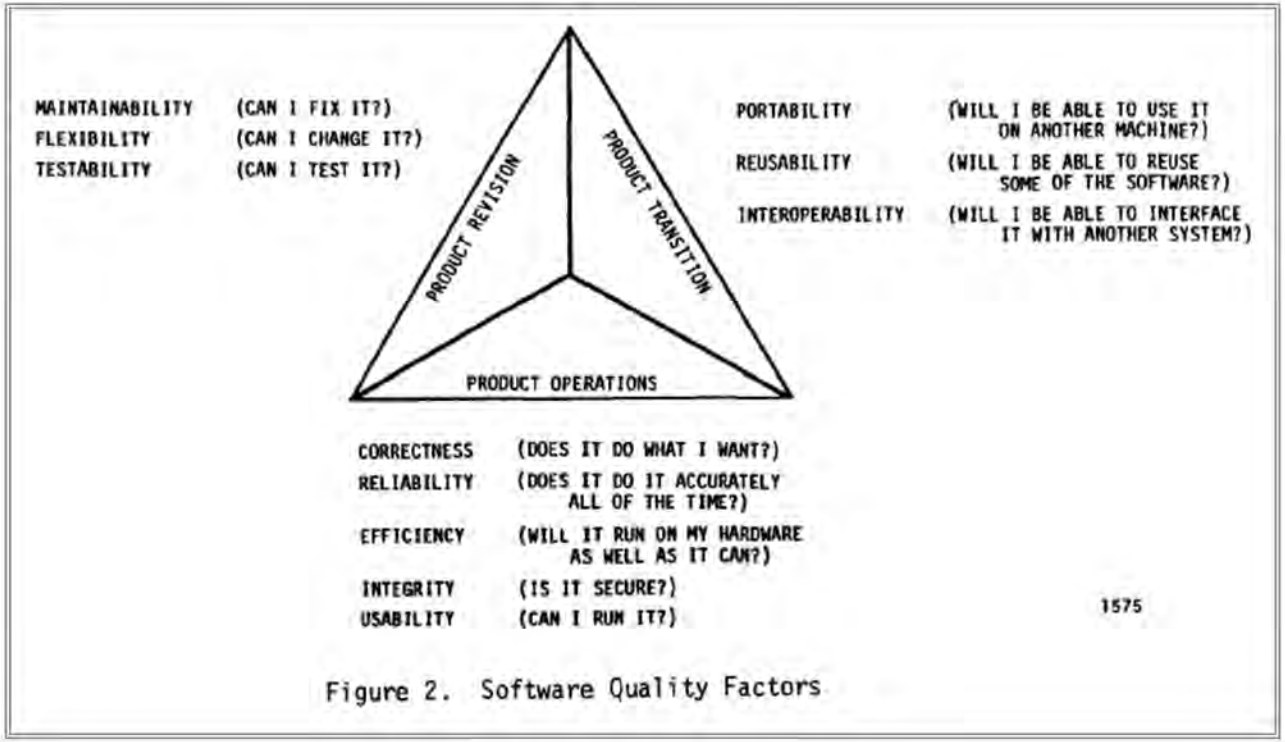
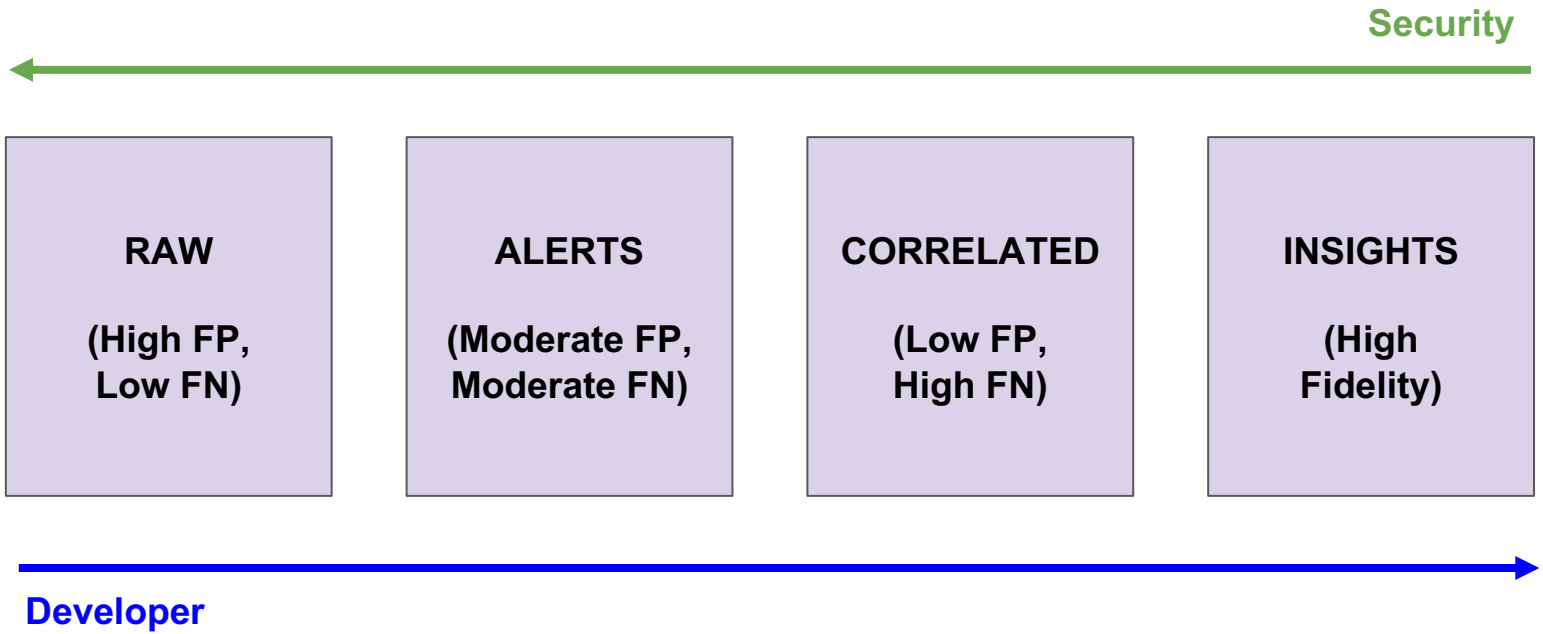


Figure 2. Software Quality Factors

“Securability”

Simplify the security domain into problem spaces

DESIGN	DEVELOP	OPERATE
• Requirement not considered • Excluded for purpose or balanced out • Misunderstood • Ideal state unknown • New attack strategies	• Bad component parts • Poor coding skills • Significant complexity • Iterative misses • Priorities not clear • Existing debt • Poor dependencies	• Leftover manual tasks • Missing capabilities • Unclear security design or strategy • Poor visibility/feedback • Lack of accountability • Not enough rigor


Security
RAW
(High FP, Low FN)
ALERTS
(Moderate FP, Moderate FN)
CORRELATED
(Low FP, High FN)
INSIGHTS
(High Fidelity)
Developer

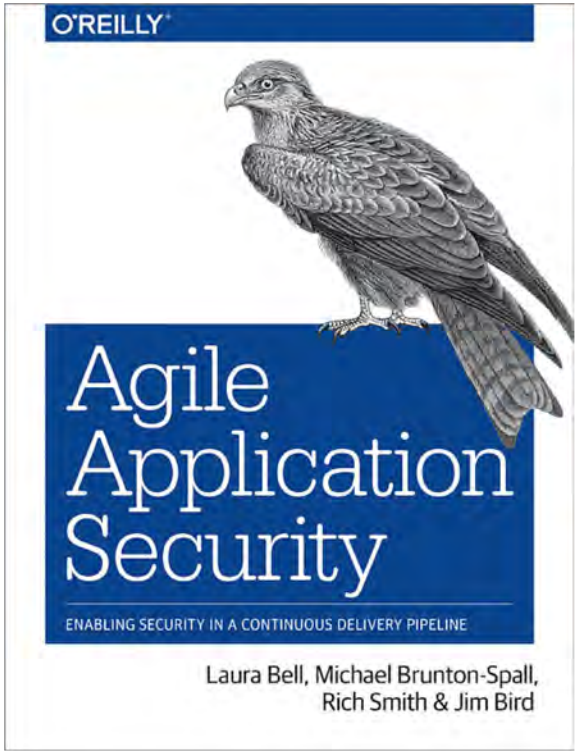
$$\frac{\#\ \text{of exploits}}{n\ \text{tests run}}$$

7. Make Security as Normal

Realign on Separation of Duties

PCI DSS Requirements
6.4.2 Separation of duties between development/test and production environments

Guidance
Reducing the number of personnel with access to the production environment and cardholder data minimizes risk and helps ensure that access is limited to those individuals with a business need to know.
The intent of this requirement is to separate development and test functions from production functions. For example, a developer may use an administrator-level account with elevated privileges in the development environment, and have a separate account with user-level access to the production environment.



[Deploys] can be treated as standard or routine changes that have been pre-approved by management, and that don't require a heavyweight change review meeting.

Dear Auditor,

a love letter to auditors from devops, where we promise to make life better

View My GitHub Profile

Download **ZIP File** | Download **TAR Ball** | View On **GitHub**

Dear Auditor,

We realize that we have been changing things in a rapid fashion from Agile and DevOps to Cloud and Containers. Yes, we have been busy, and are having great success delivering faster than ever, with better quality and supporting the business response to competitive pressures. This isn't just icing on the cake, the only sustainable advantage in our industries is the ability to meet customer demands faster, more reliably than our competitors.

With all this growth, we made a mistake, we forgot to bring you along for the ride. That is totally our bad, but we want to make it right. We want to make some new commitments.

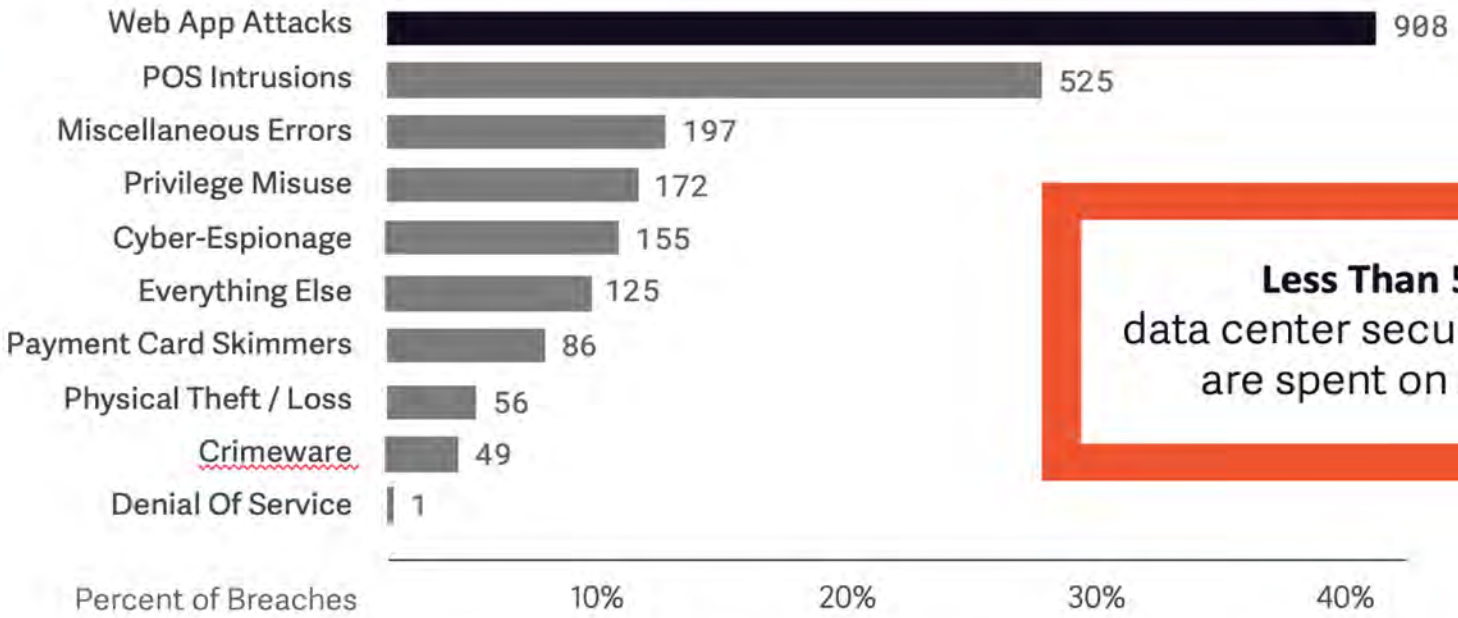
- We will bring you along
- We will be fully transparent about our development process
- We do realize that we own the risks
- We will maintain an open channel of discussion to demonstrate to you how we manage risks with our modern development practices

For example, you have told us that you are concerned about "Separation of Duties" in agile and DevOps practices, and we heard you! We think we have a better way to manage this and risks now. Having everything in version control, enforcing peer review for every change, releasing via a secure pipeline, restricting production access, and monitoring unauthorized changes in production systems should address your concern.

The DevOps community has been experimenting quite a bit over the last number of years and common practice represents the collective wisdom across many companies, industries, and countries.

Web App Attacks Are the #1 Source of Data Breaches

Attacks are Up 300% from 2014 – Incumbent Products Aren't Solving the Problem



Less Than 5% of data center security budgets are spent on AppSec

Sources: Gartner, Verizon

RSAConference2019

Are you ready to advance?

8. Track Adversary Interest

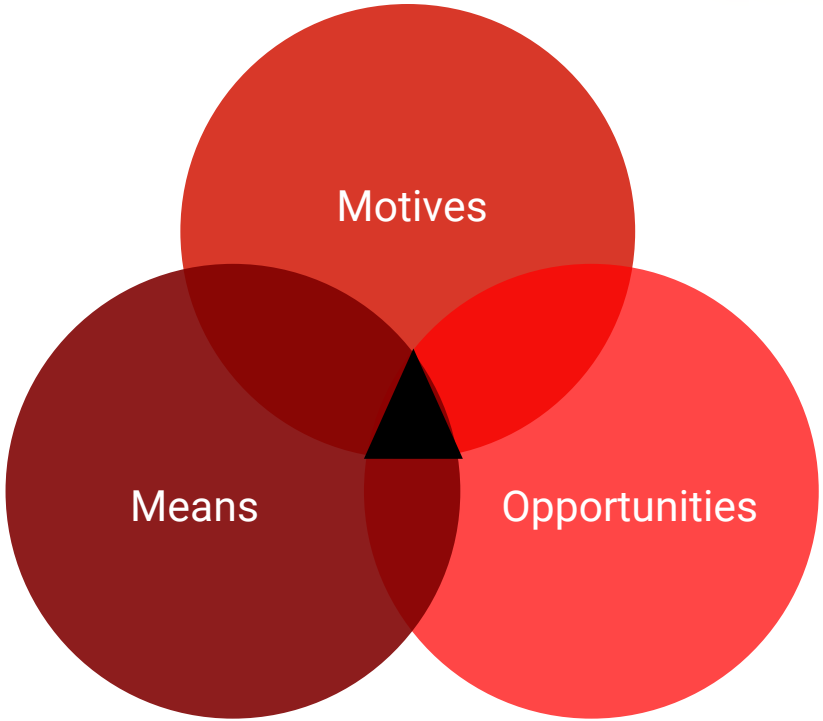
Security Facts

Original Lines of Code	300
Open Source Components	25
Type: **Embedded** Version	1.0
Intended Version Lifetime/Expiration	02/2020
Organization Security Trend at Release	3.2
Security Degradation Rating	A
Required Monthly Customer Maintainence	2

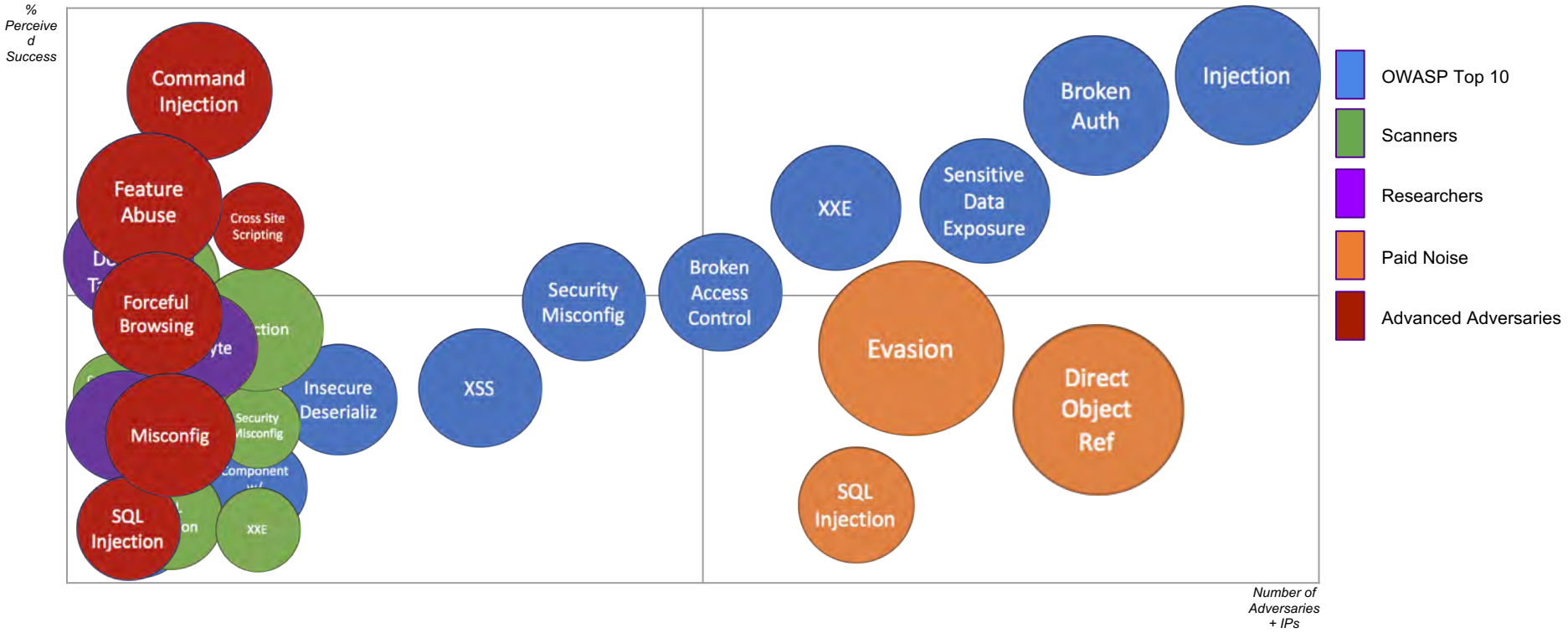
	% Control Values
Adversary Interest	97%
~~Residual Risk~~	~~8%~~
Preventative Measures	93%
Access Control	100%
Encryption	95%
Tamper	91%
Detective Measures	99%
Remote	99%
Local	99%

NIST	99%	■	OPNGBK	91%
PCI DSS	92%	■		

* All values are based on modeled Abuse and FMEA cases for this class of device and applicable implementation patterns. Your results may fluctuate according to intended business risk profile and residual risk tolerances that allow for some controls to be less restrictive. Actual results may also vary with creative use or experimental implementation.

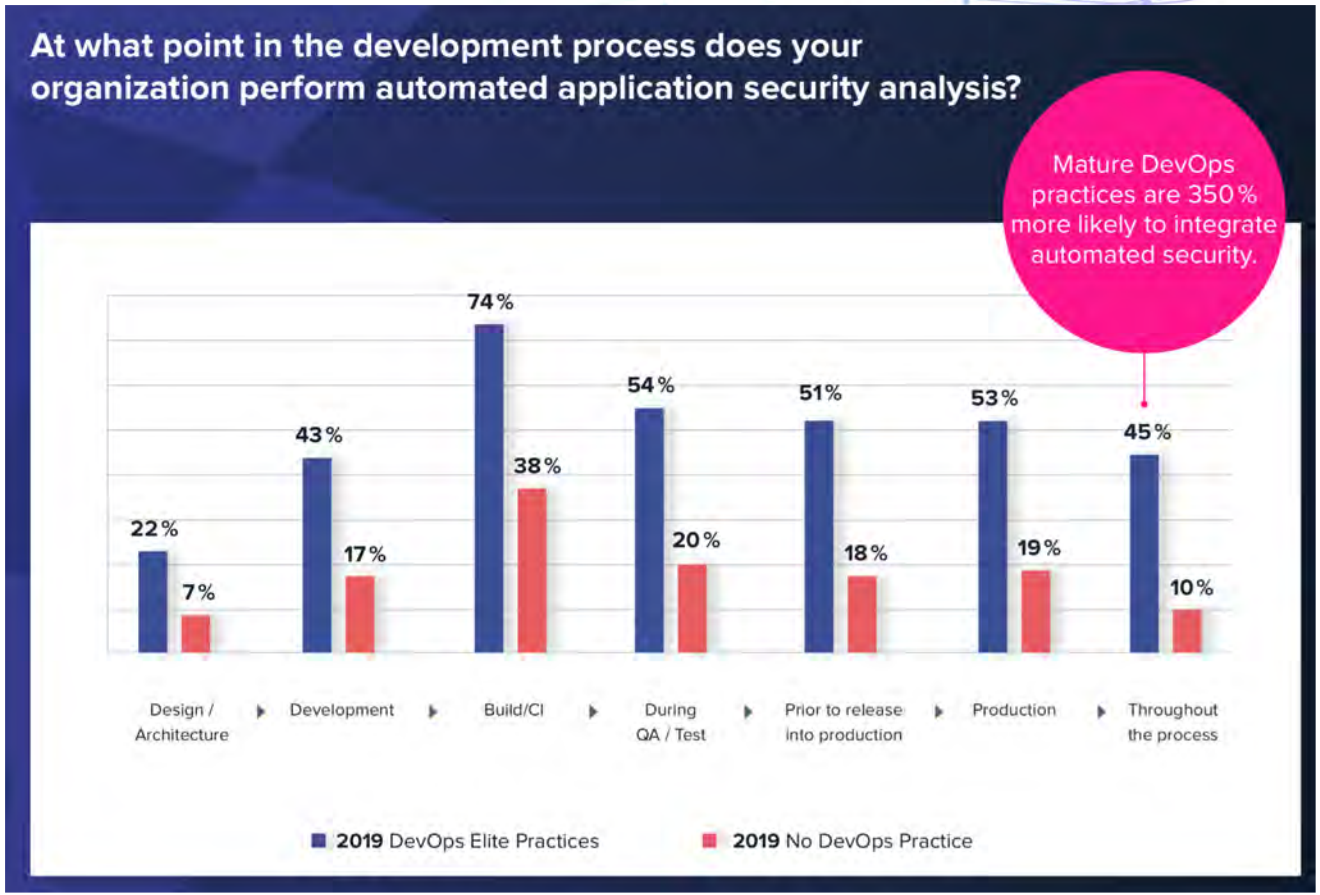

Motives
Means
Opportunities

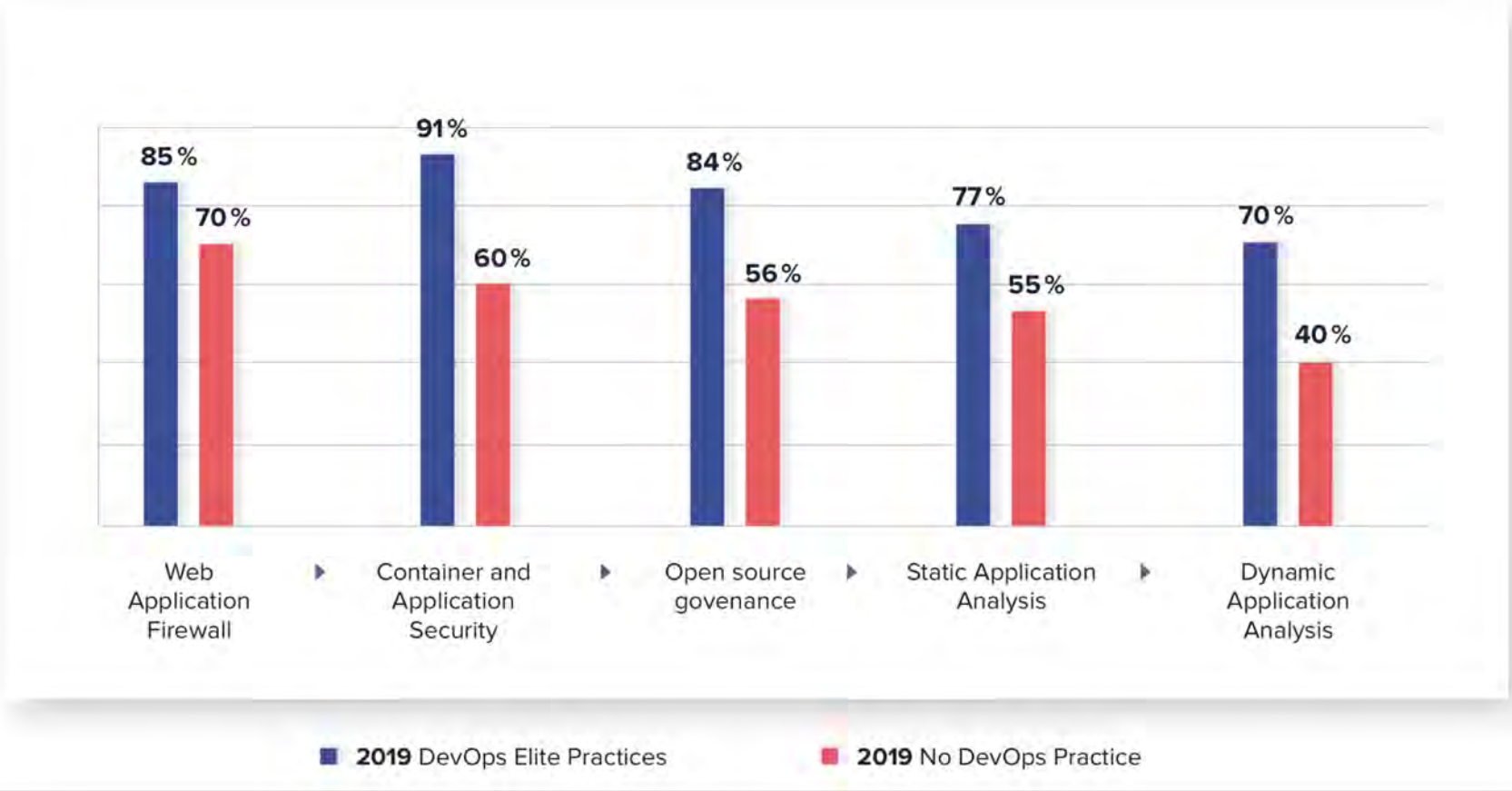
OWASP vs. Real World



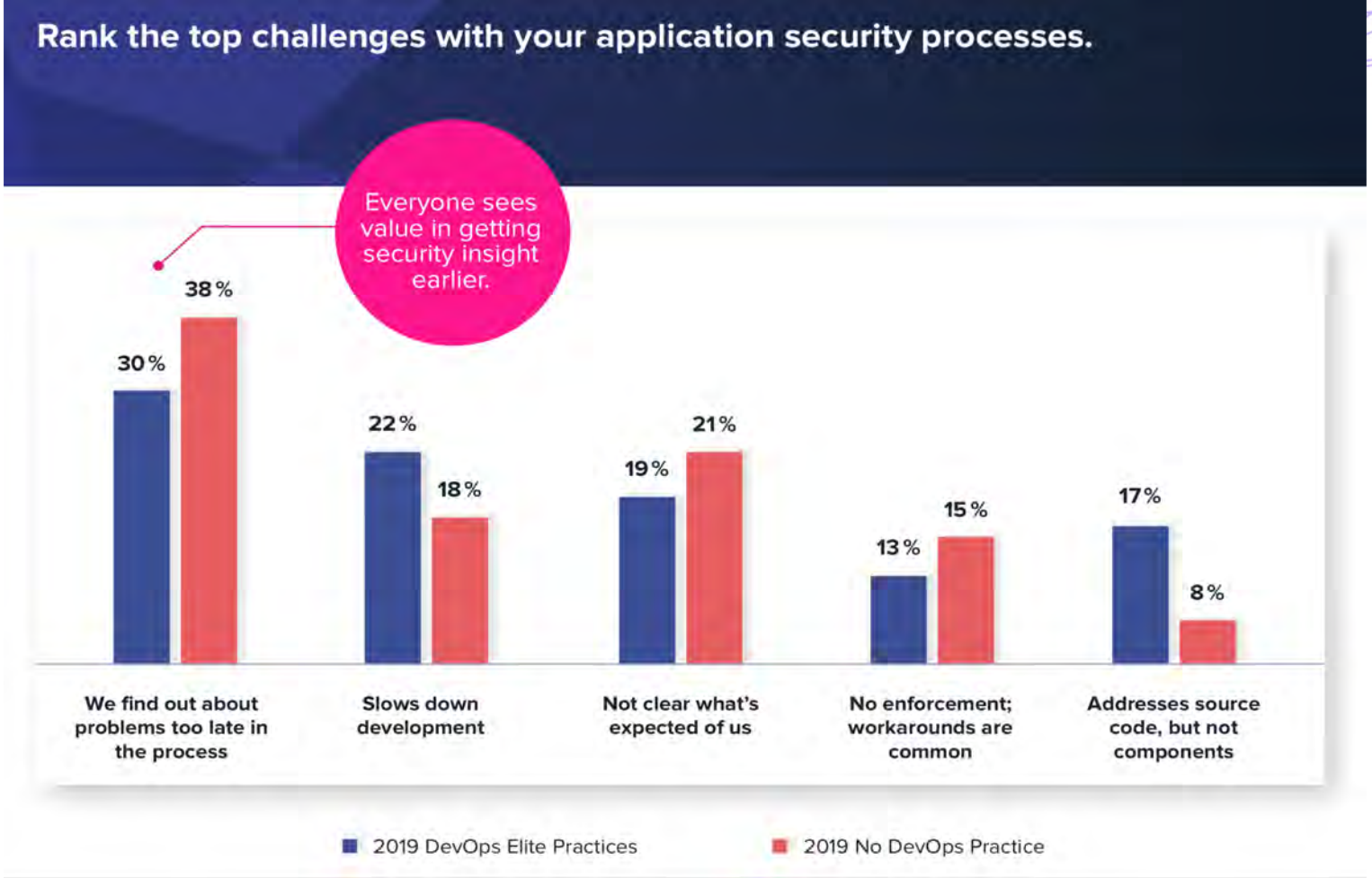
9. Create Security Observability


sonatype
DEVSECOPS
COMMUNITY
SURVEY 2019
cloudbees
Carnegie Mellon University
Software Engineering Institute
Signal Sciences
Twistlock
9TH BIT


At what point in the development process does your organization perform automated application security analysis?
Mature DevOps practices are 350% more likely to integrate automated security.
22%
7%
43%
17%
74%
38%
54%
20%
51%
18%
53%
19%
45%
10%
Design / Architecture
Development
Build/CI
During QA / Test
Prior to release into production
Production
Throughout the process
2019 DevOps Elite Practices
2019 No DevOps Practice


85%
70%
91%
60%
84%
56%
77%
55%
70%
40%
Web Application Firewall
Container and Application Security
Open source govenance
Static Application Analysis
Dynamic Application Analysis
2019 DevOps Elite Practices
2019 No DevOps Practice

Security Observability gives applications the ability to expose the attacks that are happening below the surface with feedback to devs, ops, and security.

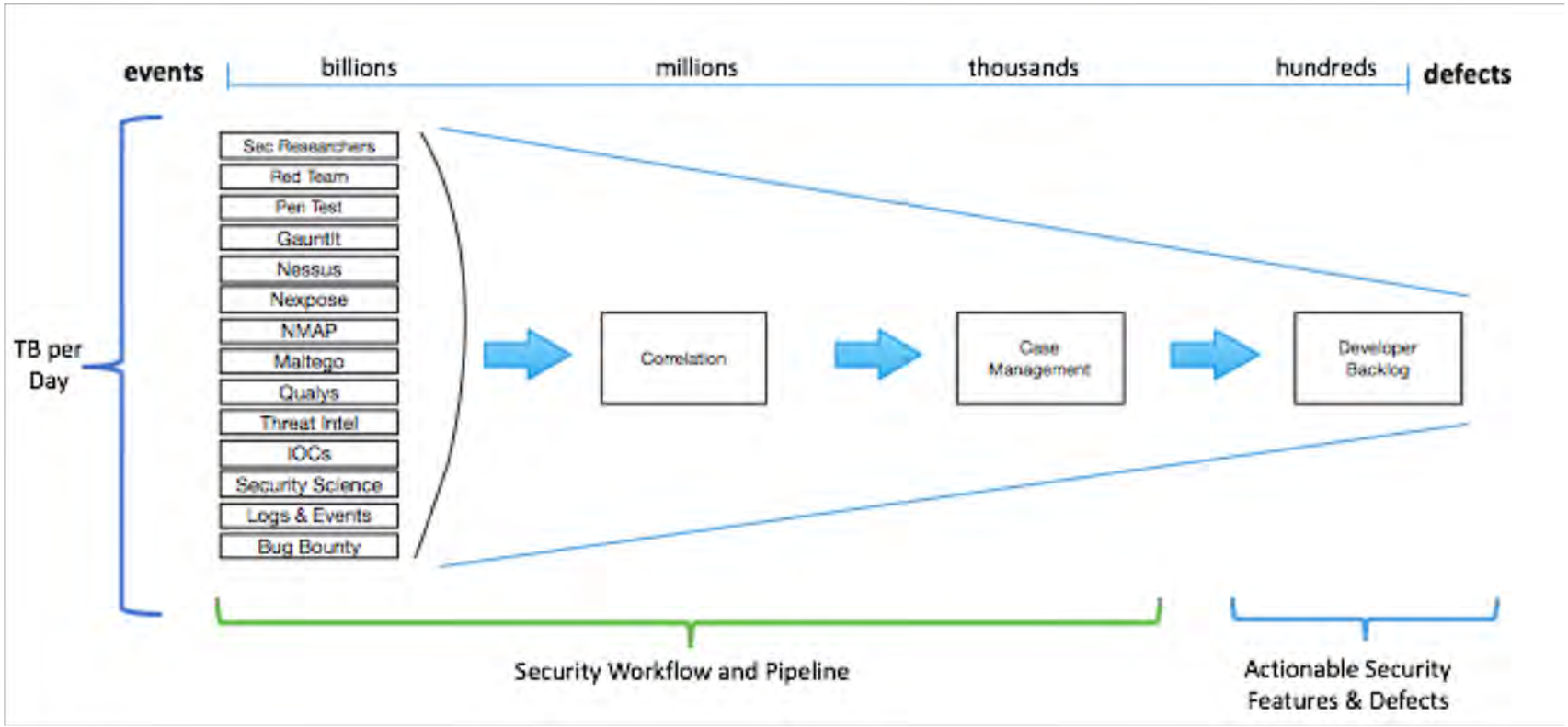

Rank the top challenges with your application security processes.
Everyone sees value in getting security insight earlier.
30%
38%
22%
18%
19%
21%
13%
15%
17%
8%
We find out about problems too late in the process
Slows down development
Not clear what's expected of us
No enforcement; workarounds are common
Addresses source code, but not components
2019 DevOps Elite Practices
2019 No DevOps Practice

RSAConference2019

Are you ready to advance?

10. Build the Future

How do we change the game?



Less Guessing...

	OWASP TOP 10 App Sec Risks	Real-World Top 10 Attacks
1	Injection	Direct Object Reference
2	Broken Authentication	Forceful Browsing
3	Sensitive Data Exposure	Null Byte Attack
4	XML External Exposures (XXE)	Command Injection
5	Broken Access Control	Feature Abuse
6	Security Misconfiguration	Evasion Techniques
7	Cross Site Scripting	Subdomain Takeover
8	Insecure Deserialization	Misconfiguration
9	Using Components with Known Vulnerabilities	Cross Site Scripting
10	Insufficient Logging/Monitoring	SQL Injection

No ***Exploitable*** Escapes

Continuous

Precision

Community

Books we recommended

Agile Application Security

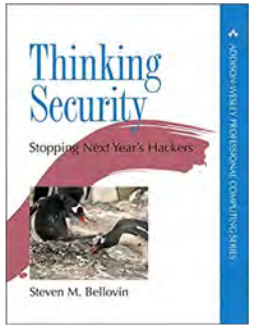
Continuous Delivery

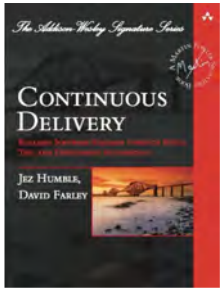
The Phoenix Project

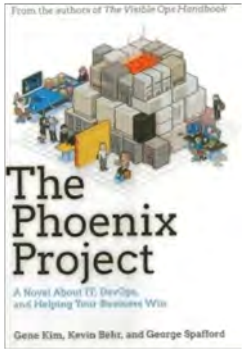
DevOps Handbook

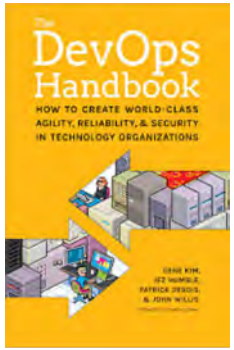
Accelerate

Thinking Security









How to apply

Next Week

- Determine where you are, understand value contribution

Next Month

- Telemetry and testing for feedback on 1-2 projects

Next 6 months

- Look for and share culture wins as they happen, get involved with DevSecOps

Get the slides now:
james@signalsciences.com

Got a good story? We're writing a book

I'm are writing a book along with James Wickett, Ernest Mueller and John Willis on DevSecOps.

We are looking for stories of DevSecOps transformations, journeys, successes and failures.

book@devsecops.org