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

Joshua Corman & Gene Kim

Rugged? DevOps COOKBOOK

Session ID: CLD-106

Session Classification: Intermediate

RSA CONFERENCE 2012

About Joshua Corman

- Director of Security Intelligence for Akamai Technologies
  - Former Research Director, Enterprise Security [The 451 Group]
  - Former Principal Security Strategist [IBM ISS]
- Industry:
  - Expert Faculty: The Institute for Applied Network Security (IANS)
  - 2009 NetworkWorld Top 10 Tech People to Know
  - Co-Founder of "Rugged Software" www.ruggedsoftware.org
  - BLOG: www.cognitivedissidents.com
- Things I've been researching:
  - Compliance vs Security
  - Disruptive Security for Disruptive Innovations
  - Chaotic Actors
  - Espionage
  - Security Metrics

About Gene Kim

- Researcher, Author
- Industry:
  - Invented and founded Tripwire, CTO (1997-2010)
  - Co-author: “Visible Ops Handbook”(2006), “Visible Ops Security” (2008)
  - Co-author: “When IT Fails: The Novel,” “The DevOps Cookbook” (Coming May 2012)
- Things I’ve been researching:
  - Benchmarked 1300+ IT organizations to test effectiveness of IT controls vs. IT performance
  - DevOps, Rugged DevOps
  - Scoping PCI Cardholder Data Environment (#FAIL)

Agenda

- Problem statement
- What is DevOps?
- What is Rugged?
- What is Rugged DevOps?
- Things you can do right away

Potentially Unfamiliar Words You Will See

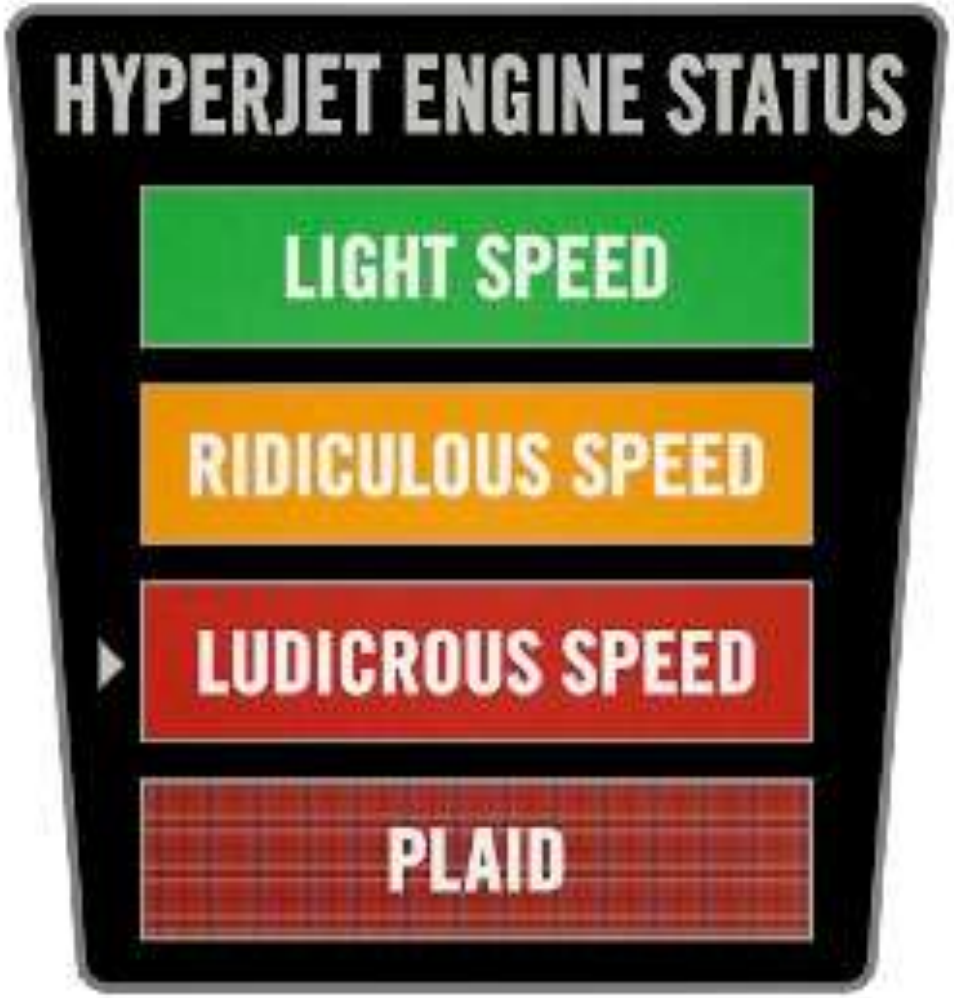
- Kanban
- Andon cord
- Sprints
- Rugged
- DevOps
- Bottleneck
- Systems thinking
- Controls reliance

Problem Statement

RSA CONFERENCE 2012

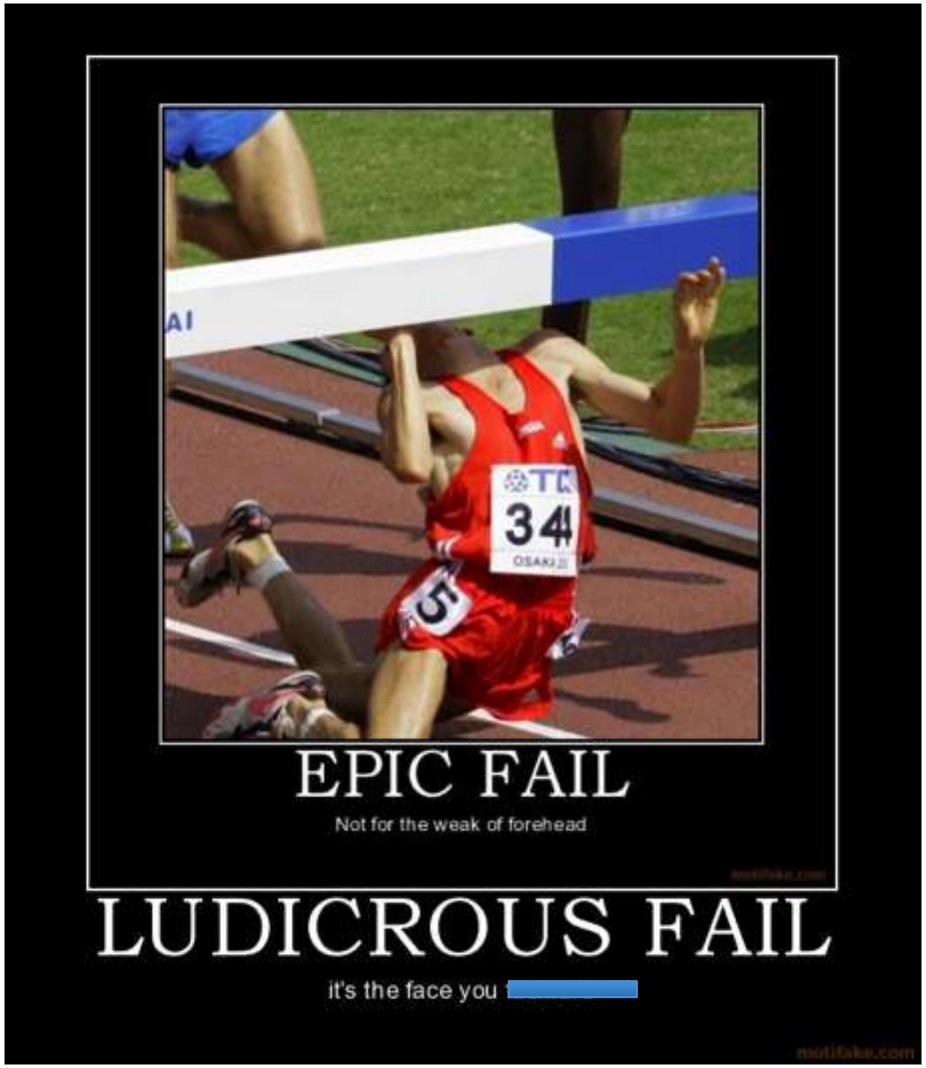
Ludicrous Speed?

Ludicrous Speed

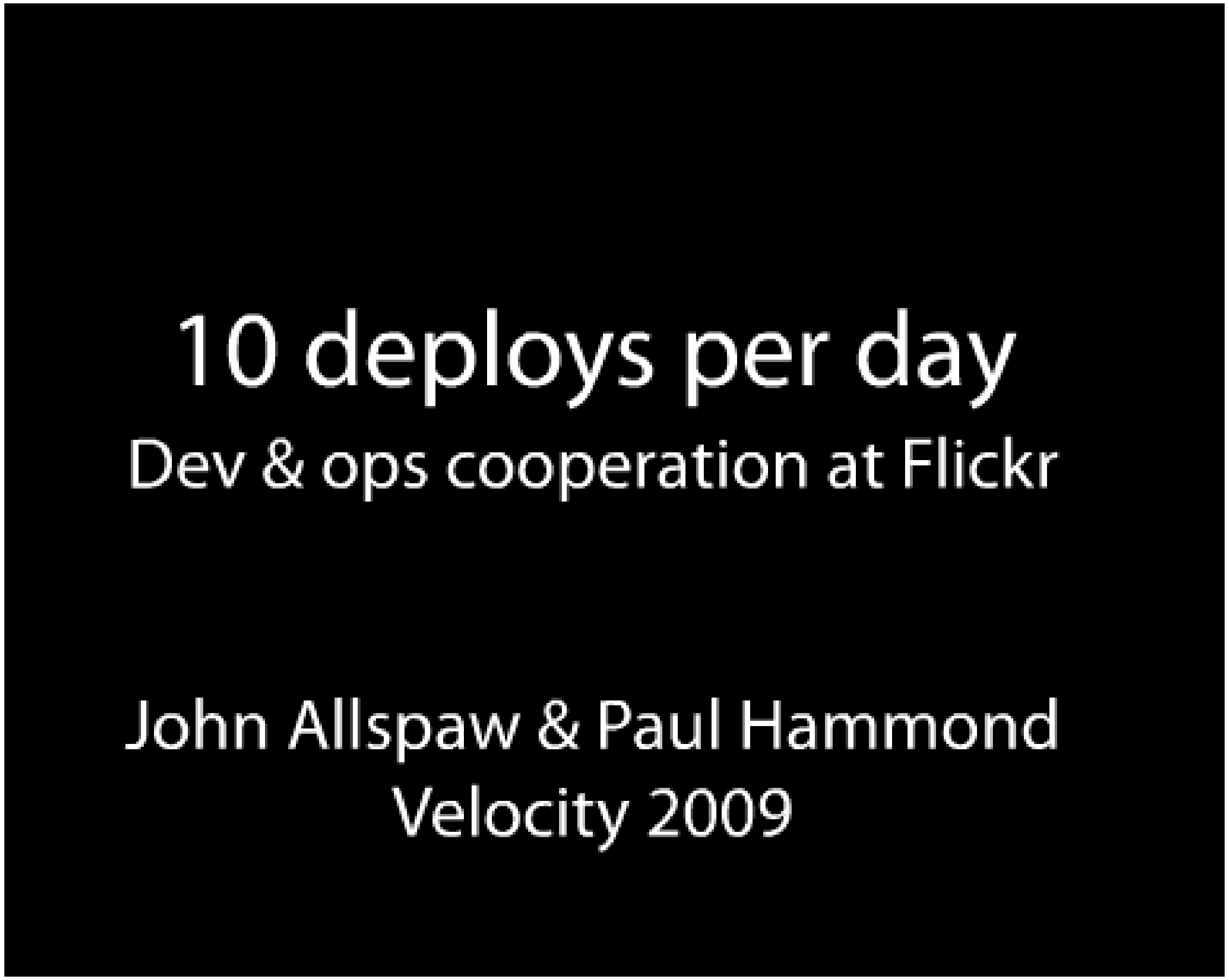


Ludicrous Speed!

Ludicrous Fail?!



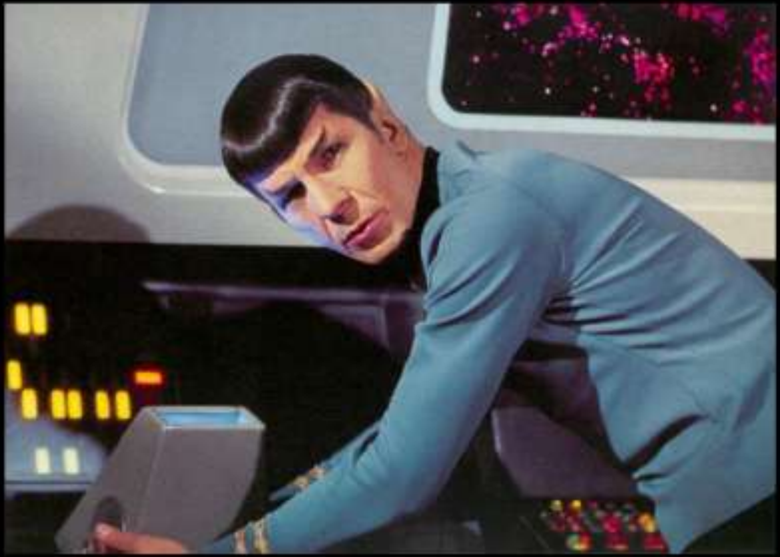
What Is DevOps?


10 deploys per day

Dev & ops cooperation at Flickr

John Allspaw & Paul Hammond

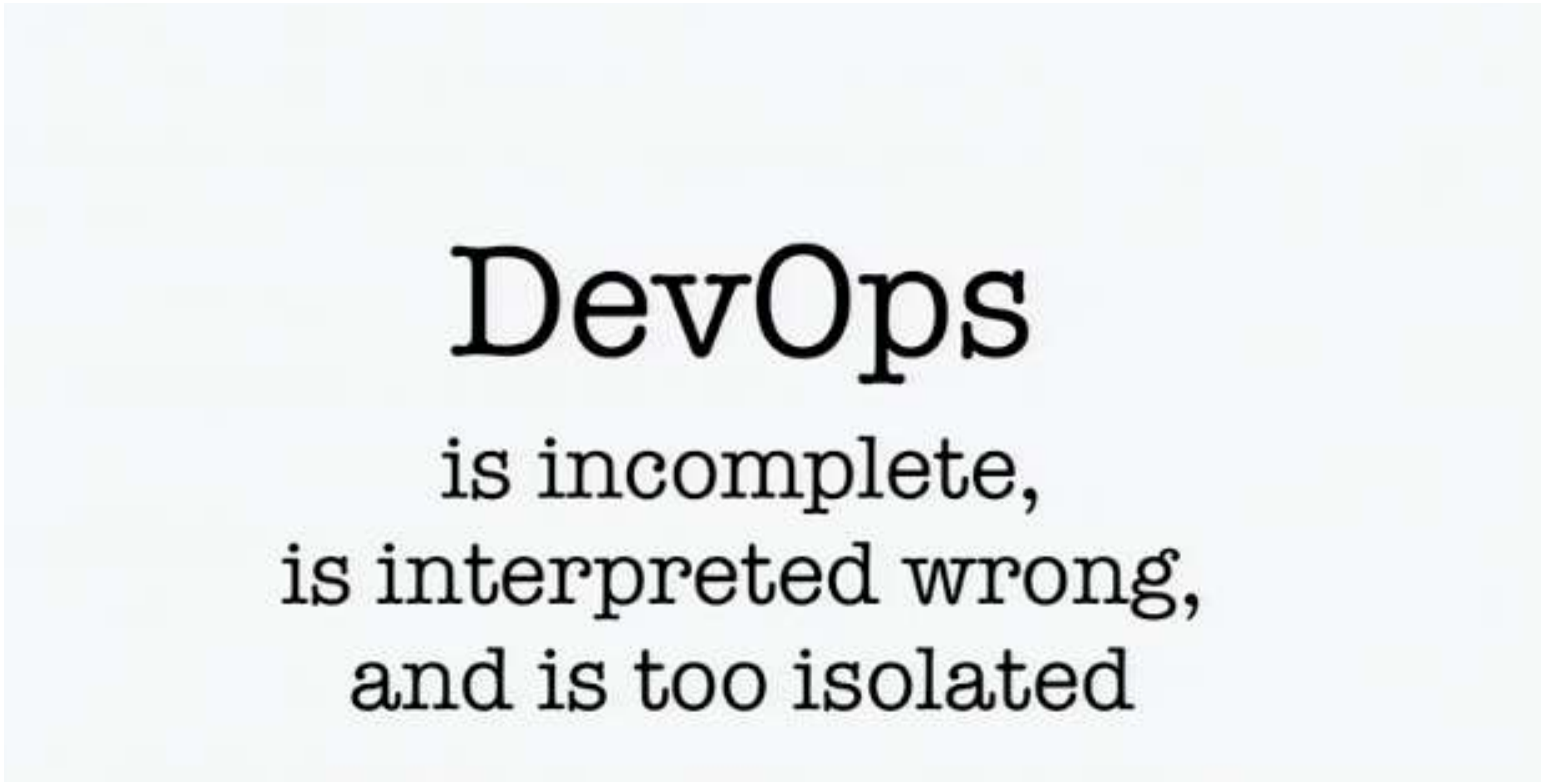
Velocity 2009



Little bit weird	Pulls levers & turns knobs
Sits closer to the boss	Easily excited
Thinks too hard	Yells a lot in emergencies

Ops who think like devs
Devs who think like ops

Dev and Ops

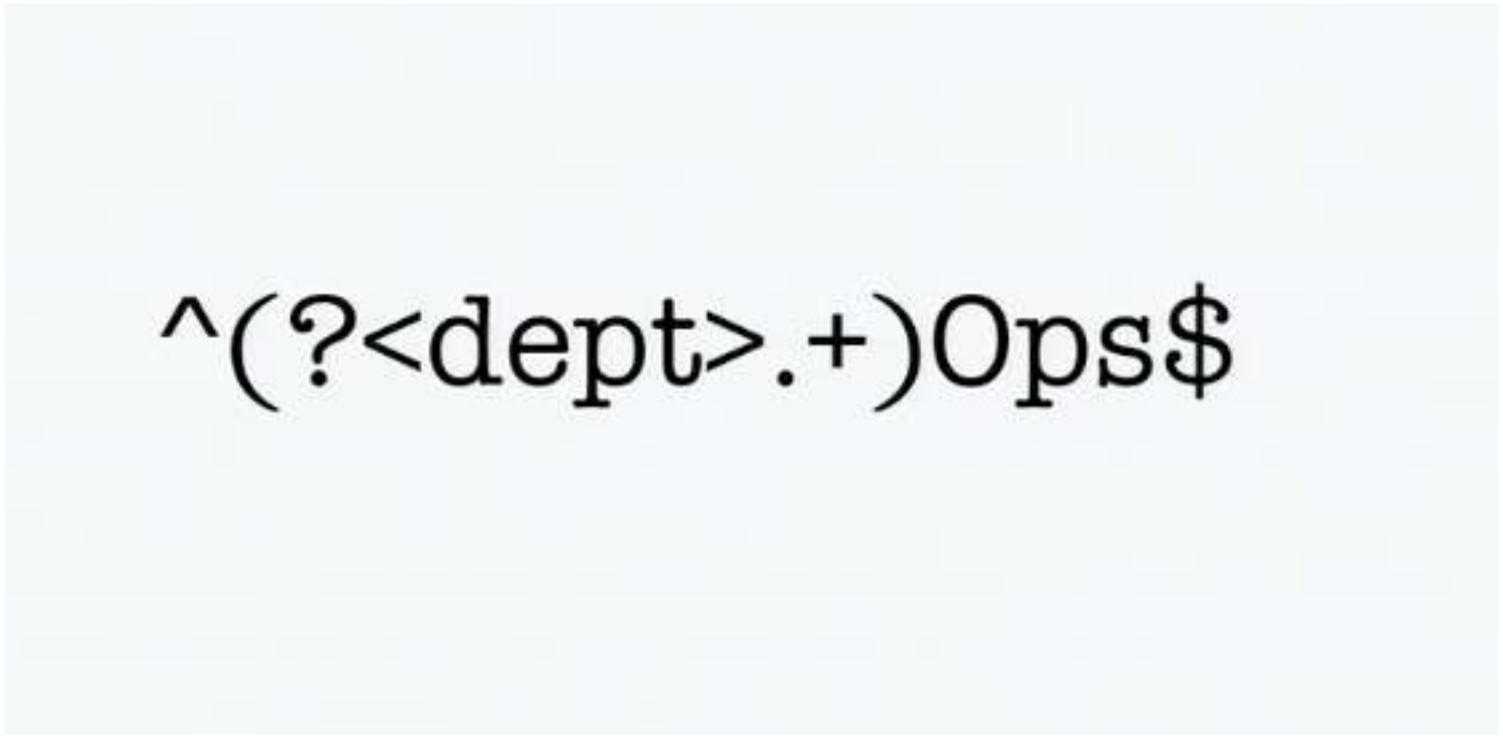


DevOps

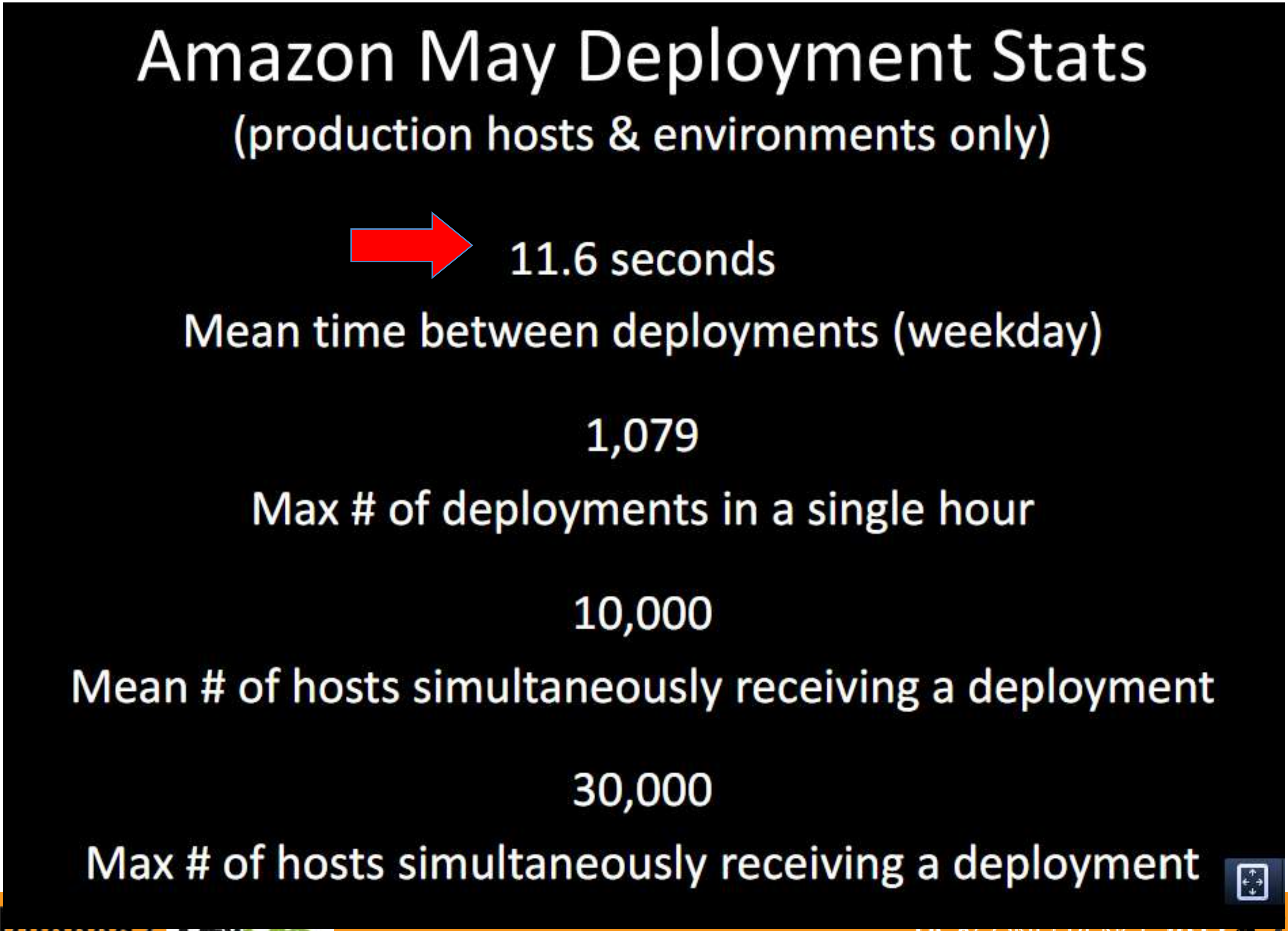
is incomplete,
is interpreted wrong,
and is too isolated

.*Ops

Source: Theo Schlossnagle



^(?<dept>.+)Ops$



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

Source: John Jenkins, Amazon.com

High Performing IT Organizations

- High performers maintain a posture of compliance
  - **Fewest** number of repeat audit findings
  - **One-third** amount of audit preparation effort
- High performers find and fix security breaches faster
  - **5 times** more likely to detect breaches by automated control
  - **5 times** less likely to have breaches result in a loss event
- When high performers implement changes…
  - **14 times more** changes
  - **One-half** the change failure rate
  - **One-quarter** the first fix failure rate
  - **10x faster** MTTR for Sev 1 outages
- When high performers manage IT resources…
  - **One-third** the amount of unplanned work
  - **8 times more** projects and IT services
  - **6 times more** applications

2007: Three Controls Predict 60% Of Performance

- To what extent does an organization define, monitor and enforce the following?
  - Standardized configuration strategy
  - Process discipline
  - Controlled access to production systems

What Is Rugged?

RSACONFERENCE2012

Rugged Software Development

Joshua Corman, David Rice, Jeff Williams

2010

RSA CONFERENCE | Where The World Talks Security

GREYLOCK PARTNERS

USA 2009 20-24 April | Moscone Center | San Francisco

RUGGED SOFTWARE

…so software not only needs to be…

hydrolift
fly without wings
SIEMENS
Koenigsegg
Color Line
NORWAY
Radisson

FAST

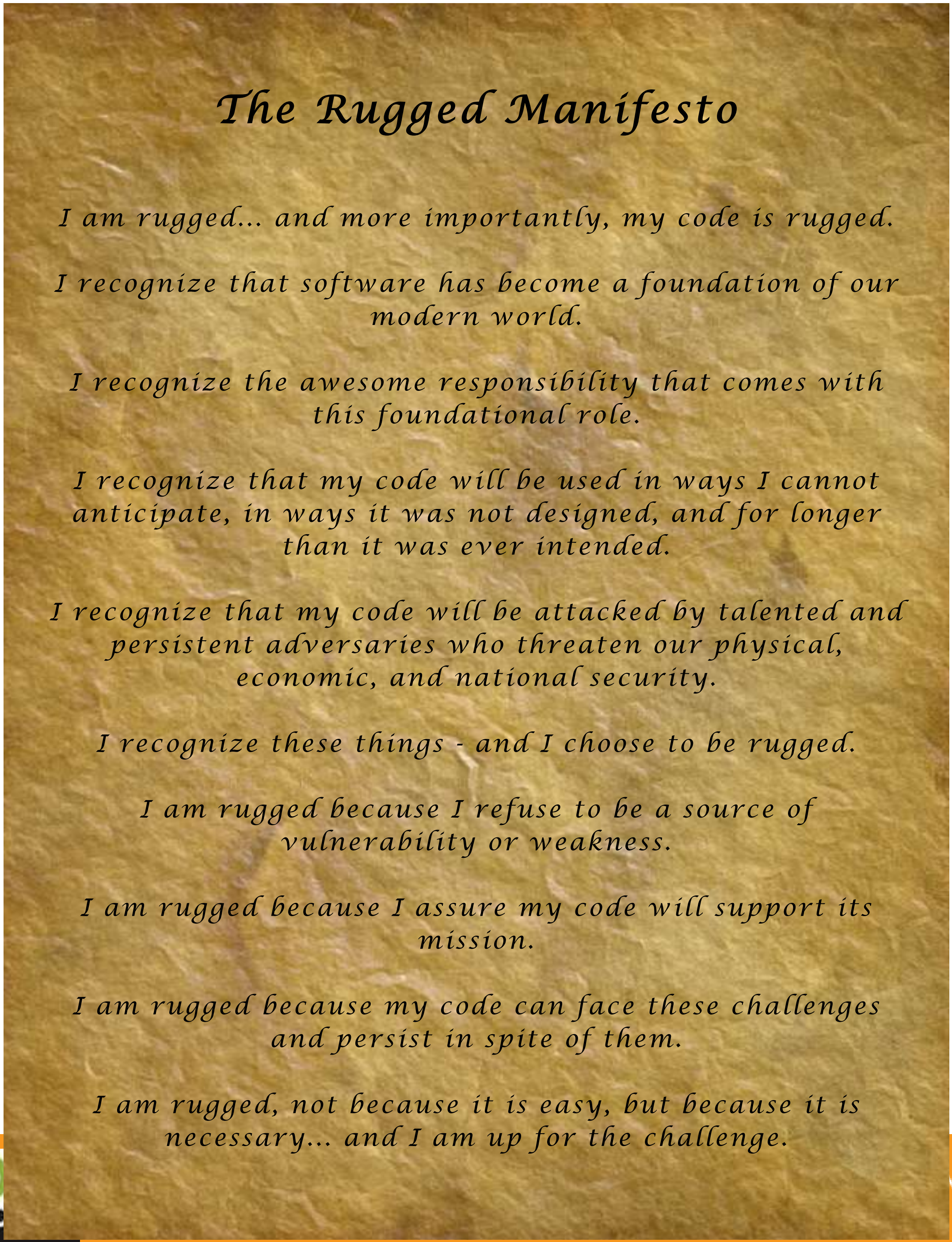
AGILE

Are You Rugged?

HARSH

UNFRIENDLY

THE MANIFESTO



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

Rugged?

CROSSTALK
The Journal of Defense Software Engineering
THE RUGGED SOFTWARE MANIFESTO

www.ruggedsoftware.org

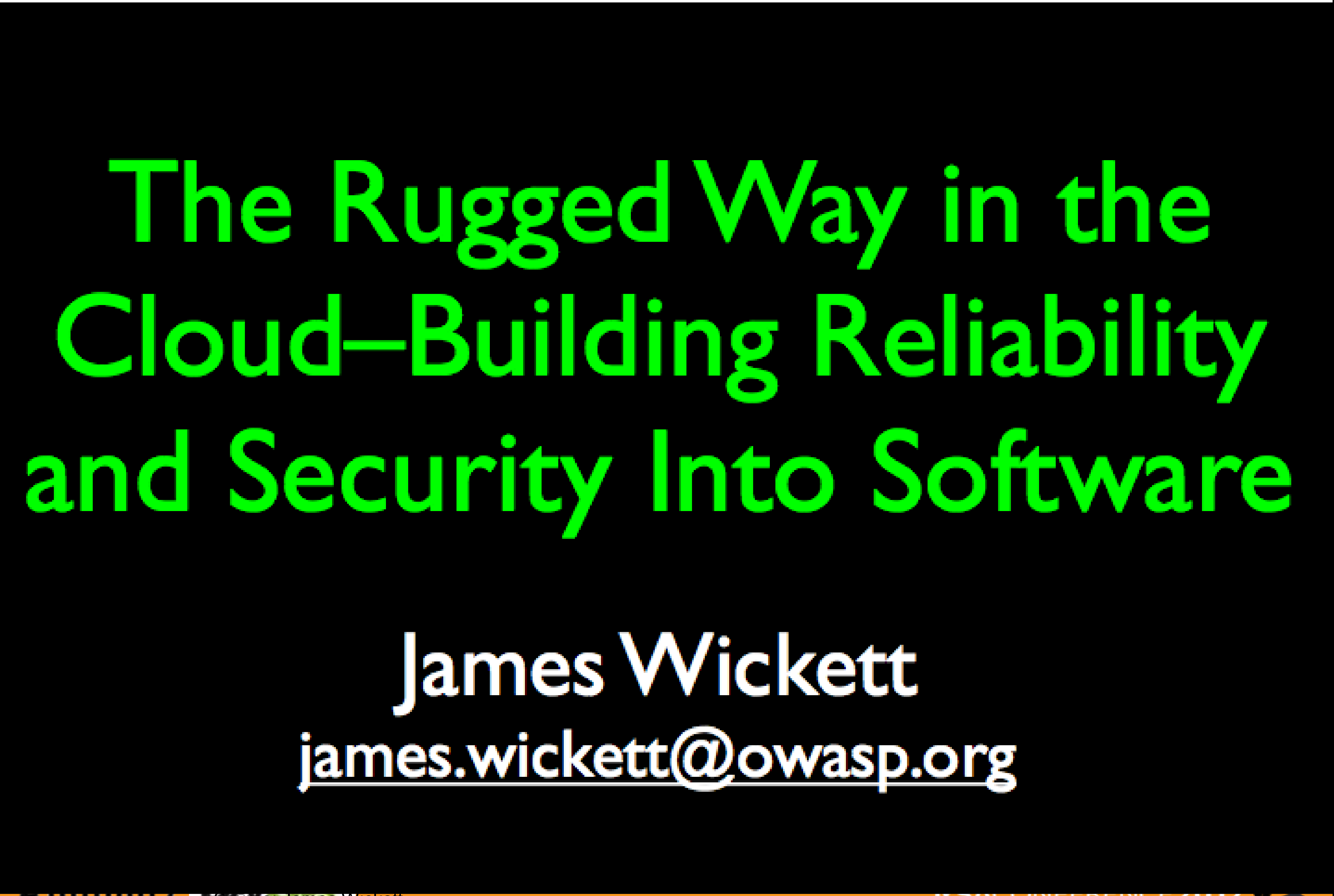
CrossTalk

http://www.crosstalkonline.org/issues/marchapril-2011.html

RSA CONFERENCE 2012

What Is Rugged DevOps?

RSACONFERENCE2012



The Rugged Way in the Cloud–Building Reliability and Security Into Software

James Wickett

james.wickett@owasp.org

Source: James Wickett

Rugged Survival Guide

- Defensible Infrastructure
- Operational Discipline
- Situational Awareness
- Countermeasures

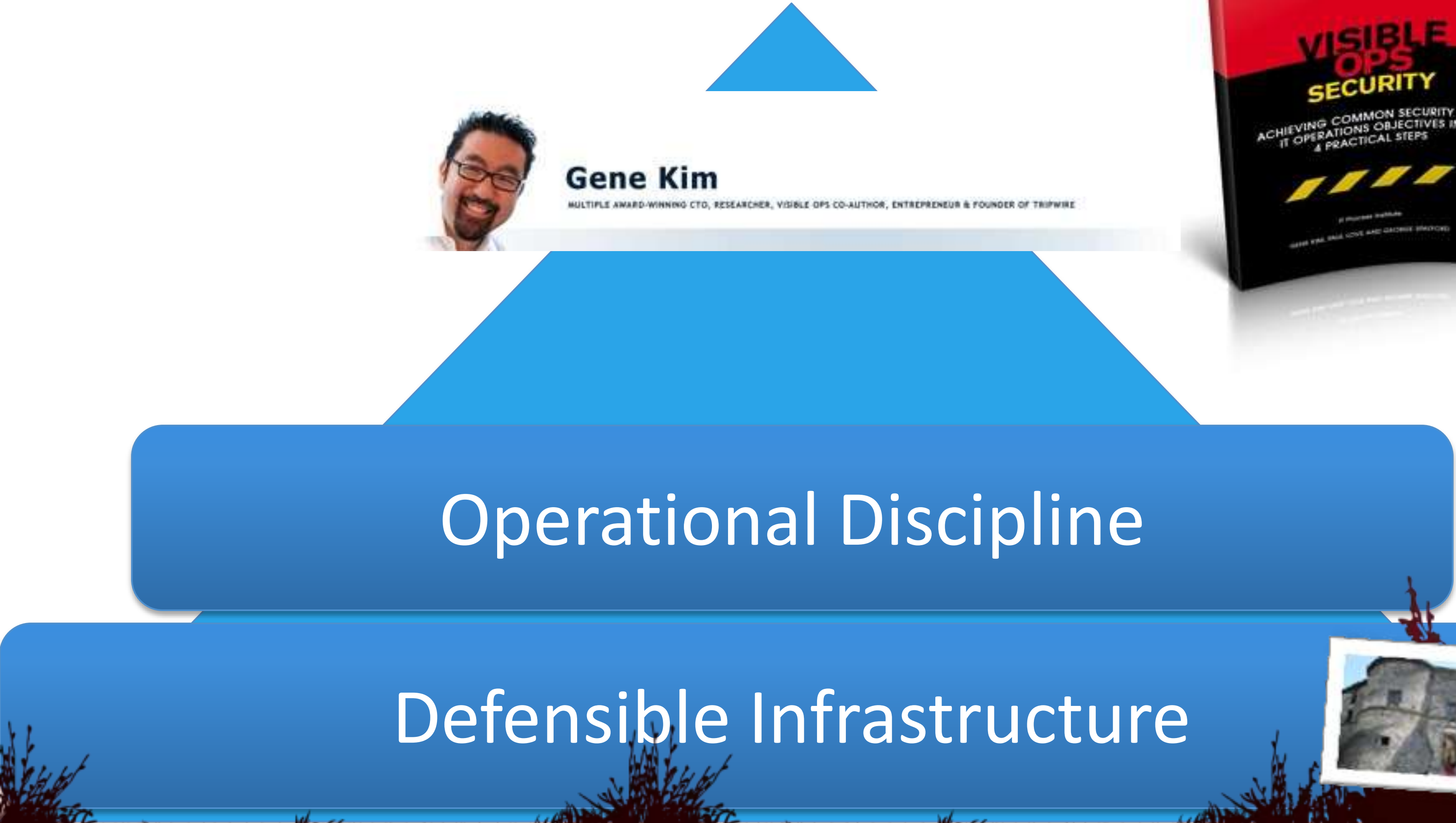
On YouTube: "PCI Zombies"

Survival Guide/Pyramid

www.ruggedsoftware.org

Defensible Infrastructure

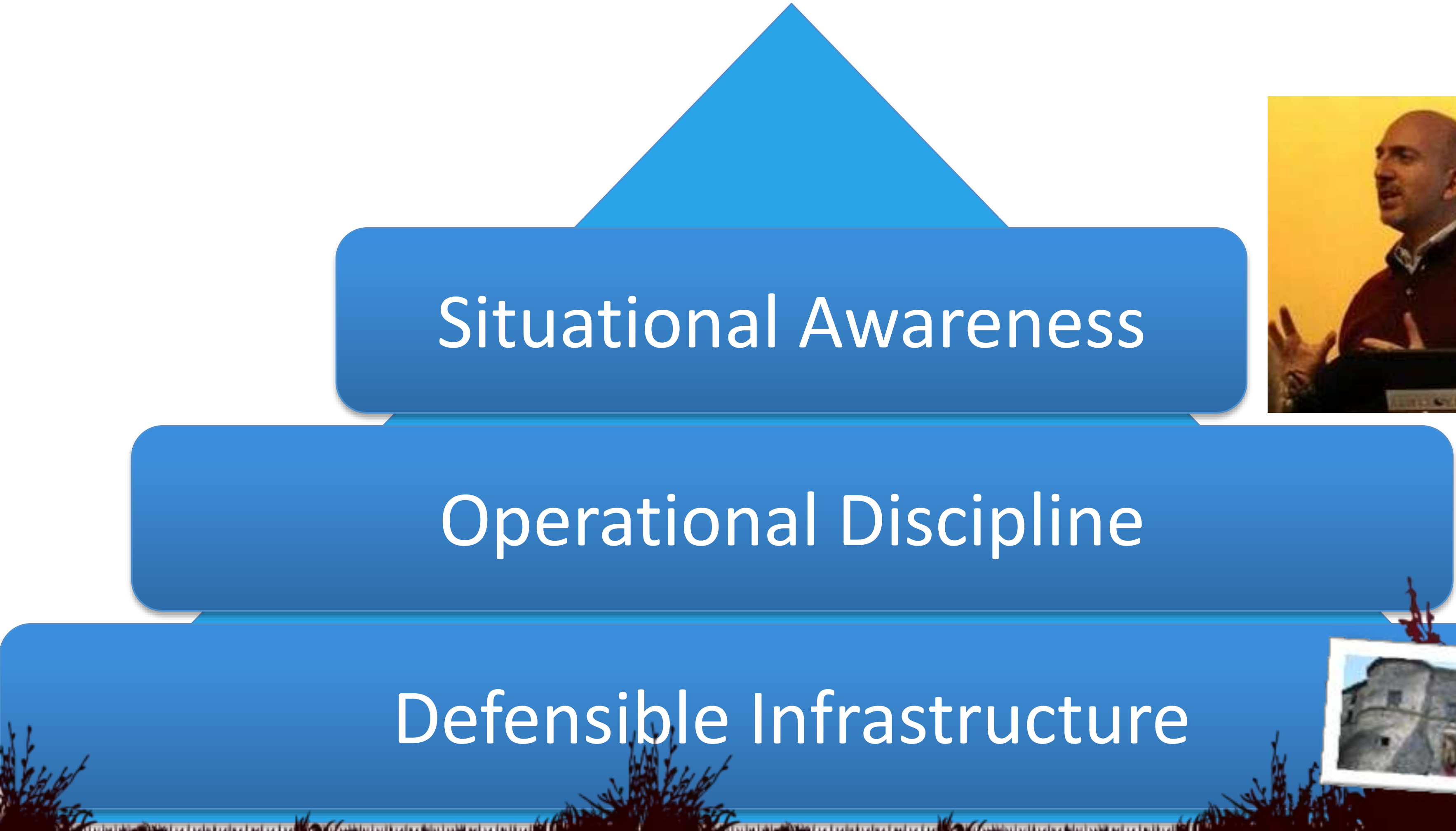
Survival Guide/Pyramid



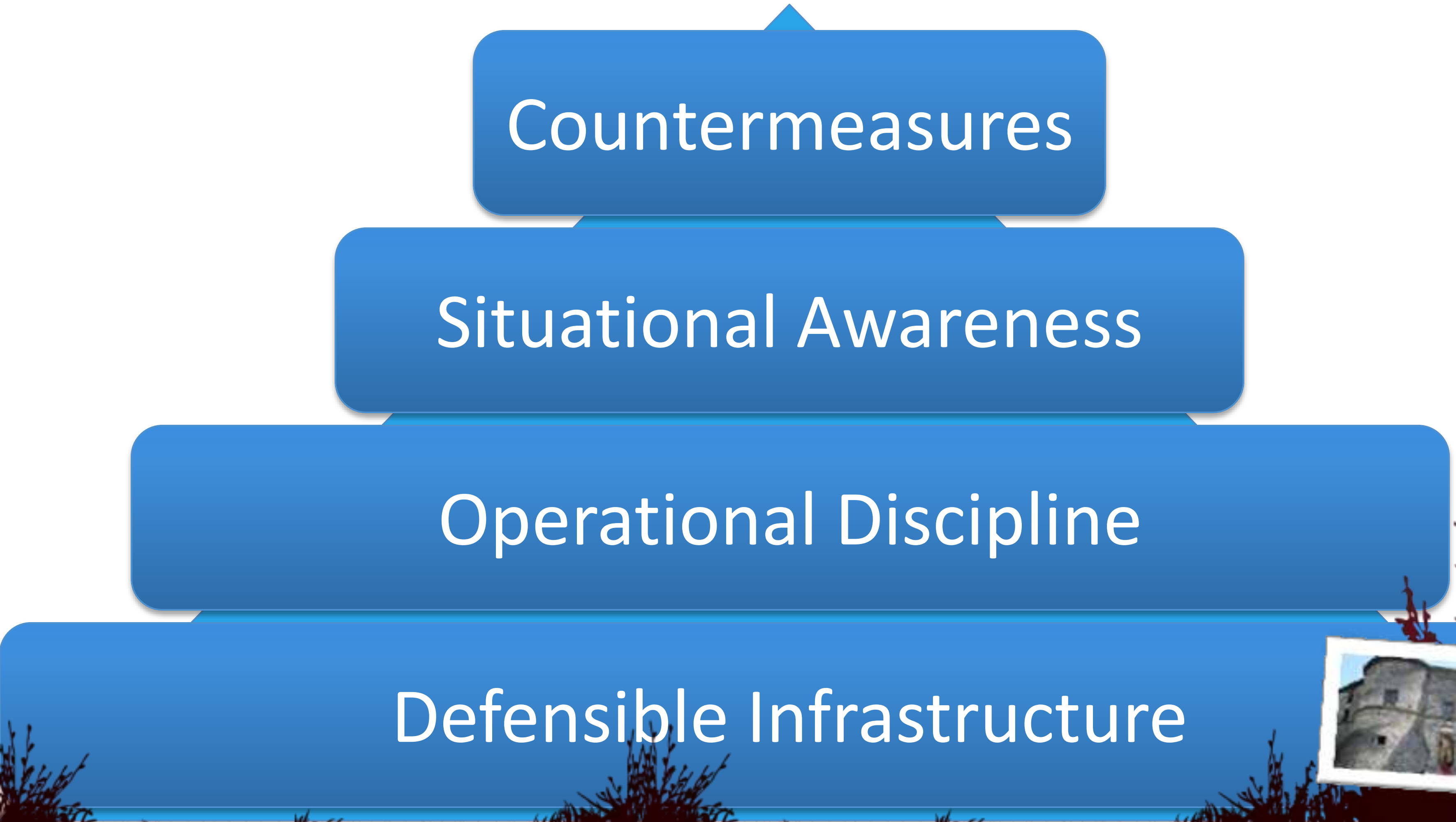
Gene Kim
MULTIPLE AWARD-WINNING CTO, RESEARCHER, VISIBLE OPS CO-AUTHOR, ENTREPRENEUR & FOUNDER OF TRIPWIRE

VISIBLE OPS
SECURITY
ACHIEVING COMMON SECURITY AND IT OPERATIONS OBJECTIVES IN 4 PRACTICAL STEPS

Survival Guide/Pyramid



Survival Guide/Pyramid

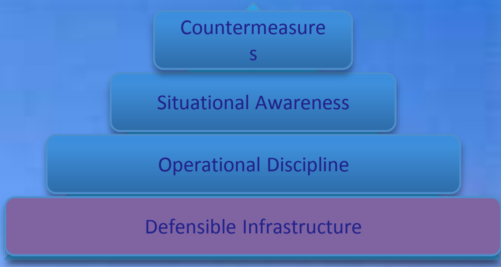


Zombie Proof Housing

http://all-that-is-interesting.com/post/4956385434/the-first-zombie-proof-house

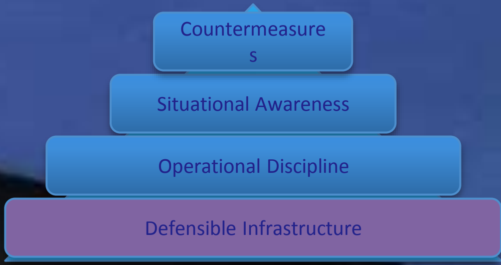
Zombie Proof Housing

http://all-that-is-interesting.com/post/4956385434/the-first-zombie-proof-house



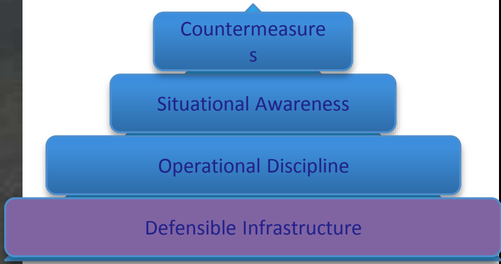
Zombie Proof Housing

http://all-that-is-interesting.com/post/4956385434/the-first-zombie-proof-house



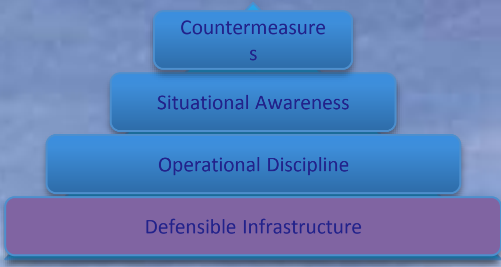
Zombie Proof Housing

http://all-that-is-interesting.com/post/4956385434/the-first-zombie-proof-house



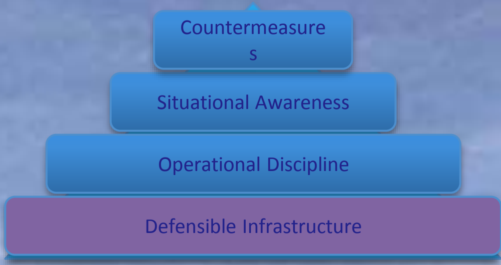
Zombie Proof Housing

http://all-that-is-interesting.com/post/4956385434/the-first-zombie-proof-house



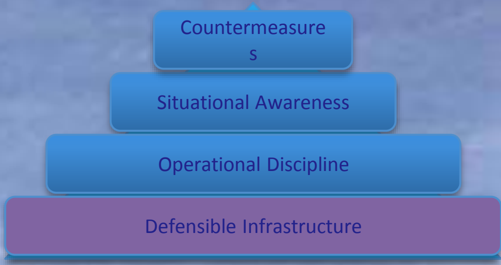
Zombie Proof Housing

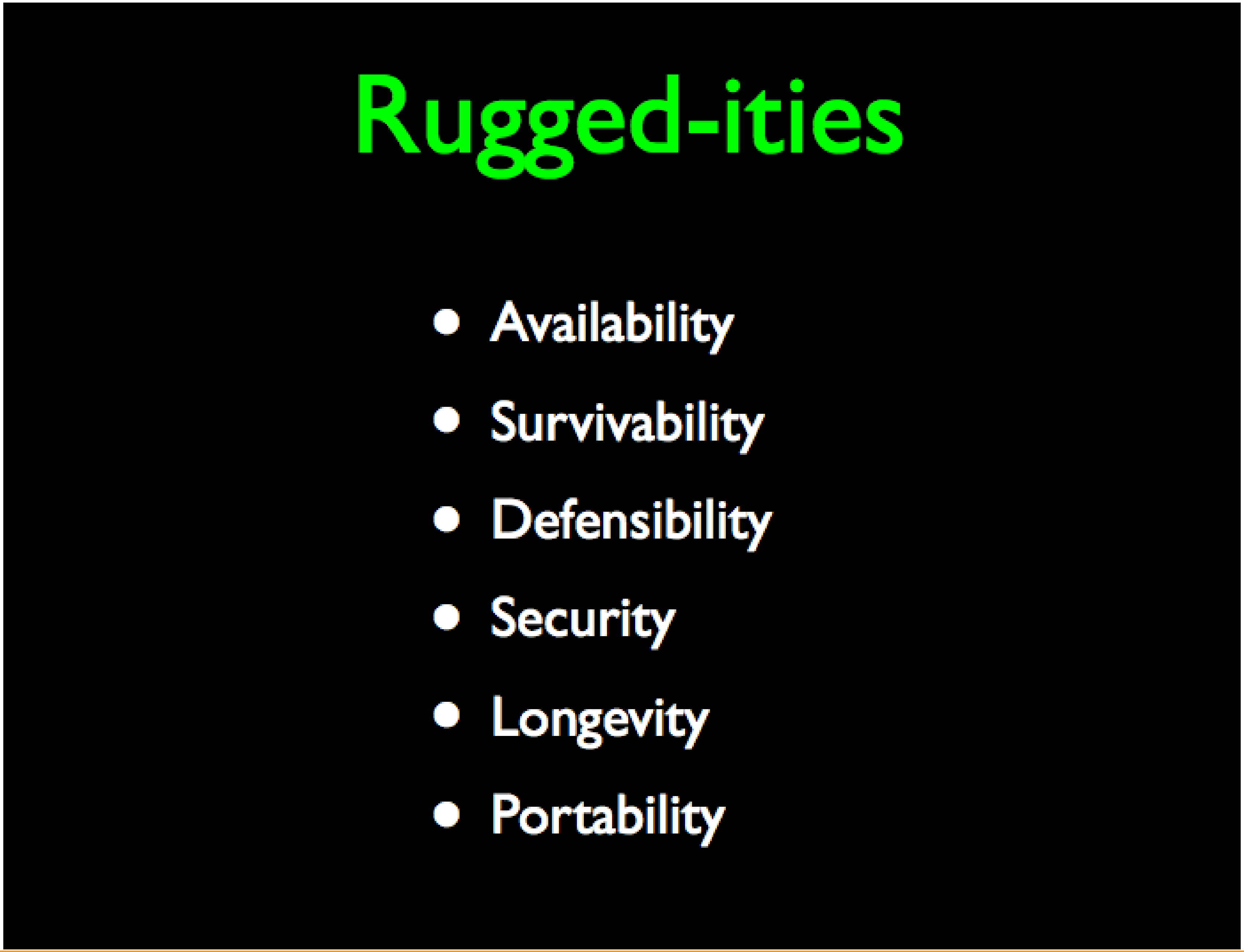
http://all-that-is-interesting.com/post/4956385434/the-first-zombie-proof-house



Zombie Proof Housing

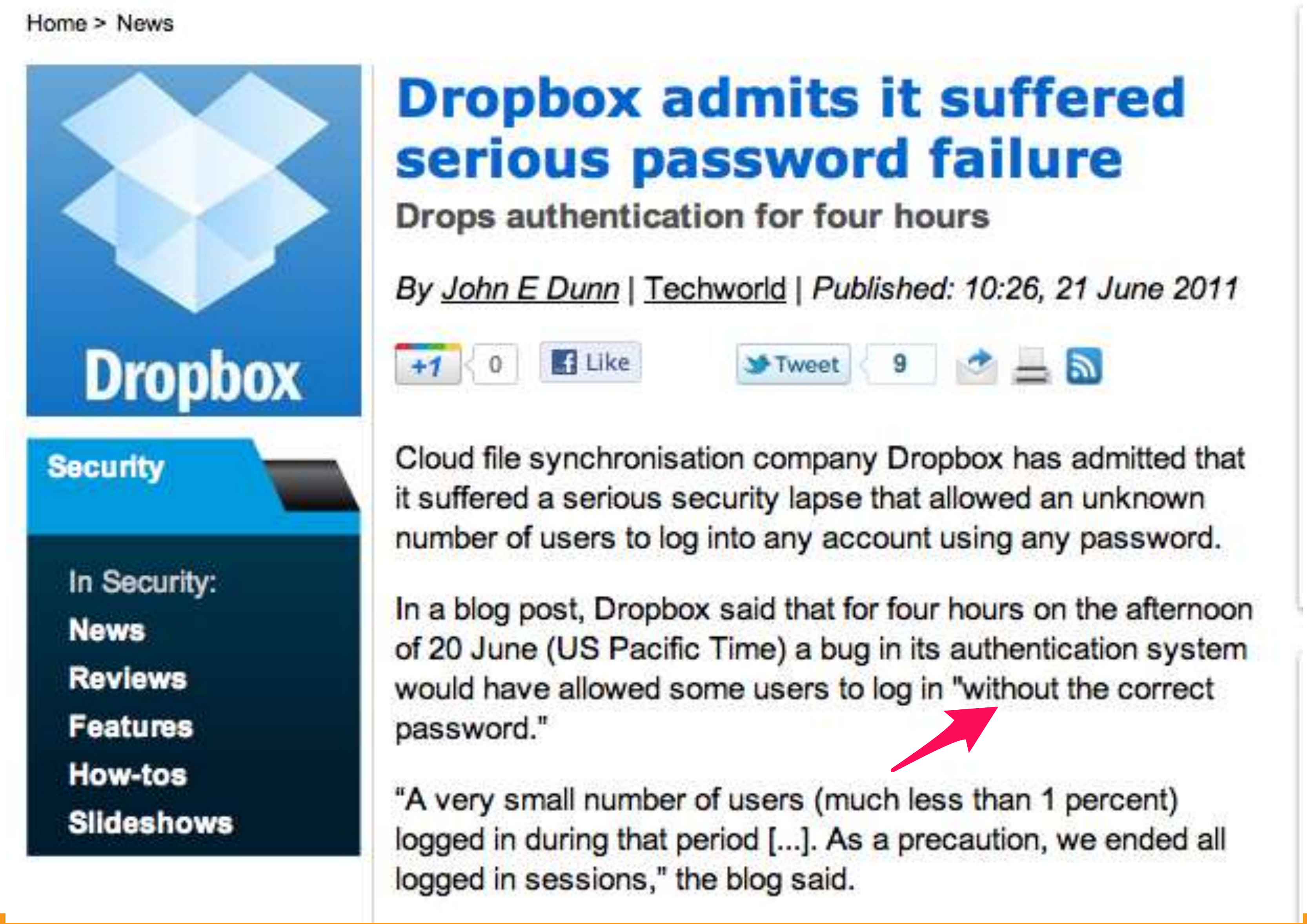
http://all-that-is-interesting.com/post/4956385434/the-first-zombie-proof-house





Rugged-ities

- Availability
- Survivability
- Defensibility
- Security
- Longevity
- Portability



Home > News

Security

In Security:
News
Reviews
Features
How-tos
Slideshows

Dropbox admits it suffered serious password failure

Drops authentication for four hours

By John E Dunn | Techworld | Published: 10:26, 21 June 2011

+1 0 Like Tweet 9

Cloud file synchronisation company Dropbox has admitted that it suffered a serious security lapse that allowed an unknown number of users to log into any account using any password.

In a blog post, Dropbox said that for four hours on the afternoon of 20 June (US Pacific Time) a bug in its authentication system would have allowed some users to log in "without the correct password."

"A very small number of users (much less than 1 percent) logged in during that period [...]. As a precaution, we ended all logged in sessions," the blog said.

DevOps: It’s A Real Movement

- I would never do another startup that didn’t employ DevOps like principles
- It’s not just startups – it’s happening in the enterprise and in public sector, too
- I believe working in DevOps environments will be a necessary skillset 5 years from now

How Do You Do Rugged DevOps?

- "DevOps Cookbook" Authors
  - Patrick DeBois, Mike Orzen, John Willis
- Goals
  - Codify how to start and finish DevOps transformations
  - How does Development, IT Operations and Infosec become dependable partners
  - Describe in detail how to replicate the transformations describe in "When IT Fails: The Novel"

Arc 1: Decrease Cycle Time Of Releases

- Create determinism in the release process
- Move packaging responsibility to development
- Release early and often
- Decrease cycle time
  - Reduce deployment times from 6 hours to 45 minutes
  - Refactor deployment process that had 1300+ steps spanning 4 weeks
- Never again “fix forward,” instead “roll back,” escalating any deviation from plan to Dev
- Ensure environments are properly built before deployment begins
- Control code and environments down the preproduction runways
- Hold Dev, QA, Int, and Staging owners accountable for integrity

Arc 2: Increase Production Resilience

- To preserve and increase throughput, elevate preventive projects and maintenance tasks
- Document all work, changes and outcomes so that it is repeatable
- Protect the flow of planned work (e.g., tickets bouncing around for weeks, causing features to slip into next sprint)
- Ops builds Agile standardized deployment stories
- Maintains adequate situational awareness so that incidents could be quickly detected and corrected
- Standardize unplanned work and escalations
- Continually seek to eradicate unplanned work and increase throughput

Arc 3: Remove Complexity, Attack Surface And Waste

- Elective complexity adds to technical debt
- Infosec (and everyone) wins when we take work out of the system
- Understand where controls reliance is placed and what matters to the business

Meeting The DevOps Leadership Team

- Typically led by Dev, QA, IT Operations and Product Management
- Our ultimate goal is to add value at every step in the flow of work
  - See the end-to-end value flow
  - Shorten and amplify feedback loops
  - Help break silos (e.g., server, networking, database)

Definition: Agile Sprints

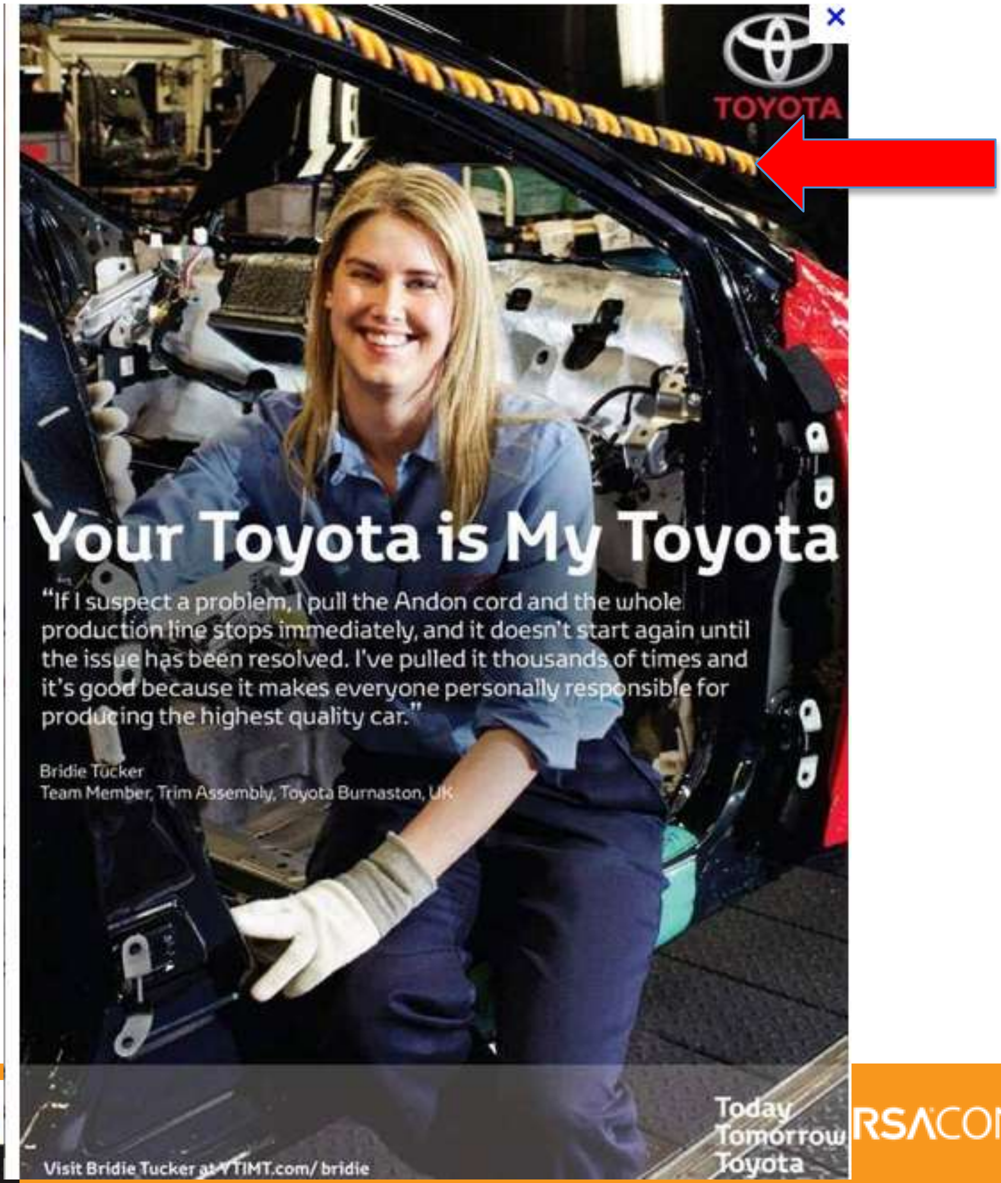
- The basic unit of development in Agile Scrums, typically between one week and one month
- At the end of each sprint, team should have potentially deliverable product

Aha Moment: shipping product implies not just code –
it's the environment, too!

Help Dev And Ops Build Code And Environments

- Dev and Ops work together in Sprint 0 and 1 to create code and environments
  - Create environment that Dev deploys into
  - Create downstream environments: QA, Staging, Production
  - Create testable migration procedures from Dev all the way to production
- Integrate Infosec and QA into daily sprint activities

Definition: Andon Cord


TOYOTA
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

Visit Bridie Tucker at TTIMT.com/ bridie
Today
Tomorrow
Toyota

Integrate Ops Into Dev

- Embed Ops person into Dev structure
  - Describes non-functional requirements, use cases and stories from Ops
  - Responsible for improving “quality at the source” (e.g., reducing technical debt, fix known problems, etc.)
  - Has special responsibility for pulling the Andon cord

Integrate Dev Into Ops

- MobBrowser case study: “Waking up developers at 3am is a great feedback loop: defects get fixed very quickly”
- Goal is to get Dev closer to the customer
  - Infosec can help determine when it’s too close (and when SOD is a requirement)

Keep Shrinking Batch Sizes

- Waterfall projects often have cycle time of one year
- Sprints have cycle time of 1 or 2 weeks
- When IT Operations work is sufficiently fast and cheap, we may decide to decouple deployments from sprint boundaries (e.g., Kanbans)

Definition: Kanban Board

- Signaling tool to reduce WIP and increase flow

IT Operations Increases Process Rigor

- Standardize deployment
- Standardize unplanned work: make it repeatable
- Modify first response: ensure constrained resources have all data at hand to diagnose
- Elevate preventive activities to reduce incidents

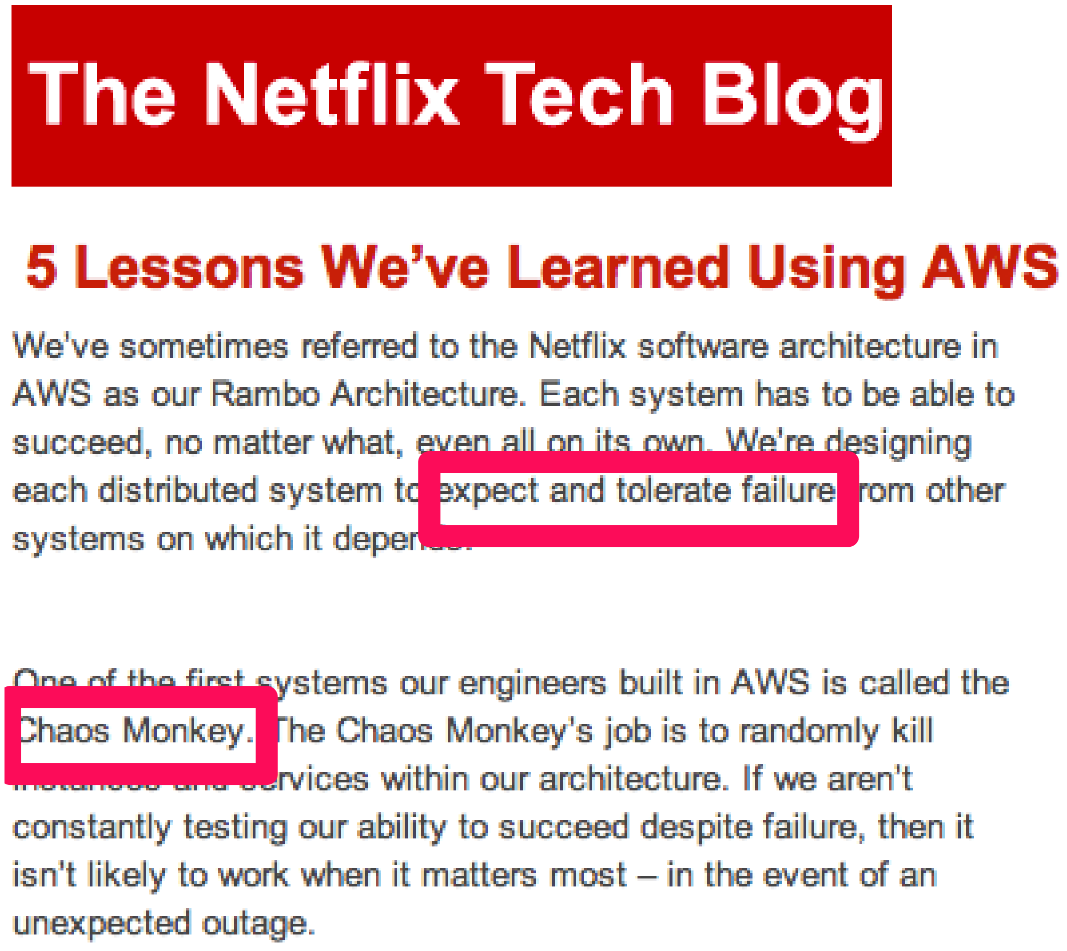
Help Development…

- Help them see downstream effects
  - Unplanned work comes at the expense of planned work
  - Technical debt retards feature throughput
  - Environment matters as much as the code
- Allocate time for fault modeling, asking “what could go wrong?” and implementing countermeasures

Help QA…

- Ensure test plans cover not only code functionality, but also:
  - Suitability of the environment the code runs in
  - The end-to-end deployment process
- Help find variance…
  - Functionality, performance, configuration
  - Duration, wait time and handoff errors, rework, …

Help IT Operations…


The Netflix Tech Blog

5 Lessons We've Learned Using AWS

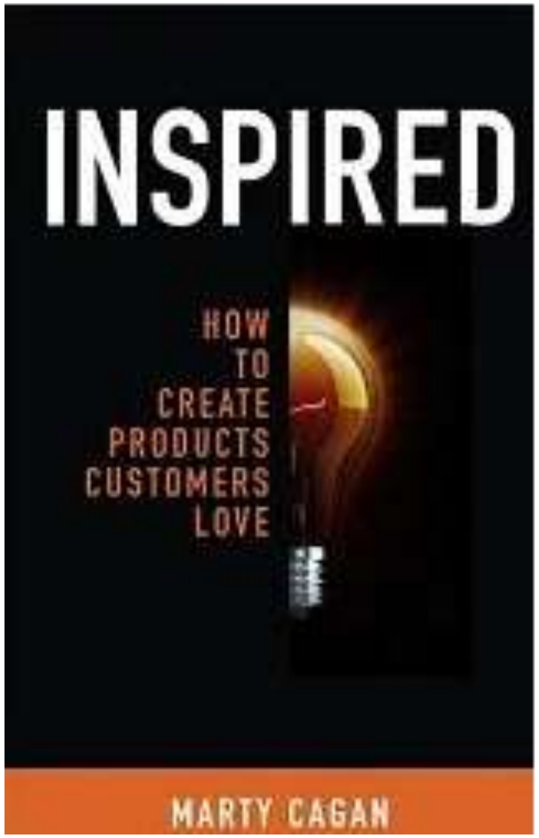
We've sometimes referred to the Netflix software architecture in AWS as our Rambo Architecture. Each system has to be able to succeed, no matter what, even all on its own. We're designing each distributed system to expect and tolerate failure from other systems on which it depends.

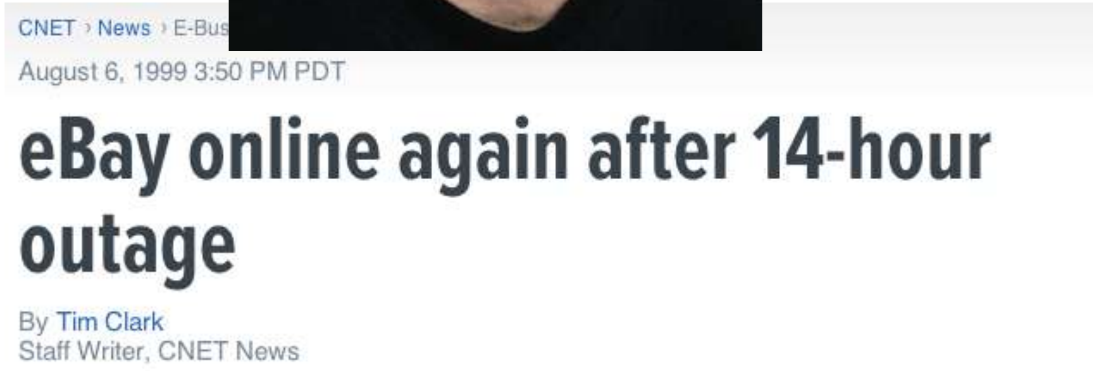
One of the first systems our engineers built in AWS is called the Chaos Monkey. The Chaos Monkey's job is to randomly kill instances and services within our architecture. If we aren't constantly testing our ability to succeed despite failure, then it isn't likely to work when it matters most – in the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

- "The best way to avoid failure is to fail constantly"
- Harden the production environment
- Have scheduled drills to "crash the data center"
- Create your "chaos monkeys" to introduce faults into the system (e.g., randomly kill processes, take out servers, etc.)
- Rehearse and improve responding to unplanned work
  - NetFlix: Hardened AWS service
  - StackOverflow
  - Amazon firedrills (Jesse Allspaw)
  - The Monkey (Mac)

You Don’t Choose Chaos Monkey… Chaos Monkey Chooses You

Help Product Management…




CNET › News › E-Bus

August 6, 1999 3:50 PM PDT

eBay online again after 14-hour outage

By Tim Clark
Staff Writer, CNET News

Lesson: Allocate 20% of Dev cycles to paying down technical debt

What Does Rugged DevOps Feel Like?

Case Studies And Early Indicators

- Almost every major Internet online services company
- VERACODE Rapid SaaS Fix Blog Post
  - http://www.veracode.com/blog/2012/01/vulnerability-response-done-right/
- Pervasive Monitoring
  - Analytics at LinkedIn viewed by CEO daily: LinkedIn Engineering: “The Birth Of inGraphs: Eric The Intern”

Applying RuggedDevOps

RSACONFERENCE2012

Things To Put Into Practice Tomorrow

- Identify your Dev/Ops/QA/PM counterparts
- Discuss your mutual interdependence and shared objectives
- Harden and instrument the production builds
- Integrate automated security testing into the build and deploy mechanisms
- Create your Evil/Hostile/Fuzzy Chaos Monkey
- Cover your untested branches
- Enforce the 20% allocation of Dev cycles to non-functional requirement

Resources

- From the IT Process Institute www.itpi.org
  - Both Visible Ops Handbooks
  - ITPI IT Controls Performance Study
- Rugged Software by Corman, et al: http://ruggedsoftware.org
- "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation" by Humble, Farley
- Follow us…
  - @JoshCorman, @RealGeneKim
  - mailto:genek@realgenekim.me
  - http://realgenekim.me/blog

Interested In “The DevOps Cookbook?”

Give Gene your business card, and get exclusive access to the first 100 pages of "*When IT Fails: The Novel*" and "*The DevOps Cookbook*" for free

We’ll send it to you as soon as it’s ready!

Thank You

RSA CONFERENCE 2012

Appendix

RSA CONFERENCE 2012

Common Traits of High Performers

Culture of…

Change management

- Integration of IT operations/security via problem/change management
- Processes that serve both organizational needs and business objectives
- Highest rate of effective change

Causality

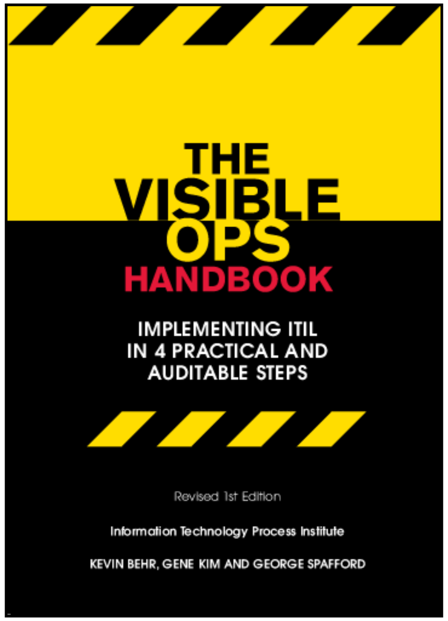
- Highest service levels (MTTR, MTBF)
- Highest first fix rate (unneeded rework)

Compliance and continual reduction of operational variance

- Production configurations
- Highest level of pre-production staffing
- Effective pre-production controls
- Effective pairing of preventive and detective controls

Visible Ops: Playbook of High Performers

- The IT Process Institute has been studying high-performing organizations since 1999
  - What is common to all the high performers?
  - What is different between them and average and low performers?
  - How did they become great?
- Answers have been codified in the Visible Ops Methodology
- The “Visible Ops Handbook” is available from the ITPI


THE VISIBLE OPS HANDBOOK
IMPLEMENTING ITIL IN 4 PRACTICAL AND AUDITABLE STEPS
Revised 1st Edition
Information Technology Process Institute
KEVIN BEHR, GENE KIM AND GEORGE SPAFFORD

www.ITPI.org

What These Breakthroughs Look Like

A Reframed IT Operations Problem Statement

- Increase flow from Dev to Production
  - Increase throughput
  - Decrease WIP
- Our goal is to create a system of operations that allows
  - Planned work to quickly move to production
  - Ensure service is quickly restored when things go wrong
  - Information security built in every stage of Development, Project Management, and IT Operations
- How does this relate to Visible Ops?
  - We focused much on “unplanned work”
  - What’s happening to all the planned work?
  - At any given time, what should IT Ops be working on?
  - Now we are focusing on the flow of planned work

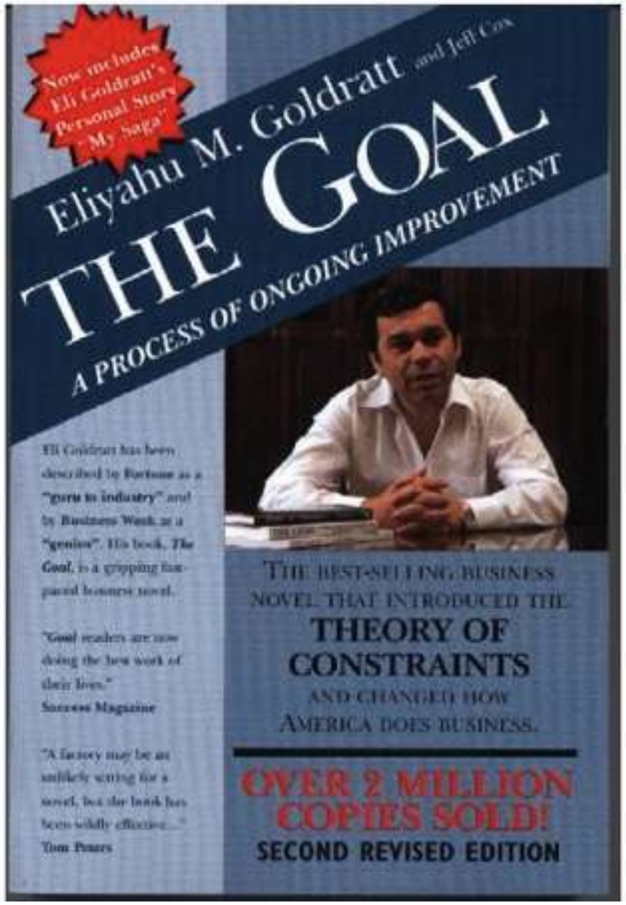
How To Reduce The Transformation Activation Energy

Framing The Moral Crusade

By The Visible Ops Team:
Gene Kim, Kevin Behr, George Spafford

The Theory of Constraints Approach To Visible Ops

- Dr. Goldratt wrote *The Goal* in 1984, describing Alex's challenge to fix his plant's cost and due date issues within 90 days
- Some tenets that went against common wisdom:
  - Every flow of work has a constraint/bottleneck
  - Any improvement not made at the bottleneck is merely an illusion
  - Fallacy of cost accounting as operational management tool


Now includes Eli Goldratt's Personal Story "My Saga"
Eliyahu M. Goldratt and Jeff Cox
THE GOAL
A PROCESS OF ONGOING IMPROVEMENT
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

Interested?

If you’re interested in *When IT Fails: The Novel* or *The DevOps Cookbook*, signup for the list at http://whenitfails.org

Or:

```
# mail genek@realgenekim.me
Subject: [ slides | research | list ]
```