Why killing your current infosec program is a good idea

a practical guide in 5 easy steps

Dave van Stein | ISACA NL Square Table

Who am I

CSSLP® Certified Secure Software Lifecycle Professional

CIPT Certified Information Privacy Technologist iapp

dvanstein@xebia.com

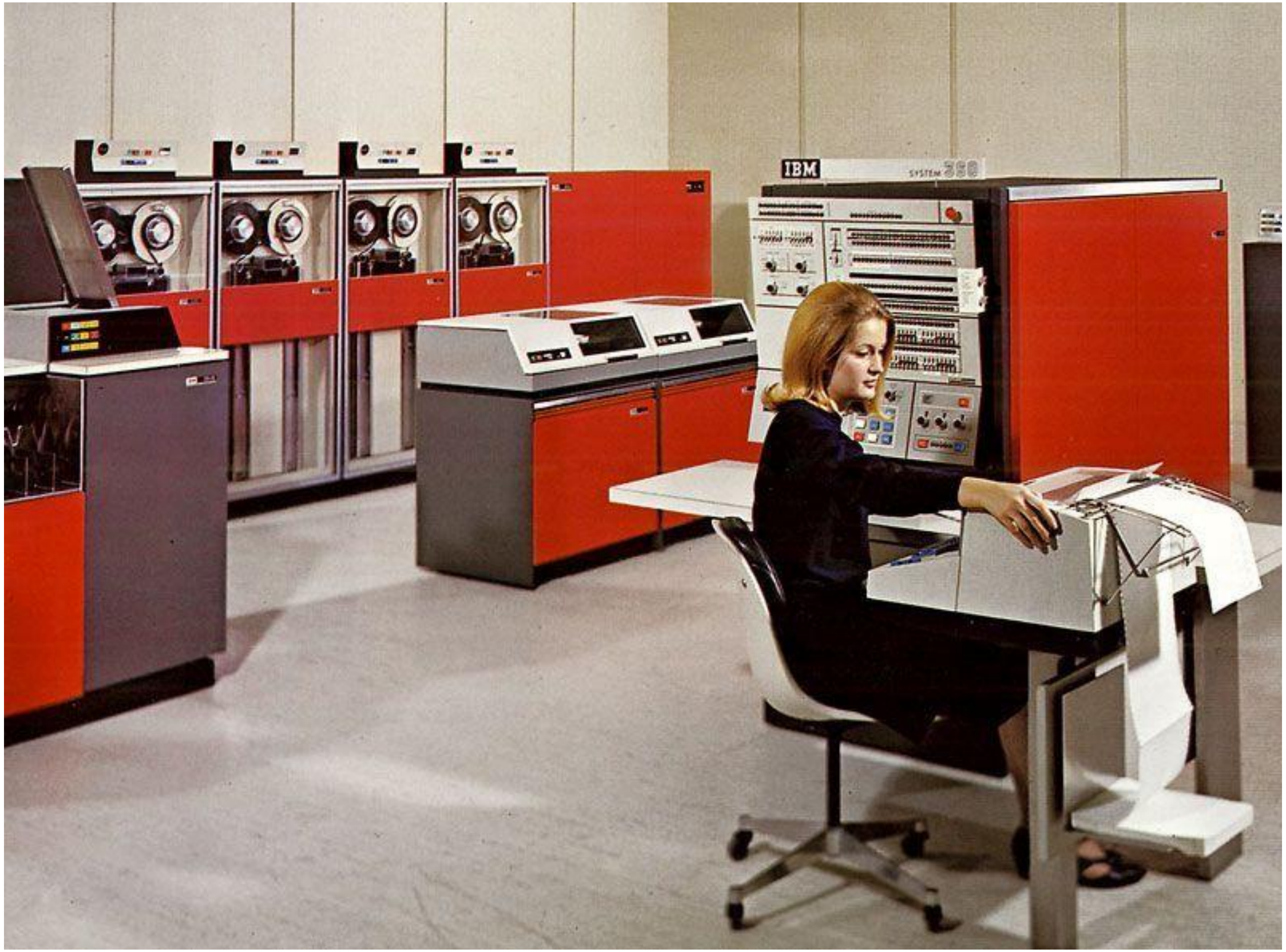
@Dave_von_S

nl.linkedin.com/in/dvstein

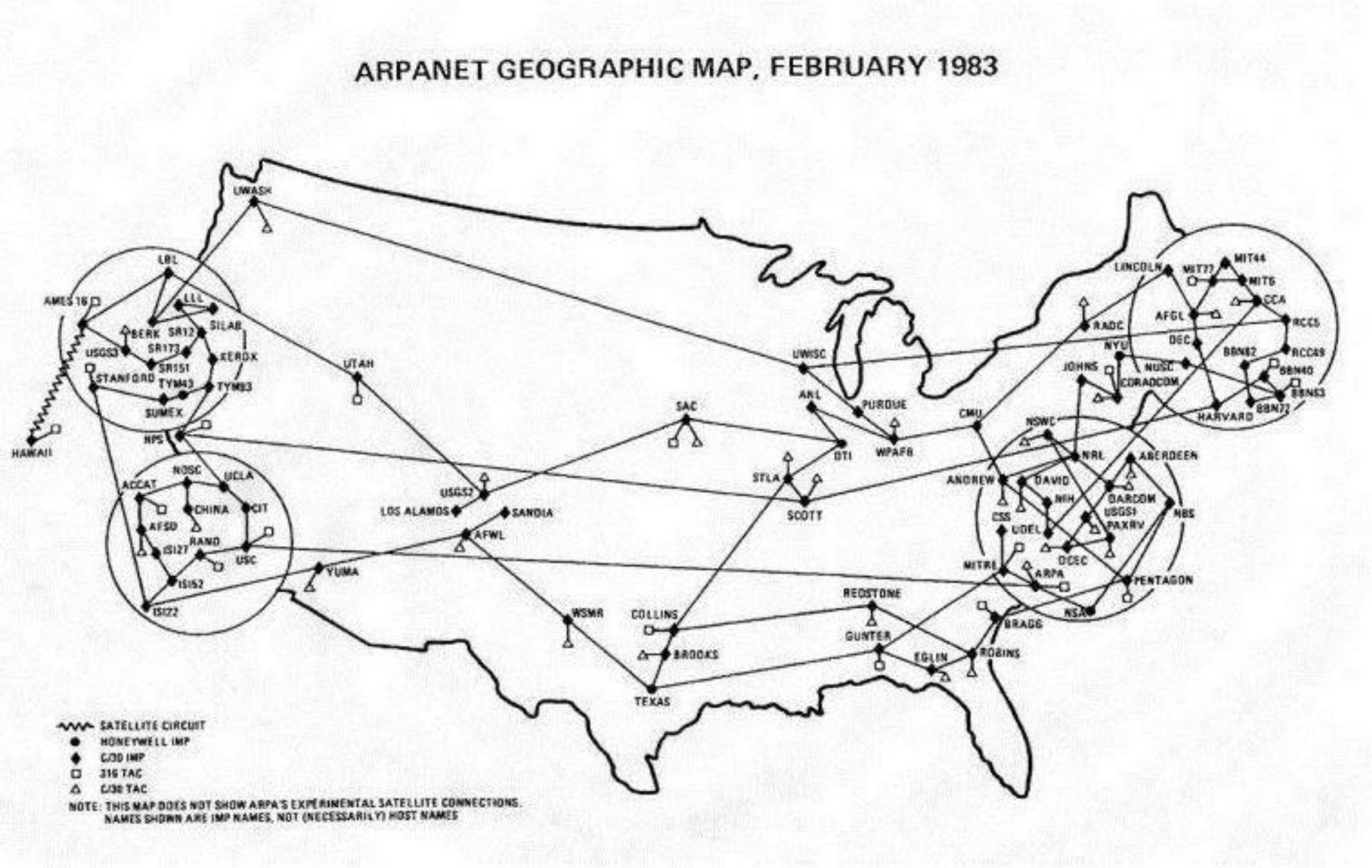
Tonight’s story: appsec

```
WP_Query( array( 'orderby
) );
have_posts() ) : $popularposts
function_exists("has_post_thumbnail")
class="col-md-6 col-sm-6 col-xs-12 sidebar
the_post_thumbnail( '', array( 'class
<?php
<?php endif; ?>
<div class="col-md-6 col-
<h3><a href="<?php
<p> <?php echo
```

How it started



1983: internet



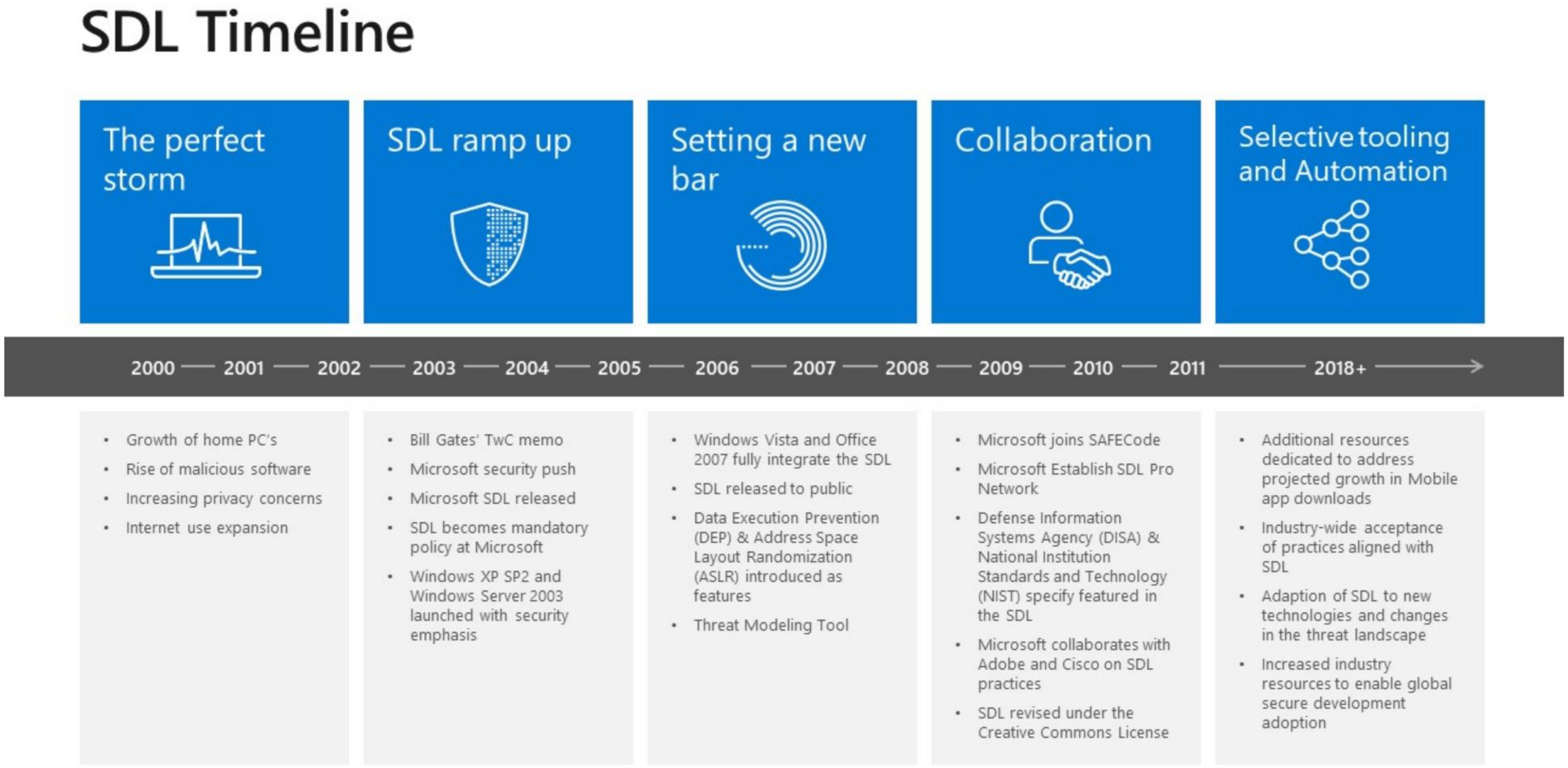
ARPANET GEOGRAPHIC MAP, FEBRUARY 1983

NOTE: THIS MAP DOES NOT SHOW ARPA'S EXPERIMENTAL SATELLITE CONNECTIONS.
NAMES SHOWN ARE IMP NAMES, NOT (NECESSARILY) HOST NAMES

1989: world wide web

1996: first CISO

2000: Microsoft releases the secure SDL



SDL Timeline

2000 — 2001 — 2002 — 2003 — 2004 — 2005 — 2006 — 2007 — 2008 — 2009 — 2010 — 2011 ——— 2018+ ——→

The perfect storm

- Growth of home PC's
- Rise of malicious software
- Increasing privacy concerns
- Internet use expansion

SDL ramp up

- Bill Gates' TwC memo
- Microsoft security push
- Microsoft SDL released
- SDL becomes mandatory policy at Microsoft
- Windows XP SP2 and Windows Server 2003 launched with security emphasis

Setting a new bar

- Windows Vista and Office 2007 fully integrate the SDL
- SDL released to public
- Data Execution Prevention (DEP) & Address Space Layout Randomization (ASLR) introduced as features
- Threat Modeling Tool

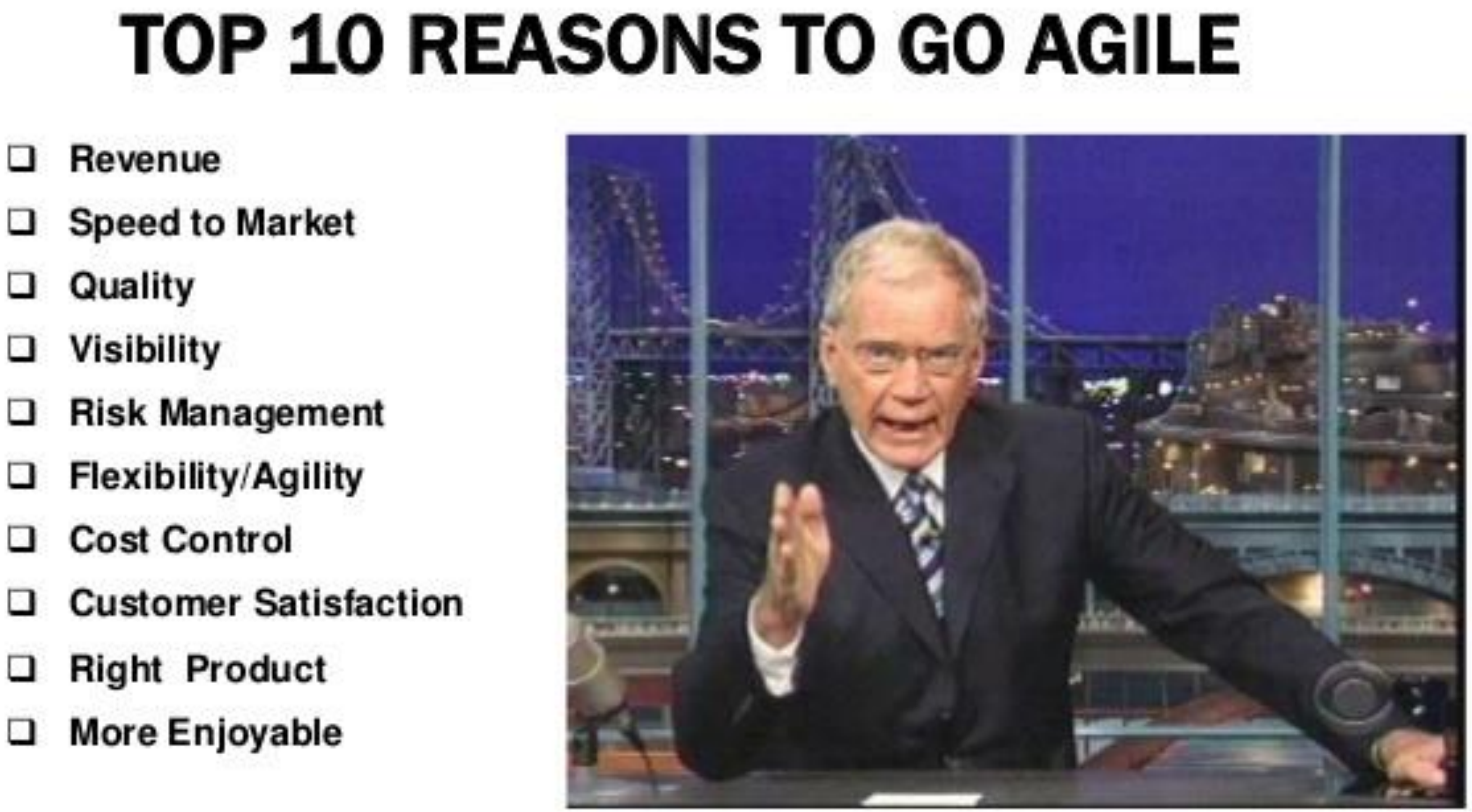
Collaboration

- Microsoft joins SAFECode
- Microsoft Establish SDL Pro Network
- Defense Information Systems Agency (DISA) & National Institution Standards and Technology (NIST) specify featured in the SDL
- Microsoft collaborates with Adobe and Cisco on SDL practices
- SDL revised under the Creative Commons License

Selective tooling and Automation

- Additional resources dedicated to address projected growth in Mobile app downloads
- Industry-wide acceptance of practices aligned with SDL
- Adaption of SDL to new technologies and changes in the threat landscape
- Increased industry resources to enable global secure development adoption

2004: the agile manifesto



TOP 10 REASONS TO GO AGILE

- □ Revenue
- □ Speed to Market
- □ Quality
- □ Visibility
- □ Risk Management
- □ Flexibility/Agility
- □ Cost Control
- □ Customer Satisfaction
- □ Right Product
- □ More Enjoyable

And here we are …

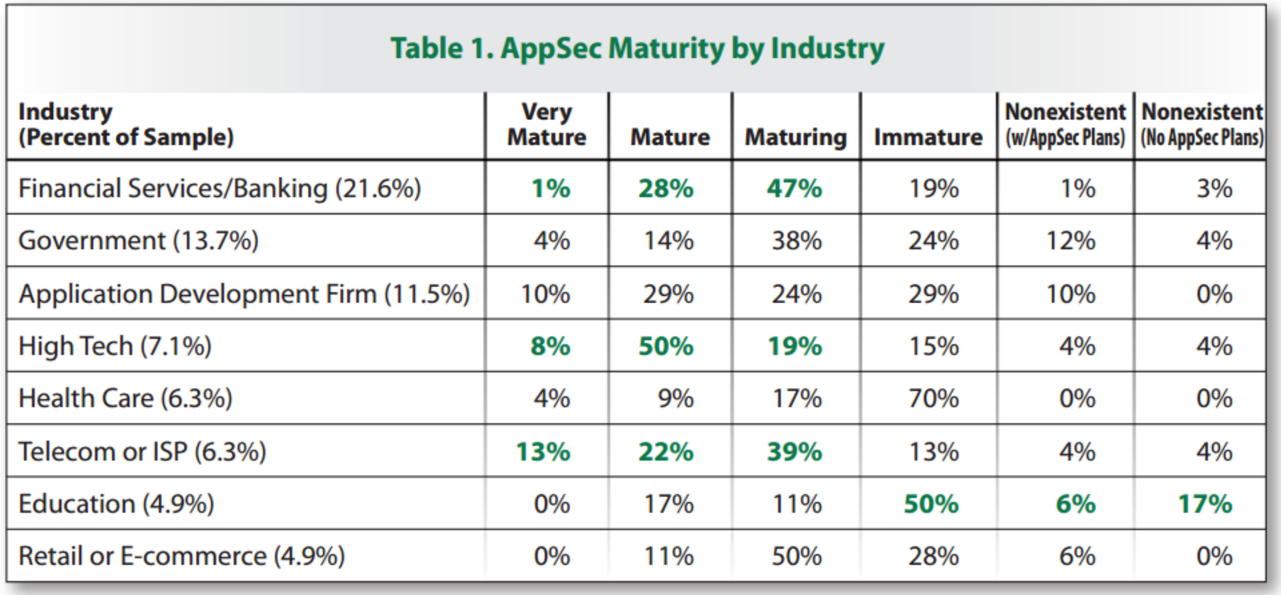
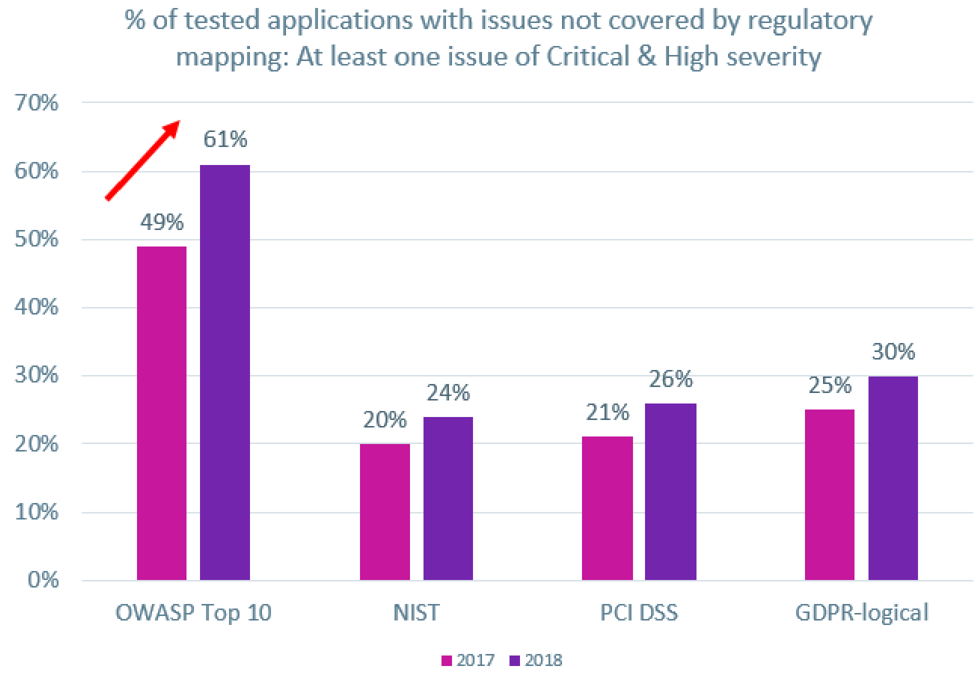


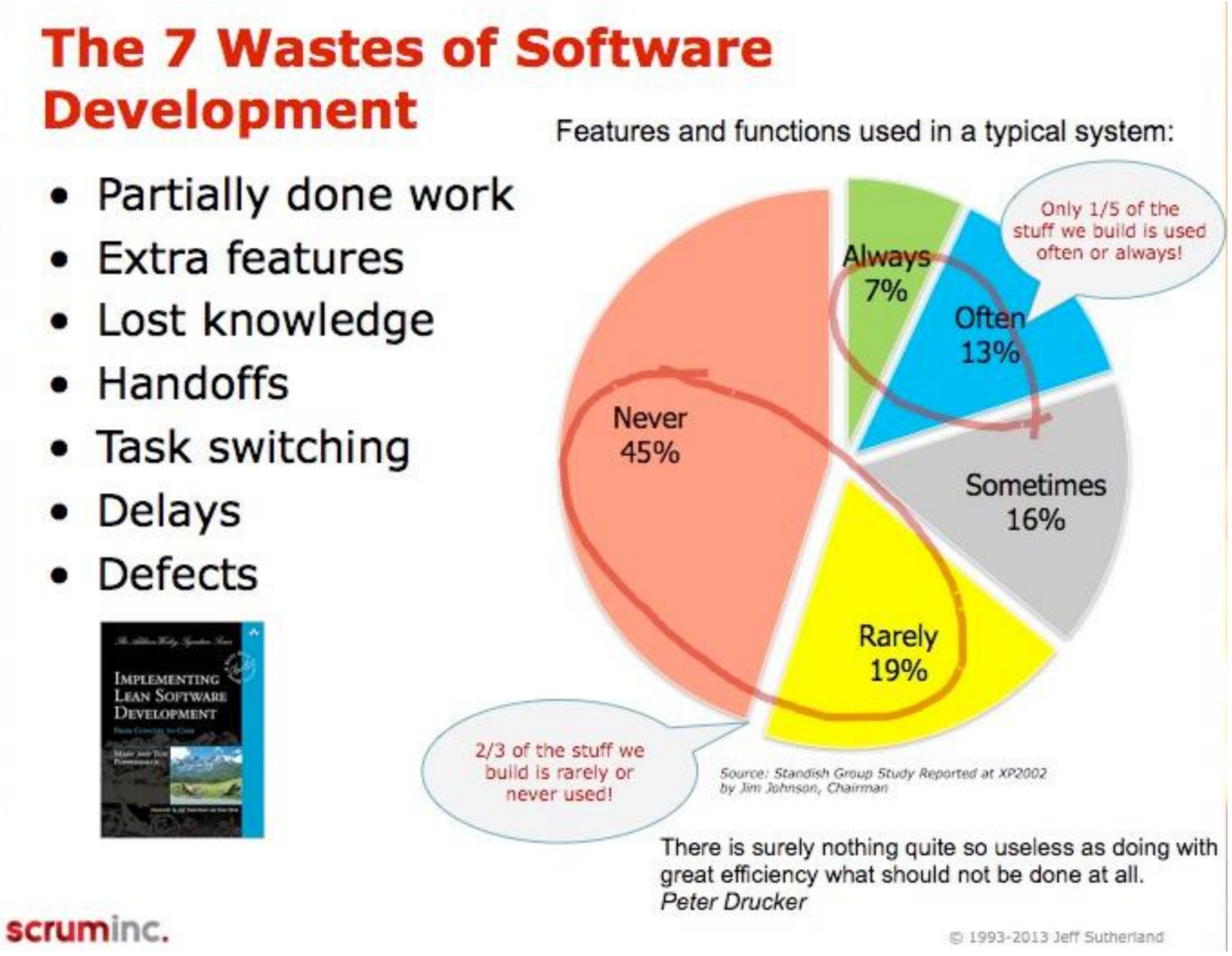
Table 1. AppSec Maturity by Industry

Industry (Percent of Sample)	Very Mature	Mature	Maturing	Immature	Nonexistent (w/AppSec Plans)	Nonexistent (No AppSec Plans)
Financial Services/Banking (21.6%)	**1%**	**28%**	**47%**	19%	1%	3%
Government (13.7%)	4%	14%	38%	24%	12%	4%
Application Development Firm (11.5%)	10%	29%	24%	29%	10%	0%
High Tech (7.1%)	**8%**	**50%**	**19%**	15%	4%	4%
Health Care (6.3%)	4%	9%	17%	70%	0%	0%
Telecom or ISP (6.3%)	**13%**	**22%**	**39%**	13%	4%	4%
Education (4.9%)	0%	17%	11%	**50%**	**6%**	**17%**
Retail or E-commerce (4.9%)	0%	11%	50%	28%	6%	0%



% of tested applications with issues not covered by regulatory mapping: At least one issue of Critical & High severity

And most of it is not even used!



The 7 Wastes of Software Development

- Partially done work
- Extra features
- Lost knowledge
- Handoffs
- Task switching
- Delays
- Defects

Features and functions used in a typical system:

Only 1/5 of the stuff we build is used often or always!

2/3 of the stuff we build is rarely or never used!

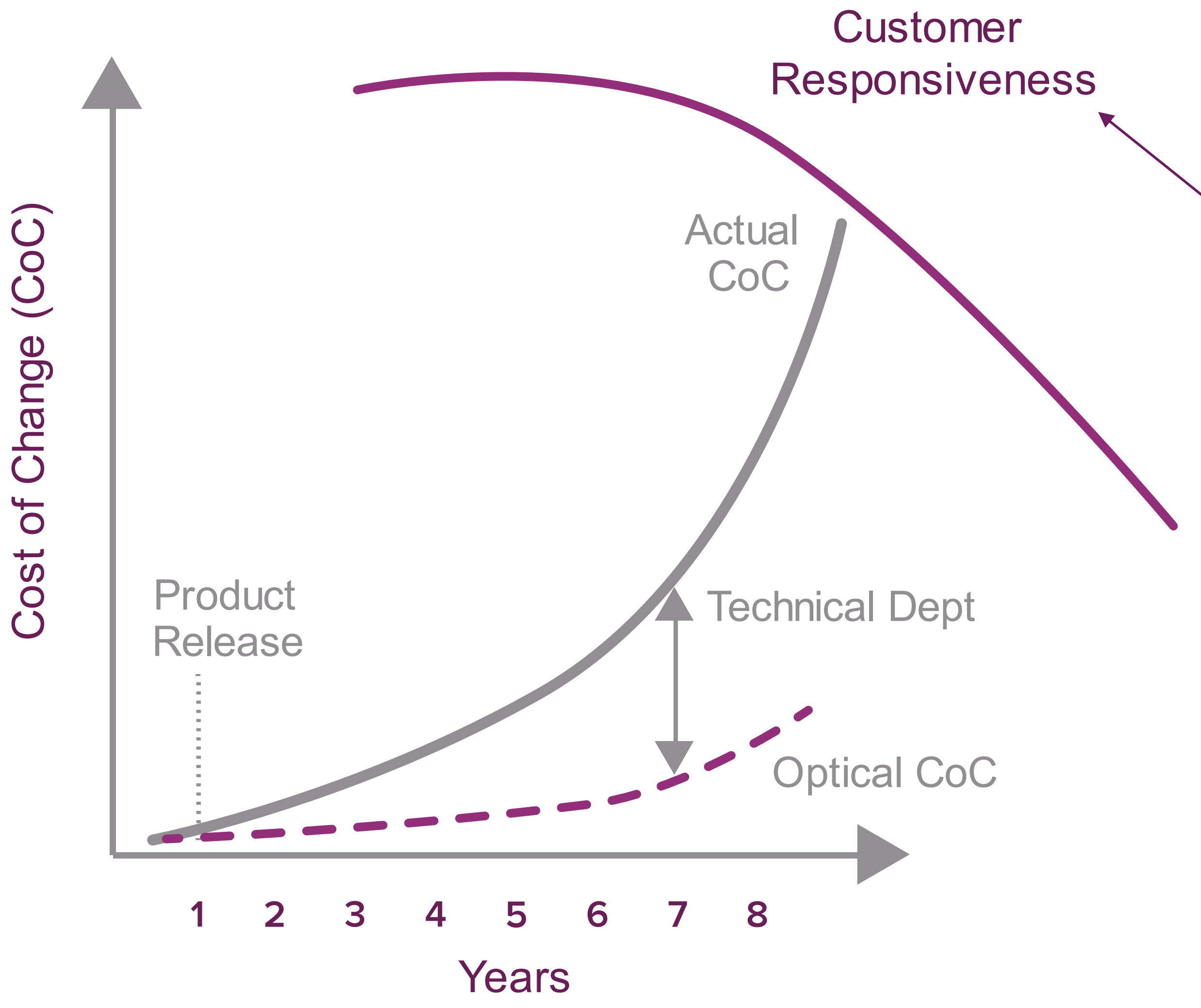
Source: Standish Group Study Reported at XP2002 by Jim Johnson, Chairman

There is surely nothing quite so useless as doing with great efficiency what should not be done at all.
Peter Drucker

scruminc.

© 1993-2013 Jeff Sutherland

And it gets worse every day



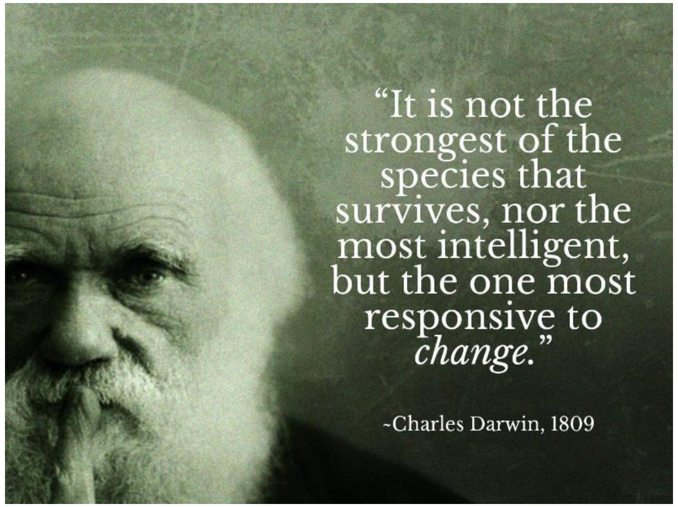
This is your ability to respond, in other words: your reason of existence

We’ve seen this movie!

“What happened?!”

Coping with change: 2 ways

The robust can handle adversity but stay the same.



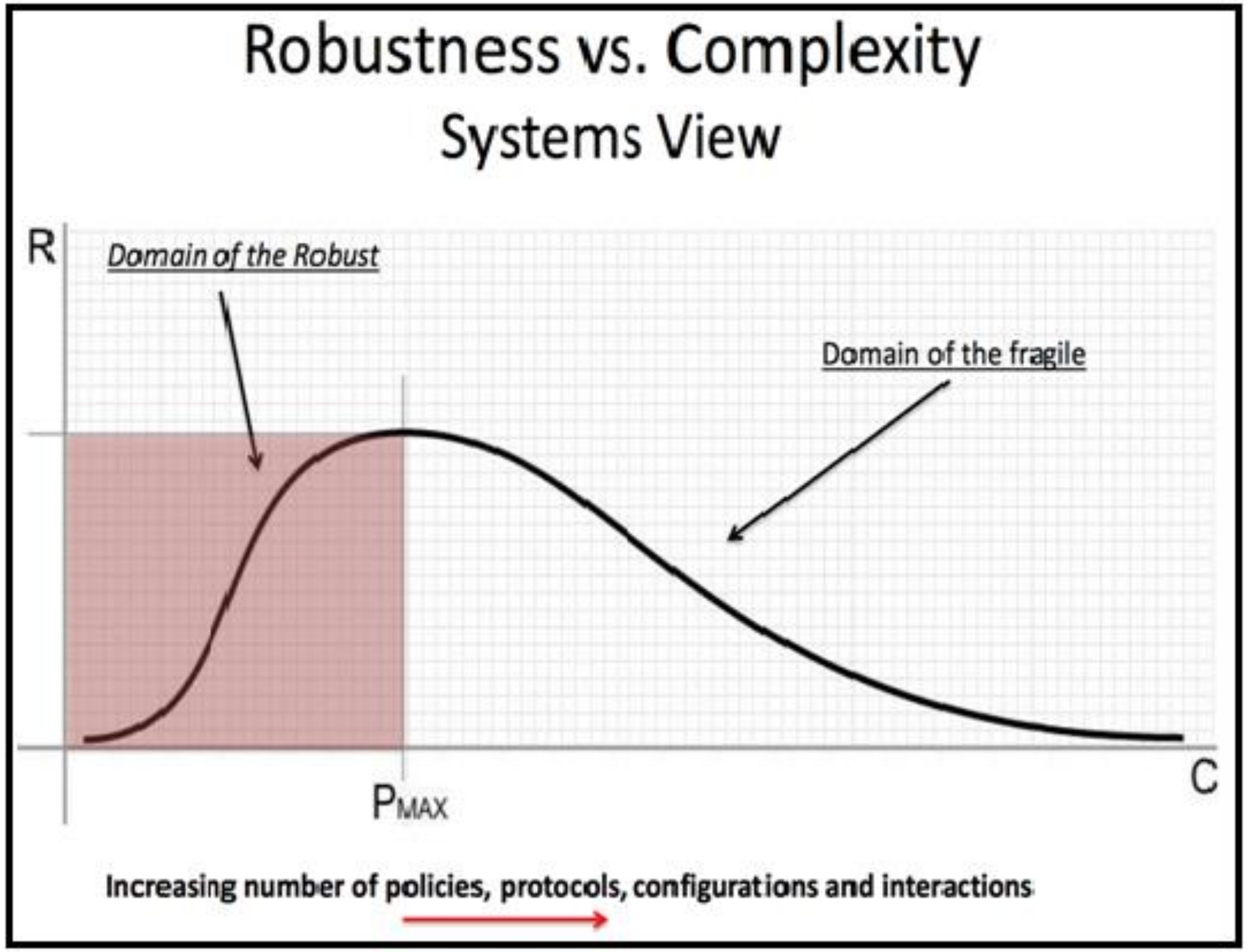
"It is not the strongest of the species that survives, nor the most intelligent, but the one most responsive to *change.*"

-Charles Darwin, 1809

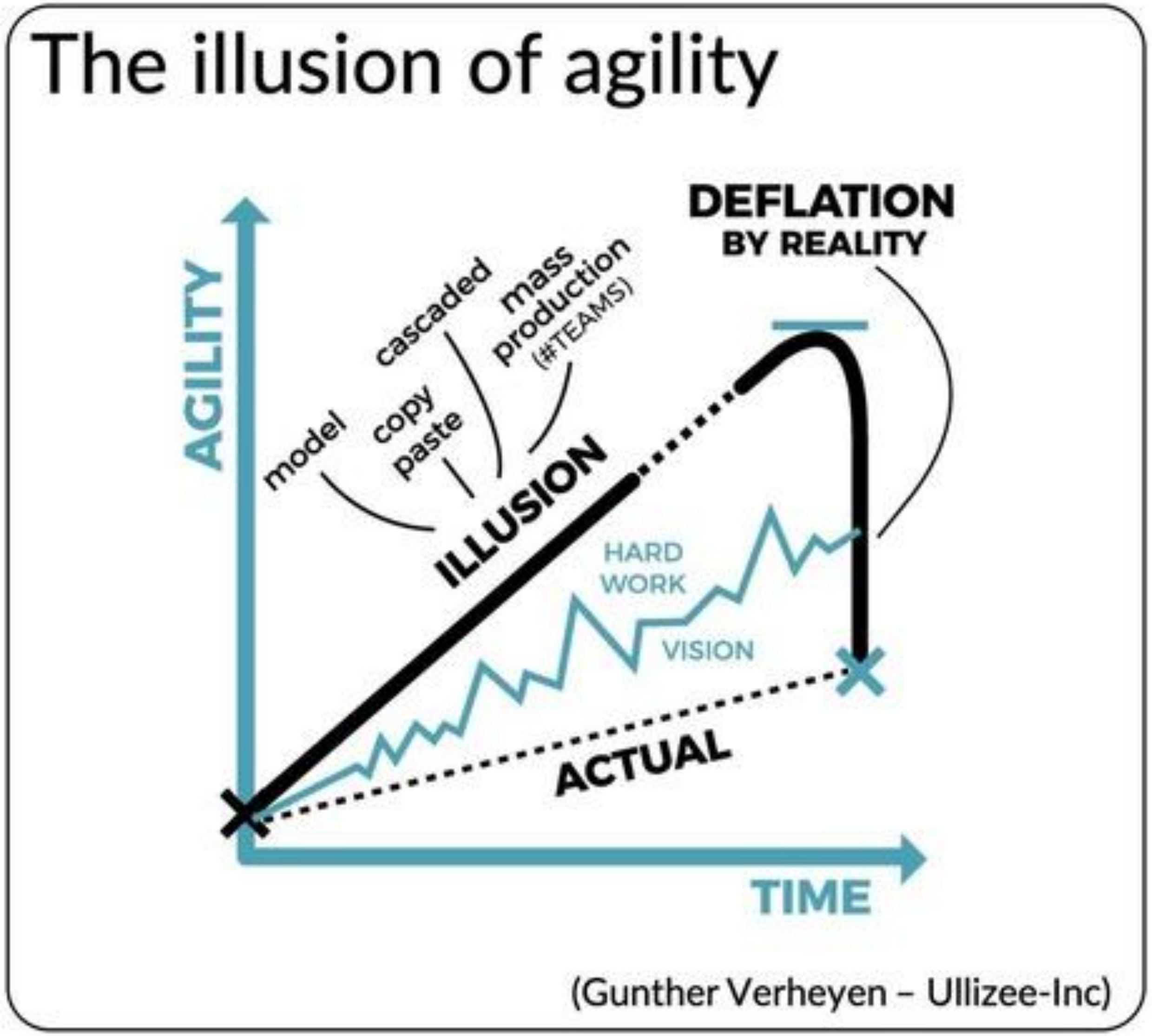
The agile move and adapt rapidly.

Development and Security: how we view ourselves

Why security is failing

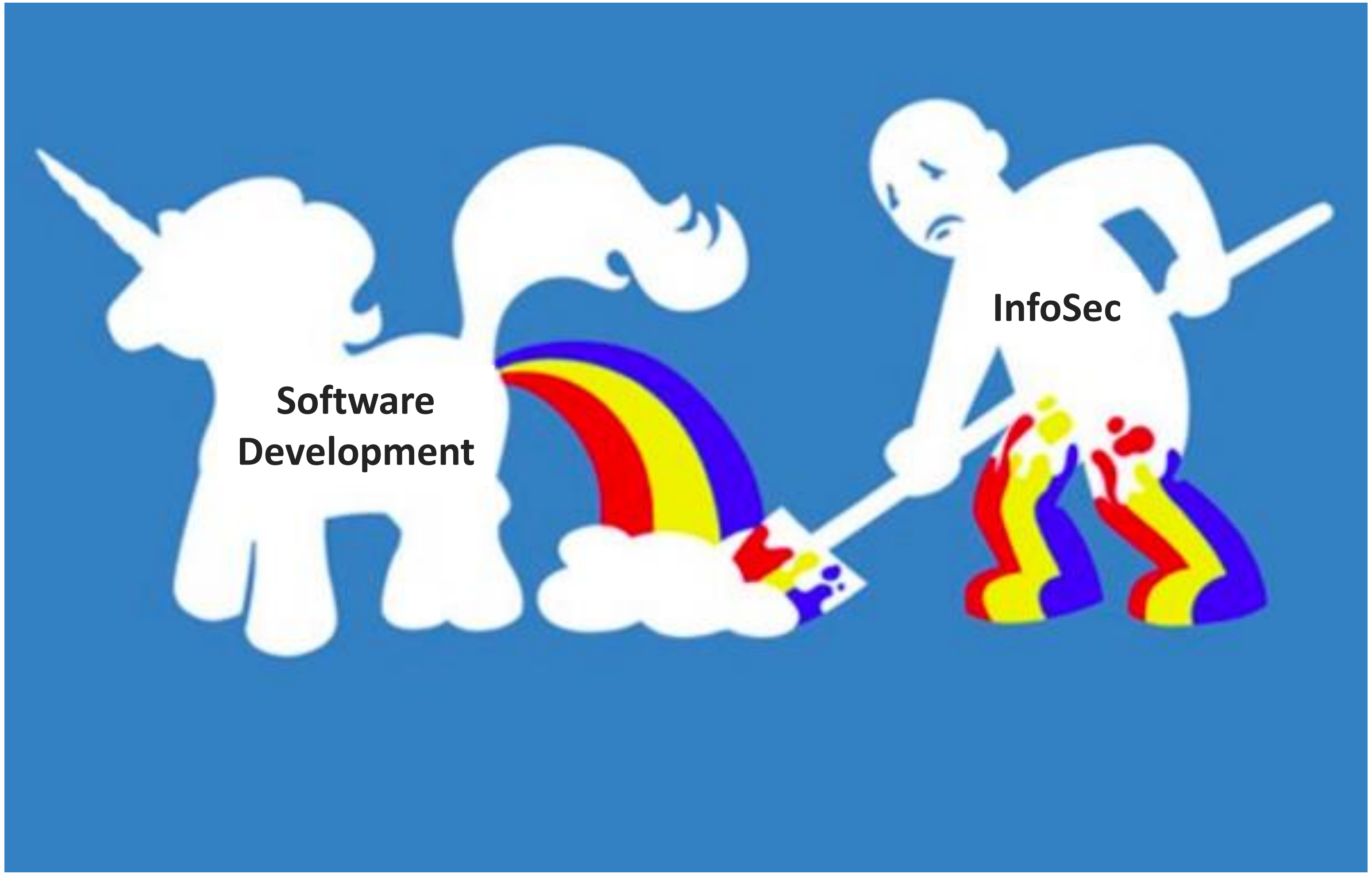


Why software development is failing



Development and Security: how we view the other

The bitter reality



WE NEED
A CHANGE

Multiple reasons

HIPAA
Health Insurance Portability and Accountability Act

Sarbanes-Oxley
Financial and Accounting Disclosure Information

London Evening Standard

News

Computer worm 'designed to blow up nuclear power stations'

A computer worm designed to blow up nuclear power stations has been uncovered by IT security experts.

... a new frontier in computer attacks, and have called it the first "cyber weapon".

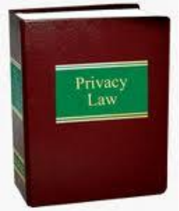
PCI Security Standards Council™

DigiD

ÎÒÊÇÖÐ'úÈË!
Beat down Imperialism of American!
Powered by
Honker Union of China
...... H.U.C ......

CERTIFIED CERTIFIED CERTIFIED
SAS 70 Type II

NATIONAL SECURITY AGENCY
UNITED STATES OF AMERICA



</xssed>
xss attacks information

ISO 9001
ISO 27001
BUREAU VERITAS
Certification
BUREAU VERITAS 1828

FDA U.S. Food and Drug Administration
Protecting and Promoting *Your* Health

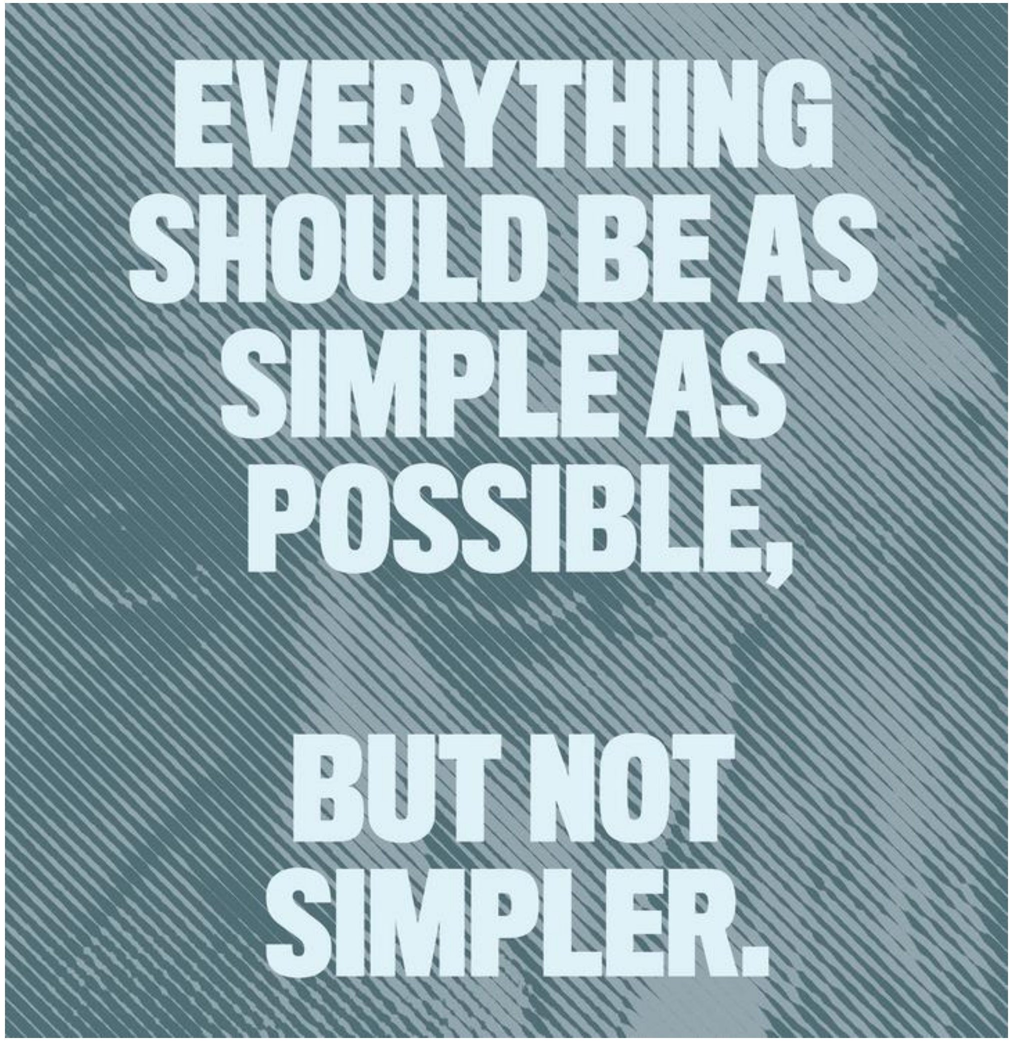
UWV werken aan perspectief

Logius
Ministerie van Binnenlandse Zaken en Koninkrijksrelaties

23% College And University Students Hack IT Systems

100% secure?

Find a new balance



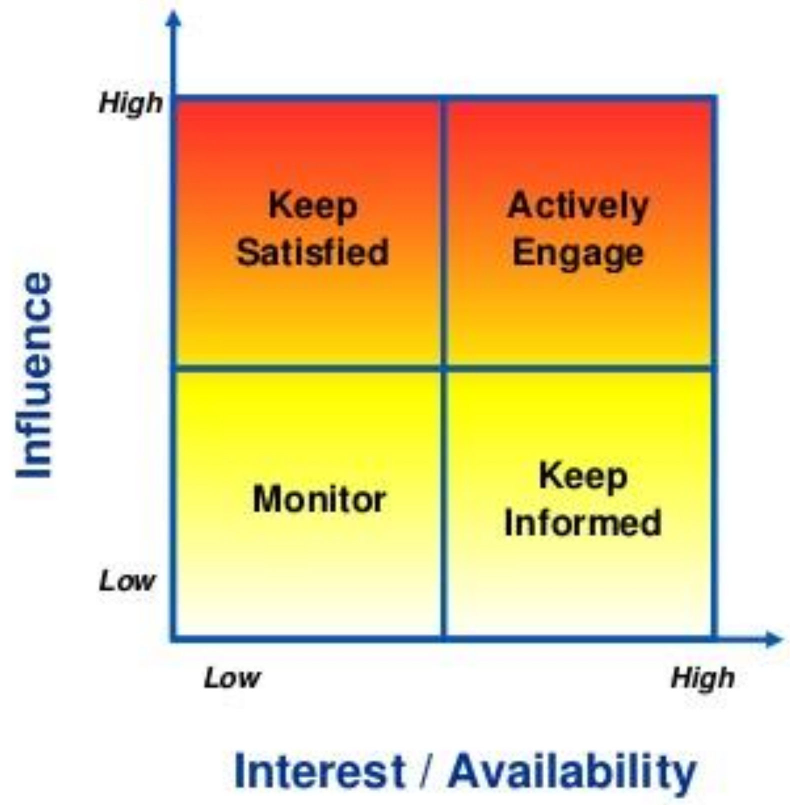
A change from this

To this

While preventing this

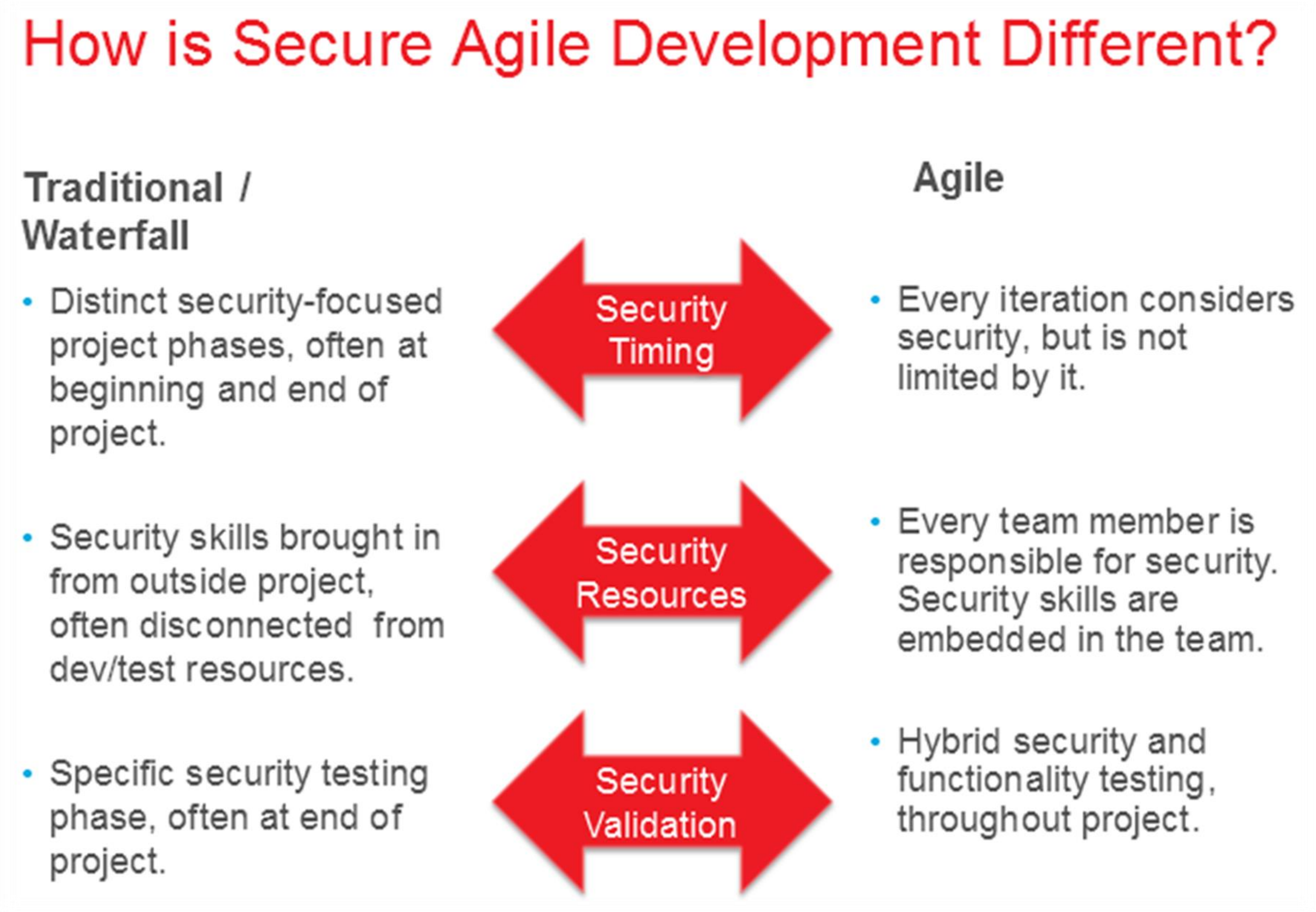
Step 1: Stop being a department

The Stakeholder Map



What is it?

- Provides a framework for managing stakeholders based on interest and influence
- Y-axis sometimes labeled “Power” (but can be a charged term)
- X-axis sometimes just labeled “Interest” (but who likes to be thought of as disinterested?)



How is Secure Agile Development Different?

Traditional / Waterfall

- Distinct security-focused project phases, often at beginning and end of project.
- Security skills brought in from outside project, often disconnected from dev/test resources.
- Specific security testing phase, often at end of project.

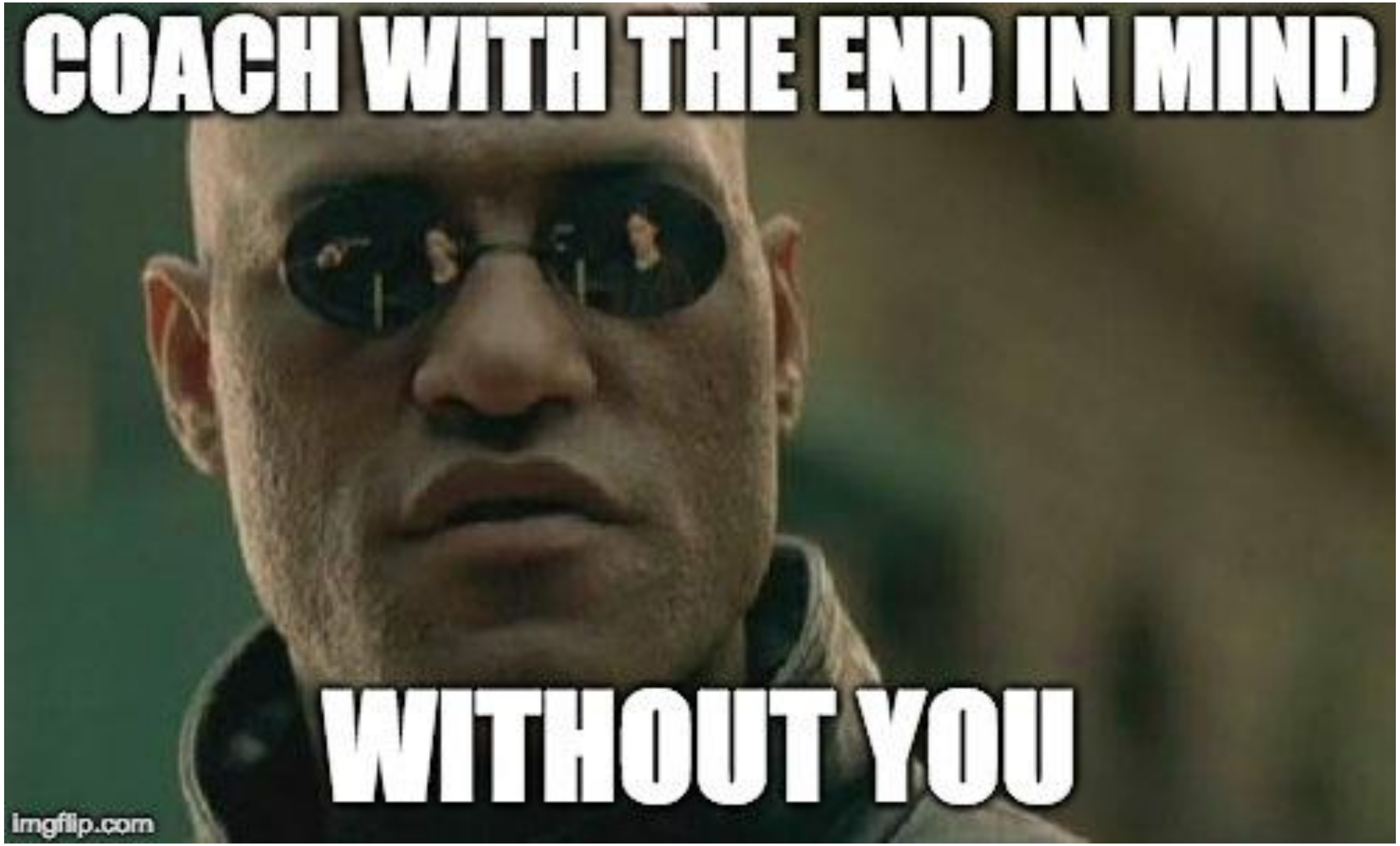
Agile

- Every iteration considers security, but is not limited by it.
- Every team member is responsible for security. Security skills are embedded in the team.
- Hybrid security and functionality testing, throughout project.

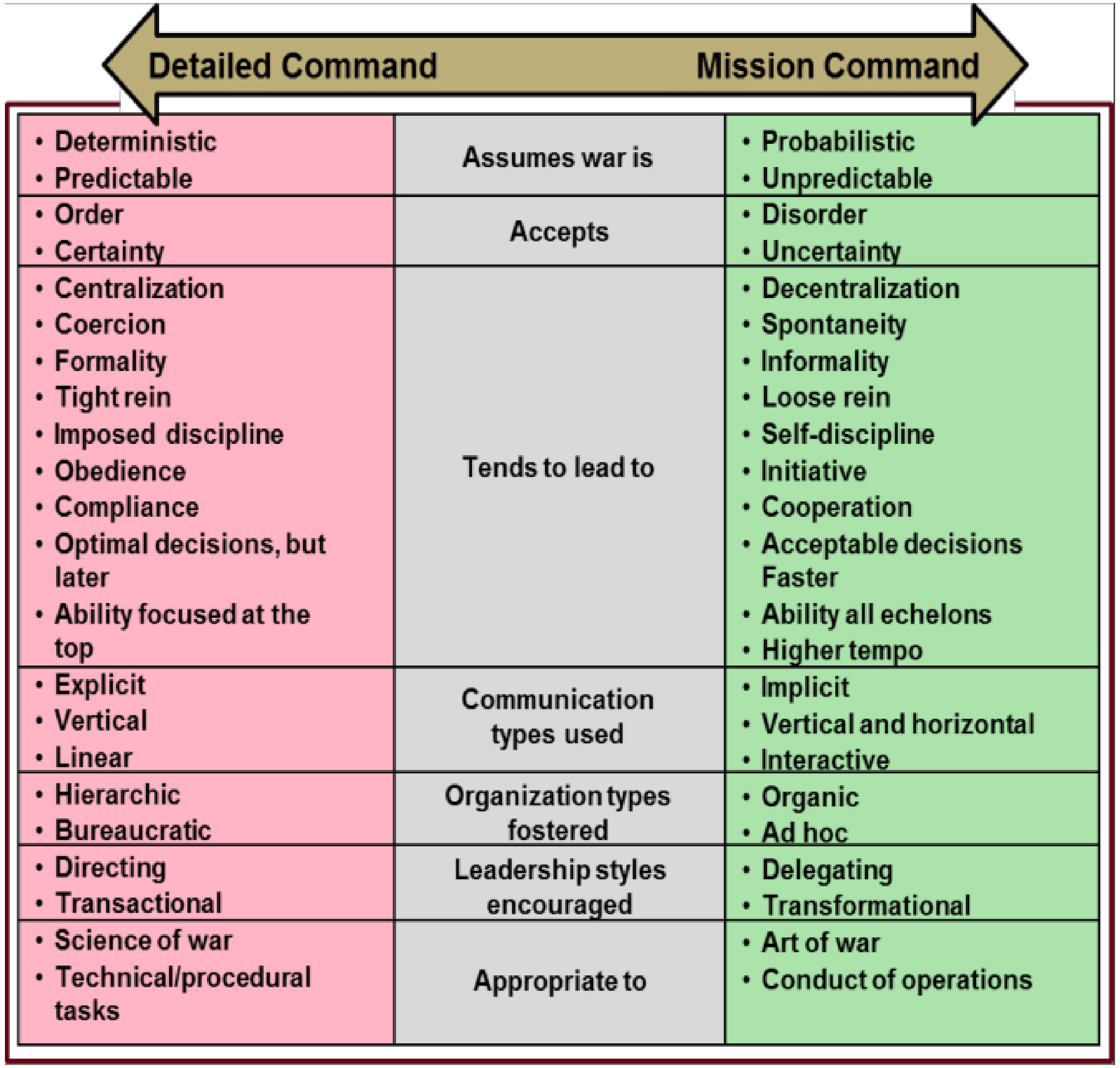
Proactive Security Strategy

- Act on ~~feelings~~ facts
- Sell based on ~~fear~~ advantages
- Security ~~costs resources~~ delivers
- Security ~~restricts~~ enables
- ~~Reactive~~ Proactive
- ~~Adhoc~~ Process
- ~~Only for specialists~~ A task for everyone

Start being an involved participant

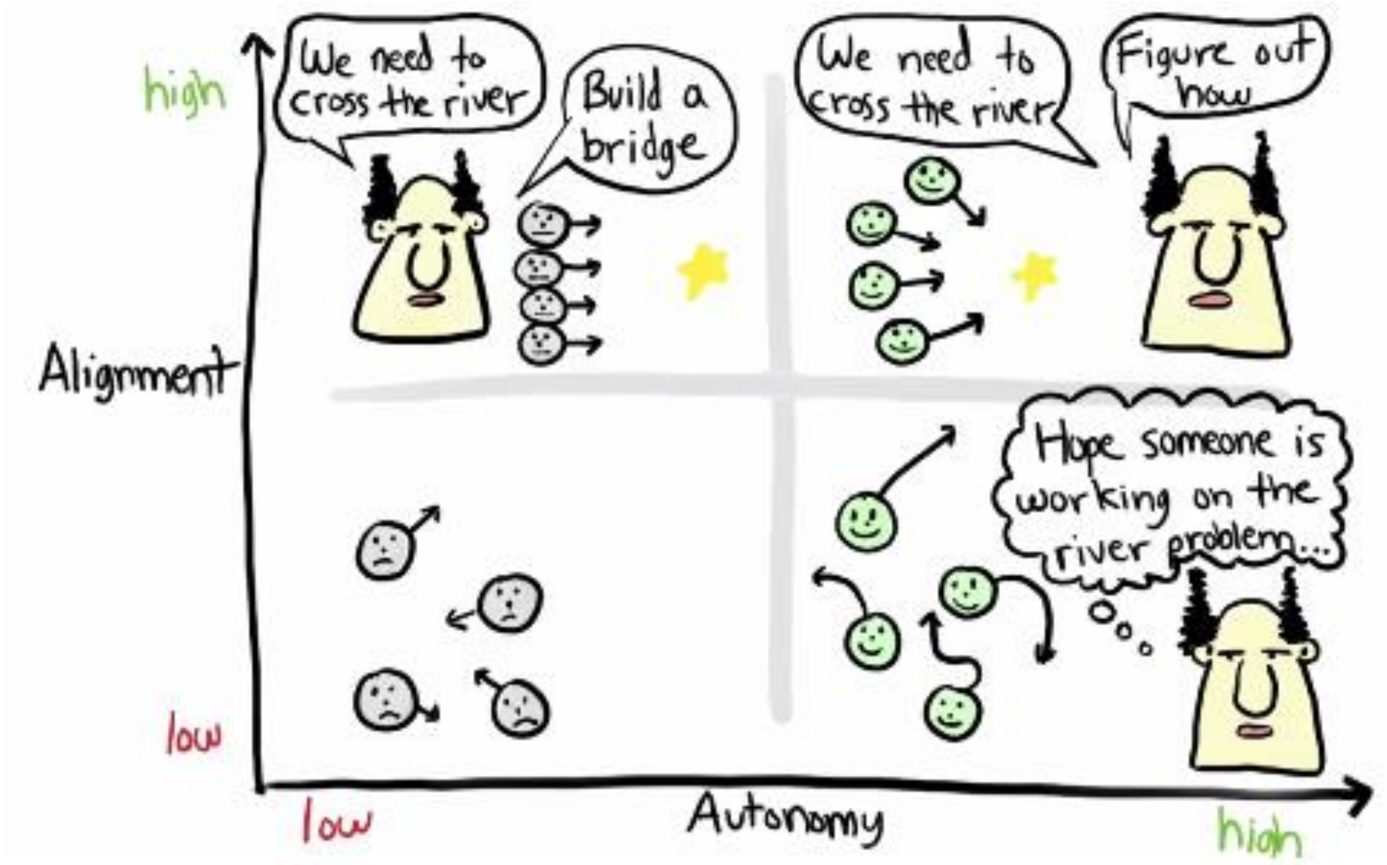


Apply military tactics

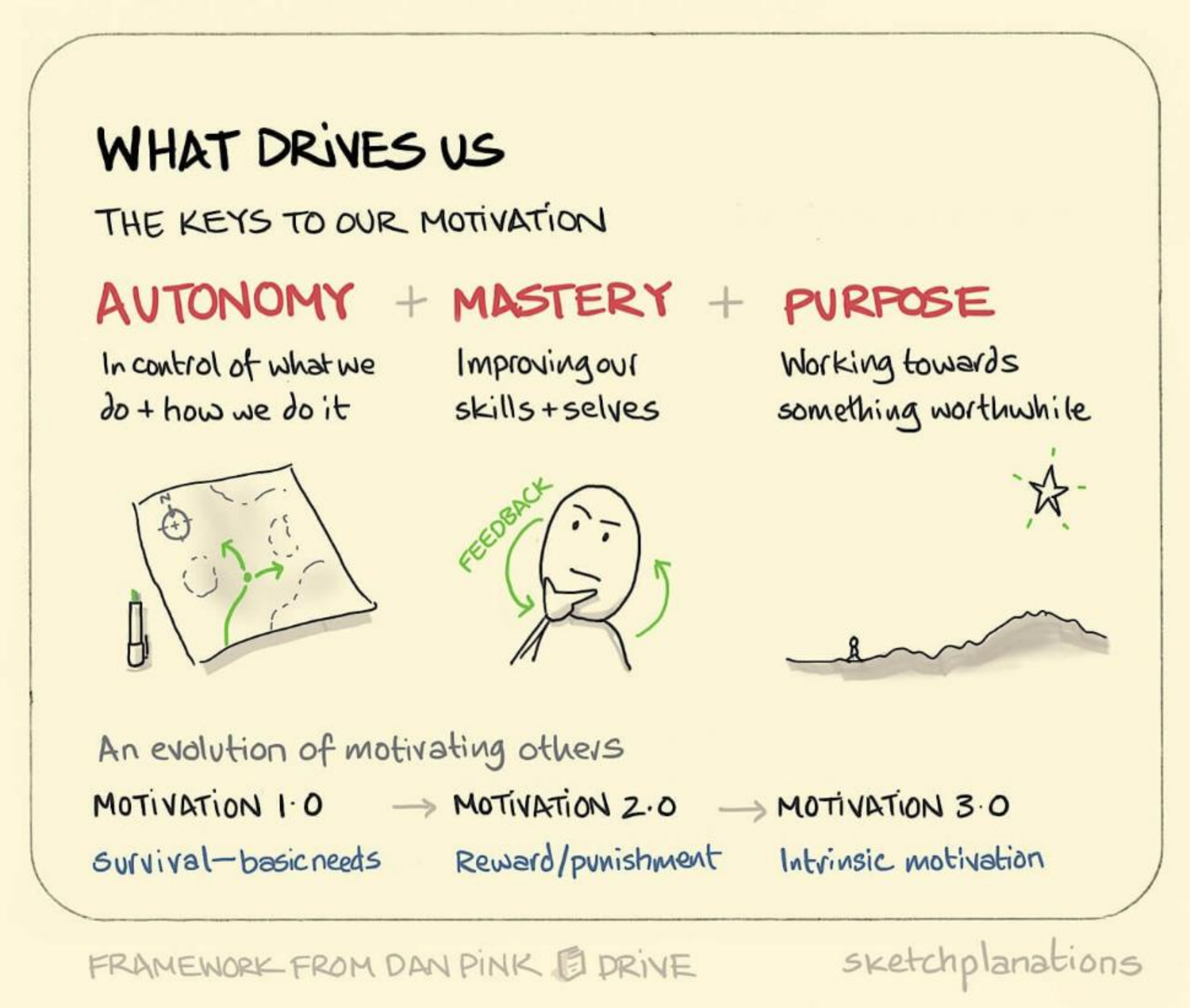


Detailed Command		Mission Command
• Deterministic • Predictable	Assumes war is	• Probabilistic • Unpredictable
• Order • Certainty	Accepts	• Disorder • Uncertainty
• Centralization • Coercion • Formality • Tight rein • Imposed discipline • Obedience • Compliance • Optimal decisions, but later • Ability focused at the top	Tends to lead to	• Decentralization • Spontaneity • Informality • Loose rein • Self-discipline • Initiative • Cooperation • Acceptable decisions Faster • Ability all echelons • Higher tempo
• Explicit • Vertical • Linear	Communication types used	• Implicit • Vertical and horizontal • Interactive
• Hierarchic • Bureaucratic	Organization types fostered	• Organic • Ad hoc
• Directing • Transactional	Leadership styles encouraged	• Delegating • Transformational
• Science of war • Technical/procedural tasks	Appropriate to	• Art of war • Conduct of operations

Modern Leadership


high
We need to cross the river
Build a bridge
We need to cross the river
Figure out how
Alignment
Hope someone is working on the river problem...
low
low
Autonomy
high

Intrinsically Motivated Teams



WHAT DRIVES US

THE KEYS TO OUR MOTIVATION

AUTONOMY + MASTERY + PURPOSE

In control of what we do + how we do it

Improving our skills + selves

Working towards something worthwhile

An evolution of motivating others

MOTIVATION 1·0 → MOTIVATION 2·0 → MOTIVATION 3·0

Survival—basic needs

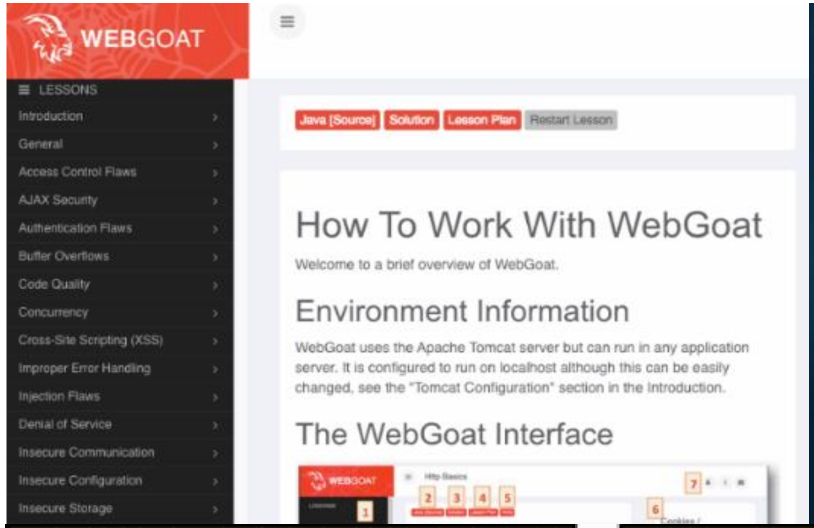
Reward/punishment

Intrinsic motivation

FRAMEWORK FROM DAN PINK DRIVE

sketchplanations

Provide challenges & incentives



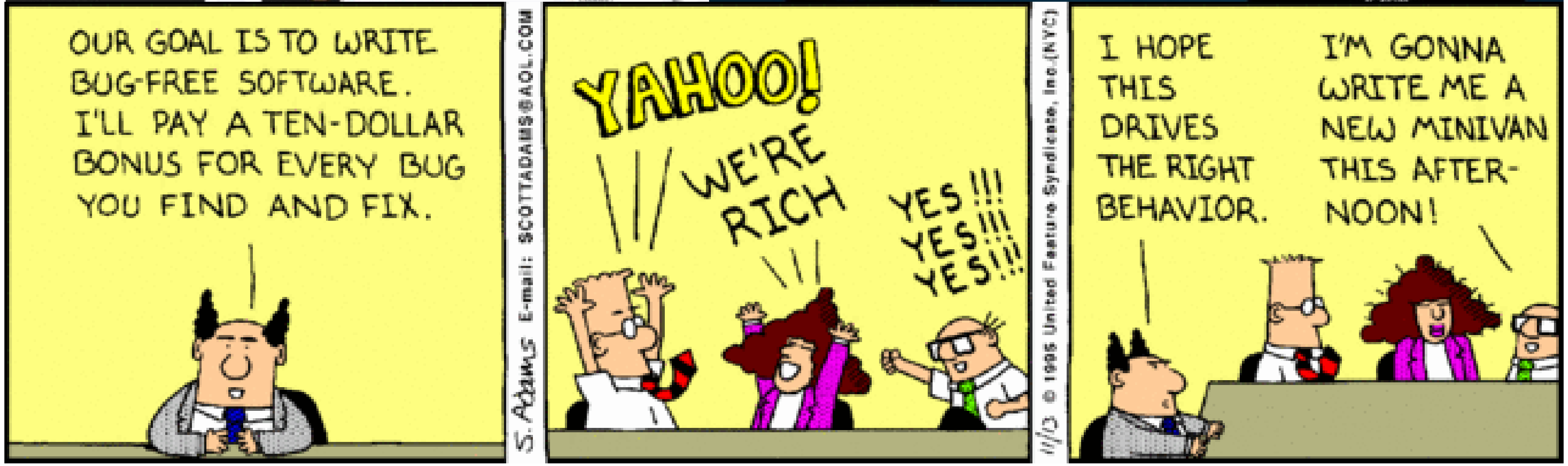
How To Work With WebGoat

Welcome to a brief overview of WebGoat.

Environment Information

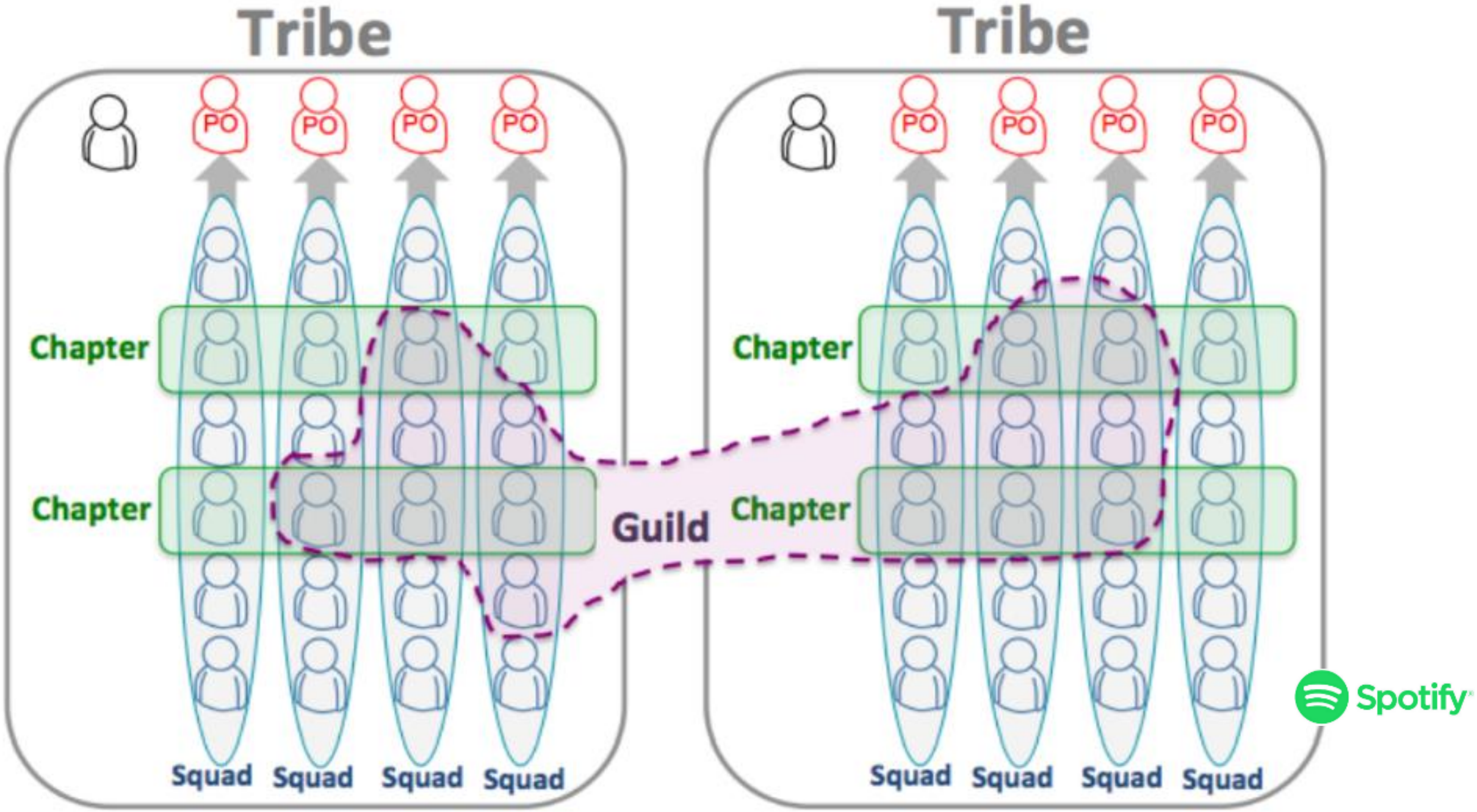
WebGoat uses the Apache Tomcat server but can run in any application server. It is configured to run on localhost although this can be easily changed, see the "Tomcat Configuration" section in the Introduction.

The WebGoat Interface



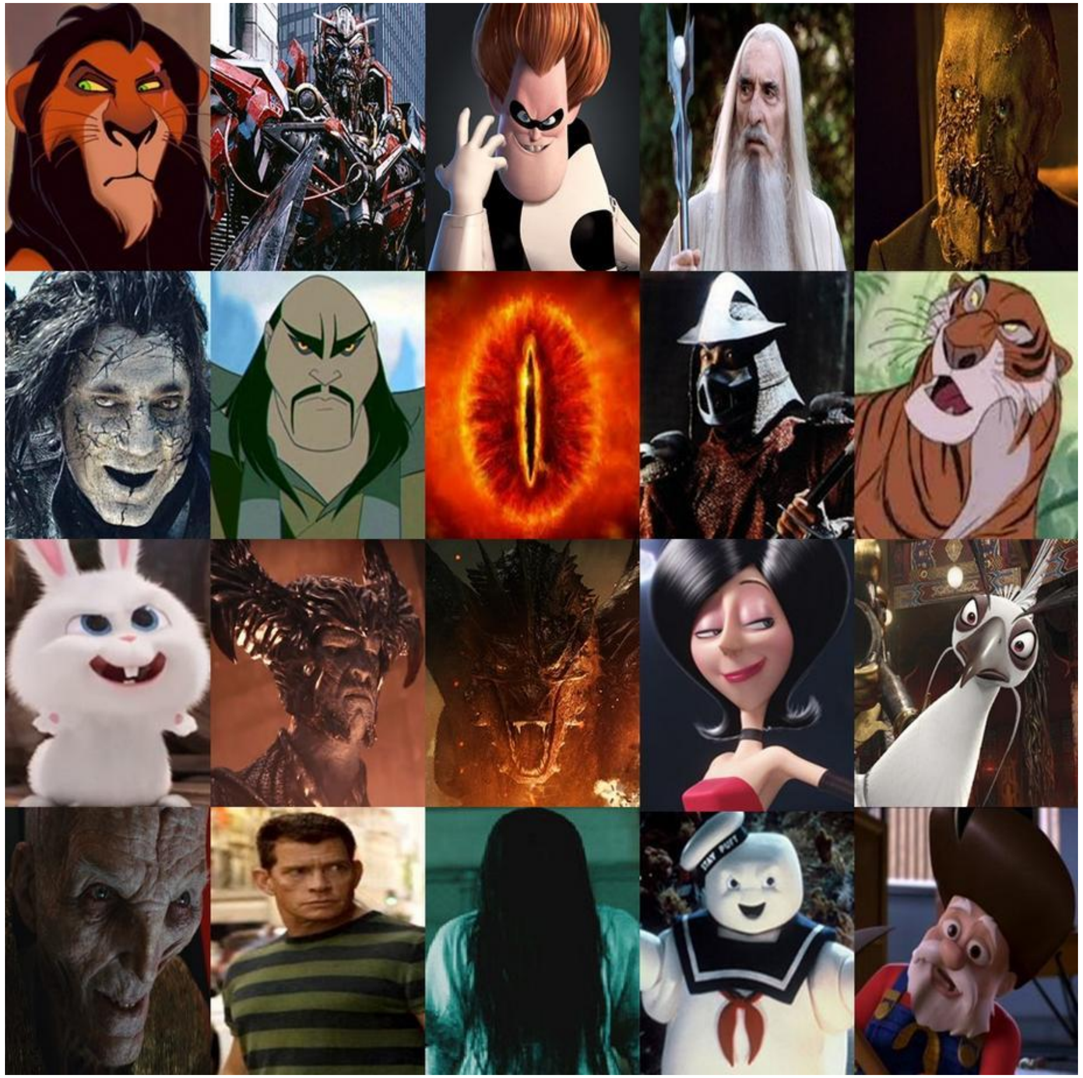
OUR GOAL IS TO WRITE BUG-FREE SOFTWARE. I'LL PAY A TEN-DOLLAR BONUS FOR EVERY BUG YOU FIND AND FIX.
S. Adams E-mail: SCOTTADAMS@AOL.COM
YAHOO!
WE'RE RICH
YES!!! YES!!! YES!!!
11/3 © 1995 United Feature Syndicate, Inc.(NYC)
I HOPE THIS DRIVES THE RIGHT BEHAVIOR.
I'M GONNA WRITE ME A NEW MINIVAN THIS AFTER-NOON!

Guild: Community of Interest

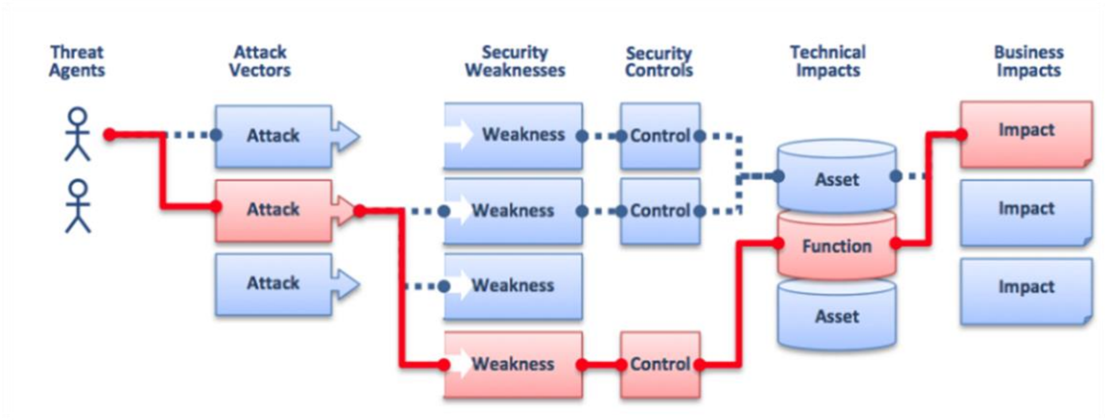


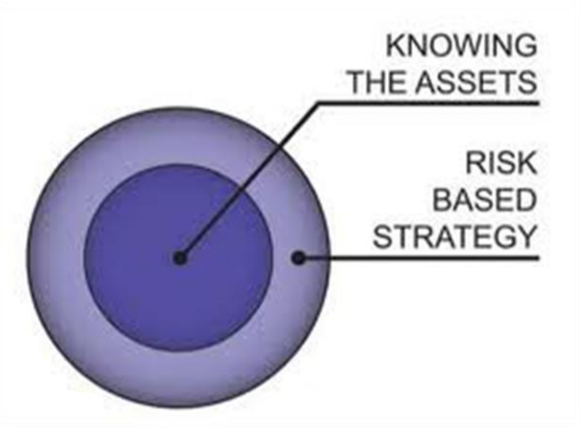
Step 2: Make risk something ordinary

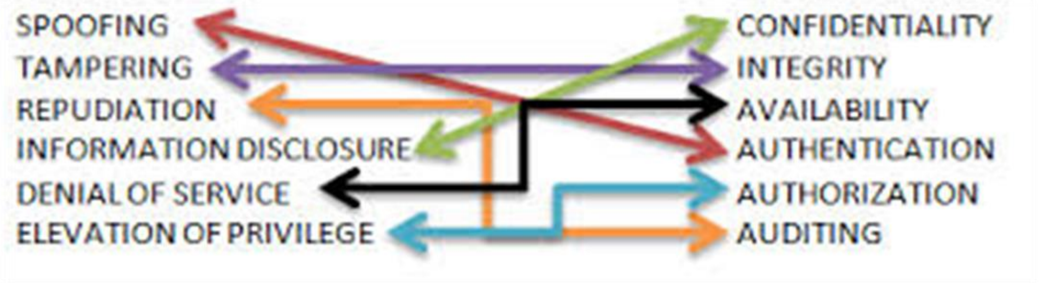
Not all users have good intentions



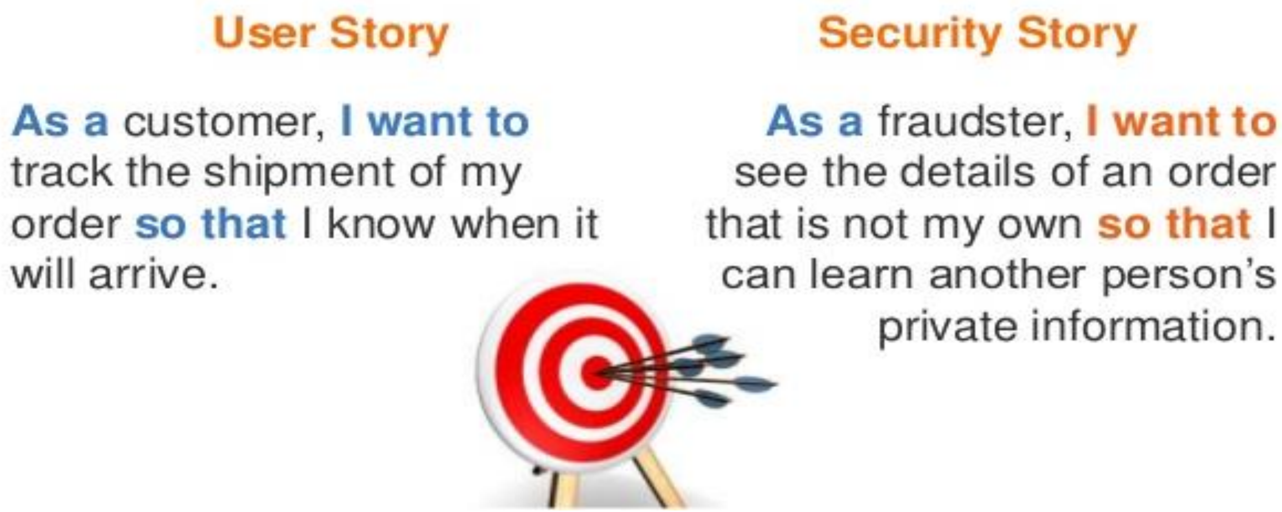
Threat model the story map







Add security user stories

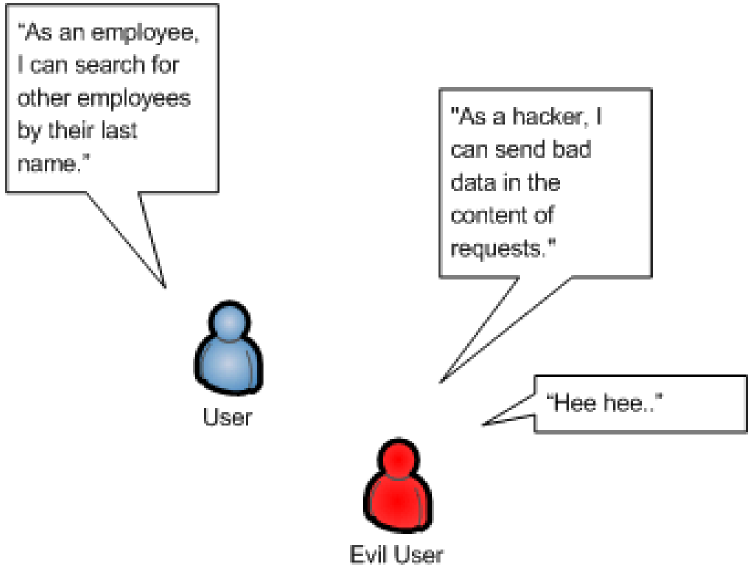


User Story

As a customer, **I want to** track the shipment of my order **so that** I know when it will arrive.

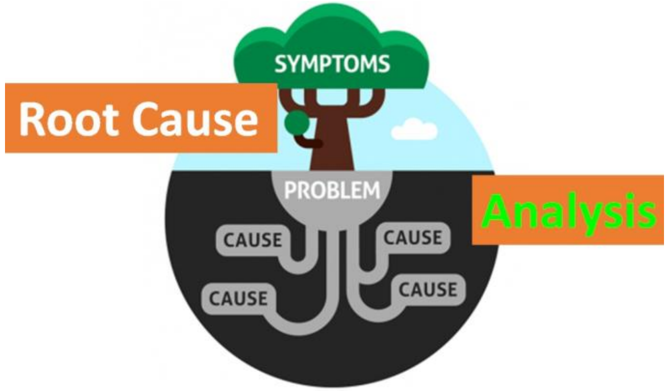
Security Story

As a fraudster, **I want to** see the details of an order that is not my own **so that** I can learn another person's private information.


"As an employee, I can search for other employees by their last name."
"As a hacker, I can send bad data in the content of requests."
"Hee hee.."
User
Evil User

Learn from mistakes

How to Conduct a **Blameless** Security Post-Mortem



The Kaizen Mindset

Seeing and prioritizing problems

Be truly prepared to:

- Uncover problems
- Accept them as a part of daily life
- Initiate an action to identify the problems that need immediate solutions

Solving problems

Be prepared to:

- Invest time and other resources
- Understand the root causes of problems
- Resolve the problems completely

Sharing lessons learned

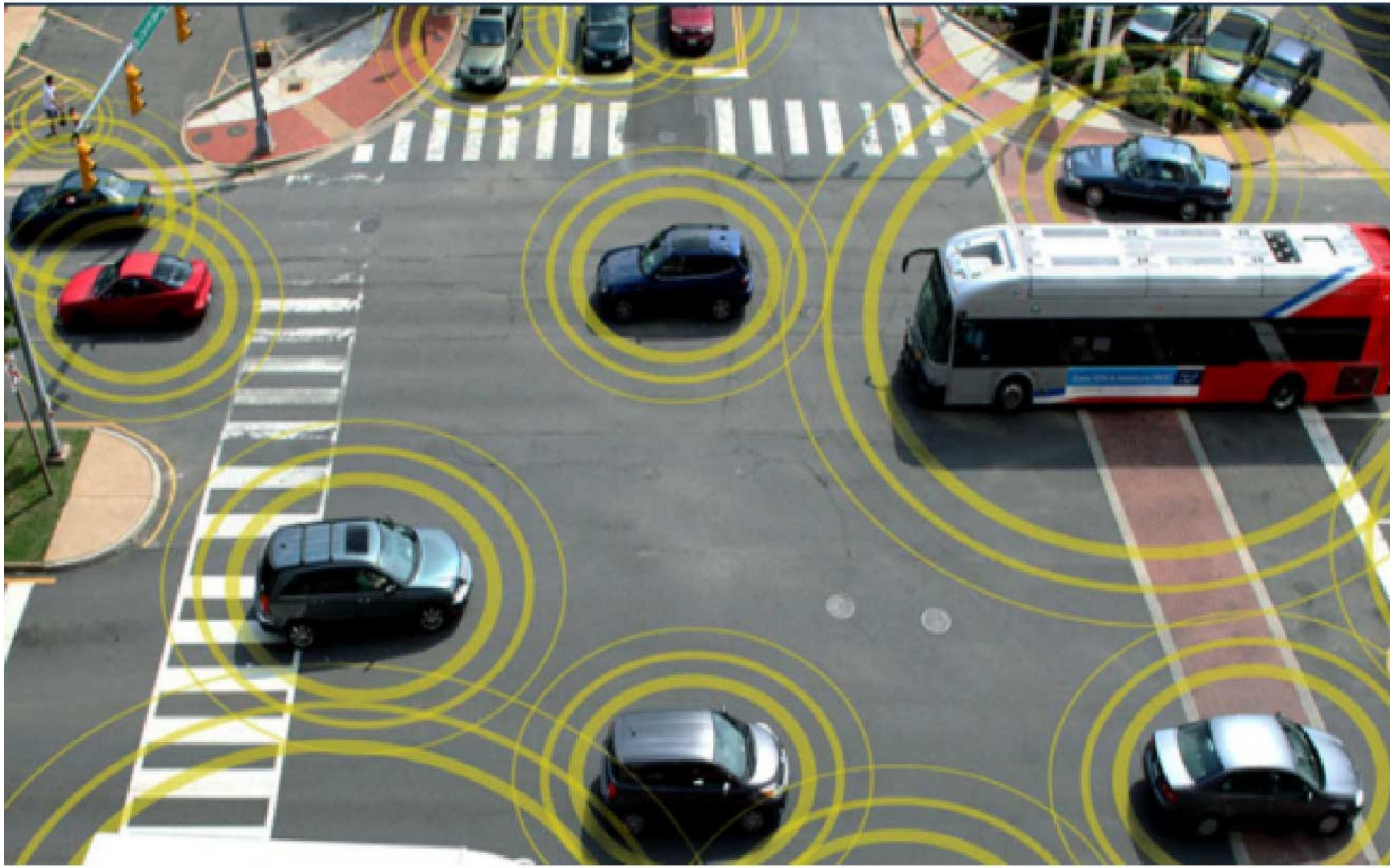
Be driven to:

- Share the lessons learned with others in the IT organization, so they can benefit from it

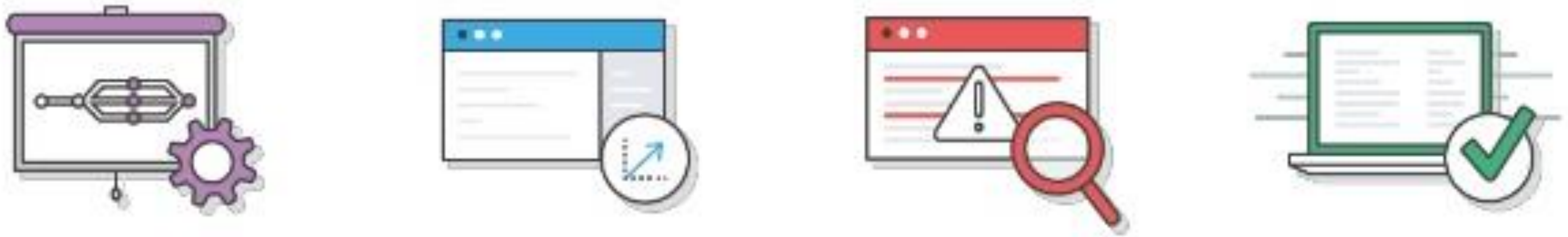
Leak
Temporary solution
Why?
Why?
Why?
Why?
Why?

Don’t wait for bad things to happen; plan them

Step 3: Decentralise security automation



Why development loves automation



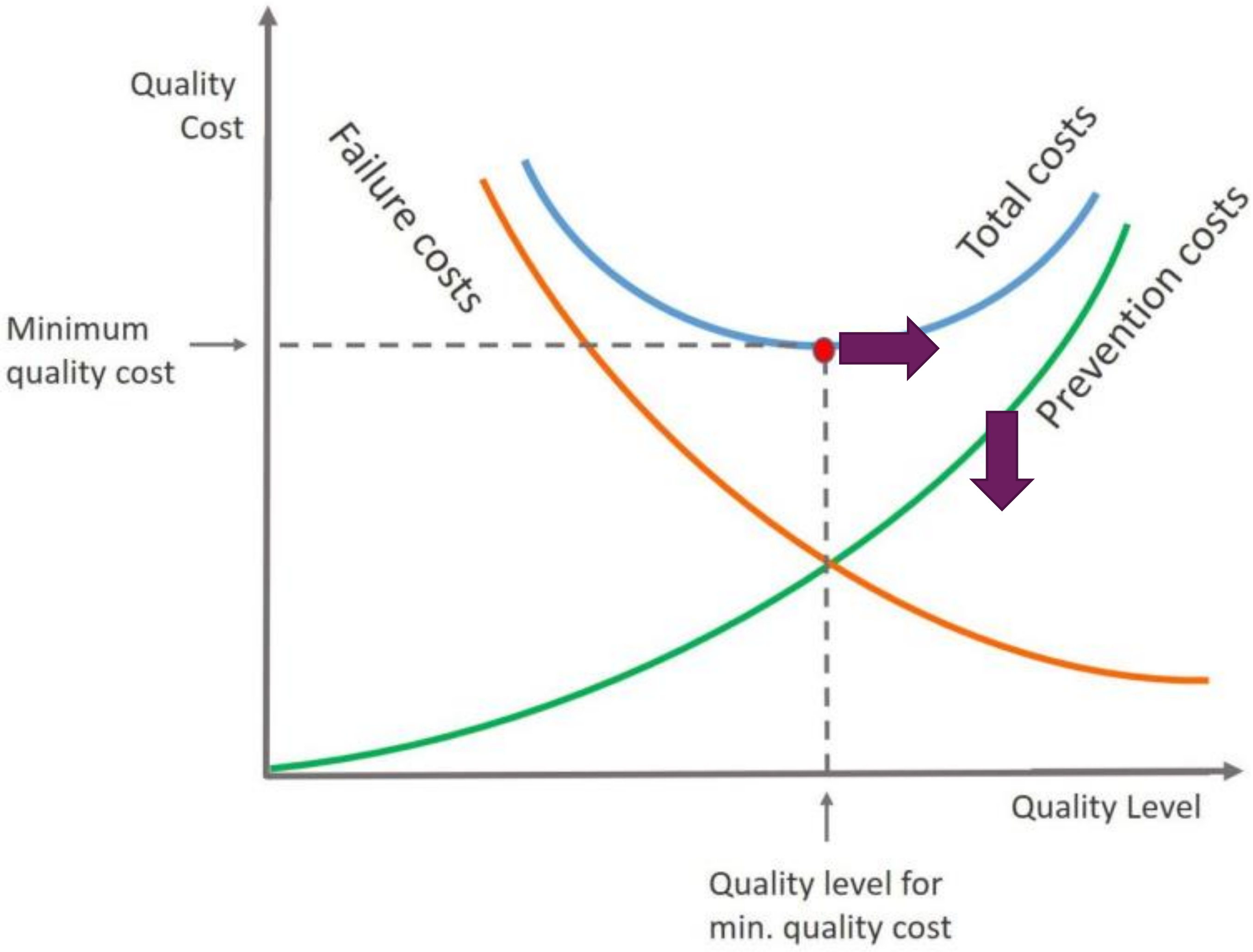
Automate the software release process

Improve developer productivity

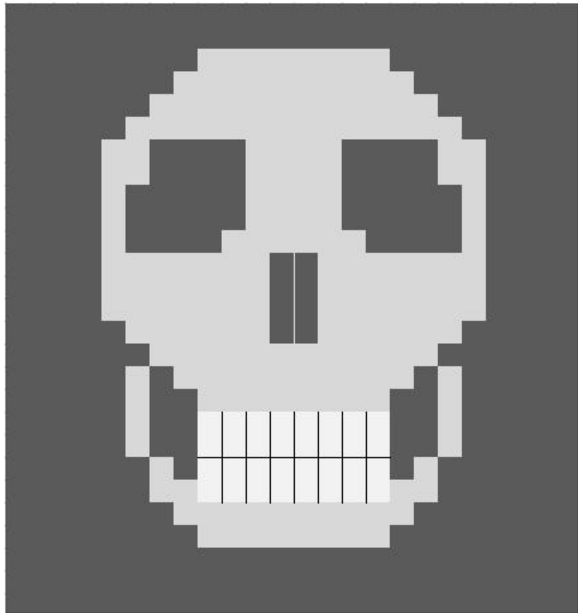
Find and address bugs quickly

Deliver updates faster

Speed vs Quality



Use the opportunity

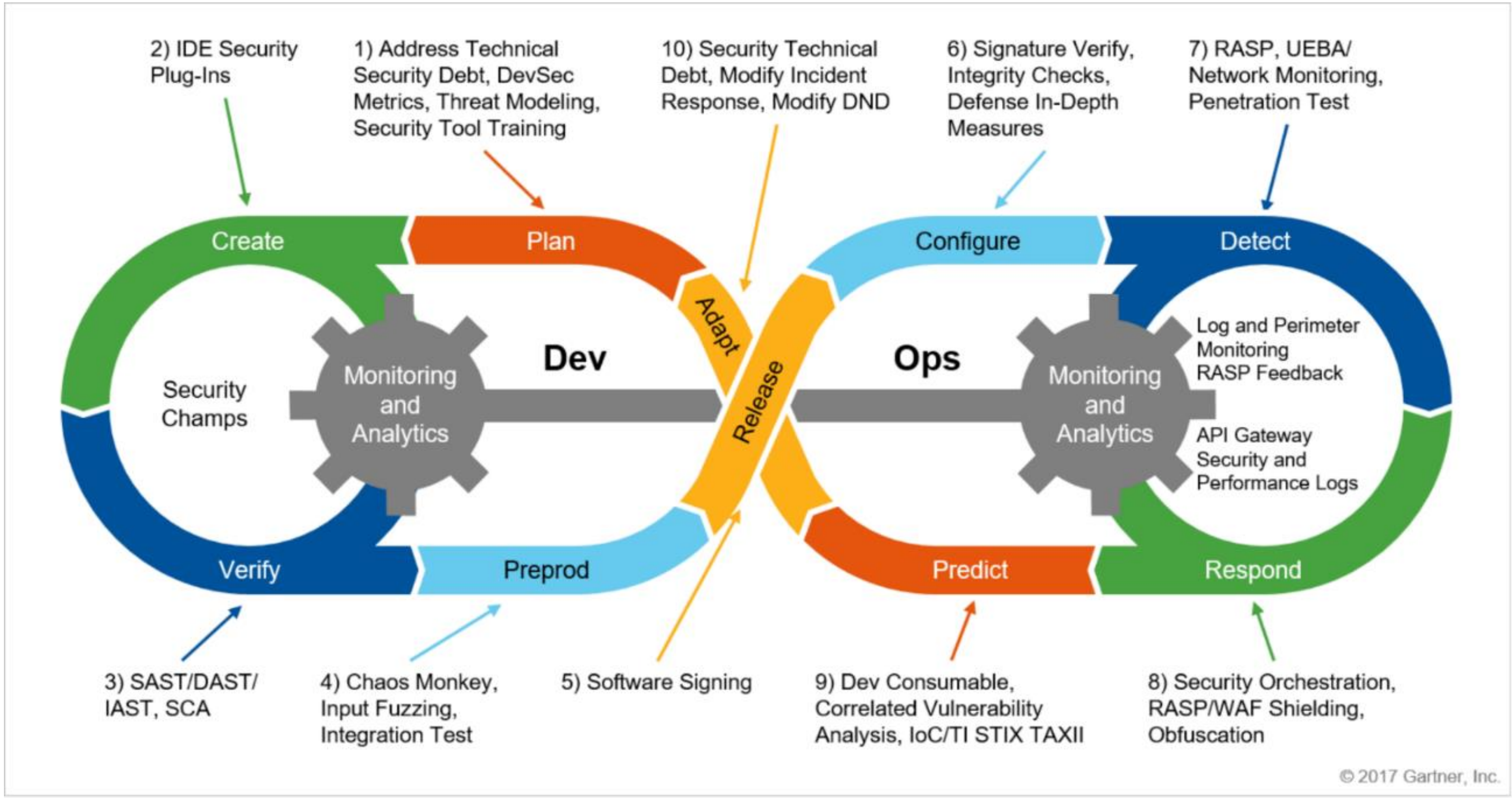


Developers want to code…

Not do your paperwork…

→ Embed & automate

Automate everything?


2) IDE Security Plug-Ins
1) Address Technical Security Debt, DevSec Metrics, Threat Modeling, Security Tool Training
10) Security Technical Debt, Modify Incident Response, Modify DND
6) Signature Verify, Integrity Checks, Defense In-Depth Measures
7) RASP, UEBA/ Network Monitoring, Penetration Test
Create
Plan
Adapt
Configure
Detect
Security Champs
Monitoring and Analytics
Dev
Release
Ops
Monitoring and Analytics
Log and Perimeter Monitoring RASP Feedback
API Gateway Security and Performance Logs
Verify
Preprod
Predict
Respond
3) SAST/DAST/ IAST, SCA
4) Chaos Monkey, Input Fuzzing, Integration Test
5) Software Signing
9) Dev Consumable, Correlated Vulnerability Analysis, IoC/TI STIX TAXII
8) Security Orchestration, RASP/WAF Shielding, Obfuscation

© 2017 Gartner, Inc.

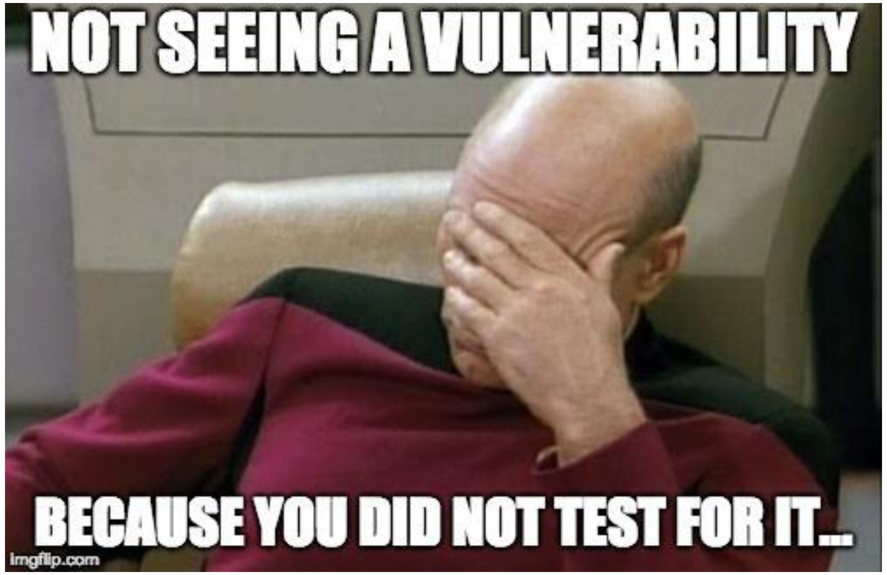
Overcomplexity

Don’t deliver complex final products…

GRUMPY CAT DOES NOT LIKE YOU
BECAUSE YOU LEFT BEHIND A COMPLEX SYSTEM
imgflip.com

Finding the balance

WAITING FOR
THE APPSECPIPELINE TO FINISH
imgflip.com


NOT SEEING A VULNERABILITY
BECAUSE YOU DID NOT TEST FOR IT...
imgflip.com

Security enabled vs secure

A SECURITY PIPELINE

Wielding the power of security tooling

A SECURE PIPELINE

Access denied?

Think about your secrets and access control!

- Security-sensitive information
- DB User/Pass
- AWS IAM Credentials
- SSL Keys
- Encryption Keys
- Personally-identifiable information (PII)
- Anything that would make the news ;)

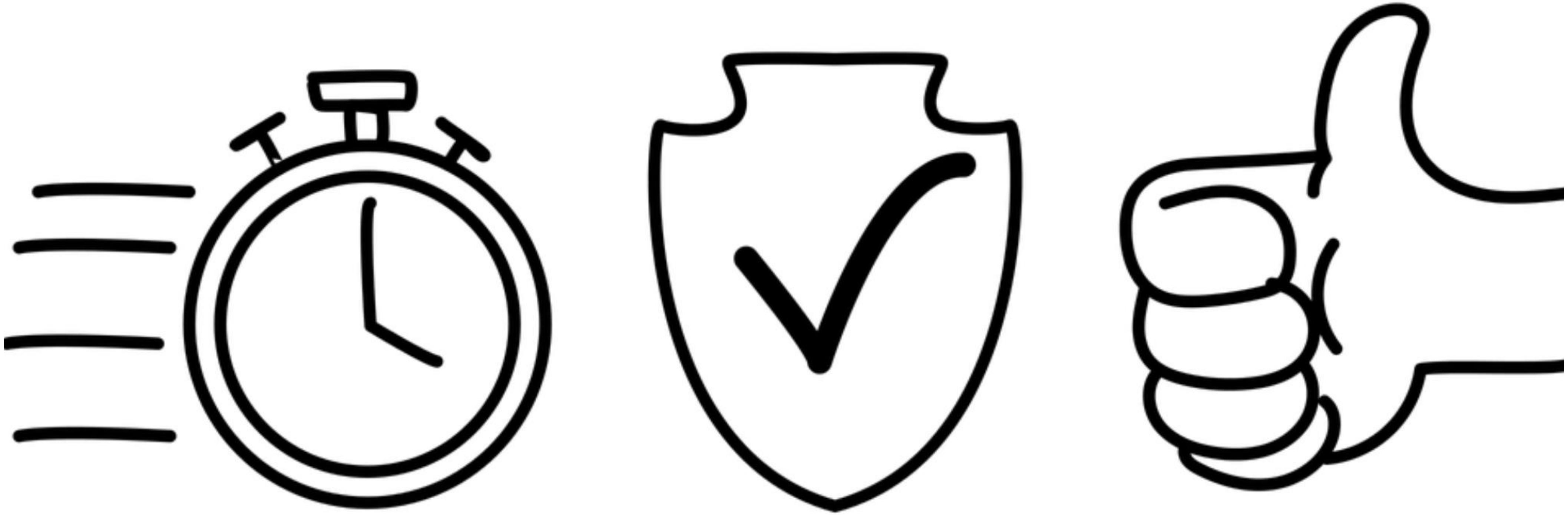
Logging

YOU CANNOT HEAR AN ATTACKER IF YOU DO NOT <u>LISTEN.</u>

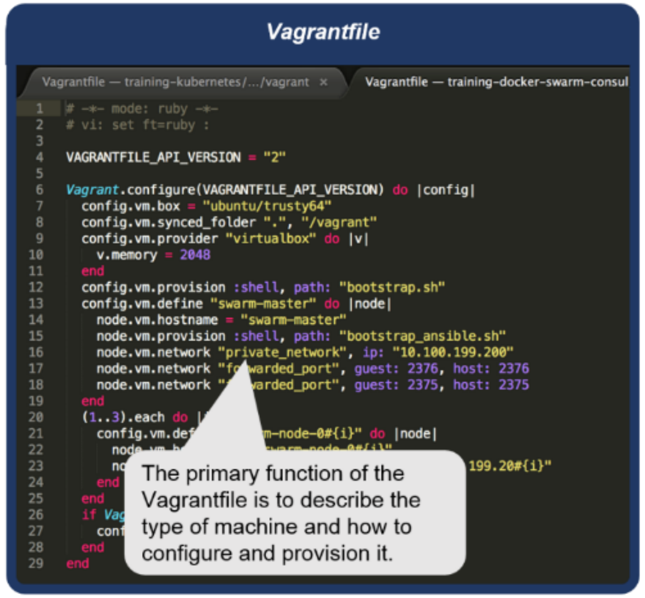
Monitoring

YOU DO NOT SEE ANYTHING IN THE DARK.

Reap the benefits!



Everything as Code

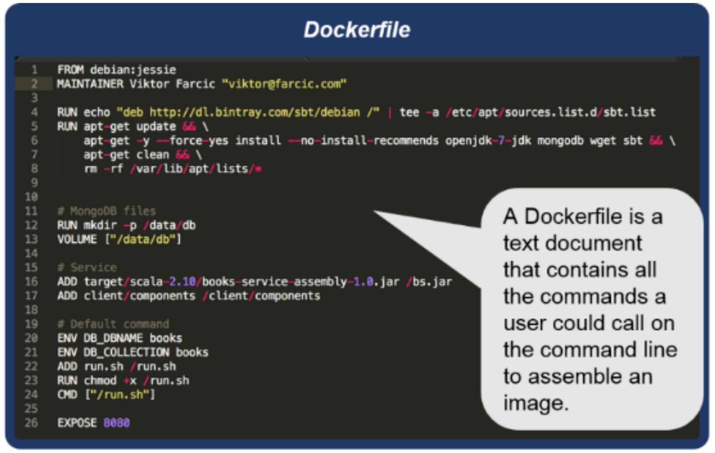


Vagrantfile

Vagrantfile — training-kubernetes/…/vagrant ×   Vagrantfile — training-docker-swarm-consul

```
# -*- mode: ruby -*-
# vi: set ft=ruby :

VAGRANTFILE_API_VERSION = "2"

Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
  config.vm.box = "ubuntu/trusty64"
  config.vm.synced_folder ".", "/vagrant"
  config.vm.provider "virtualbox" do |v|
    v.memory = 2048
  end
  config.vm.provision :shell, path: "bootstrap.sh"
  config.vm.define "swarm-master" do |node|
    node.vm.hostname = "swarm-master"
    node.vm.provision :shell, path: "bootstrap_ansible.sh"
    node.vm.network "private_network", ip: "10.100.199.200"
    node.vm.network "fo warded_port", guest: 2376, host: 2376
    node.vm.network "  warded_port", guest: 2375, host: 2375
  end
  (1..3).each do |
    config.vm.def        m-node-0#{i}" do |node|
      node.vm.h          swarm-node-0#{i}"
      no                                     199.20#{i}"
    end
  end
  if Vag
    conf
  end
end
```

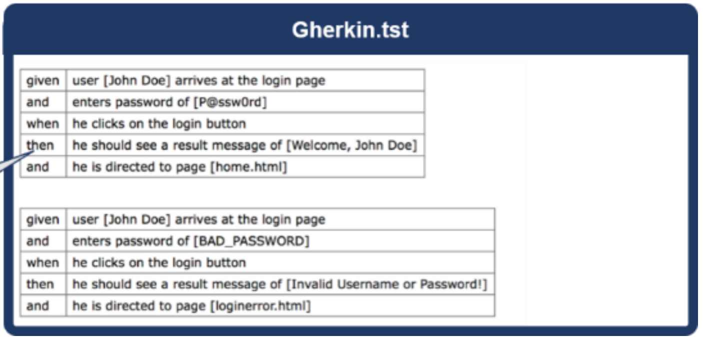
The primary function of the Vagrantfile is to describe the type of machine and how to configure and provision it.



Dockerfile

```
FROM debian:jessie
MAINTAINER Viktor Farcic "viktor@farcic.com"

RUN echo "deb http://dl.bintray.com/sbt/debian /" | tee -a /etc/apt/sources.list.d/sbt.list
RUN apt-get update && \
    apt-get -y --force-yes install --no-install-recommends openjdk-7-jdk mongodb wget sbt && \
    apt-get clean && \
    rm -rf /var/lib/apt/lists/*


# MongoDB files
RUN mkdir -p /data/db
VOLUME ["/data/db"]

# Service
ADD target/scala-2.10/books-service-assembly-1.0.jar /bs.jar
ADD client/components /client/components

# Default command
ENV DB_DBNAME books
ENV DB_COLLECTION books
ADD run.sh /run.sh
RUN chmod +x /run.sh
CMD ["/run.sh"]

EXPOSE 8080
```

A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image.



Gherkin.tst

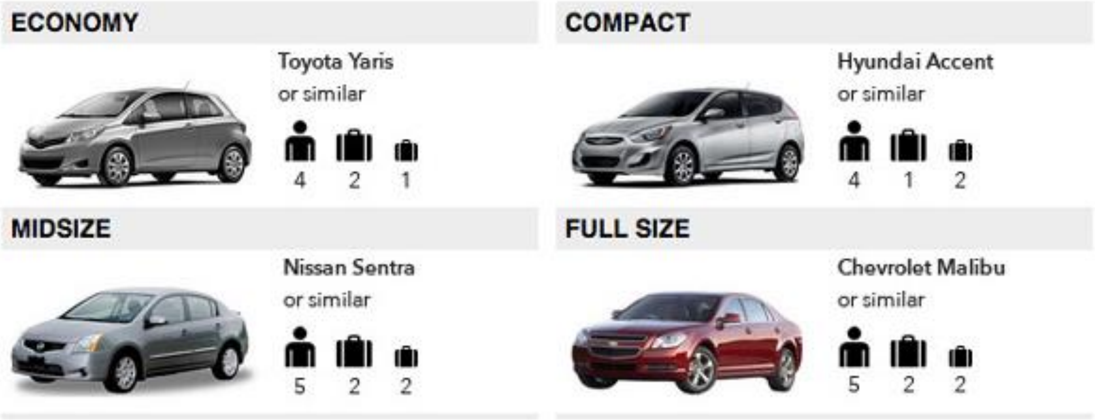
given	user [John Doe] arrives at the login page
and	enters password of [P@ssw0rd]
when	he clicks on the login button
then	he should see a result message of [Welcome, John Doe]
and	he is directed to page [home.html]

given	user [John Doe] arrives at the login page
and	enters password of [BAD_PASSWORD]
when	he clicks on the login button
then	he should see a result message of [Invalid Username or Password!]
and	he is directed to page [loginerror.html]

Gherkin serves two purposes — documentation and automated tests

~~Pets versus Cattle~~ Oldtimers and Rental cars

- Oldtimers are given names.
- They are unique.
- They are hand raised and are given proper care.
- When they get ill, they are nursed back to health.
- Flaws are worked around



ECONOMY
Toyota Yaris or similar
4 2 1

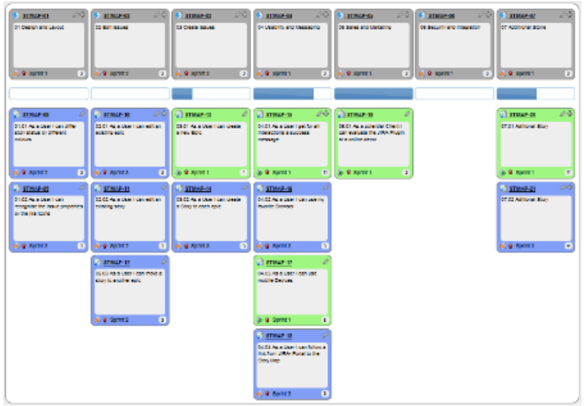
COMPACT
Hyundai Accent or similar
4 1 2

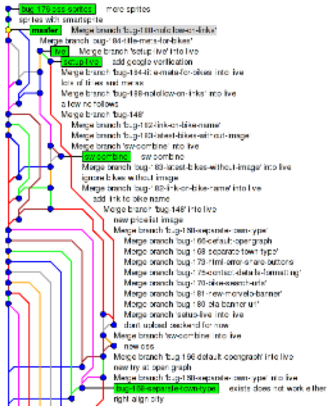
MIDSIZE
Nissan Sentra or similar
5 2 2

FULL SIZE
Chevrolet Malibu or similar
5 2 2

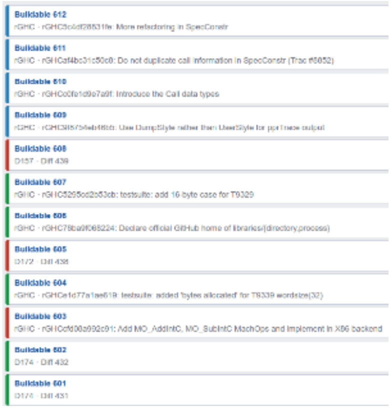
- Rentals are given numbers.
- They are (almost) identical to one another.
- They are managed as group.
- When they get ill, they are replaced.
- Flaws are unacceptable

Audit Automation





```
bug 179 css sprites   more sprites
sprites with smartsprite
master   Merge branch 'bug-180-nofollow-on-links'
Merge branch 'bug-184-title-meta-for-bikes'
live   Merge branch 'setup-live' into live
setup live   add google verification
Merge branch 'bug-184-title-meta-for-bikes' into live
lots of titles and metas
Merge branch 'bug-180-nofollow-on-links' into live
a few no follows
Merge branch 'bug-148'
Merge branch 'bug-182-link-on-bike-name'
Merge branch 'bug-183-latest-bikes-without-image'
Merge branch 'sw-combine' into live
sw combine   sw combine
Merge branch 'bug-183-latest-bikes-without-image' into live
ignore bikes without image
Merge branch 'bug-182-link-on-bike-name' into live
add link to bike name
Merge branch 'bug-148' into live
new pricelist image
Merge branch 'bug-158-separate-town-type'
Merge branch 'bug-166-default-opengraph'
Merge branch 'bug-168-separate-town-type'
Merge branch 'bug-173-html-error-share-buttons'
Merge branch 'bug-175-contact-details-formatting'
Merge branch 'bug-170-bike-search-urls'
Merge branch 'bug-181-new-marvelo-banner'
Merge branch 'bug-180-ola-banner-url'
Merge branch 'setup-live' into live
don't upload backend for now
Merge branch 'sw-combine' into live
new css
Merge branch 'bug-166-default-opengraph' into live
new try at open graph
Merge branch 'bug-158-separate-town-type' into live
bug-158-separate-town-type   exists does not work either
right align city
```



Buildable 612
rGHC - rGHC5c4df28531fe: More refactoring in SpecConstr

Buildable 611
rGHC - rGHCaf4bc31c50c0: Do not duplicate call information in SpecConstr (Trac #8852)

Buildable 610
rGHC - rGHCc0fe1d9e7a9f: Introduce the Call data types

Buildable 609
rGHC - rGHC3f8754eb4666: Use DumpStyle rather than UserStyle for pprTrace output

Buildable 608
D157 - Diff 439

Buildable 607
rGHC - rGHC5295cd2b53cb: testsuite: add 16 byte case for T9329

Buildable 606
rGHC - rGHC78ba9f065224: Declare official GitHub home of libraries/{directory,process}

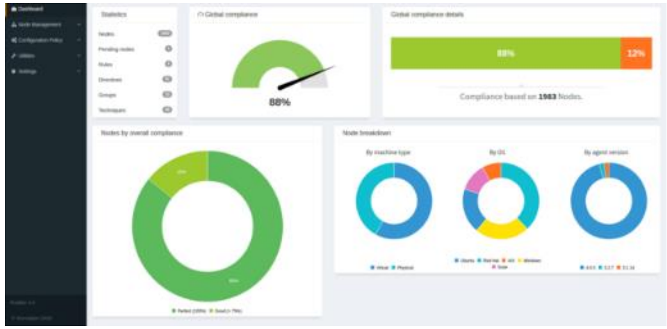
Buildable 605
D172 - Diff 438

Buildable 604
rGHC - rGHCe1d77a1ae619: testsuite: added 'bytes allocated' for T8339 wordsize(32)

Buildable 603
rGHC - rGHCcfd08a992c91: Add MO_AddIntC, MO_SubIntC MachOps and implement in X86 backend

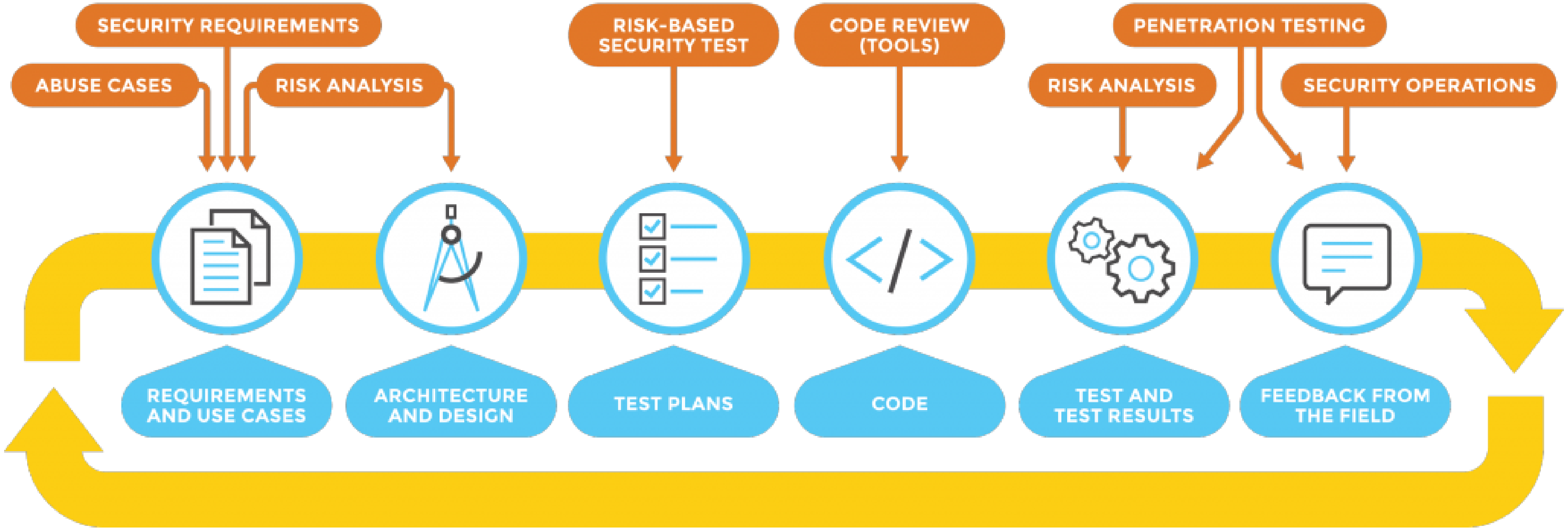
Buildable 602
D174 - Diff 432

Buildable 601
D174 - Diff 431

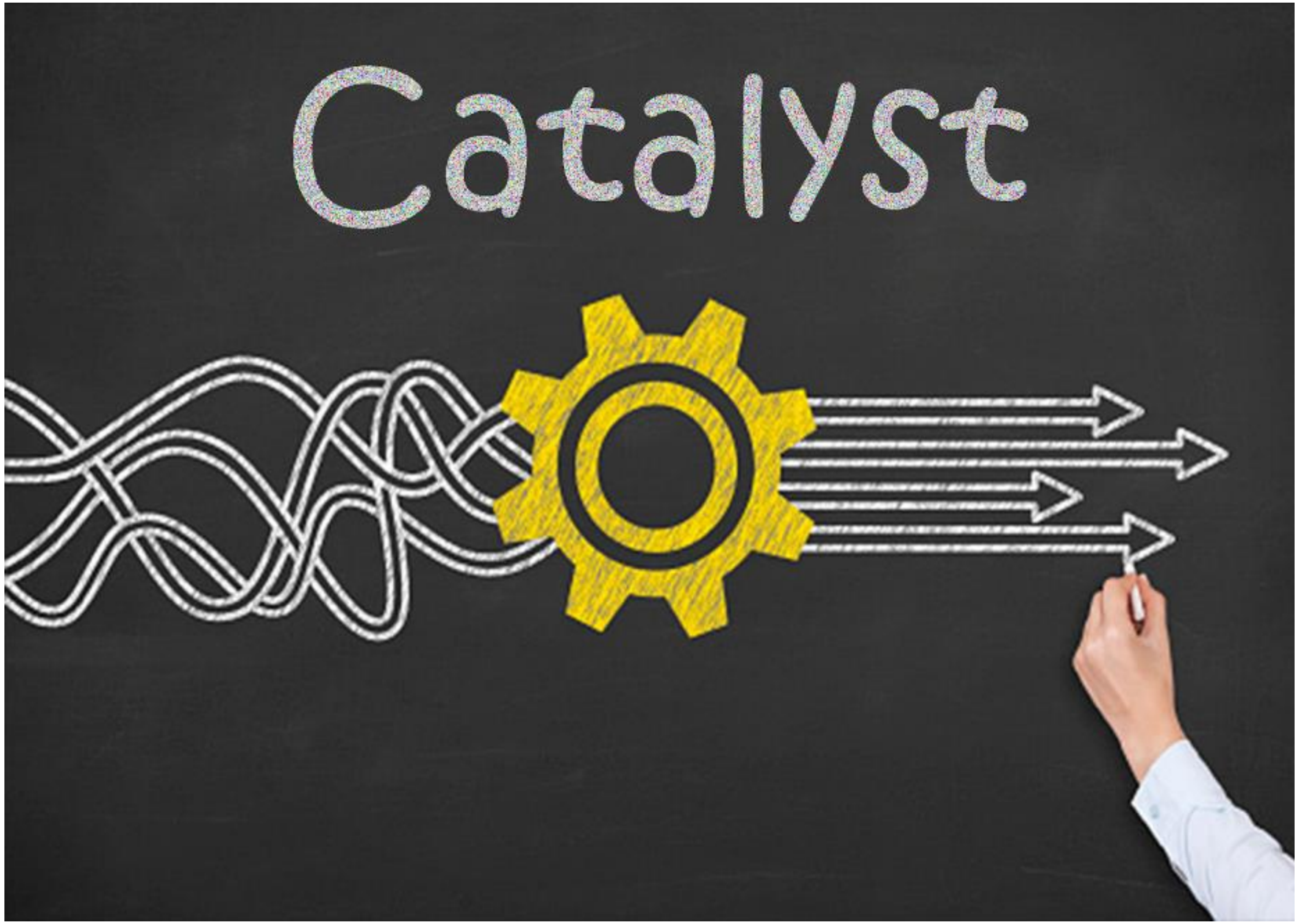


Global compliance
88%
Global compliance details
88%
12%
Compliance based on 1983 Nodes.
Nodes by overall compliance
Node breakdown

Continuous security



Step 4: Make security the fast route



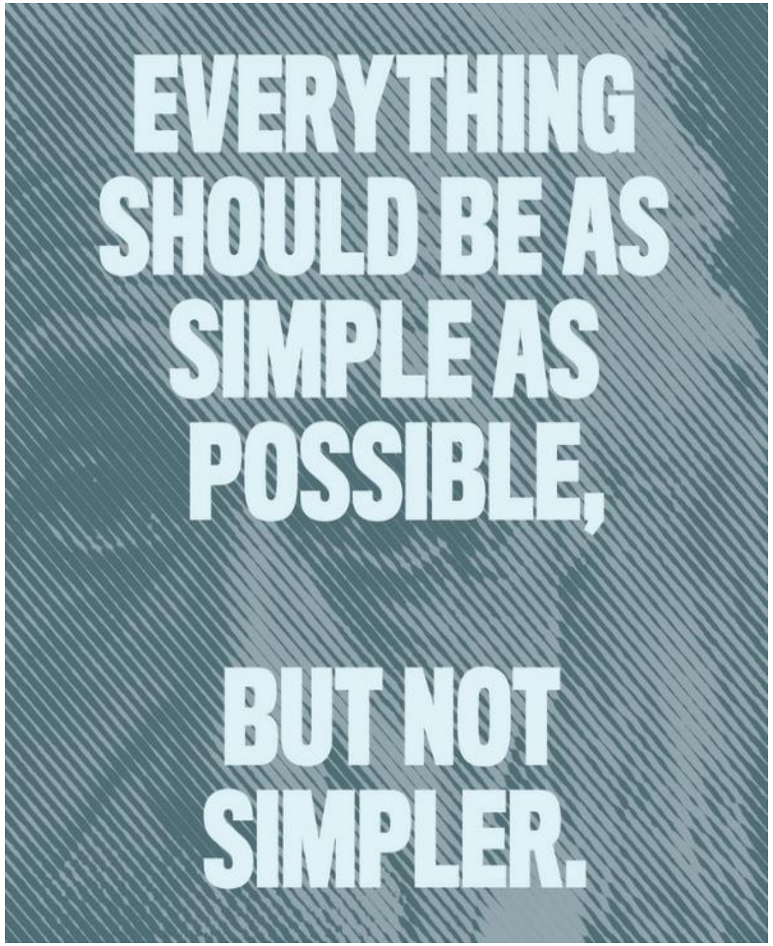
Agile Manifesto

We are uncovering better ways of developing software by doing it an helping others do it. Through this work we have come to value:

Individuals and interactions	over	**Processes and tools**
Working software	over	**Comprehensive documentation**
Customer collaboration	over	**Contract negotiation**
Responding to change	over	**Following a plan**

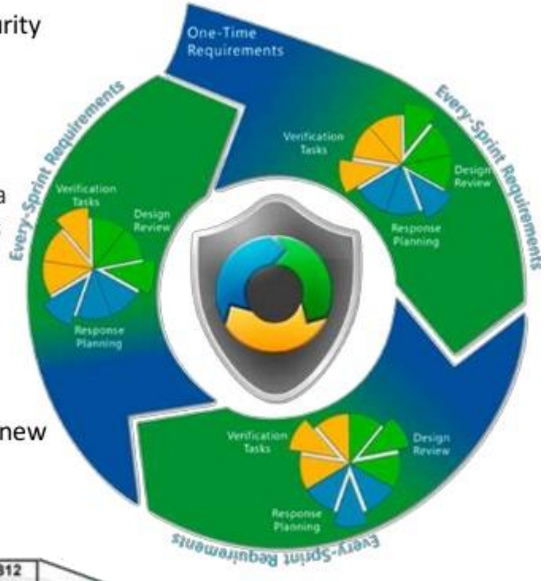
That is, while there is value in the items on the right, we value the items on the left more.

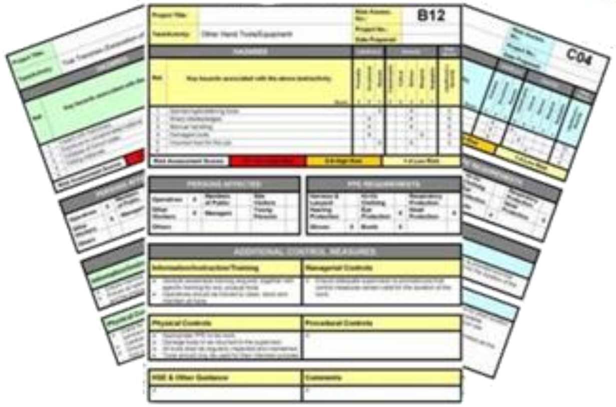
Focus on added value



Mapping SDL to Agile

- **Every-Sprint practices:** Essential security practices that should be performed in every release.
- **Bucket practices:** Important security practices that must be completed on a regular basis but can be spread across multiple sprints during the project lifetime.
- **One-Time practices:** Foundational security practices that must be established once at the start of every new Agile project.





Set simple ground rules

Recipe for a Safe Kitchen

Ingredients:

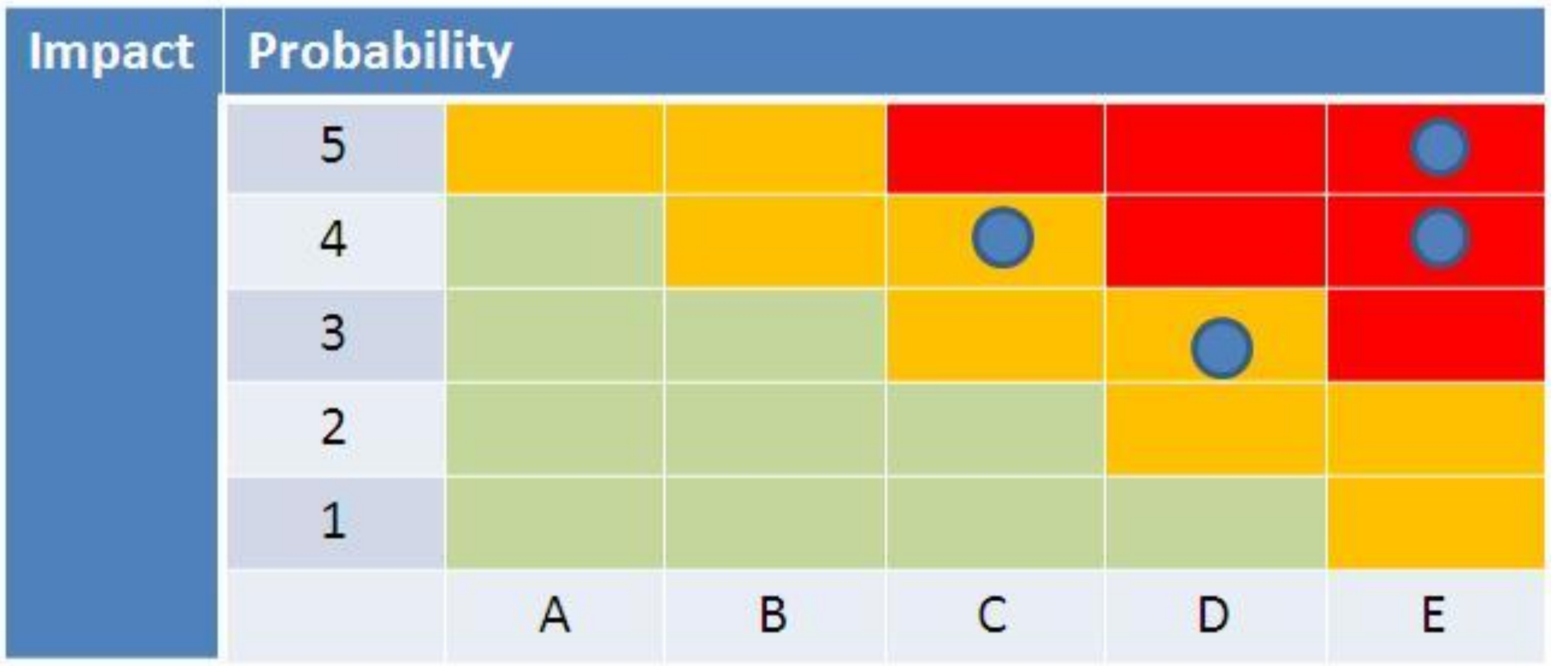
Prepare a “kid-free zone” of at least 3 feet (1 meter) around the stove.

Reduce chances of a fire. Keep anything that can catch fire away from stovetop.

Never dash out while cooking. Keep an eye on what you fry. Always cook with a lid beside your pan. If you have a fire, slide lid over pan and turn off burner.

Prep your kitchen by having a working smoke alarm. Keep smoke alarms at least 10 feet (3 meters) from the stove to reduce false alarms.

Risk Self Assessment



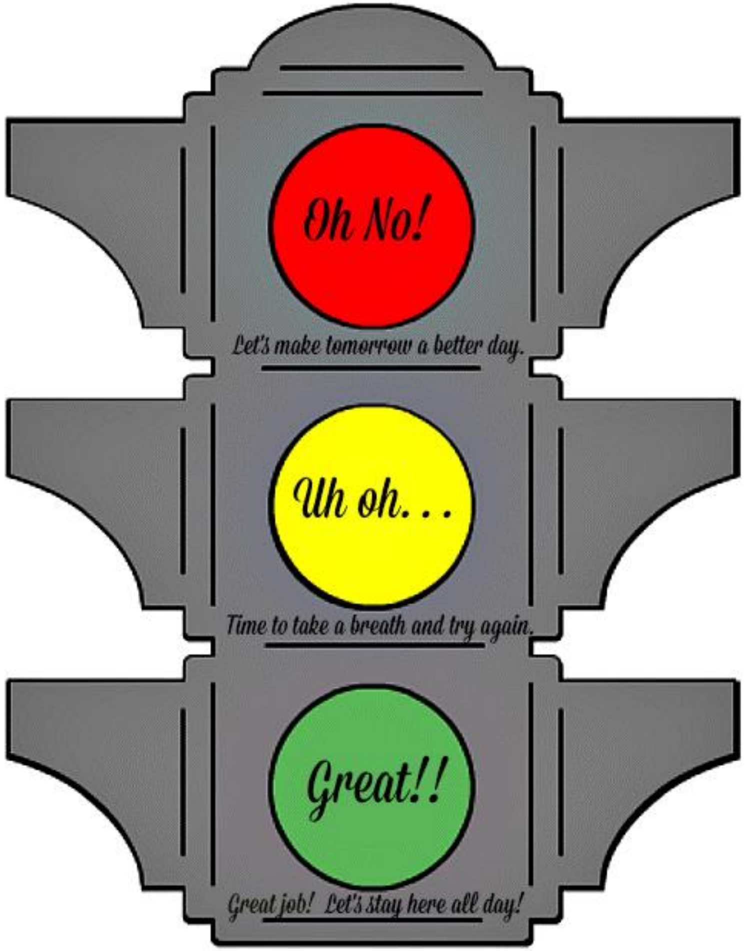
Impact	Probability				
5	🟧	🟧	🟥	🟥	🟥 ●
4	🟩	🟧	🟧 ●	🟥	🟥 ●
3	🟩	🟩	🟧	🟧 ●	🟥
2	🟩	🟩	🟩	🟧	🟧
1	🟩	🟩	🟩	🟩	🟧
	A	B	C	D	E

Risk Probability and Impact Assessment

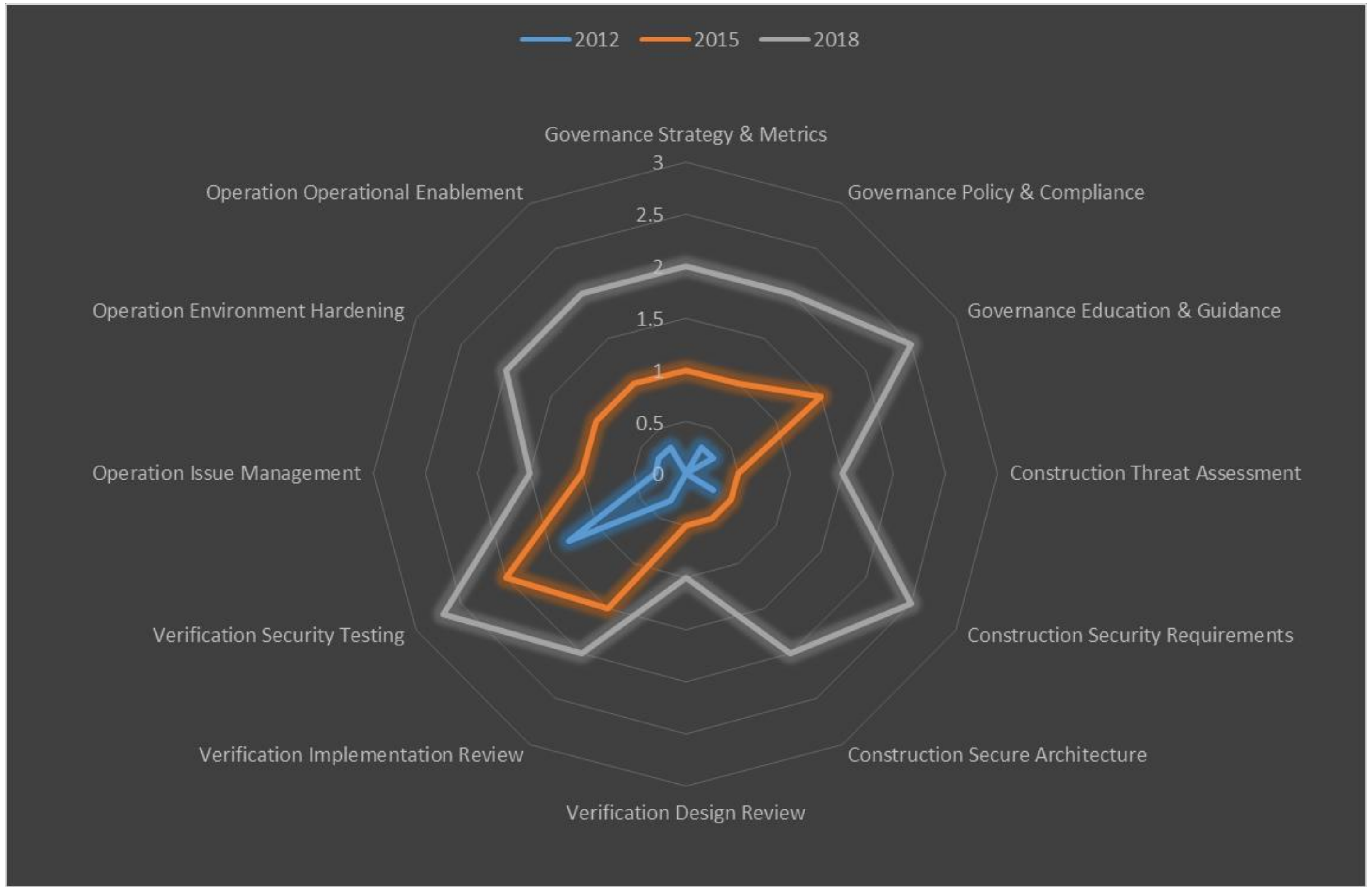
Probability: A – Rare; B – Unlikely; C- Possible; D – likely; E – Frequent

Impact: 1= Up to $100K; 2= up to $1MM; 3= up to $5MM; 4= up to $10MM; 5 =>$10MM

Define thresholds


Oh No!
Let's make tomorrow a better day.
Uh oh...
Time to take a breath and try again.
Great!!
Great job! Let's stay here all day!

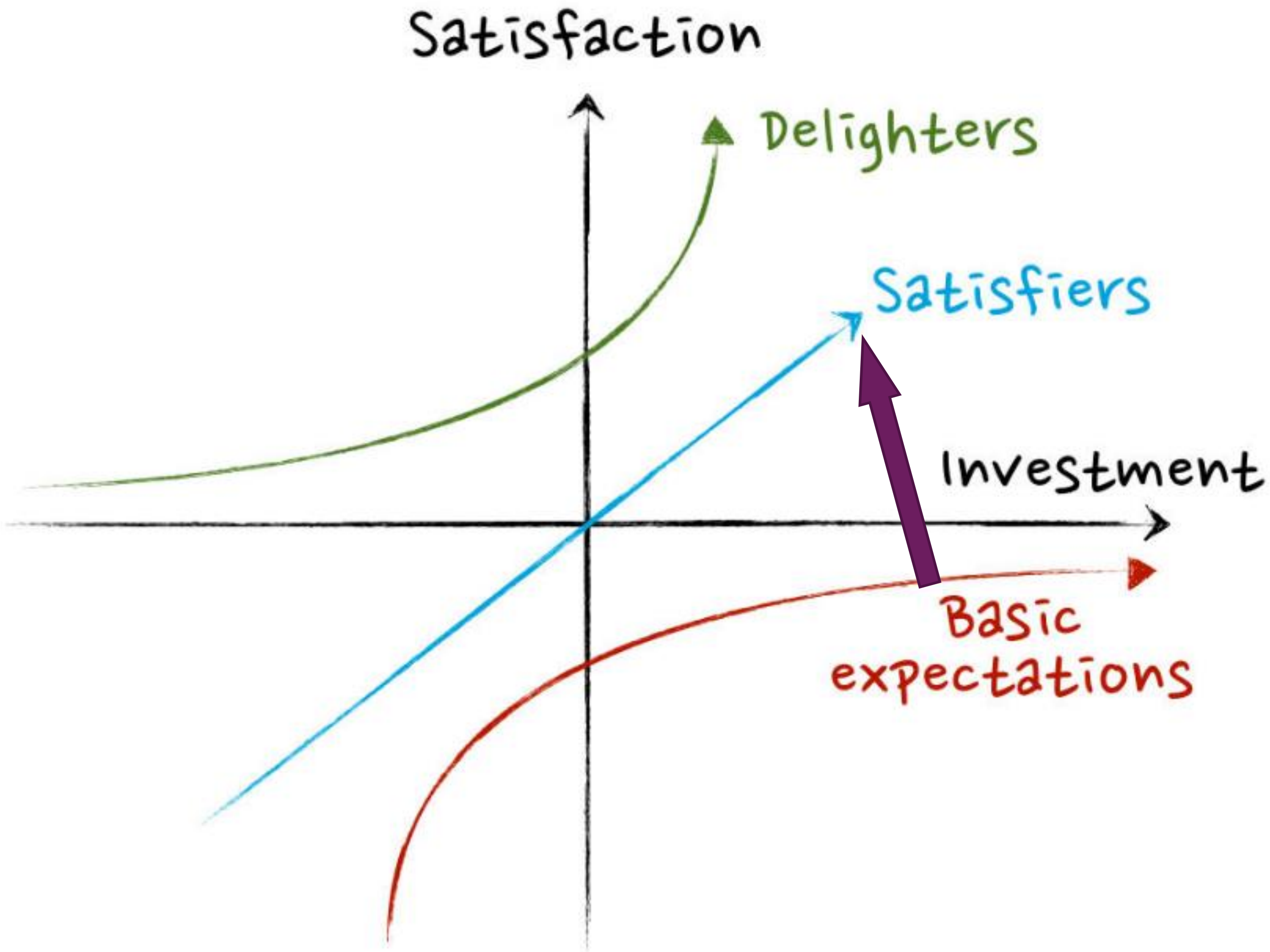
Use maturity models to track improvement



Enable flow

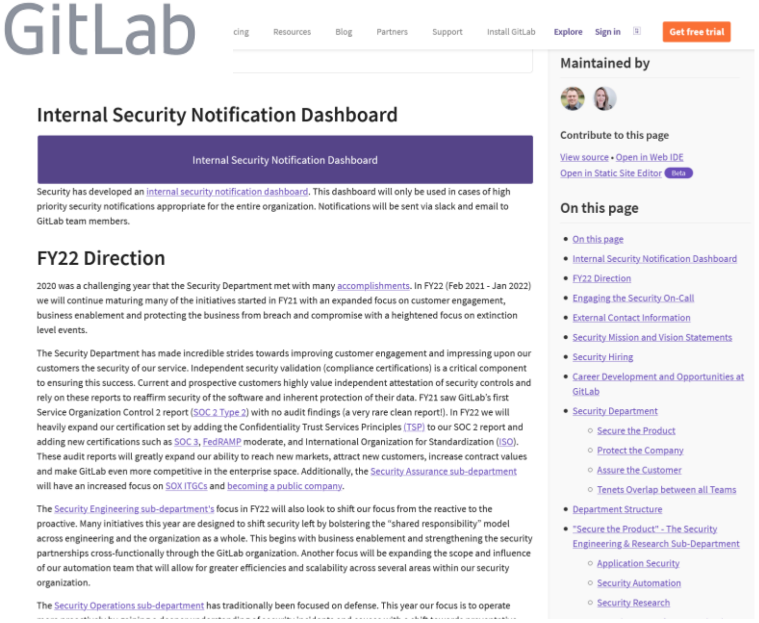
Step 5: Security as marketing tool

The challenge of being a security stakeholder



Xebia Security

Make it public



Internal Security Notification Dashboard

Internal Security Notification Dashboard

Security has developed an internal security notification dashboard. This dashboard will only be used in cases of high priority security notifications appropriate for the entire organization. Notifications will be sent via slack and email to GitLab team members.

FY22 Direction

2020 was a challenging year that the Security Department met with many accomplishments. In FY22 (Feb 2021 - Jan 2022) we will continue maturing many of the initiatives started in FY21 with an expanded focus on customer engagement, business enablement and protecting the business from breach and compromise with a heightened focus on extinction level events.

The Security Department has made incredible strides towards improving customer engagement and impressing upon our customers the security of our service. Independent security validation (compliance certifications) is a critical component to ensuring this success. Current and prospective customers highly value independent attestation of security controls and rely on these reports to reaffirm security of the software and inherent protection of their data. FY21 saw GitLab's first Service Organization Control 2 report (SOC 2 Type 2) with no audit findings (a very rare clean report!). In FY22 we will heavily expand our certification set by adding the Confidentiality Trust Services Principles (TSP) to our SOC 2 report and adding new certifications such as SOC 3, FedRAMP moderate, and International Organization for Standardization (ISO). These audit reports will greatly expand our ability to reach new markets, attract new customers, increase contract values and make GitLab even more competitive in the enterprise space. Additionally, the Security Assurance sub-department will have an increased focus on SOX ITGCs and becoming a public company.

The Security Engineering sub-department's focus in FY22 will also look to shift our focus from the reactive to the proactive. Many initiatives this year are designed to shift security left by bolstering the "shared responsibility" model across engineering and the organization as a whole. This begins with business enablement and strengthening the security partnerships cross-functionally through the GitLab organization. Another focus will be expanding the scope and influence of our automation team that will allow for greater efficiencies and scalability across several areas within our security organization.

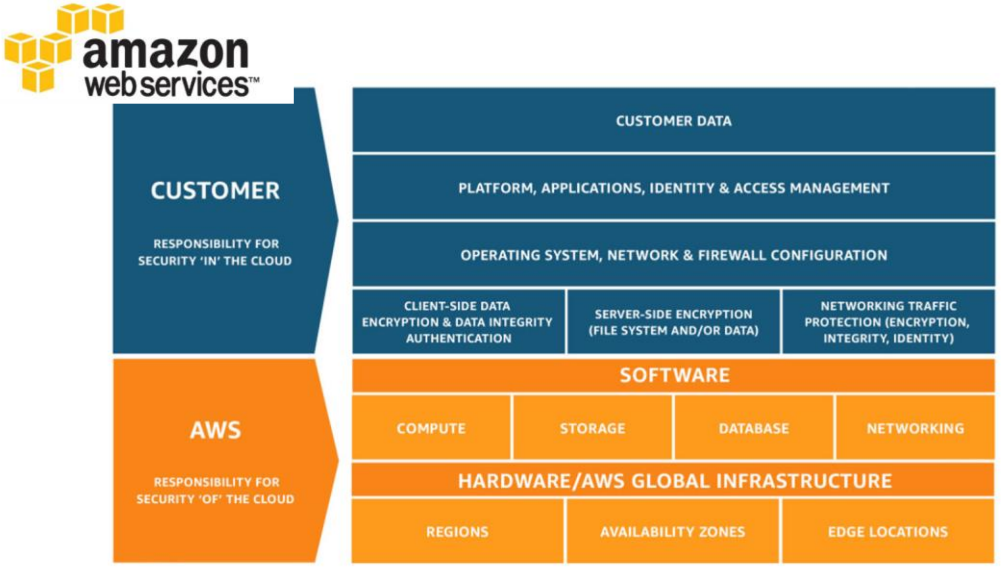
The Security Operations sub-department has traditionally been focused on defense. This year our focus is to operate

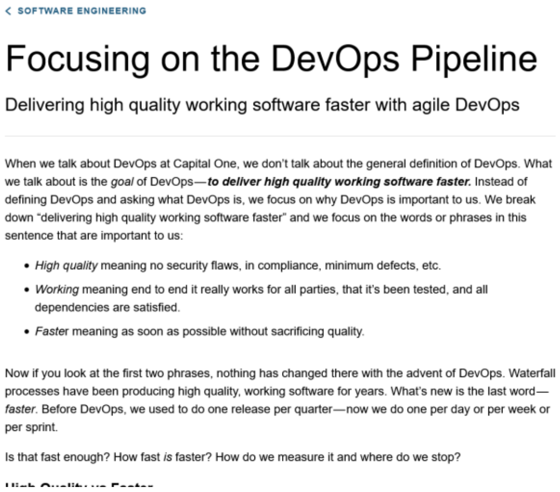
Maintained by

Contribute to this page

View source • Open in Web IDE
Open in Static Site Editor Beta

On this page


amazon web services™
CUSTOMER
RESPONSIBILITY FOR SECURITY 'IN' THE CLOUD
CUSTOMER DATA
PLATFORM, APPLICATIONS, IDENTITY & ACCESS MANAGEMENT
OPERATING SYSTEM, NETWORK & FIREWALL CONFIGURATION
CLIENT-SIDE DATA ENCRYPTION & DATA INTEGRITY AUTHENTICATION
SERVER-SIDE ENCRYPTION (FILE SYSTEM AND/OR DATA)
NETWORKING TRAFFIC PROTECTION (ENCRYPTION, INTEGRITY, IDENTITY)
AWS
RESPONSIBILITY FOR SECURITY 'OF' THE CLOUD
SOFTWARE
COMPUTE
STORAGE
DATABASE
NETWORKING
HARDWARE/AWS GLOBAL INFRASTRUCTURE
REGIONS
AVAILABILITY ZONES
EDGE LOCATIONS


< SOFTWARE ENGINEERING

Focusing on the DevOps Pipeline

Delivering high quality working software faster with agile DevOps

When we talk about DevOps at Capital One, we don't talk about the general definition of DevOps. What we talk about is the *goal* of DevOps—***to deliver high quality working software faster.*** Instead of defining DevOps and asking what DevOps is, we focus on why DevOps is important to us. We break down "delivering high quality working software faster" and we focus on the words or phrases in this sentence that are important to us:

- *High quality* meaning no security flaws, in compliance, minimum defects, etc.
- *Working* meaning end to end it really works for all parties, that it's been tested, and all dependencies are satisfied.
- *Faster* meaning as soon as possible without sacrificing quality.

Now if you look at the first two phrases, nothing has changed there with the advent of DevOps. Waterfall processes have been producing high quality, working software for years. What's new is the last word—*faster*. Before DevOps, we used to do one release per quarter—now we do one per day or per week or per sprint.

Is that fast enough? How fast *is* faster? How do we measure it and where do we stop?

High Quality vs Faster

Summary

XEBIA
Christopher's
BIG
CHERRY
WHOLE CHERRY CENTER
NET.WT. 1-9/16 OZ.

Security Agile Manifesto

Leaning in over Always Saying "No"

Data & Security Science over Fear, Uncertainty and Doubt

Open Contribution & Collaboration over Security-Only Requirements

Consumable Security Services with APIs over Mandated Security Controls

Business Driven Security Scores over Rubber Stamp Security

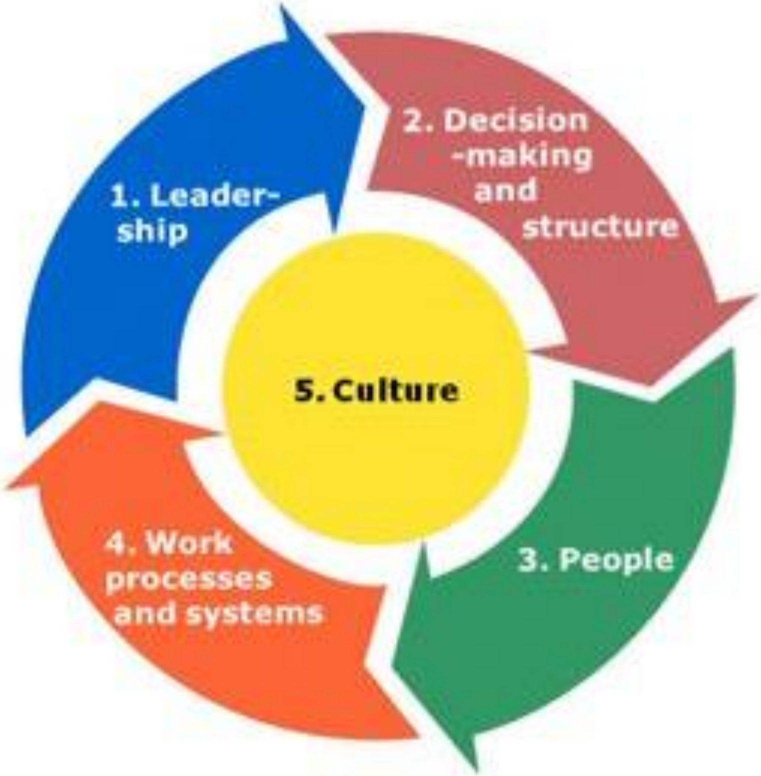
Team Based Exploit Testing over Relying on Scans & Theoretical Vulnerabilities

Proactive Security Monitoring over Reacting after being Informed of an Incident

Shared Threat Intelligence over Keeping Info to Ourselves

Compliance Operations over Clipboards & Checklists

A successful cooperation


1. Leader-ship
2. Decision-making and structure
5. Culture
3. People
4. Work processes and systems

Key Characteristics

- Clear vision and priorities
- Cohesive leadership team

- Clear roles and accountabilities for decisions
- Organizational structure that supports objectives

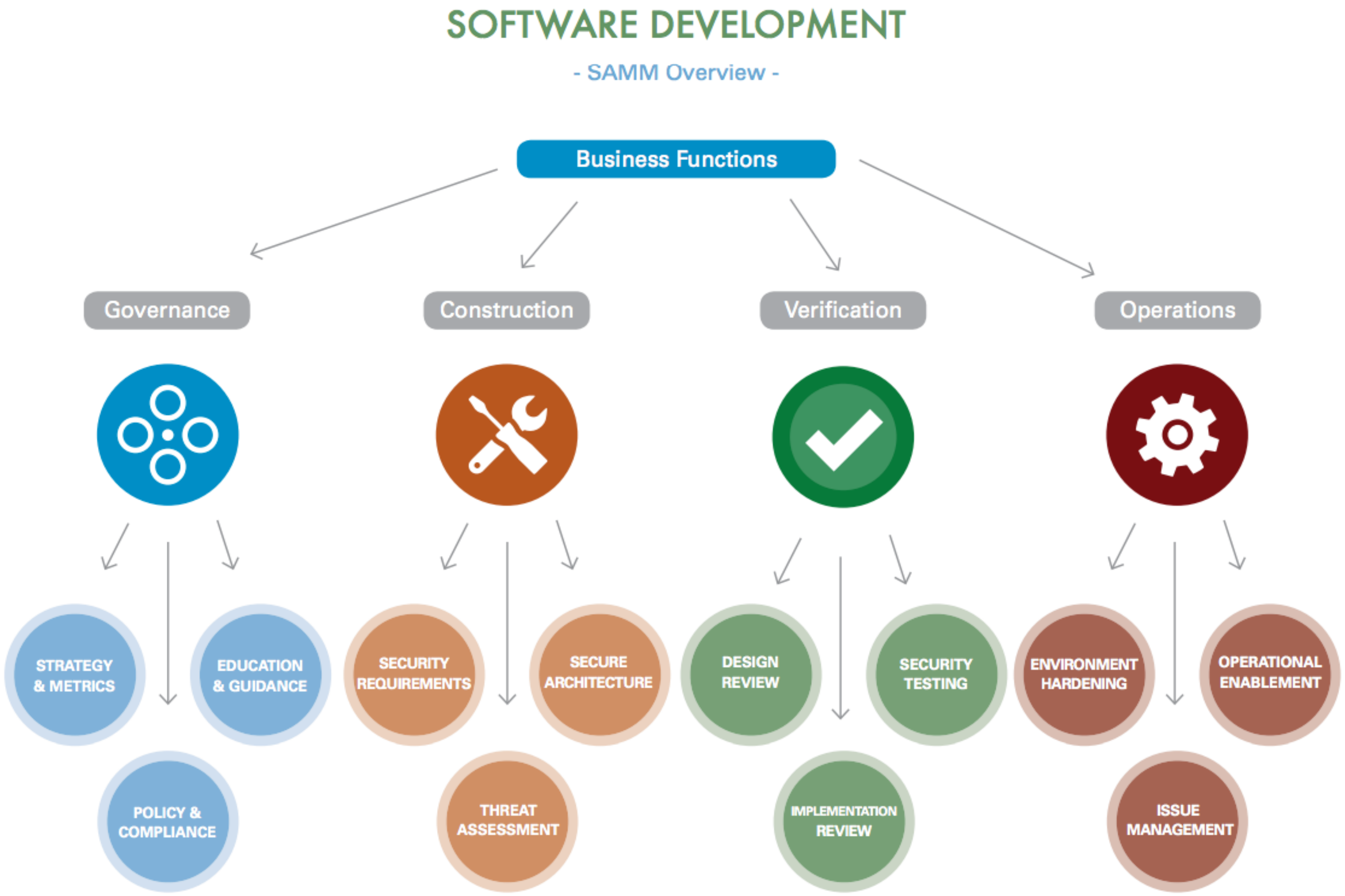
- Organizational and individual talent necessary for success
- Performance measures and incentives aligned to objectives

- Superior execution of programmatic work processes
- Effective and efficient support processes and systems

- 'High performance' values and behaviors
- Capacity to change

Source: Bain & Company organizational toolkit and Bridgespan analysis

Don’t reinvent the wheel



SOFTWARE DEVELOPMENT

- SAMM Overview -

Business Functions

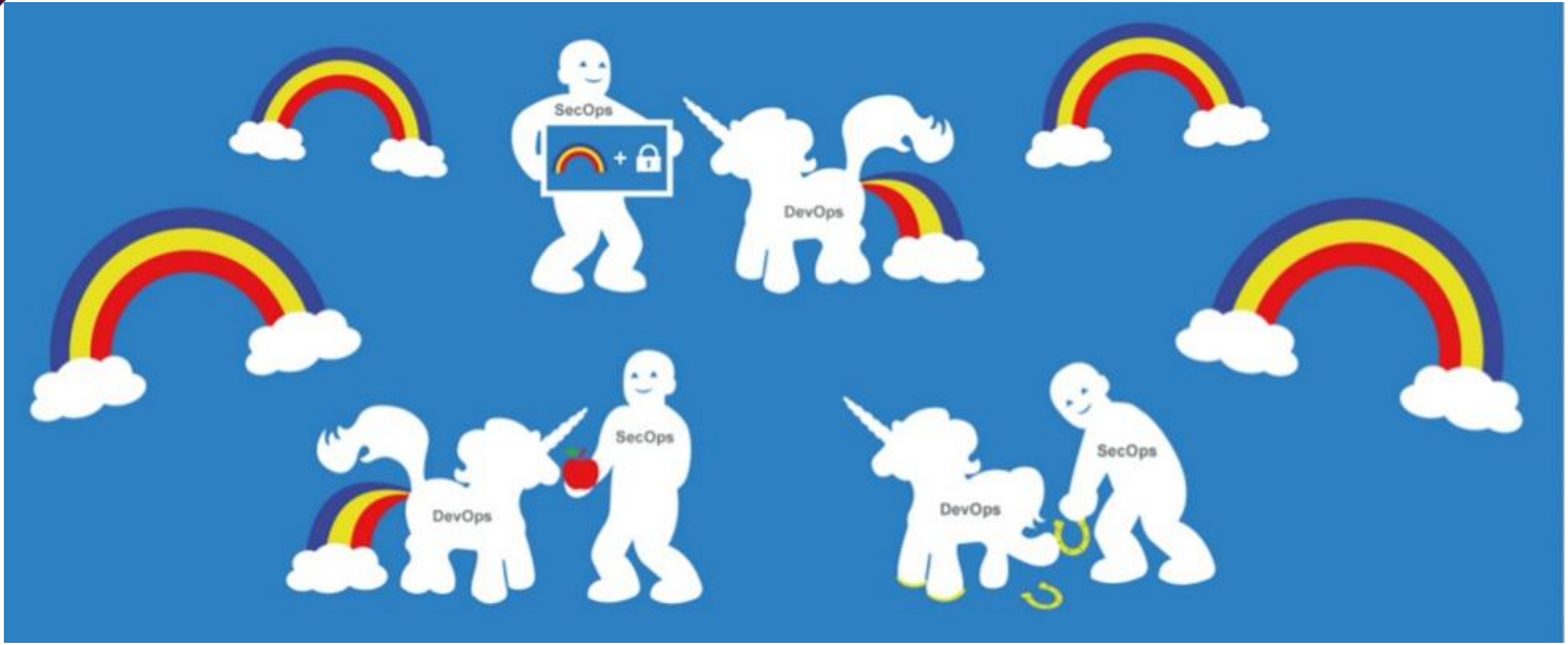
Governance
- STRATEGY & METRICS
- POLICY & COMPLIANCE
- EDUCATION & GUIDANCE

Construction
- SECURITY REQUIREMENTS
- THREAT ASSESSMENT
- SECURE ARCHITECTURE

Verification
- DESIGN REVIEW
- IMPLEMENTATION REVIEW
- SECURITY TESTING

Operations
- ENVIRONMENT HARDENING
- ISSUE MANAGEMENT
- OPERATIONAL ENABLEMENT

Utopia here we come!



Remarks & Questions