Business Value of CI, CD, & DevSecOps

Scaling Up to Billion User Global SoS Using Containerized Cloud-based Microservices

Dr. David F. Rico, PMP, CSEP, EBAS, BAF, FCP, FCT, ACP, CSM, SAFe, DevOps, AWS

Website: http://davidfrico.com • LinkedIn: http://linkedin.com/in/davidfrico • Twitter: @dr_david_f_rico

Agile Cost of Quality: http://www.davidfrico.com/agile-vs-trad-coq.pdf
DevOps Return on Investment (ROI): http://davidfrico.com/rico-devops-roi.pdf

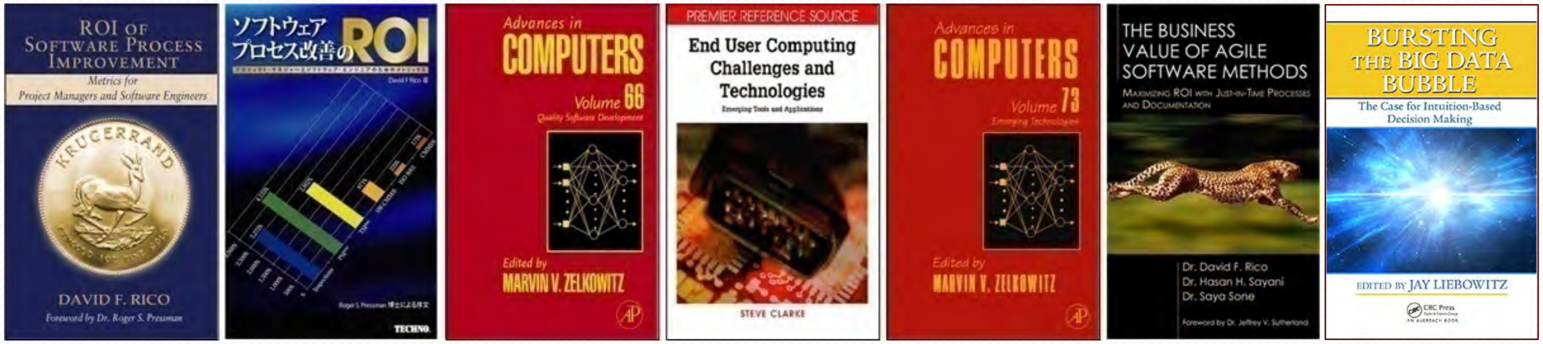
Dave's NEW Business Agility Video: http://www.youtube.com/watch?v=hTvtsAkL8xU
Dave's NEWER Scaled Agile Framework SAFe 4.5 Video: http://youtu.be/1TAuCRq5a34
Dave's NEWEST Development Operations Security Video: http://youtu.be/OBAdu4_t2EU
Dave's BRAND-NEW ROI of Lean Thinking Principles Video: http://youtu.be/wkMfaPAxO6E
Dave's REALLY-NEW ROI of Evolutionary Design Principles Video: http://youtu.be/TcXI26ClRb0
Dave's EXTREMELY-NEW ROI of Organizational Agility Principles Video: http://youtu.be/HOzDM5krtes

DoD Fighter Jets versus Amazon Web Services: http://davidfrico.com/dod-agile-principles.pdf
Principles of Collaborative Contracts: http://davidfrico.com/collaborative-contract-principles.pdf
Principles of Lean Organizational Leadership: http://davidfrico.com/lean-leadership-principles.pdf
Principles of Evolutionary Architecture: http://davidfrico.com/evolutionary-architecture-principles.pdf
Principles of CI, CD, & DevOps - Development Operations: http://davidfrico.com/devops-principles.pdf
Principles of SAFe Transformations - Scaled Agile Framework: http://davidfrico.com/safe-principles.pdf
Principles of Maximizing SAFe ROI - Scaled Agile Framework: http://davidfrico.com/safe-roi-principles.pdf
Principles of Lean-Agile - Contract Statements of Work (SOW): http://davidfrico.com/agile-sow-principles.pdf
Principles of Department of Defense (DoD) – Cloud Computing: http://davidfrico.com/dod-cloud-principles.pdf

Economic Value of Agile Businesses, Enterprises & Organizations - http://davidfrico.com/value-of-business-agility.pdf

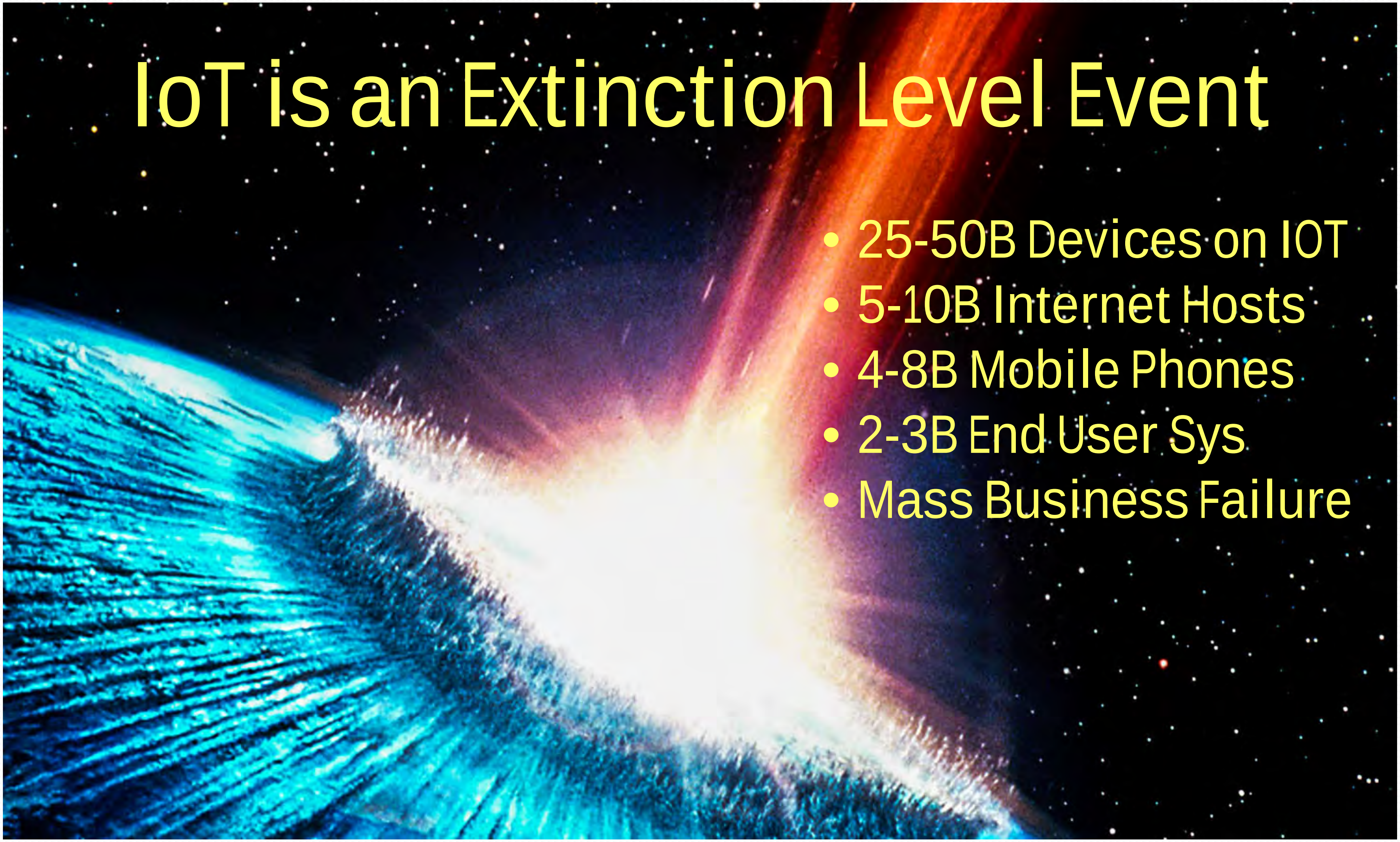
Author Background

- □ Management consultant 39+ years of IT experience
- □ B.S. Comp. Sci., M.S. Soft. Eng., & D.M. Info. Sys.
- ☞ □ Large IT projects in U.S., Far/Mid-East, & Europe


ROI of Software Process Improvement
Metrics for Project Managers and Software Engineers
KRUGERRAND
David F. Rico
Foreword by Dr. Roger S. Pressman
ソフトウェアプロセス改善のROI
TECHNO
Advances in COMPUTERS
Volume 66
Quality Software Development
Edited by Marvin V. Zelkowitz
AP
PREMIER REFERENCE SOURCE
End User Computing Challenges and Technologies
Emerging Tools and Applications
Steve Clarke
Advances in COMPUTERS
Volume 73
Emerging Technologies
Edited by Marvin V. Zelkowitz
AP
The Business Value of Agile Software Methods
Maximizing ROI with Just-in-Time Processes and Documentation
Dr. David F. Rico
Dr. Hasan H. Sayani
Dr. Saya Sone
Foreword by Dr. Jeffrey V. Sutherland
Bursting the Big Data Bubble
The Case for Intuition-Based Decision Making
Edited by Jay Liebowitz
CRC Press

- ✓ Career IT project management, systems and software engineering PROCESS strategist.
- ✓ Supported numerous billion-dollar enterprise digital transformation initiatives for 35+ years.
- ✓ Clients multi-billion government agencies, Fortune 500 conglomerates, and international IT firms.
- ✓ Included NASA's Space Station, Japanese Firms, Navy Fighters, NRO Satellites, and Intel Clouds, etc.
- ✓ Supported Digital Transformations at leading energy, healthcare, financial, and DoD enterprises and firms.
- ✓ Supported virtual casefile systems, data warehouses, data lakes, cloud migrations, and enterprise architectures.
- ✓ Specialized in Lean, Agile, Scrum, Scaled Agile Framework (SAFe), CI, CD, DevOps, DevSecOps, and Cloud Computing.
- ✓ Quickstart SAFe rollouts for critical portfolios, solutions, programs, projects, and new product development initiatives.
- ✓ Provides one-on-one and small group coaching services for C-levels, directors, managers, tech leaders, and developers.
- ✓ Skills include Lean, Agile, Scrum, SAFe, DevSecOps, Agile assessments, metrics, toolsets, dashboards, and case studies.
- ✓ Public speaker, author, blogger, trainer and holds over 15 professional certifications including SAFe SPC 5.0 and AWS CCP.
- ✓ Supported HHS, CMS, IRS, Exelon, ODNI IC-CIO, Intel, DoD, DoJ, USPS, NASA, DARPA, DISA, U.S. Air Force, Army, and Navy.

Internet of Things—Dinosaur Killer



IoT is an Extinction Level Event

- 25-50B Devices on IOT
- 5-10B Internet Hosts
- 4-8B Mobile Phones
- 2-3B End User Sys
- Mass Business Failure

DevSecOps—What is it?

- Dev-Ops (dĕv′ŏps) Early, iterative, & automated combo of development & operations; Incremental deployment
  - *An approach embracing principles & values of* lean thinking, product development flow, & agile methods
  - Early, collaborative, *and* automated *form of* incremental *development, integration, system, & operational testing*
  - *Design method that supports* collaboration, teamwork, iterative development, & responding to change
  - *Multi-tiered automated framework for* TDD, Continuous Integration, Continuous Delivery, DevOps, & **AppSec**
  - ☞ *Maximizes* **BUSINESS VALUE** *of organizations, portfolios, & projects by enabling buyers-suppliers to* scale globally

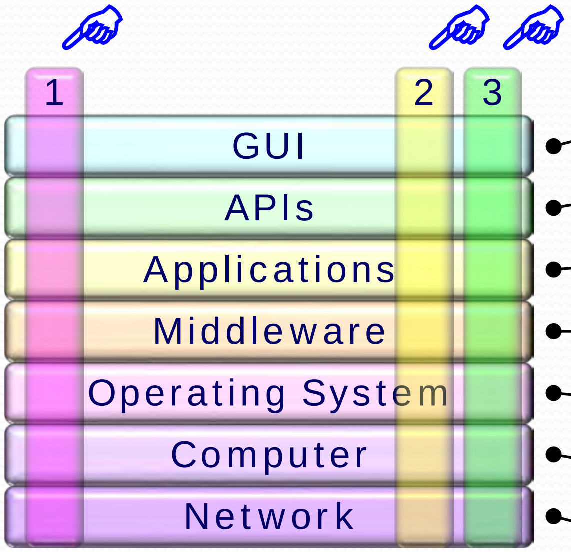
Crispin, L., & Gregory, J. (2009). *Agile testing: A practical guide for testers and agile teams*. Boston, MA: Addison-Wesley.
Crispin, L., & Gregory, J. (2015). *More agile testing: Learning journeys for the whole team*. Boston, MA: Addison-Wesley.

DevSecOps—How it works?

- □ Requirements are implemented in slices vs. layers
- □ User needs with higher business value are done first
- □ Reduces cost & risk while increasing business success

Agile

- Faster
- Early ROI
- Lower Costs
- Fewer Defects
- Manageable Risk
- Better Performance
- **Smaller Attack Surface**



Traditional

- Late
- No Value
- Cost Overruns
- Very Poor Quality
- Uncontrollable Risk
- Slowest Performance
- **More Security Incidents**

- JIT, Just-enough architecture
- Early, in-process system V&V
- Fast continuous improvement
- Scalable to systems of systems
- Maximizes successful outcomes

MINIMIZES

Seven Wastes

1. **Rework**
2. **Motion**
3. **Waiting**
4. **Inventory**
5. **Transportation**
6. **Overprocessing**
7. **Overproduction**

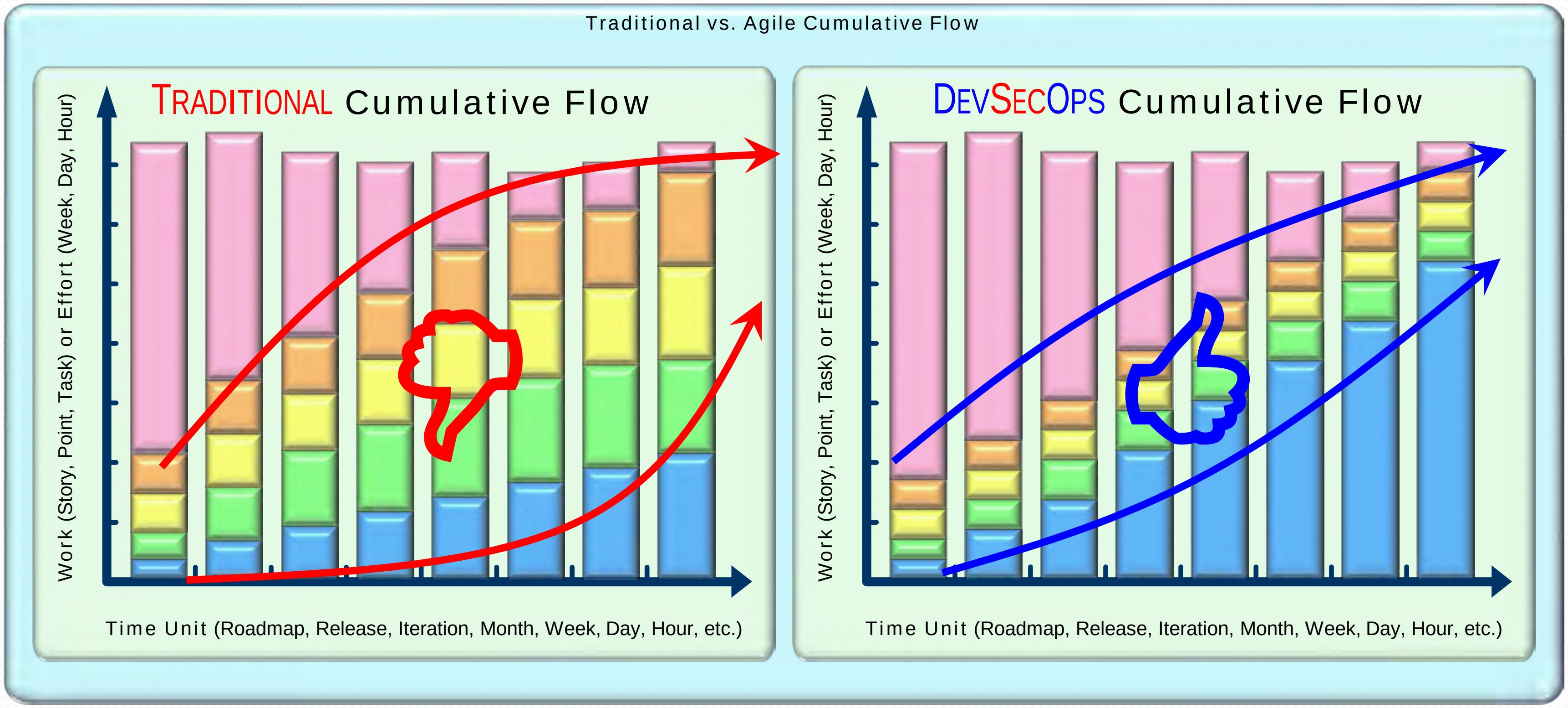
MAXIMIZES

- Myth of perfect architecture
- Late big-bang integration tests
- Year long improvement cycles
- Breaks down on large projects
- Undermines business success

Shore, J. (2011). Evolutionary design illustrated. *Norwegian Developers Conference, Oslo, Norway*.

DevSecOps—Workflow Results

- □ Late big bang integration increases WIP backlog
- □ Agile testing early and often reduces WIP backlog
- ☞ □ Improves workflow and reduces WIP & lead times



Anderson, D. J. (2004). *Agile management for software engineering*. Upper Saddle River, NJ: Pearson Education.
Anderson, D. J. (2010). *Kanban: Successful evolutionary change for your technology business*. Sequim, WA: Blue Hole Press.

DevSecOps—MMF, MVP, MVA, etc.

- □ Methods to "scope" project, product, or system
- □ "Key" is smallest possible scope with highest value
- □ Reduces cost, risk, time, failure, & tech. obsolescence

MINIMUM MARKETABLE FEATURE - MMF -

- ✓ Advantage
- ✓ Difference
- ✓ Revenue
- ✓ Profit
- ✓ Savings
- ✓ Brand
- ✓ Loyalty

MINIMUM VIABLE PRODUCT - MVP -

- ✓ Goal
- ✓ Process
- ✓ Features
- ✓ Priorities
- ✓ Story Map
- ✓ Architecture

STORY MAP OR IMPACT MAP - SM or IM -

- ✓ Goal
- ✓ Actors
- ✓ Impacts
- ✓ Deliverables
- ✓ Measures
- ✓ Milestones

VISION STATEMENT - VS -

- ✓ For *<customer>*
- ✓ Who *<needs it>*
- ✓ The *<product>*
- ✓ Is a *<benefit>*
- ✓ That *<customer>*
- ✓ Unlike *<other>*
- ✓ Ours *<different>*

MICRO-SERVICE - MS -

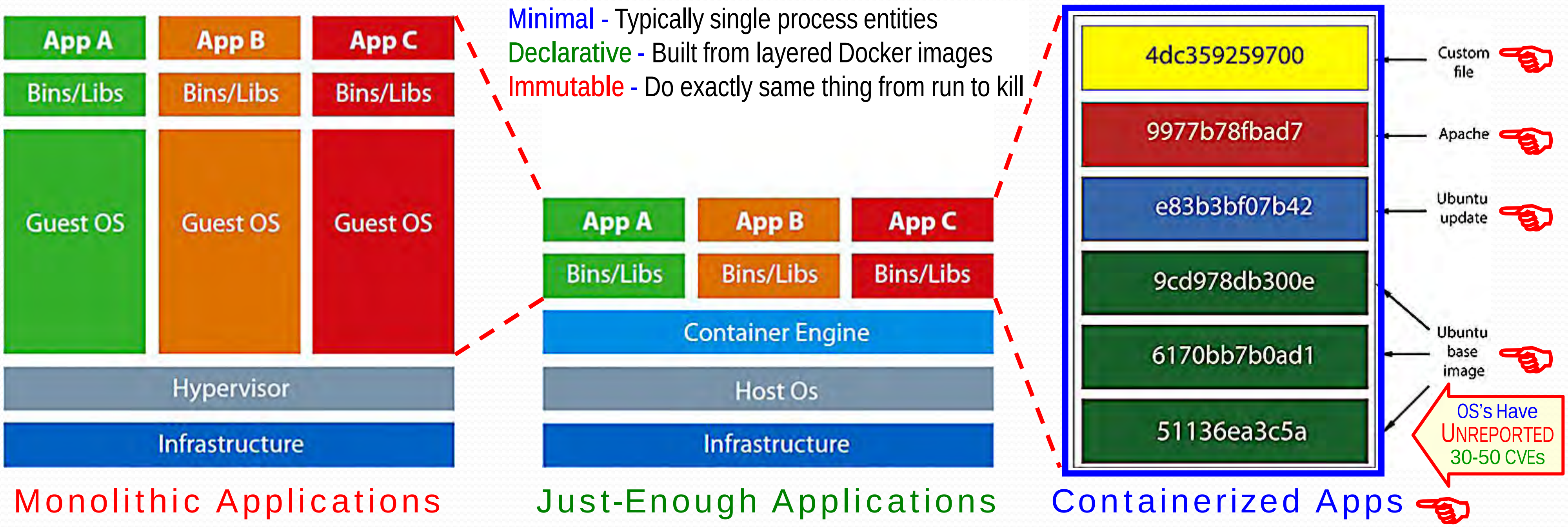
- ✓ Purpose
- ✓ Automated
- ✓ Unique
- ✓ Independent
- ✓ Resilient
- ✓ Ecosystem
- ✓ Consumer

INCREASES TESTABILITY, QUALITY, RELIABILITY, SECURITY, MORALE, MAINTAINABILITY, & SUCCESS

Denne, M., & Cleland-Huang, J. (2004). *Software by numbers*: *Low-risk, high-return development*. Santa Clara, CA: Sun Microsystems.
Ries, E. (2011). *The lean startup*: *How today's entrepreneurs use continuous innovation*. New York, NY: Crown Publishing.
Patton, J. (2014). *User story mapping*: *Discover the whole story, build the right product*. Sebastopol, CA: O'Reilly Media.
Layton, M. C., & Maurer, R. (2011). *Agile project management for dummies*. Hoboken, NJ: Wiley Publishing.
Krause, L. (2014). *Microservices*: *Patterns and applications*. Paris, France: Lucas Krause.

DevSecOps—Microservices

- □ Lightweight, fast, disposable virtual environments
- □ Set of isolated processes running on shared kernel
- □ Efficient way for building, delivering, & running apps



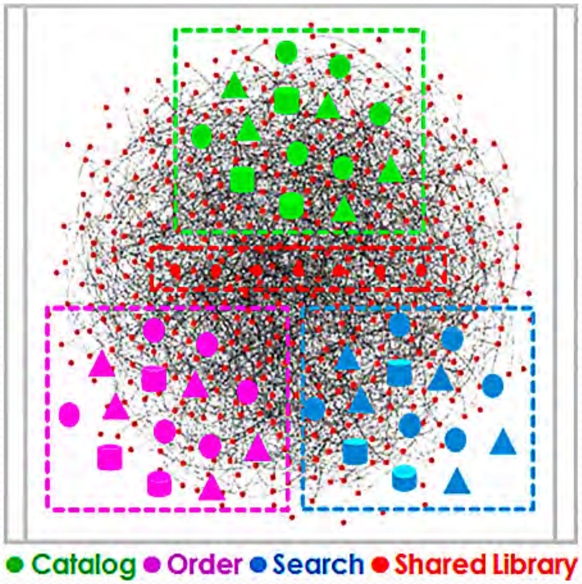
Minimal - Typically single process entities
Declarative - Built from layered Docker images
Immutable - Do exactly same thing from run to kill

- Small autonomous services that work together
- Self-contained process that provides a unique capability
- Loosely coupled service oriented architecture with bounded contexts
- Small independent processes communicating with each other using language-agnostic APIs
- Fined-grained independent services running in their own processes that are developed and deployed independently
- Suite of services running in their own process, exposing APIs, and doing one thing well (independently developed and deployable)
- Single app as a suite of small services, each running in its own process and communicating with lightweight mechanisms (HTTP APIs)

Krause, L. (2014). *Microservices: Patterns and applications*. Paris, France: Lucas Krause.

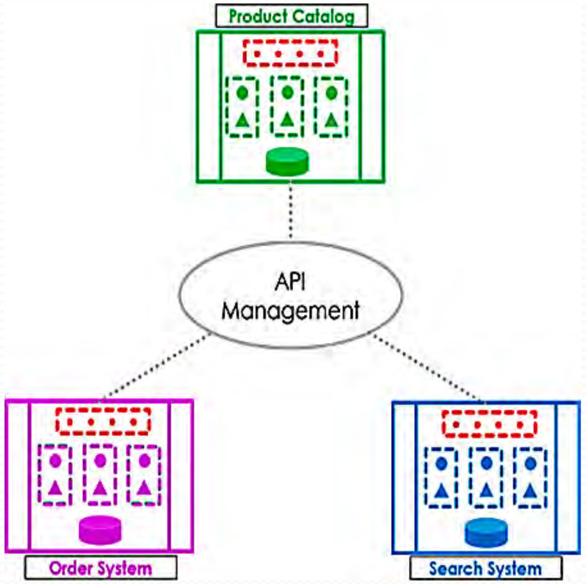
DevSecOps—Monolith to µServices

Domain Driven Design



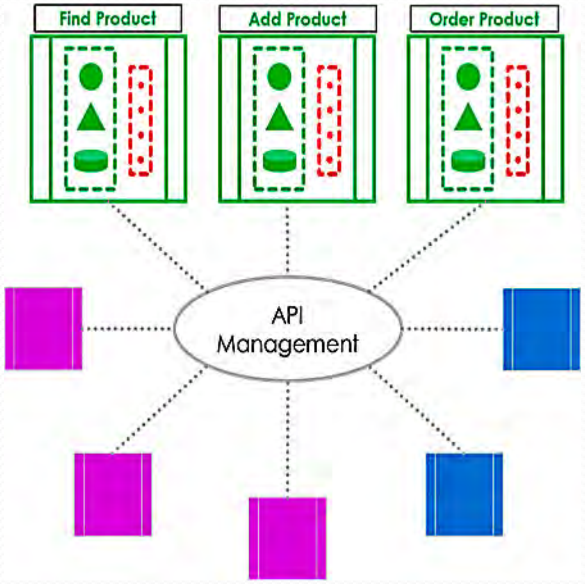
- Aligned to Business
- Better Organized
- Shared Libraries
- Fewer Dependencies
- Portable/Changeable
- Faster Testing
- Enables Scaled Agile Teams

Service-Based Architecture



- Separately Deployable Systems
- Shared Database per System
- Decoupled Business Systems
- Fewer Defects/Breaking Bugs
- More Development Options
- More Infrastructure Options
- Enables Small Agile Teams

Microservice Architecture



- Decoupled Business Functions
- Local Database per Service
- Separately Deployable Services
- CI, CD, and Fast Deployments
- Release on Demand/Fast Recovery
- Container Ready and Cloud Ready
- Enables Tiny Two-Pizza Teams

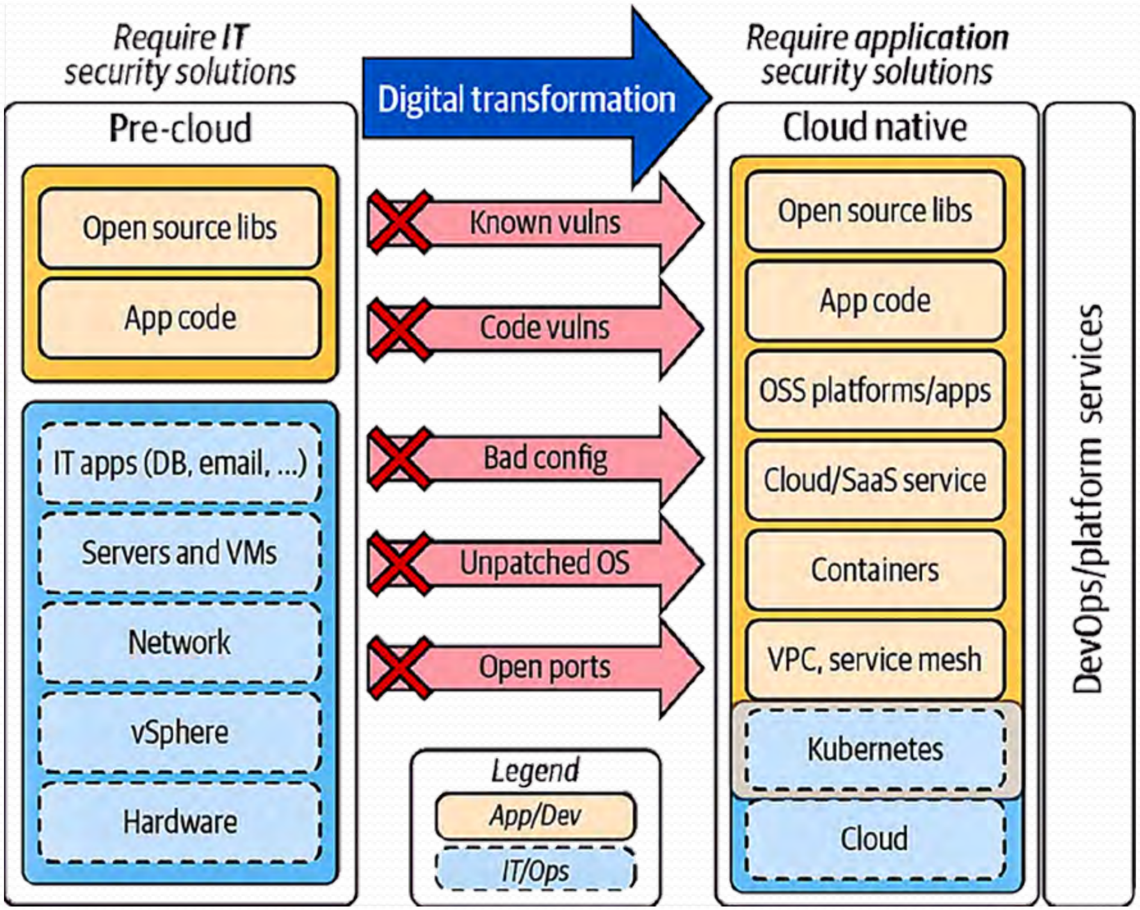
- **Reverse Conway's Law**
- **Use Strangler Application Pattern**
- **Test Within Domains (vs. Across Domains)**
- **Avoid Canonical and Master Data Definitions**
- **Not All Monoliths Are Evil (However, Most Are)**
- **Plan to Re-Architect in Five Years (Moore's Law)**
- **Lean-Agile practices rarely scale to high-risk solutions**

Rix, M. (2019). Conquering the monolith: Architecting for DevOps and release on demand. *SAFe Summit Europe, Hague, Netherlands*.

Newman, S. (2019). *Monolith to microservices*: *Evolutionary patterns to transform your monolith*. Sebastopol, CA: O'Reilly.

DevSecOps—Cloud Native µServices

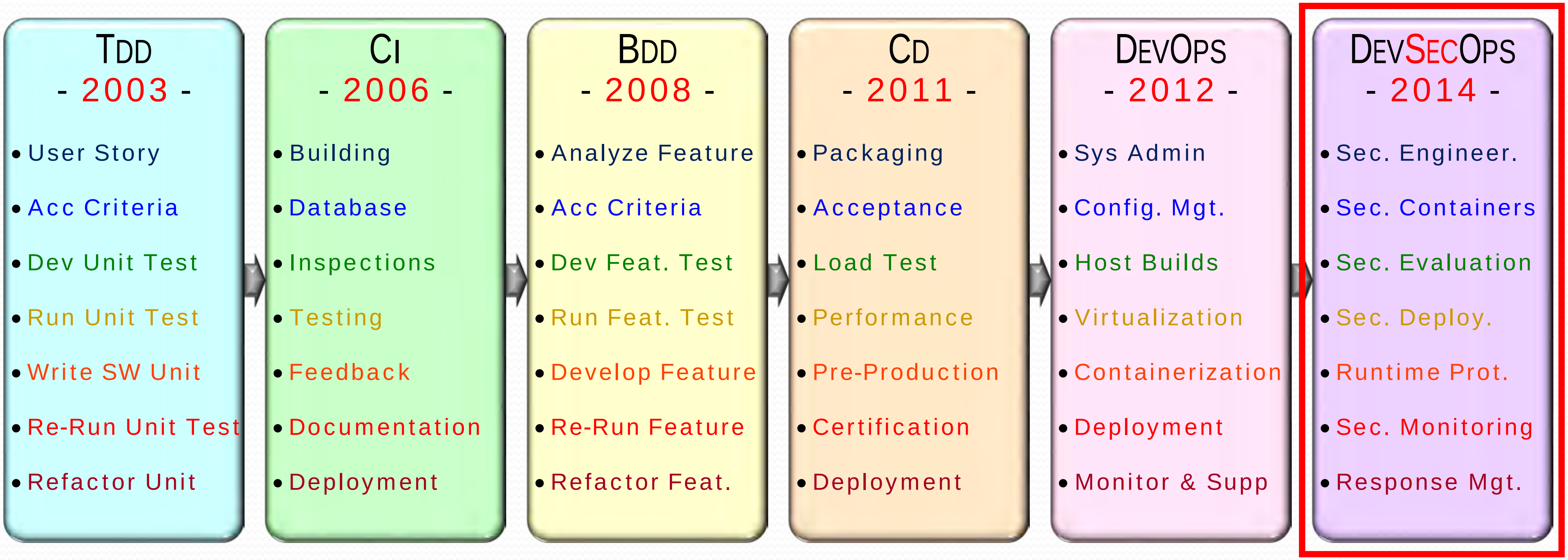
- □ Cloud native microservices have security concerns
- □ Developers must first concentrate on code appsec
- ☞ □ Then focus on middleware, VMs, & network sec



Podjarny, G. (2021). Cloud native application security: Embracing developer-first security for the cloud era. Sebastopol, CA: O'Reilly.

DevSecOps—Evolution

- □ Numerous models of lean-agile testing emerging
- □ Based on principles of lean & agile one piece flow
- ☞ □ Include software, hardware, system, & port. testing



TDD - 2003 -

- User Story
- Acc Criteria
- Dev Unit Test
- Run Unit Test
- Write SW Unit
- Re-Run Unit Test
- Refactor Unit

CI - 2006 -

- Building
- Database
- Inspections
- Testing
- Feedback
- Documentation
- Deployment

BDD - 2008 -

- Analyze Feature
- Acc Criteria
- Dev Feat. Test
- Run Feat. Test
- Develop Feature
- Re-Run Feature
- Refactor Feat.

CD - 2011 -

- Packaging
- Acceptance
- Load Test
- Performance
- Pre-Production
- Certification
- Deployment

DEVOPS - 2012 -

- Sys Admin
- Config. Mgt.
- Host Builds
- Virtualization
- Containerization
- Deployment
- Monitor & Supp

DEVSECOPS - 2014 -

- Sec. Engineer.
- Sec. Containers
- Sec. Evaluation
- Sec. Deploy.
- Runtime Prot.
- Sec. Monitoring
- Response Mgt.

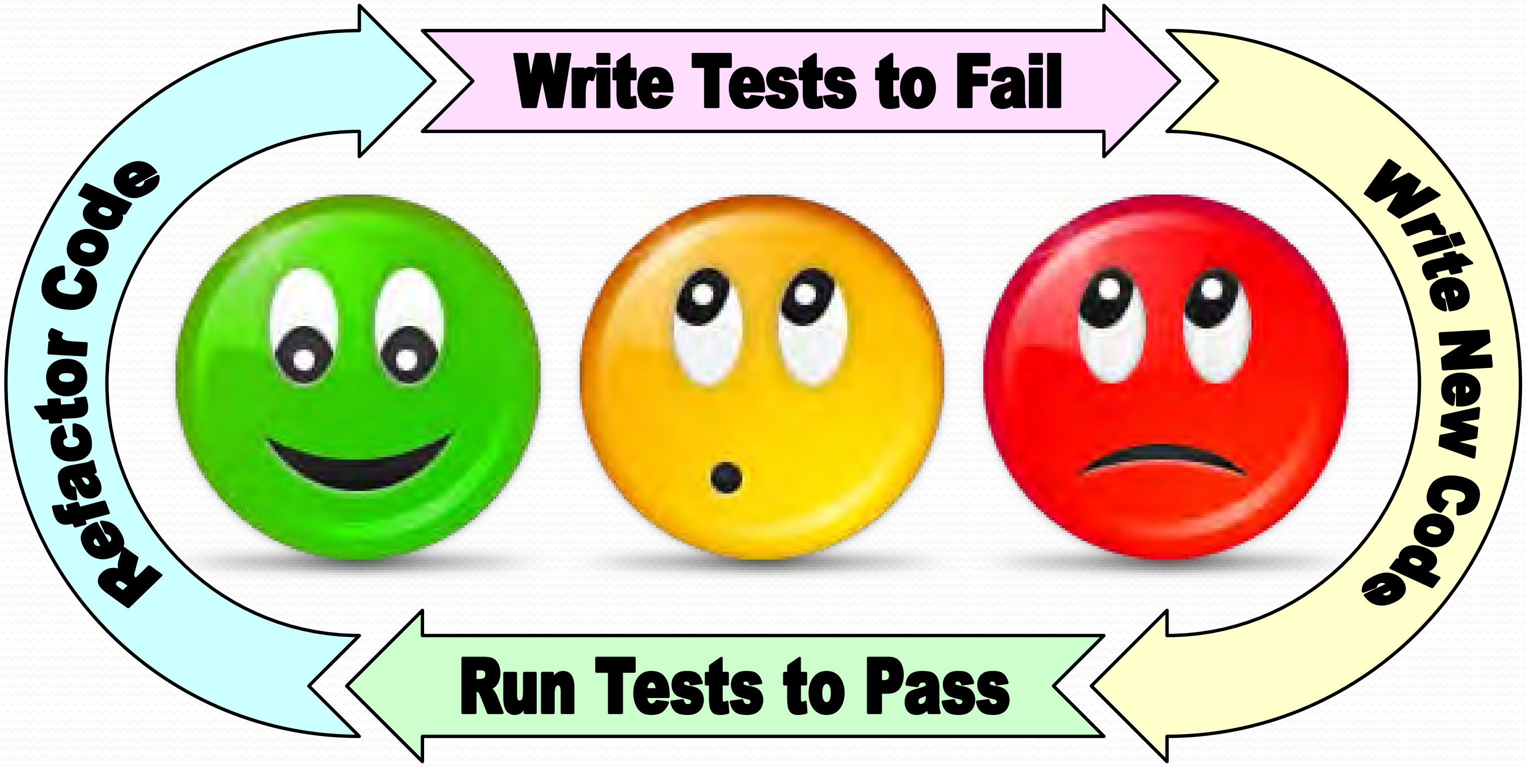
Beck, K. (2003). *Test-driven development*: *By example*. Boston, MA: Addison-Wesley.
Duvall, P., Matyas, S., & Glover, A. (2006). *Continuous integration*. Boston, MA: Addison-Wesley.
Barker, K., & Humphries, C. (2008). *Foundations of rspec*: *Behavior driven development with ruby and rails*. New York, NY: Apress.
Humble, J., & Farley, D. (2011). *Continuous delivery*. Boston, MA: Pearson Education.
Huttermann, M. (2012). *Devops for developers*: *Integrate development and operations the agile way*. New York, NY: Apress.
Bird, J. (2016). *Devopssec*: *Delivering secure software through continuous delivery*. Sebastopol, CA: O'Reilly Media.

STAGE 1—Test Driven Development

□ Term coined by Kent Beck in 2003

□ Consists of writing all tests before design

☞ □ Ensures all components are verified and validated



Write Tests to Fail

Write New Code

Run Tests to Pass

Refactor Code

Beck, K. (2003). *Test-driven development*: *By example*. Boston, MA: Addison-Wesley.

STAGE 1—Test Driven Develop.

- □ Agile TDD consists of seven broad practices
- □ Document test criteria, tests, software units, etc.
- ☞ □ Include refactoring, verification, optimization, etc.

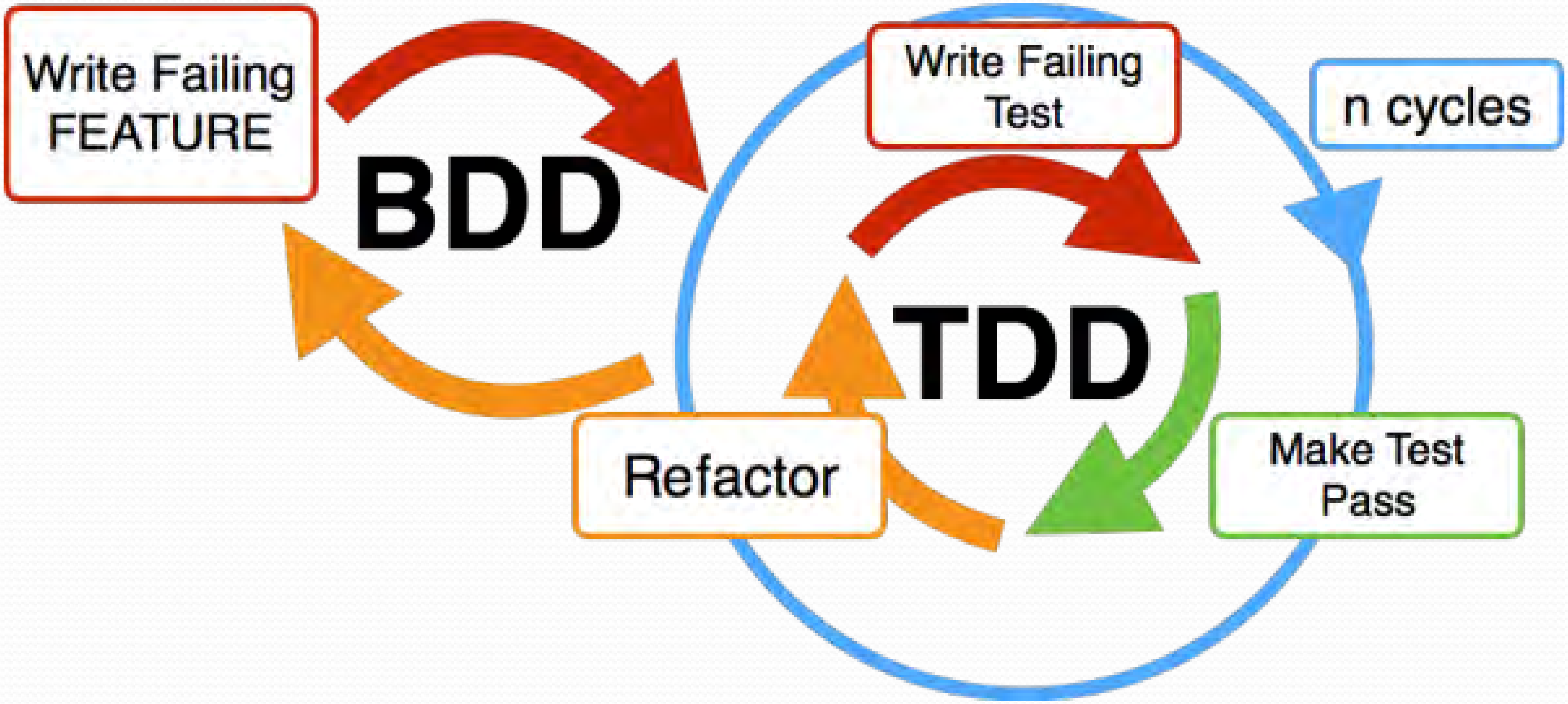
Practice	Description
User Story	Read story, analyze meaning, ask questions, and clarify understanding
Acc Criteria	Identify, verify, and document acceptance criteria for each user story
Dev Test	Design, develop, code, and verify automated unit test for user story
Run Test	Run automated unit test to verify that it fails the first time (sanity check)
Dev Unit	Design, develop, code, and verify the software unit to satisfy user story
Rerun Test	Rerun automated unit test to see if code satisfies automated unit test
Refactor Unit	Refine, reduce, and simplify code to remove waste and optimize performance

Beck, K. (2003). *Test-driven development*: *By example*. Boston, MA: Addison-Wesley.

STAGE 2—Behavior Driven Develop.

- □ Term coined by Dan North in 2006
- □ Consists of writing feature tests before design
- ☞ □ Ensures all system features are verified and validated

☞



Smart, J. F. (2014). *BDD in action*: *Behavior-driven development for the whole software lifecycle*. Shelter Island, NY: Manning Publications.

STAGE 2—Behavior Driven Dev.

- □ Agile BDD consists of seven broad practices
- □ Document test criteria, tests, syst. features, etc.
- ☞ □ Include refactoring, verification, optimization, etc.

Practice	Description
Feature	Read feature, analyze meaning, ask questions, and clarify understanding
Acc Criteria	Identify, verify, and document acceptance criteria for each feature
Dev Test	Design, develop, code, and verify automated feature test for feature
Run Test	Run automated feature test to verify that it fails the first time (sanity check)
Dev Feature	Design, develop, code, and verify the feature software to satisfy feature
Rerun Test	Rerun automated feature test to see if code satisfies automated feature test
Refac Feature	Refine, reduce, and simplify code to remove waste and optimize performance

Smart, J. F. (2014). *BDD in action*: *Behavior-driven development for the whole software lifecycle*. Shelter Island, NY: Manning Publications.

STAGE 3—Continuous Integration

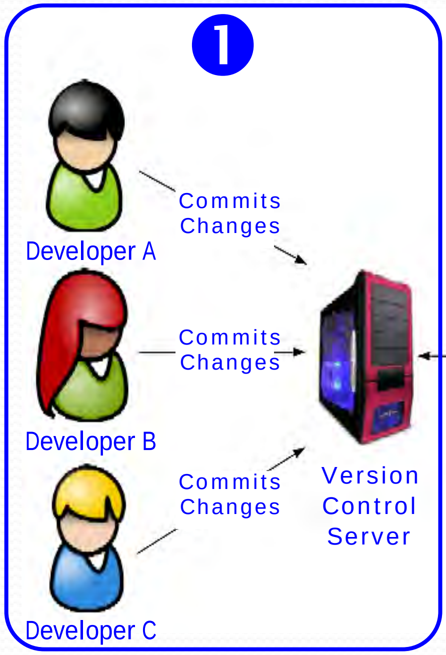
- □ Term coined by Martin Fowler circa 1998
- □ User needs designed & developed one-at-a-time
- ☞ □ Changes automatically detected, built, & fully-tested

☞ Early, Automated, Fast, Efficient, & Repeatable

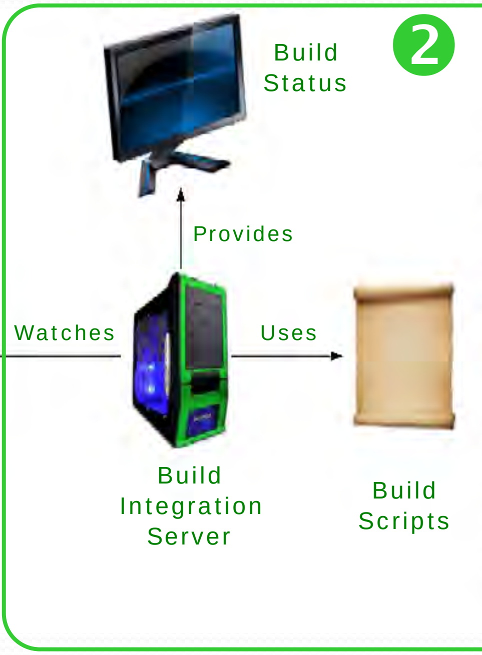
☞ Thousands of Tests Continuously Executed

☞ Lean, Waste Free, Low WIP, No Deadlocked Test Queues

❶



❷



- ✓ Builds
- ✓ Database
- ✓ Analysis
- ✓ Testing
- ✓ Reporting
- ✓ Documentation
- ✓ Deployment

❸

- ALL DEVELOPERS RUN PRIVATE BUILDS
- DEVELOPERS COMMIT CODE TO VERSION CONTROL
- INTEGRATION BUILDS OCCUR SEVERAL TIMES PER DAY
- 100% OF SYSTEM TESTS MUST PASS FOR EVERY BUILD
- A SHIPPABLE PRODUCT RESULTS FROM EVERY BUILD
- FIXING BROKEN BUILDS IS OF THE HIGHEST PRIORITY
- REPORTS AUTOMATICALLY GENERATED & REVIEWED

☞ Constant Readiness State & CM Control

☞ No More Late Big Bang Integration

☞ Rapidly & Successfully Dev. Complex Systems

Humble, J., & Farley, D. (2011). *Continuous delivery*. Boston, MA: Pearson Education.
Duvall, P., Matyas, S., & Glover, A. (2006). *Continuous integration*. Boston, MA: Addison-Wesley.

STAGE 3—Continuous Integration

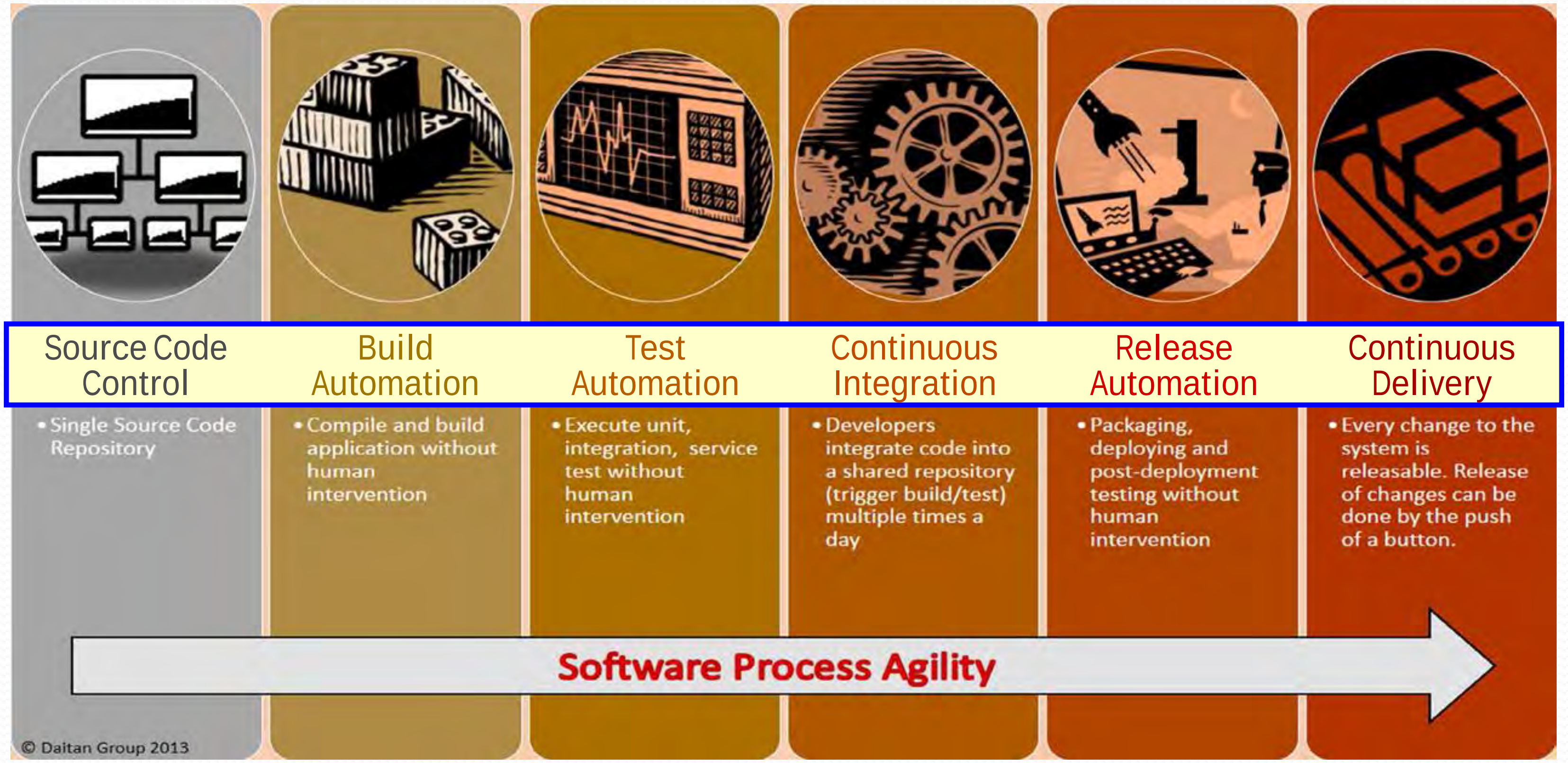
- □ Agile CI consists of seven broad practices
- □ Automated build, database, inspection, tests, etc.
- ☞ □ Include reporting, documentation, deployment, etc.

Practice	Description
Building	Frequently assembling products and services to ensure delivery readiness
Database	Frequently generating/analyzing database schemas, queries, and forms
Inspections	Frequently performing automated static analysis of product/service quality
Testing	Frequently performing automated dynamic product and service evaluation
Feedback	Frequently generating automated status reports/messages for all stakeholders
Documentation	Frequently performing automated technical/customer document generation
Deployment	Frequently performing automated delivery of products/services to end users

Duvall, P., Matyas, S., & Glover, A. (2006). *Continuous integration*: *Improving software quality and reducing risk*. Boston, MA: Addison-Wesley.
Humble, J., & Farley, D. (2011). *Continuous delivery*. Boston, MA: Pearson Education.

STAGE 4—Continuous Delivery

- □ Created by Jez Humble of ThoughtWorks in 2011
- □ Includes CM, build, testing, integration, release, etc.
- ☞ □ Goal is one-touch automation of deployment pipeline



☞ Source Code Control | Build Automation | Test Automation | Continuous Integration | Release Automation | Continuous Delivery

- Single Source Code Repository
- Compile and build application without human intervention
- Execute unit, integration, service test without human intervention
- Developers integrate code into a shared repository (trigger build/test) multiple times a day
- Packaging, deploying and post-deployment testing without human intervention
- Every change to the system is releasable. Release of changes can be done by the push of a button.

CoQ

- 80% MS Tst
- 8/10 No Val
- $24B in 90s
- Rep by CD
- Not Add MLK

Humble, J., & Farley, D. (2011). *Continuous delivery*. Boston, MA: Pearson Education.
Duvall, P., Matyas, S., & Glover, A. (2006). *Continuous integration*. Boston, MA: Addison-Wesley.
Ohara, D. (2012). *Continuous delivery and the world of devops*. San Francisco, CA: GigaOM Pro.

Stage 4—Continuous Delivery

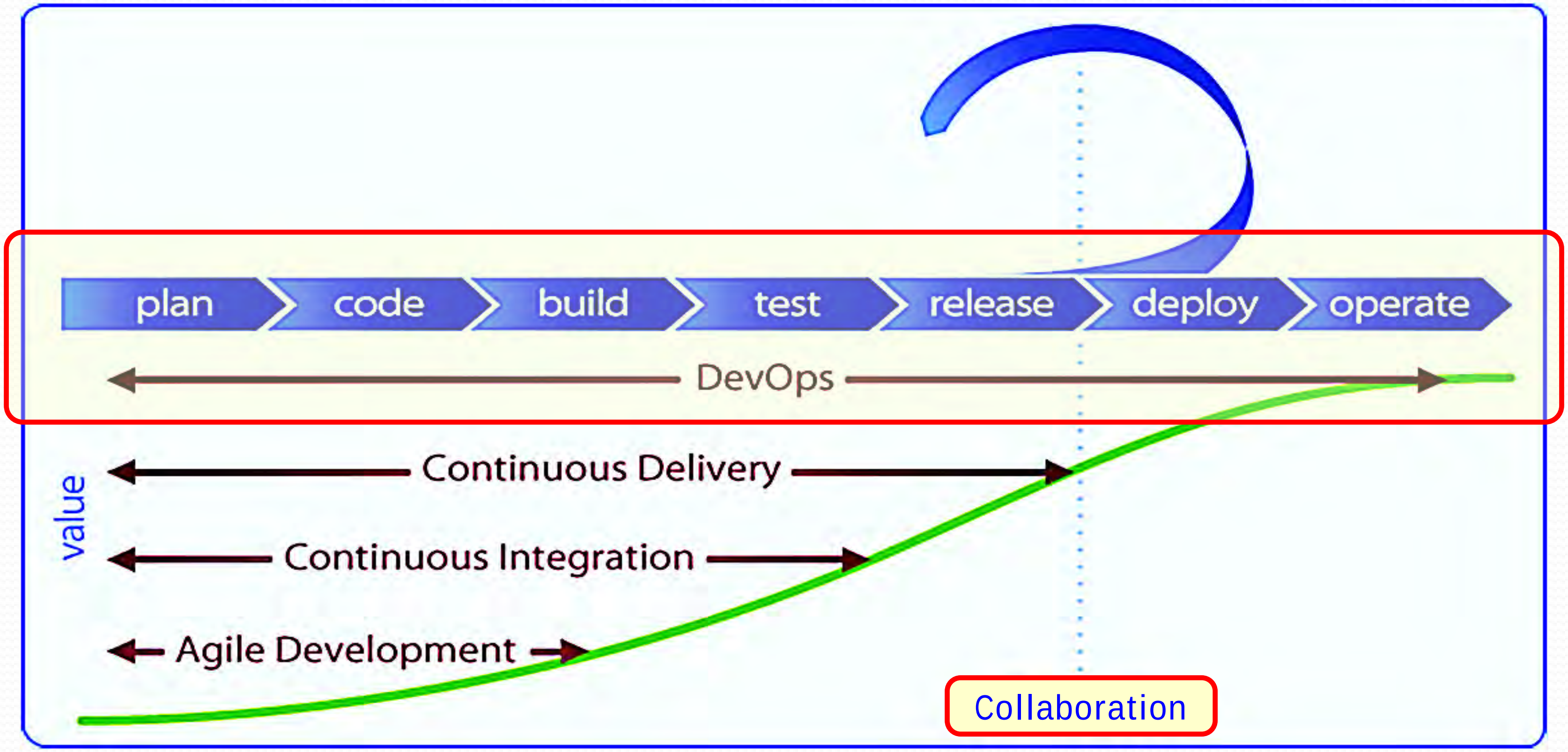
- Agile CD consists of seven broad practices
- Automated acceptance, load, performance, etc.
- ☞ Include packaging, pre-production test, C&A, etc.

Practice	Description
Packaging	Frequently generating system images for pre-production testing & checkout
Acceptance	Frequently performing automated system & user acceptance testing
Load Test	Frequently performing automated system load, stress, & capacity testing
Performance	Frequently performing automated system user & technical performance testing
Pre-Production	Frequently performing automated pre-production tests prior to final deployment
Certification	Frequently performing automated system certification & accreditation tests
Deployment	Frequently generating product images for pre-deployment testing & checkout

Mukherjee, J. (2015). *Continuous delivery pipeline*: *Where does it choke*. Charleston, SC: CreateSpace.
Swartout, P. (2014). *Continuous delivery and devops*: *A quickstart guide*. Birmingham, UK: Packt Publishing.

STAGE 5—Development Operations

- □ Created by Patrick Debois of Jedi BVBA in 2007
- □ Collaboration of developers & infrastructure people
- ☞ □ Goal to automate the deployment to end-user devices



Bass, L., Weber, I., & Zhu, L. (2015). *Devops*: *A software architect's perspective*. Old Tappan, NJ: Pearson Education.
Gruver, G., & Mouser, T. (2015). *Leading the transformation*: *Applying agile and devops at scale*. Portland, OR: IT Revolution Press.
Humble, J., Molesky, J., & O'Reilly, B. (2015). *Lean enterprise*: *How high performance organizations innovate at scale*. Sebastopol, CA: O'Reilly Media.

STAGE 5—Development Operations

- □ Agile DevOps consists of seven broad practices
- □ Automated sys admin, CM, building, monitor, etc.
- ☞ □ Include virtualization, containerize, deployment, etc.

Practice	Description
Sys Admin	Frequently performing automated system administration tasks, i.e., scripting
Config. Mgt.	Frequently performing automated infrastructure config. mgt./version control
Host Builds	Frequently performing automated system and server host builds and config.
Virtualization	Frequently performing automated system, server, & net virtualization services
Containerize	Frequently performing automated software and Microservices containerization
Deployment	Frequently generating final end-user system & software images for distribution
Monitor & Supp	Frequently performing automated metrics collection & deployment monitoring

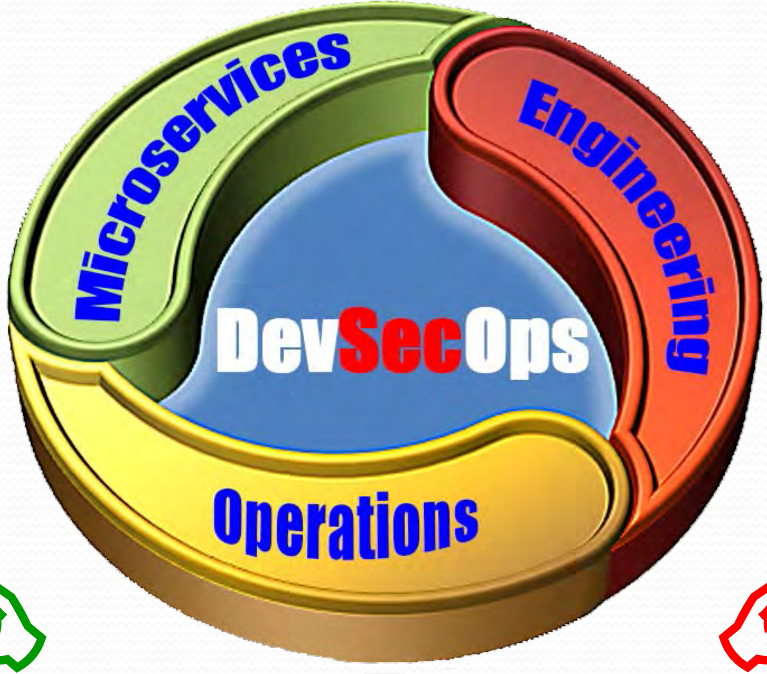
Duffy, M. (2015). *Devops automation cookbook*: *Over 120 recipes coverying key automation techniques*. Birmingham, UK: Packt Publishing.
Farcic, V. (2016). *The devops 2.0 toolkit*: *Automating the continuous deployment pipelines with containerized microservices*. Victoria, CA: LeanPub.

STAGE 6—Development Sec Operations

- □ DevSecOps coined by Shannon Lietz in 2014
- □ Rugged devops, secdevops, devopssec, devsecops
- ☞ □ Microservices, security engineering & operations keys

Secure Microservices

- Docker App
- Docker Bins
- Docker Files
- Docker Images
- Docker Scanning
- Docker Registry
- Docker Host
- Docker Hub
- Docker Monitoring



Secure Engineering

- Security Champions
- Security Planning
- Security Training
- Security Requirements
- Security Architecture
- Security Analysis
- Security Testing
- Security Review
- Security Response

Secure Operations

- Activity Logging
- Event Monitoring
- Configuration Mgt.
- Patch Management
- User Access Control
- Privilege Management
- Vulnerability Mgt.
- Response Mgt.
- Performance Mgt.

Bird, J. (2016). *Devopssec: Delivering secure software through continuous delivery*. Sebastopol, CA: O'Reilly Media.

Stage 6—Development Sec Operations

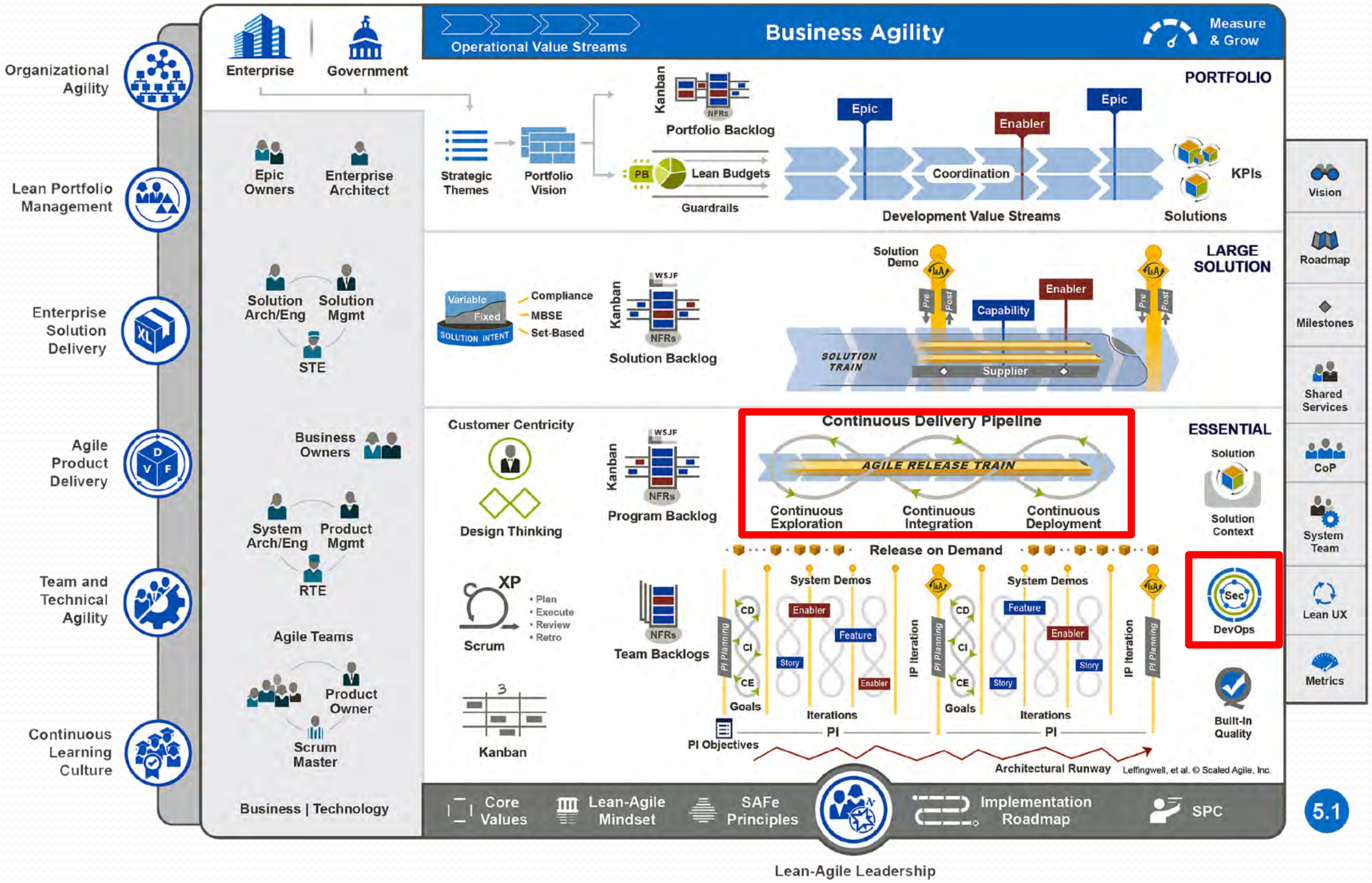
- □ DevSecOps consists of seven broad practices
- □ Automated secure build, analysis, & deployment
- ☞ □ Includes containerization, engineering & operations

Practice	Description
Engineering	Frequently performing “baked-in” lean and agile security engineering practices
Containers	Frequently performing automated microservices containerization practices
Evaluation	Frequently performing automated static and dynamic vulnerability analysis
Deployment	Frequently performing automated digitally signed security deployment practices
Protection	Frequently performing automated real-time self-security protection practices
Monitoring	Frequently performing automated real-time security monitoring practices
Responses	Frequently performing automated trigger-based rollback response practices

Bird, J. (2016). *Devopssec*: *Delivering secure software through continuous delivery*. Sebastopol, CA: O'Reilly Media.

STAGE 7—Enterprise DevSecOps

- □ SE framework by Dean Leffingwell of Rally in 2007
- □ Newest version leaner, meaner, lighter, and simpler
- ☞ □ Experimental bottoms-up DevOps-based innovation



Leffingwell, D. (2007). *Scaling software agility*: *Best practices for large enterprises*. Boston, MA: Pearson Education.

STAGE 7—Enterprise DevSecOps

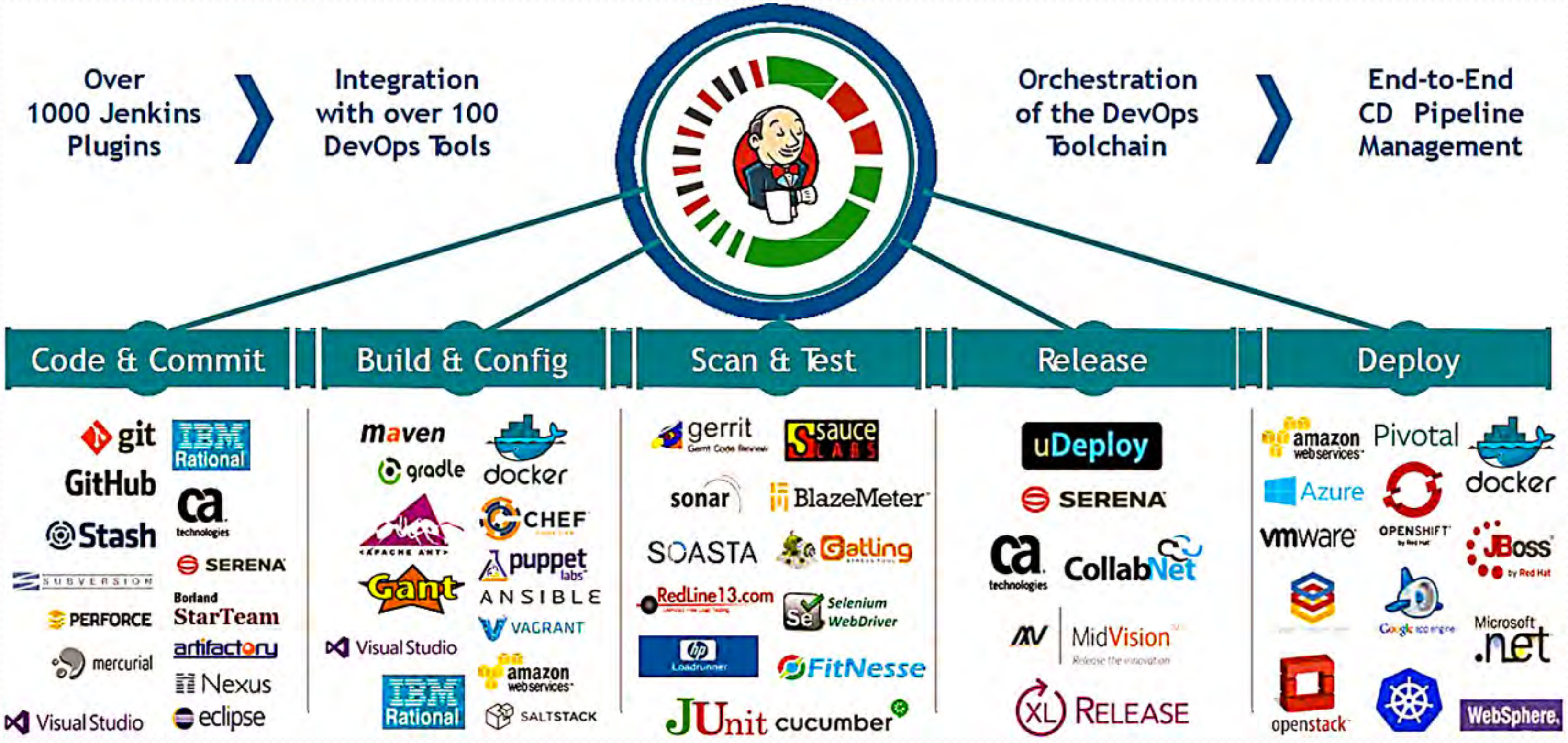
- □ Ent. DevSecOps consists of seven broad practices
- □ Automated experiments, measures, feedback, etc.
- ☞ □ Includes Lean UX, experiments, DevSecOps, etc.

Practice	Description
Themes	Capturing strategic goals and objectives as objectives and key results
Epics	Synthesizing epic hypothesis statements to quickly realize strategic themes
Lean UX	Using low-cost, lightweight user experience techniques to quickly scope needs
Experiments	Quickly developing/deploying lightweight business experiments to production
DevSecOps	Applying DevSecOps principles, practices, and tools for business experiments
Feedback	Quickly gather measurable feedback from markets, customer, and end users
Pivot/Persevere	Be prepared to pivot to a new business experiments when new data emerges

Leffingwell, D. (2018). *SAFe reference guide: Scaled agile framework for lean enterprises*. Boston, MA: Pearson.
Knaster, R. (2018). *SAFe distilled: Applying the scaled agile framework for lean enterprises*. Boston, MA: Pearson.

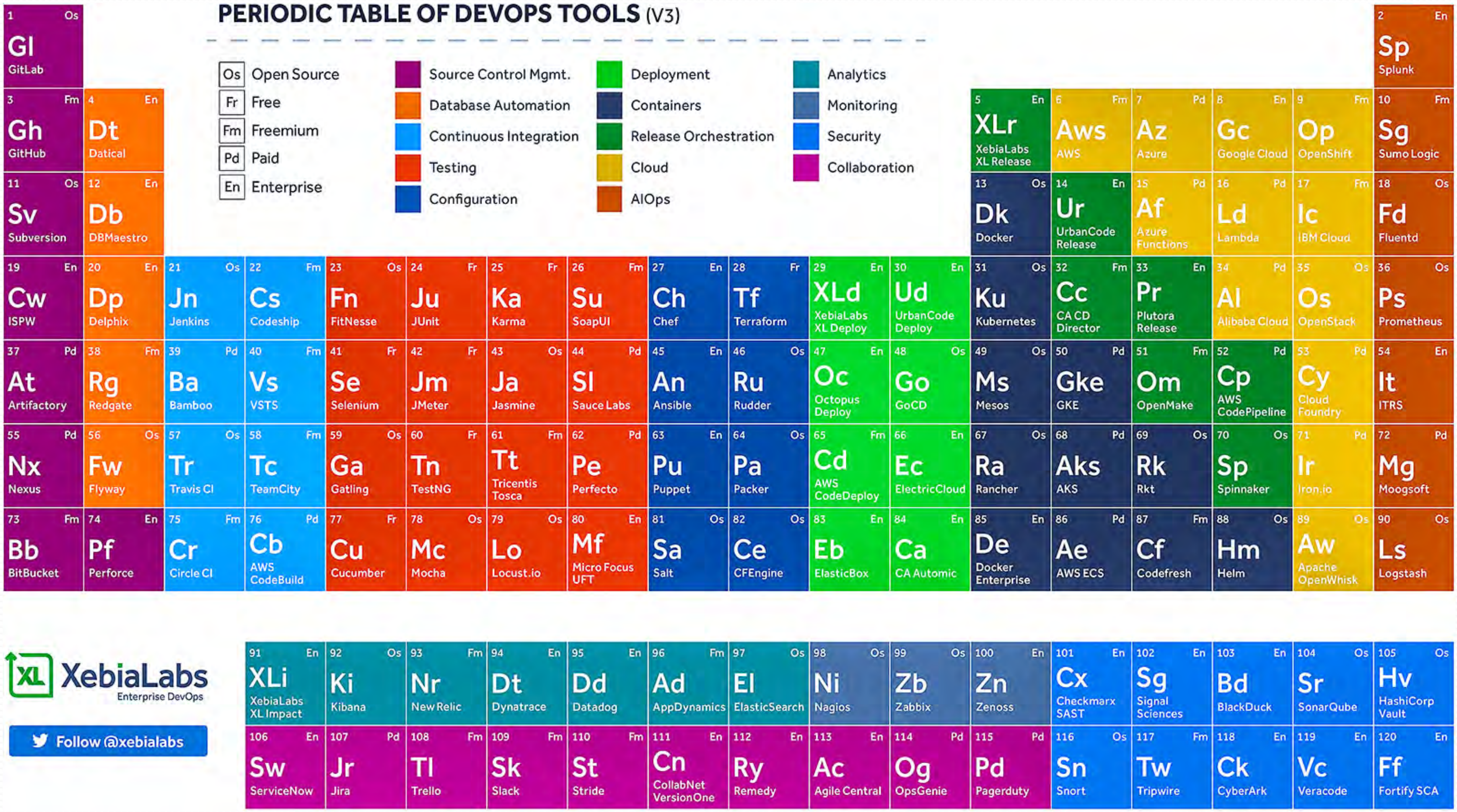
DevSecOps—Basic DevOps Tools

- □ Numerous tools to automate DevOps pipeline
- □ People can piece together toolset along with hubs
- ☞ □ Tools include version control, testing, & deployment



Juengst, D. (2015). *Deliver better software faster*: *With the cloudbees jenkins platform*. San Francisco, CA: CloudBees.
Weeks, D. E. (2014). *Devops and continuous delivery reference architectures* (*volume 1 & 2*). Fulton, MD: Sonatype.

DevSecOps—Periodic Table



PERIODIC TABLE OF DEVOPS TOOLS (V3)

Os	Open Source
Fr	Free
Fm	Freemium
Pd	Paid
En	Enterprise

Source Control Mgmt.
Database Automation
Continuous Integration
Testing
Configuration
Deployment
Containers
Release Orchestration
Cloud
AIOps
Analytics
Monitoring
Security
Collaboration

1	Os	Gl	GitLab
2	En	Sp	Splunk
3	Fm	Gh	GitHub
4	En	Dt	Datical
5	En	XLr	XebiaLabs XL Release
6	Fm	Aws	AWS
7	Pd	Az	Azure
8	En	Gc	Google Cloud
9	Fm	Op	OpenShift
10	Fm	Sg	Sumo Logic
11	Os	Sv	Subversion
12	En	Db	DBMaestro
13	Os	Dk	Docker
14	En	Ur	UrbanCode Release
15	Pd	Af	Azure Functions
16	Pd	Ld	Lambda
17	Fm	Ic	IBM Cloud
18	Os	Fd	Fluentd
19	En	Cw	ISPW
20	En	Dp	Delphix
21	Os	Jn	Jenkins
22	Fm	Cs	Codeship
23	Os	Fn	FitNesse
24	Fr	Ju	JUnit
25	Fr	Ka	Karma
26	Fm	Su	SoapUI
27	En	Ch	Chef
28	Fr	Tf	Terraform
29	En	XLd	XebiaLabs XL Deploy
30	En	Ud	UrbanCode Deploy
31	Os	Ku	Kubernetes
32	Fm	Cc	CA CD Director
33	En	Pr	Plutora Release
34	Pd	Al	Alibaba Cloud
35	Os	Os	OpenStack
36	Os	Ps	Prometheus
37	Pd	At	Artifactory
38	Fm	Rg	Redgate
39	Pd	Ba	Bamboo
40	Fm	Vs	VSTS
41	Fr	Se	Selenium
42	Fr	Jm	JMeter
43	Os	Ja	Jasmine
44	Pd	Sl	Sauce Labs
45	En	An	Ansible
46	Os	Ru	Rudder
47	En	Oc	Octopus Deploy
48	Os	Go	GoCD
49	Os	Ms	Mesos
50	Pd	Gke	GKE
51	Fm	Om	OpenMake
52	Pd	Cp	AWS CodePipeline
53	Pd	Cy	Cloud Foundry
54	En	It	ITRS
55	Pd	Nx	Nexus
56	Os	Fw	Flyway
57	Os	Tr	Travis CI
58	Fm	Tc	TeamCity
59	Os	Ga	Gatling
60	Fr	Tn	TestNG
61	Fm	Tt	Tricentis Tosca
62	Pd	Pe	Perfecto
63	En	Pu	Puppet
64	Os	Pa	Packer
65	Fm	Cd	AWS CodeDeploy
66	En	Ec	ElectricCloud
67	Os	Ra	Rancher
68	Pd	Aks	AKS
69	Os	Rk	Rkt
70	Os	Sp	Spinnaker
71	Pd	Ir	Iron.io
72	Pd	Mg	Moogsoft
73	Fm	Bb	BitBucket
74	En	Pf	Perforce
75	Fm	Cr	Circle CI
76	Pd	Cb	AWS CodeBuild
77	Fr	Cu	Cucumber
78	Os	Mc	Mocha
79	Os	Lo	Locust.io
80	En	Mf	Micro Focus UFT
81	Os	Sa	Salt
82	Os	Ce	CFEngine
83	En	Eb	ElasticBox
84	En	Ca	CA Automic
85	En	De	Docker Enterprise
86	Pd	Ae	AWS ECS
87	Fm	Cf	Codefresh
88	Os	Hm	Helm
89	Os	Aw	Apache OpenWhisk
90	Os	Ls	Logstash
91	En	XLi	XebiaLabs XL Impact
92	Os	Ki	Kibana
93	Fm	Nr	New Relic
94	En	Dt	Dynatrace
95	En	Dd	Datadog
96	Fm	Ad	AppDynamics
97	Os	El	ElasticSearch
98	Os	Ni	Nagios
99	Os	Zb	Zabbix
100	En	Zn	Zenoss
101	En	Cx	Checkmarx SAST
102	En	Sg	Signal Sciences
103	En	Bd	BlackDuck
104	Os	Sr	SonarQube
105	Os	Hv	HashiCorp Vault
106	En	Sw	ServiceNow
107	Pd	Jr	Jira
108	Fm	Tl	Trello
109	Fm	Sk	Slack
110	Fm	St	Stride
111	En	Cn	CollabNet VersionOne
112	En	Ry	Remedy
113	En	Ac	Agile Central
114	Pd	Og	OpsGenie
115	Pd	Pd	Pagerduty
116	Os	Sn	Snort
117	Fm	Tw	Tripwire
118	En	Ck	CyberArk
119	En	Vc	Veracode
120	En	Ff	Fortify SCA

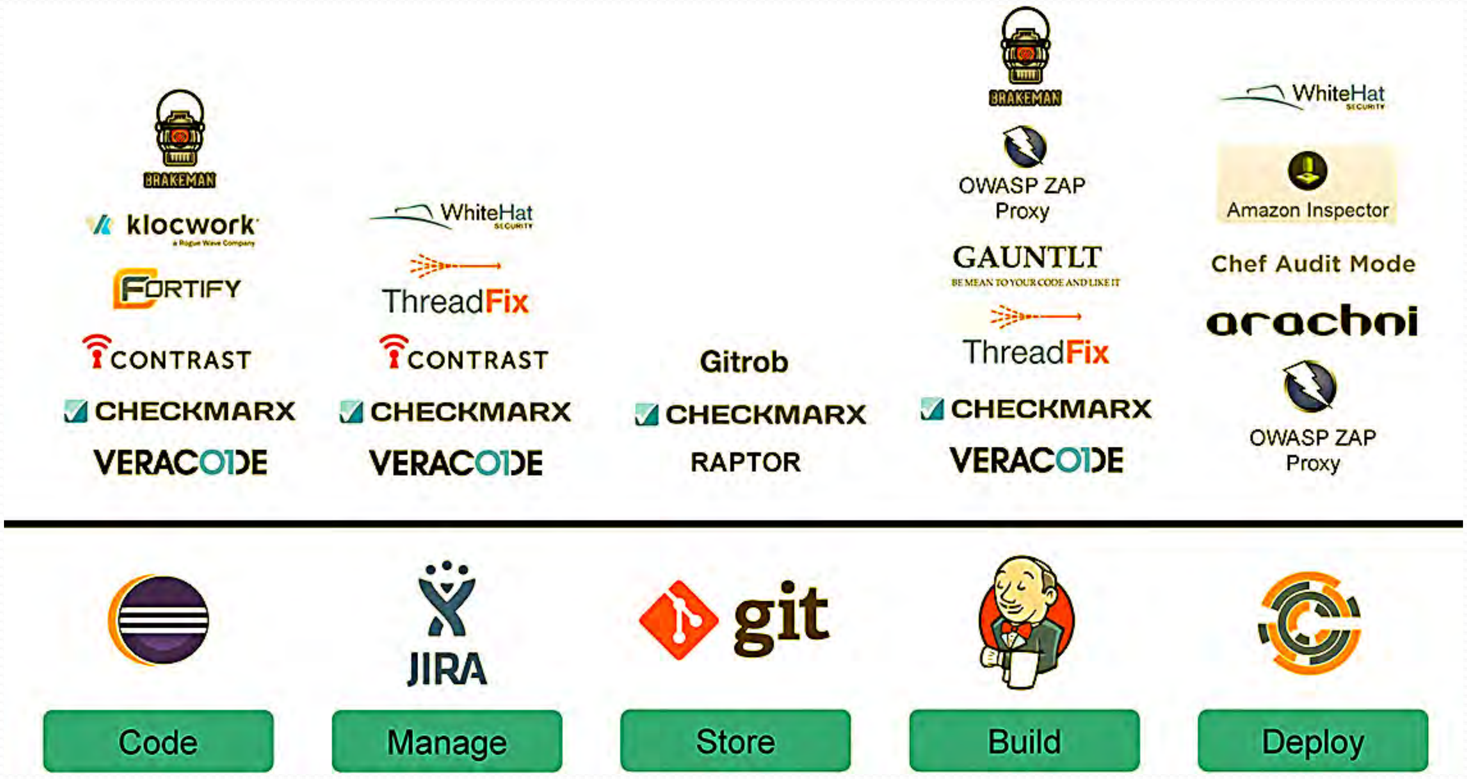
XebiaLabs Enterprise DevOps

Follow @xebialabs

XebiaLabs. (2018). *Periodic table of devops tools*. Retrieved April 11, 2016, from https://xebialabs.com/periodic-table-of-devops-tools.
Weeks, D. E. (2017). *Devops and continuous delivery reference architectures* (*volume 1 & 2*). Fulton, MD: Sonatype.

DevSecOps—Basic Security Tools

- □ Many tools emerging for DevOps application security
- □ Begins-ends with microservices—tiny attack surface
- ☞ □ Includes containers, testing, & real-time monitoring



Tesauro, M. (2016). *Taking appsec to 11*: *Appsec pipelines, devops, and making things better*. Denver, CO: SnowFROC 2016.
Weeks, D. E. (2014). *Devops and continuous delivery reference architectures* (*volume 1 & 2*). Fulton, MD: Sonatype.

Pre-Commit

Security activities before code is checked in to version control

Threat Modeling/Attack Mapping:
- Attacker personas
- Evil user stories
- Raindance
- Mozilla Rapid Risk Assessment
- OWASP ThreatDragon

Security and Privacy Stories:
- OWASP ASVS
- SAFECode Security Stories

IDE Security Plugins:
- DevSkim
- FindSecurityBugs
- Puma Scan
- SonarLint

Pre-Commit Security Hooks:
- git-hound
- git-secrets
- Repo-supervisor
- ThoughtWorks Talisman

Secure Coding Standards:
- CERT Secure Coding Standards
- OWASP Proactive Controls

Manual and Peer Reviews:
- Gerrit
- GitHub pull request
- GitLab merge request
- Review Board

Commit (Continuous Integration)

Fast, automated security checks during the build and Continuous Integration steps

Static Code Analysis (SCA):
- FindSecurityBugs
- Brakeman
- ESLint
- Phan

Security Unit Tests:
- JUnit
- Mocha
- xUnit

Infrastructure as Code Analysis:
- ansible-lint
- Foodcritic
- puppet-lint
- cfn_nag

Dependency Management:
- OWASP Dependency Check
- Bundler-Audit
- Gemnasium
- PHP Security Checker
- Retire.JS
- Node Security Platform

Container Security:
- Actuary
- Anchore
- Clair
- Dagda
- Docker Bench
- Falco

Container Hardening:
- Bane
- CIS Benchmarks
- grsecurity

Acceptance (Continuous Delivery)

Automated security acceptance, functional testing, and deep out-of-band scanning during Continuous Delivery

Infrastructure as Code:
- Ansible
- Chef
- Puppet
- SaltStack
- Terraform
- Vagrant

Immutable Infrastructure:
- Docker
- rkt

Security Scanning:
- Arachni
- nmap
- sqlmap
- sslyze
- ZAP
- ssh_scan

Cloud Configuration Management:
- AWS CloudFormation
- Azure Resource Manager
- Google Cloud Deployment Manager

Security Acceptance Testing:
- BDD-Security
- Gauntlt
- Mittn

Infrastructure Tests:
- Serverspec
- Test Kitchen

Infrastructure Compliance Checks:
- HubbleStack
- InSpec

Production (Continuous Deployment)

Security checks before, during, and after code is deployed to production

Security Smoke Tests:
- ZAP Baseline Scan
- nmap
- ssllabs-scan

Configuration Safety Checks:
- AWS Config
- AWS Trusted Advisor
- Microsoft Azure Advisor
- Security Monkey
- OSQuery

Secrets Management:
- Ansible Vault
- Blackbox
- Chef Vault
- Docker Secrets
- Hashicorp Vault
- Pinterest Knox

Cloud Secrets Management:
- AWS KMS
- Azure Key Vault
- Google Cloud KMS

Cloud Security Testing:
- CloudSploit
- Nimbostratus

Server Hardening:
- dev-sec.io
- SIMP

Host Intrusion Detection System (HIDS):
- fail2ban
- OSSEC
- Samhain

Operations

Continuous security monitoring, testing, audit, and compliance checks

Fault Injection:
- Chaos Kong
- Chaos Monkey

Cyber Simulations:
- Game day exercises
- Tabletop scenarios

Penetration Testing:
- Attack-driven defense
- Bug Bounties
- Red team exercises

Threat Intelligence:
- Diamond Model
- Kill Chain
- STIX
- TAXII

Continuous Scanning:
- OpenSCAP
- OpenVAS
- Prowler
- Scout2
- vuls

Blameless Postmortems:
- Etsy Morgue

Continuous Monitoring:
- grafana
- graphite
- statsd
- seyren
- sof-elk
- ElastAlert
- 411

Cloud Monitoring:
- CloudWatch
- CloudTrail
- Reddalert

Cloud Compliance:
- Cloud Custodian
- Compliance Monkey
- Forseti Security

Secure DevOps Toolchain

Building a DevSecOps Program (CALMS)

Culture

Break down barriers between Development, Security, and Operations through education and outreach

Automation

Embed self-service automated security scanning and testing in continuous delivery

Lean

Value stream analysis on security and compliance processes to optimize flow

Measurement

Use metrics to shape design and drive decisions

Sharing

Share threats, risks, and vulnerabilities by adding them to engineering backlogs

Start Your DevOps Metrics Program

- Number of high-severity vulnerabilities and how long they are open
- Build and deployment cycle time
- Automated test frequency and coverage
- Scanning frequency and coverage
- Number of attacks (and attackers) hitting your application

First Steps in Automation

- Build a security smoke test (e.g., ZAP Baseline Scan)
- Conduct negative unit testing to get off of the happy path
- Attack your system before somebody else does (e.g., Gauntlt)
- Add hardening steps into configuration recipes (e.g., dev-sec.io)
- Harden and test your CI/CD pipelines and do not rely on developer-friendly defaults

Learn to build, deliver, and deploy modern applications using secure DevOps and cloud principles, practices, and tools.

DEV540: Secure DevOps and Cloud Application Security

www.sans.org/DEV540

SANS AppSec
APPLICATION & SOFTWARE SECURITY

SANS APPSEC CURRICULUM

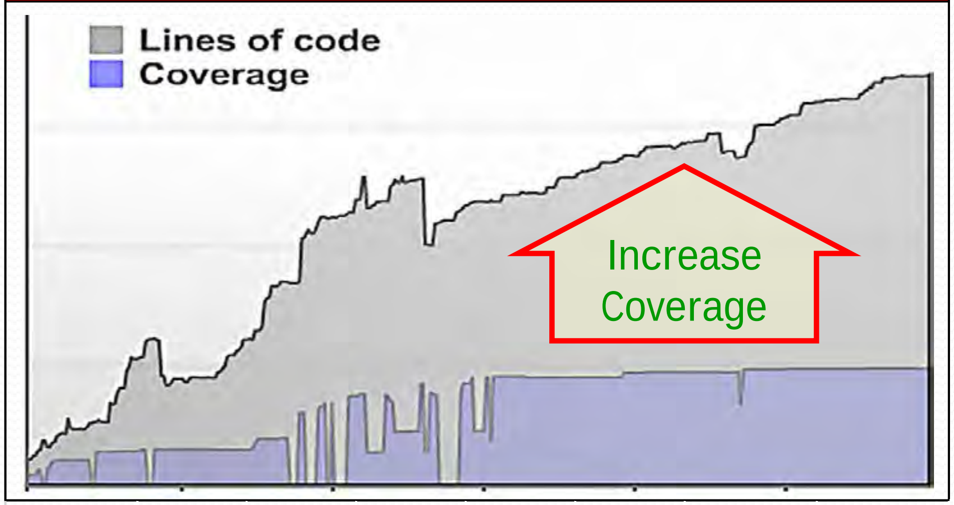
PLATFORM SECURITY	CORE	SPECIALIZATION
DEV531 Defending Mobile Applications Security Essentials	STH.DEVELOPER Application Security Awareness Modules	SEC542 Web App Penetration Testing and Ethical Hacking GWAPT
DEV541 Secure Coding in Java/JEE GSSP-JAVA	DEV522 Defending Web Applications Security Essentials GWEB	SEC642 Advanced Web App Penetration Testing, Ethical Hacking, and Exploitation Techniques
DEV544 Secure Coding in .NET GSSP-NET	DEV534 Secure DevOps: A Practical Introduction	ASSESSMENT
	DEV540 Secure DevOps and Cloud Application Security	AppSec CyberTalent Assessment sans.org/appsec-assessment

Poster contributors:
- Ben Allen
- Jim Bird
- David Deatherage
- Mark Geeslin
- Eric Johnson
- Frank Kim
- Jason Lam
- Gregory Leonard
- Dr. Johannes Ullrich

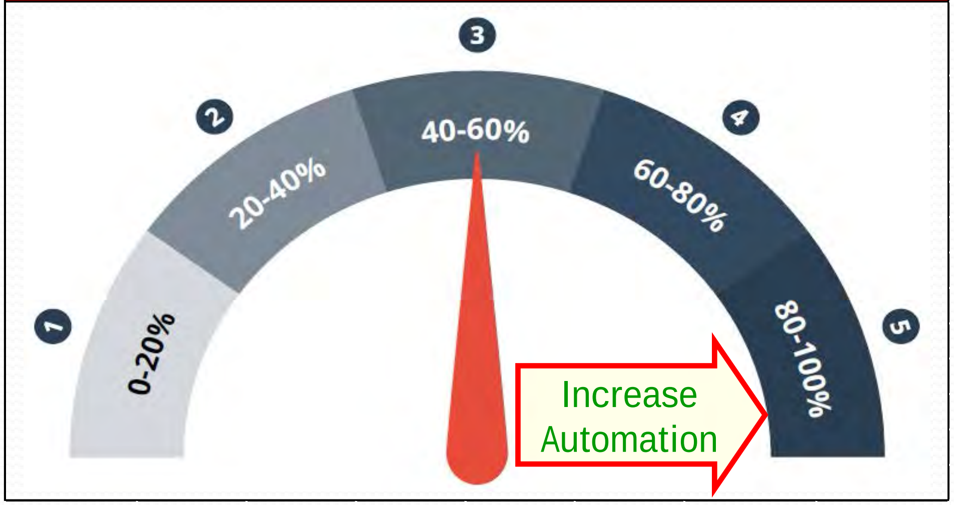
SANS

DevSecOps—Basic DevOps Metrics

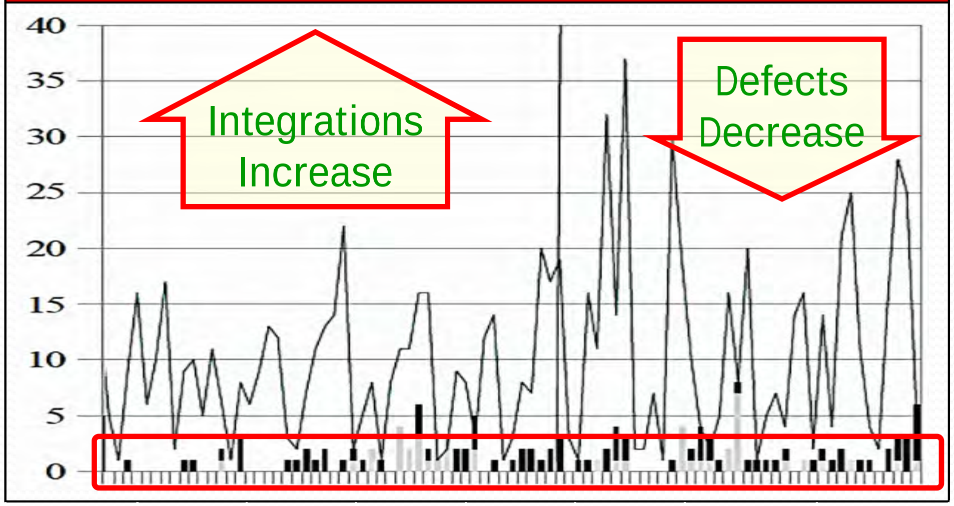
Test Coverage



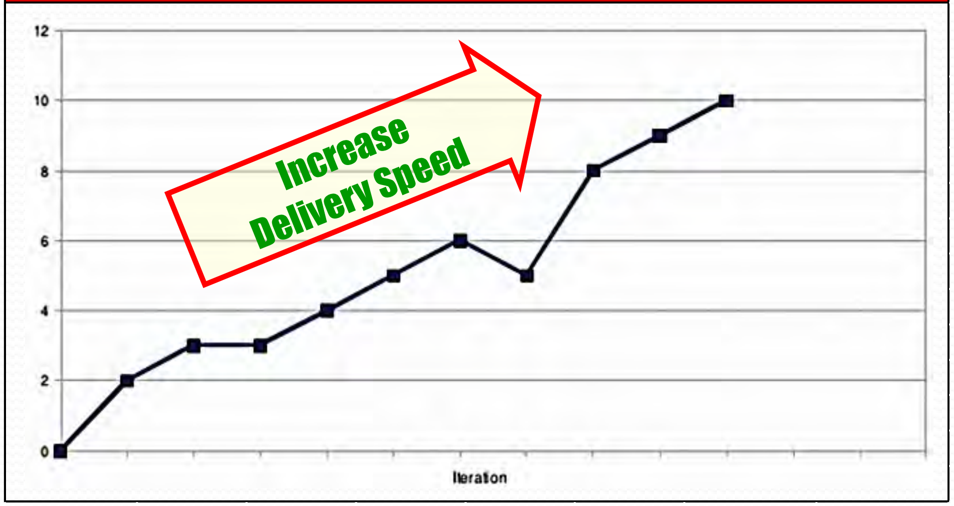
Test Automation



Integration Builds



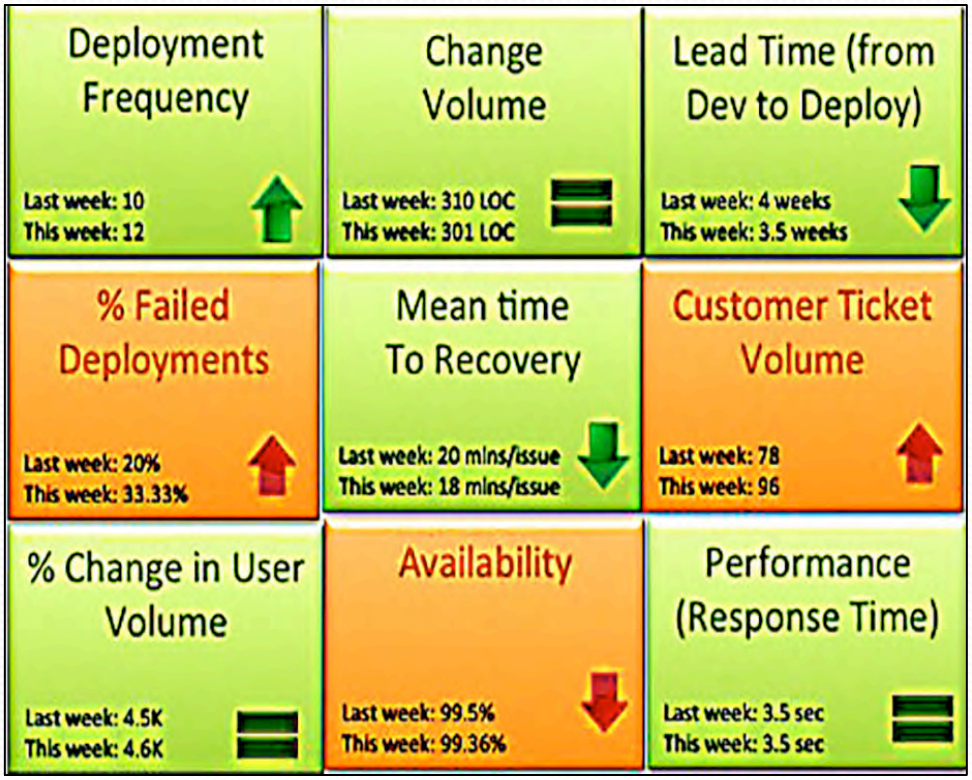
Running Tested Features



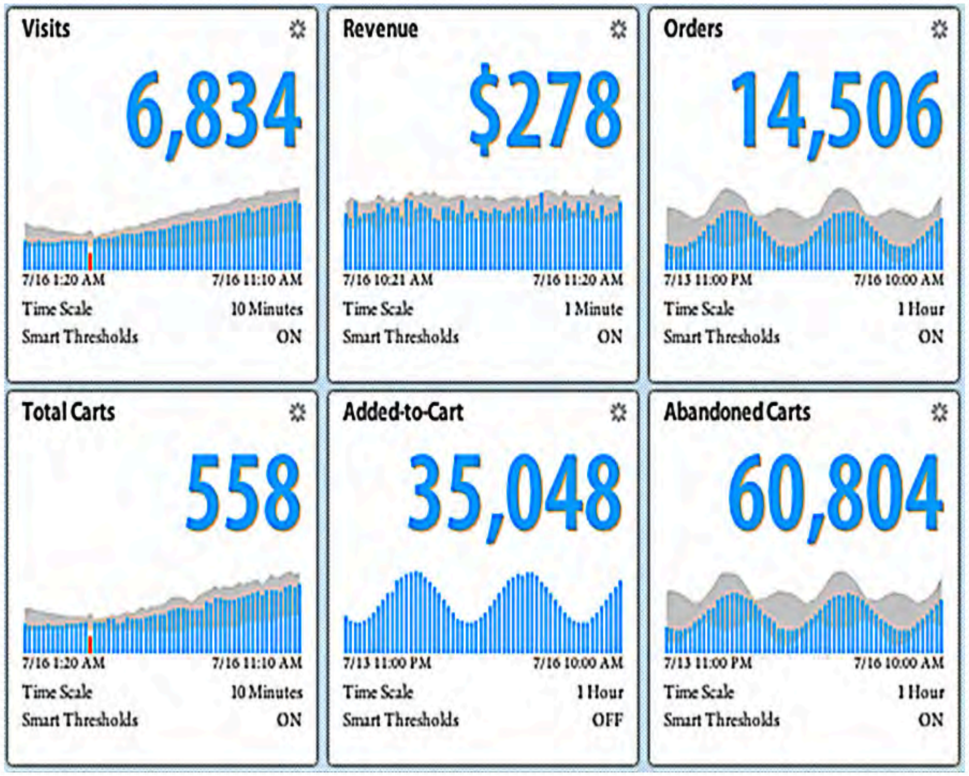
Duvall, P., Matyas, S., & Glover, A. (2006). *Continuous integration: Improving software quality and reducing risk*. Boston, MA: Addison-Wesley.
Jones, C. L., et al. (2020). *Continuous Iterative Development Measurement Framework*. Picatinny Arsenal, NJ: US Army ARDEC.

DevSecOps—Advanced Metrics

- □ DevOps metrics gaining in widespread popularity
- □ Hybrid of development & IT operations measures
- ☞ □ Includes code, deployment & e-business analytics



Deployment Frequency Last week: 10 This week: 12 ↑	Change Volume Last week: 310 LOC This week: 301 LOC =	Lead Time (from Dev to Deploy) Last week: 4 weeks This week: 3.5 weeks ↓
% Failed Deployments Last week: 20% This week: 33.33% ↑	Mean time To Recovery Last week: 20 mins/issue This week: 18 mins/issue ↓	Customer Ticket Volume Last week: 78 This week: 96 ↑
% Change in User Volume Last week: 4.5K This week: 4.6K =	Availability Last week: 99.5% This week: 99.36% ↓	Performance (Response Time) Last week: 3.5 sec This week: 3.5 sec =

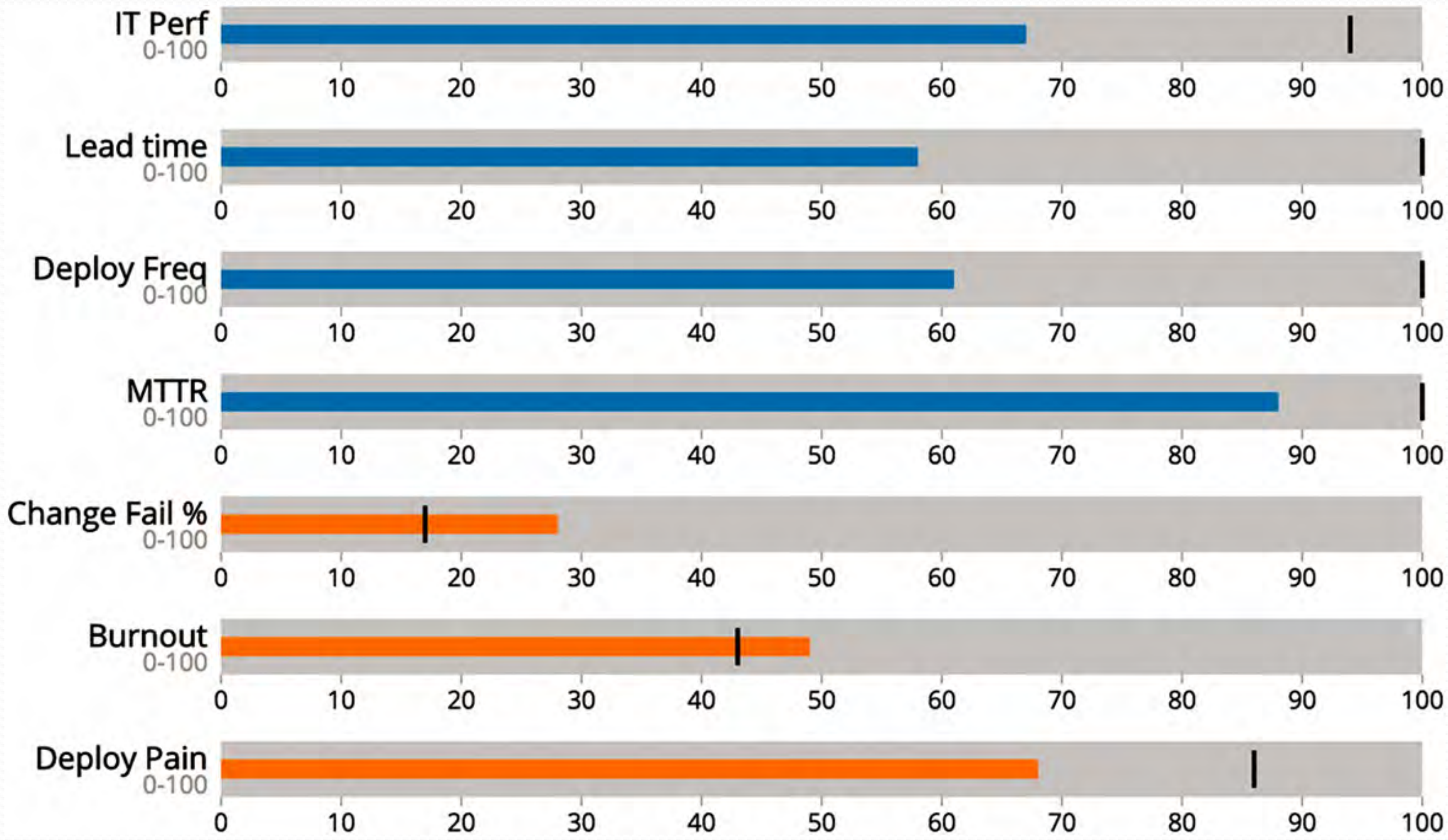


Panel	Value	Start	End	Time Scale	Smart Thresholds
Visits	6,834	7/16 1:20 AM	7/16 11:10 AM	10 Minutes	ON
Revenue	$278	7/16 10:21 AM	7/16 11:20 AM	1 Minute	ON
Orders	14,506	7/13 11:00 PM	7/16 10:00 AM	1 Hour	ON
Total Carts	558	7/16 1:20 AM	7/16 11:10 AM	10 Minutes	ON
Added-to-Cart	35,048	7/13 11:00 PM	7/16 10:00 AM	1 Hour	OFF
Abandoned Carts	60,804	7/13 11:00 PM	7/16 10:00 AM	1 Hour	ON

Velasquez, N. F. (2014). *State of devops report*. Portland, OR: Puppet Labs, Inc.
Jones, C. L., et al. (2020). *Continuous Iterative Development Measurement Framework*. Picatinny Arsenal, NJ: US Army ARDEC.

DevSecOps—Assessments

- □ Industry leading DevOps assessments are emerging
- □ DORA Technology DevOps Assessment is popular
- ☞□ Includes speed, deployments, reliability & morale

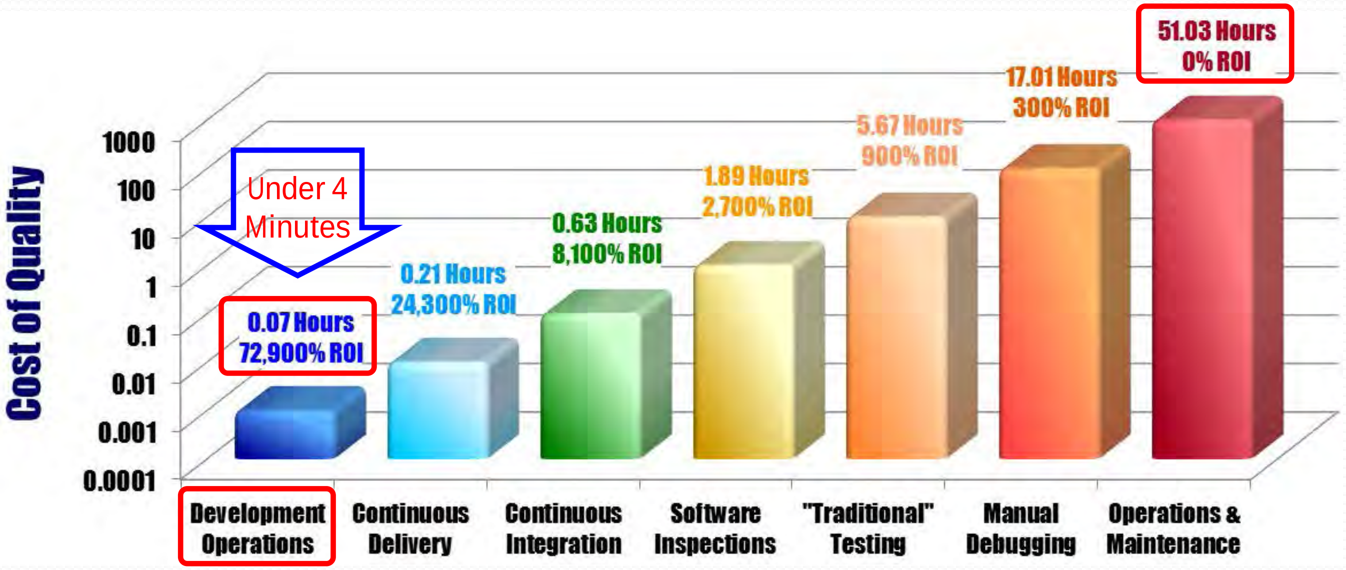


Kim, G., Forsgren, N., & Humble, J. (2017). *The DORA technology performance assessment*. Portland, OR: DevOps Research.

DevSecOps—Cost of Quality

- DevSecOps is orders-of-magnitude more efficient
- Based on millions of automated tests run in seconds
- One-touch auto-delivery to billions of global end-users

Activity	Def	CoQ	DevOps Economics	Hours	ROI
Development Operations	100	0.001	100 Defects x 70% Efficiency x 0.001 Hours	0.070	72,900%
Continuous Delivery	30	0.01	30 Defects x 70% Efficiency x 0.01 Hours	0.210	24,300%
Continuous Integration	9	0.1	9 Defects x 70% Efficiency x 0.1 Hours	0.630	8,100%
Software Inspections	3	1	2.7 Defects x 70% Efficiency x 1 Hours	1.890	2,700%
"Traditional" Testing	0.81	10	0.81 Defects x 70% Efficiency x 10 Hours	5.670	900%
Manual Debugging	0.243	100	0.243 Defects x 70% Efficiency x 100 Hours	17.010	300%
Operations & Maintenance	0.073	1,000	0.0729 Defects x 70% Efficiency x 1,000 Hours	51.030	n/a



4,500 x Faster than Code Inspections

Rico, D. F. (2016). *Devops cost of quality (CoQ)*: *Phase-based defect removal model*. Retrieved May 10, 2016, from http://davidfrico.com

DevSecOps—HP Case Study

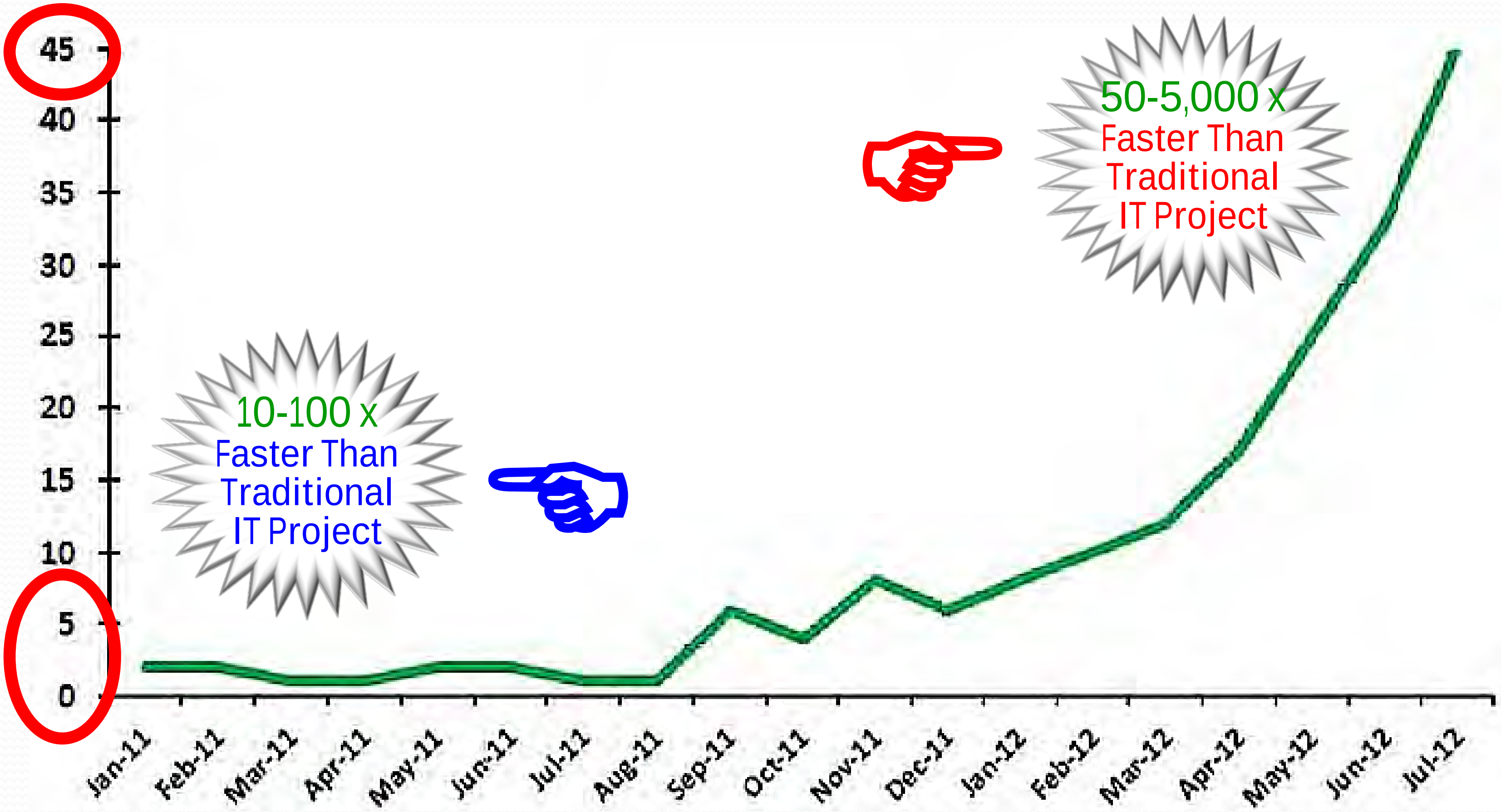
- □ Hewlett-Packard is a major user of CI, CD, & DevOps
- □ 400 engineers developed 10 million LOC in 4 years
- ☞ □ Major gains in testing, deployment, & innovation

Type	Metric	Manual	DevOps	Major Gains
Cycle Time Improvements	Build Time	40 Hours	3 Hours	13 x
	No. Builds	1-2 per Day	10-15 per Day	8 x
	Feedback	1 per Day	100 per Day	100 x
	Regression Testing	240 Hours	24 Hours	10 x
Development Cost Effort Distribution	Integration	10%	2%	5 x
	Planning	20%	5%	4 x
	Porting	25%	15%	2 x
	Support	25%	5%	5 x
	Testing	15%	5%	3 x
	Innovation	5%	40%	8 x

Gruver, G., Young, M. & Fulghum, P. (2013). *A practical approach to large-scale agile development*. Upper Saddle River, NJ: Pearson Education.

DevSecOps—Dot Com Case Studies

- □ Assembla went from 2 to 45 releases every month
- □ 15K Google developers run 150 million tests per day
- ☞ □ 30K+ Amazon developers deliver 136K releases a day



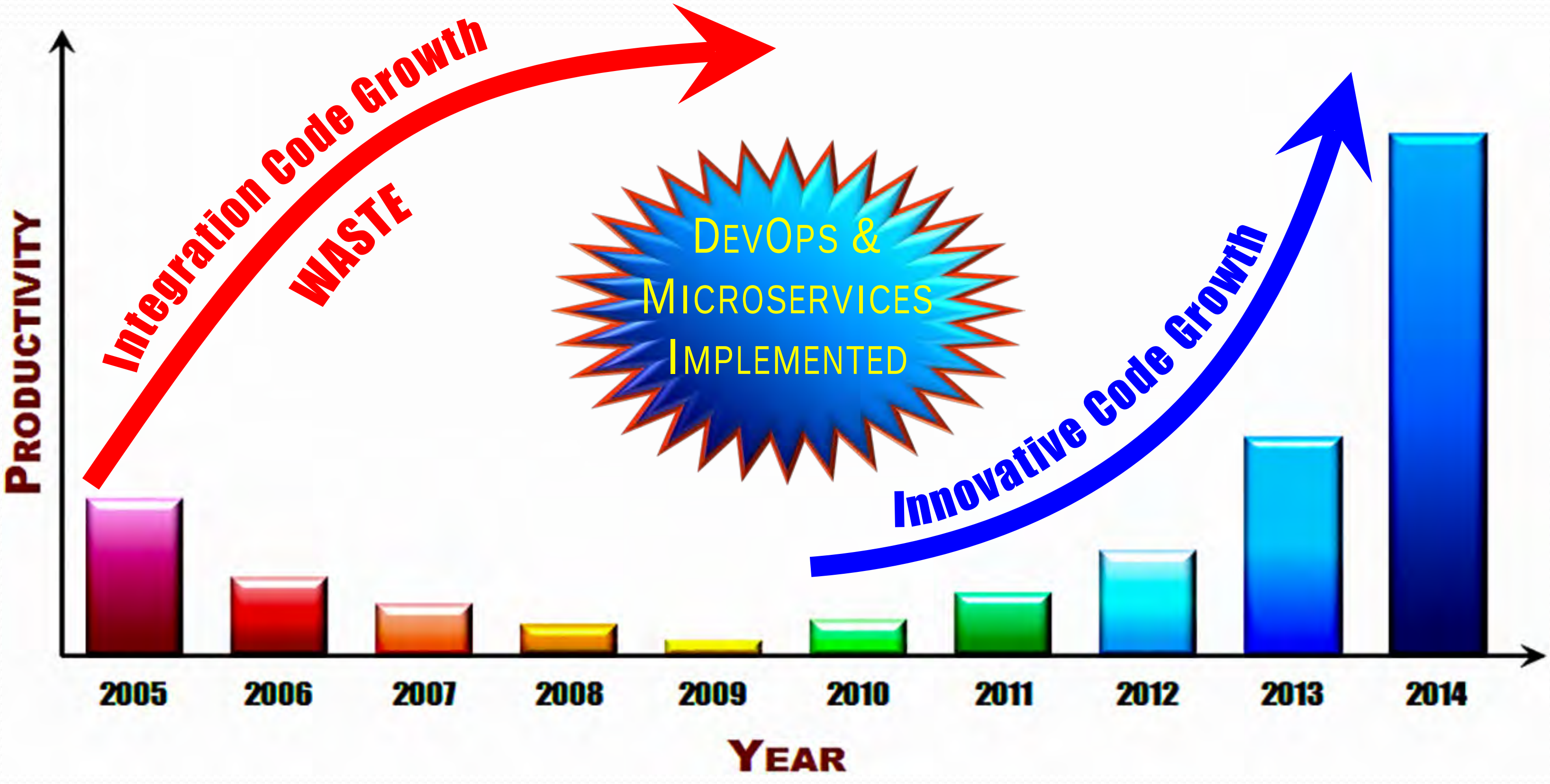
10-100 x Faster Than Traditional IT Project

50-5,000 x Faster Than Traditional IT Project

Singleton, A. (2014). *Unblock*: *A guide to the new continuous agile*. Needham, MA: Assembla, Inc.

DevSecOps—Blackboard Case Study

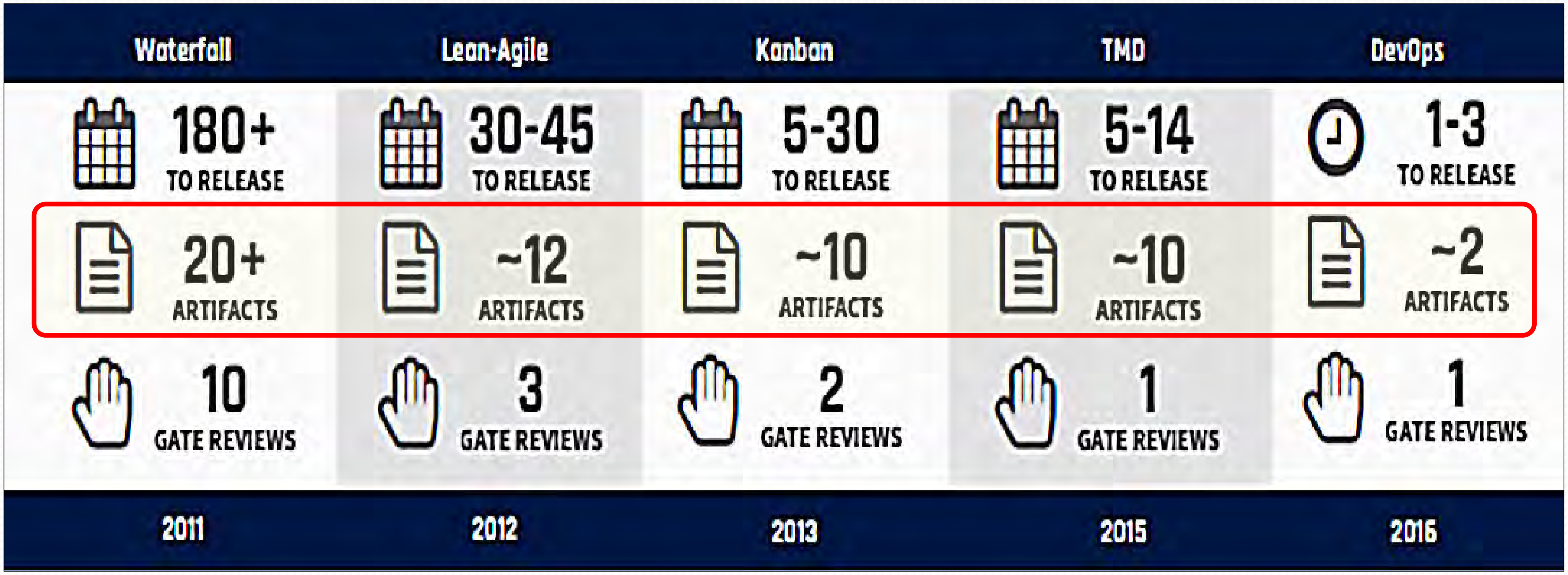
- □ Productivity **STOPS** due to excessive integration
- □ Implements DevOps & Microservices around 2010
- ☞ □ Waste elimination, productivity & innovation skyrocket



Ashman, D. (2014). Blackboard: Keep your head in the clouds. *Proceedings of the 2014 Enterprise DevOps Summit, San Francisco, California, USA.*

DevSecOps—U.S. DHS Case Study

- □ 1st gen replete with large portfolios & governance
- □ 2nd-3rd gen yield minor incremental improvements
- ☞ □ <u>4th-5th gen enables big order-of-magnitude impacts</u>



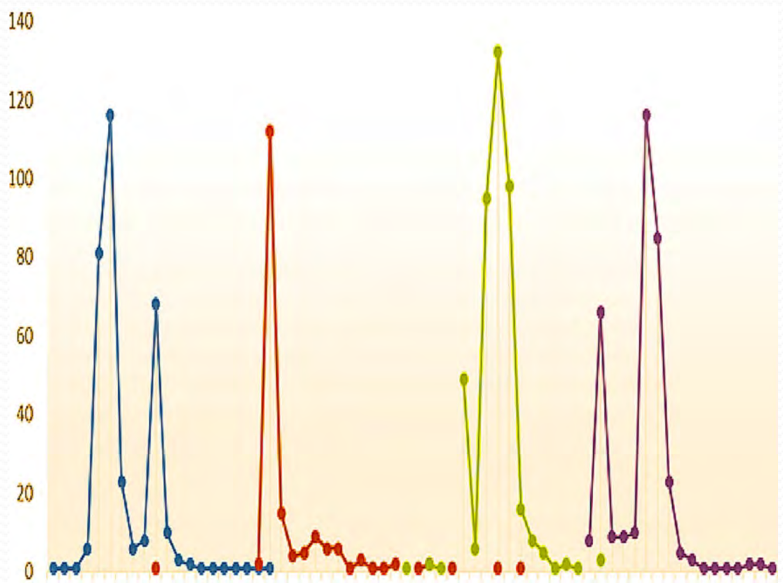
❶	❷	❸	❹	❺
Waterfall	Lean-Agile	Kanban	TMD	DevOps
180+ TO RELEASE	30-45 TO RELEASE	5-30 TO RELEASE	5-14 TO RELEASE	1-3 TO RELEASE
20+ ARTIFACTS	~12 ARTIFACTS	~10 ARTIFACTS	~10 ARTIFACTS	~2 ARTIFACTS
10 GATE REVIEWS	3 GATE REVIEWS	2 GATE REVIEWS	1 GATE REVIEWS	1 GATE REVIEWS
2011	2012	2013	2015	2016

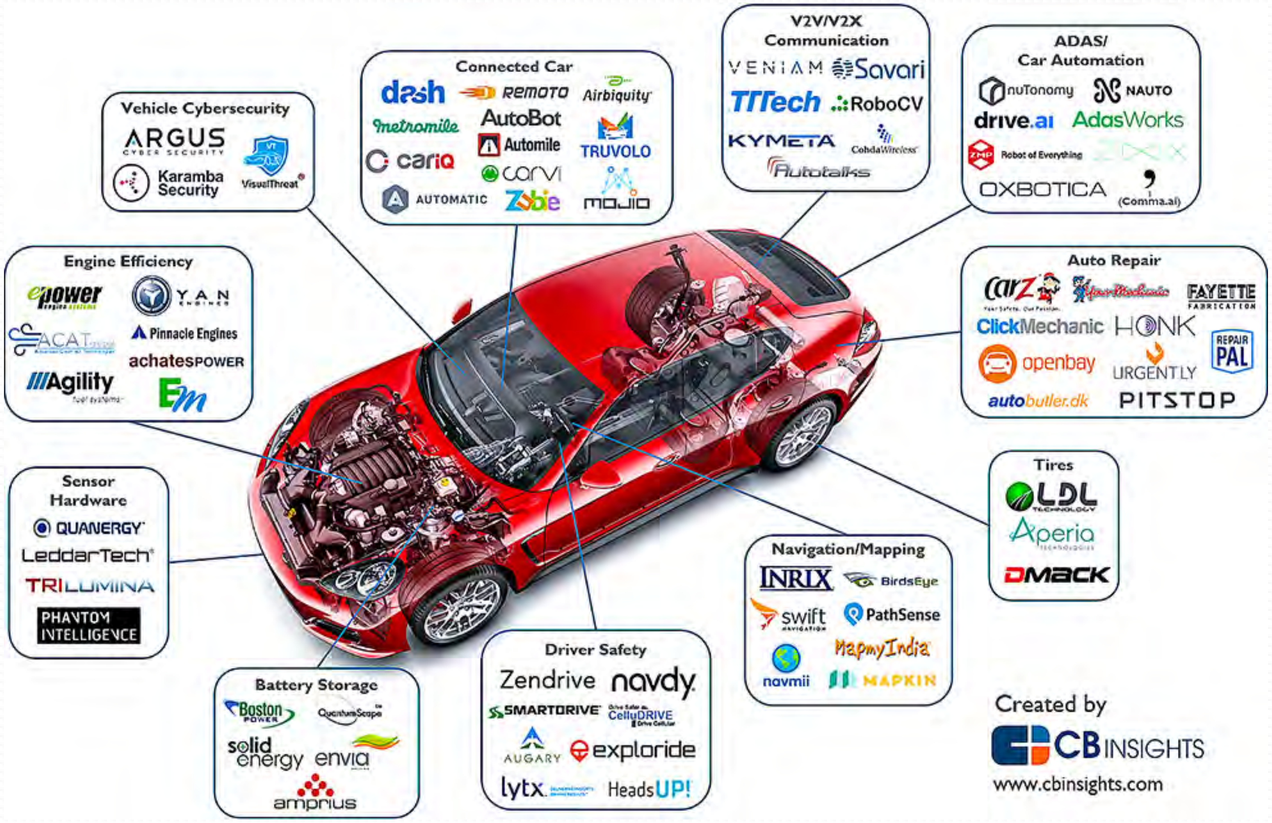
Manual Governance → Automated Governance

Denayer, L. (2017). *U.S. DHS citizenship and immigration services*: *USCIS agile development*. Washington, DC. iSDLC Seminar.

DevSecOps—Tesla Software Updates

- □ Tesla vehicle models are all electric automobiles
- □ Tesla autos have 100-200 million lines of code
- ☞□ Tesla performs up to 130 deployments per day





Choksi, N. (2016). *How software lifecycle integration and devops are transforming car development*. *Goto Conference, Copenhagen, Denmark*.
Vost, S., & Wagner, S. (2016). *Towards continuous integration and continuous delivery in the automotive industry*. Ithaca, NY: Cornell University.
Farley, D. (2021). *How Tesla's software disrupted the car industry*. Retrieved July 17, 2021, from http://youtu.be/ZMWAlPRhiwY

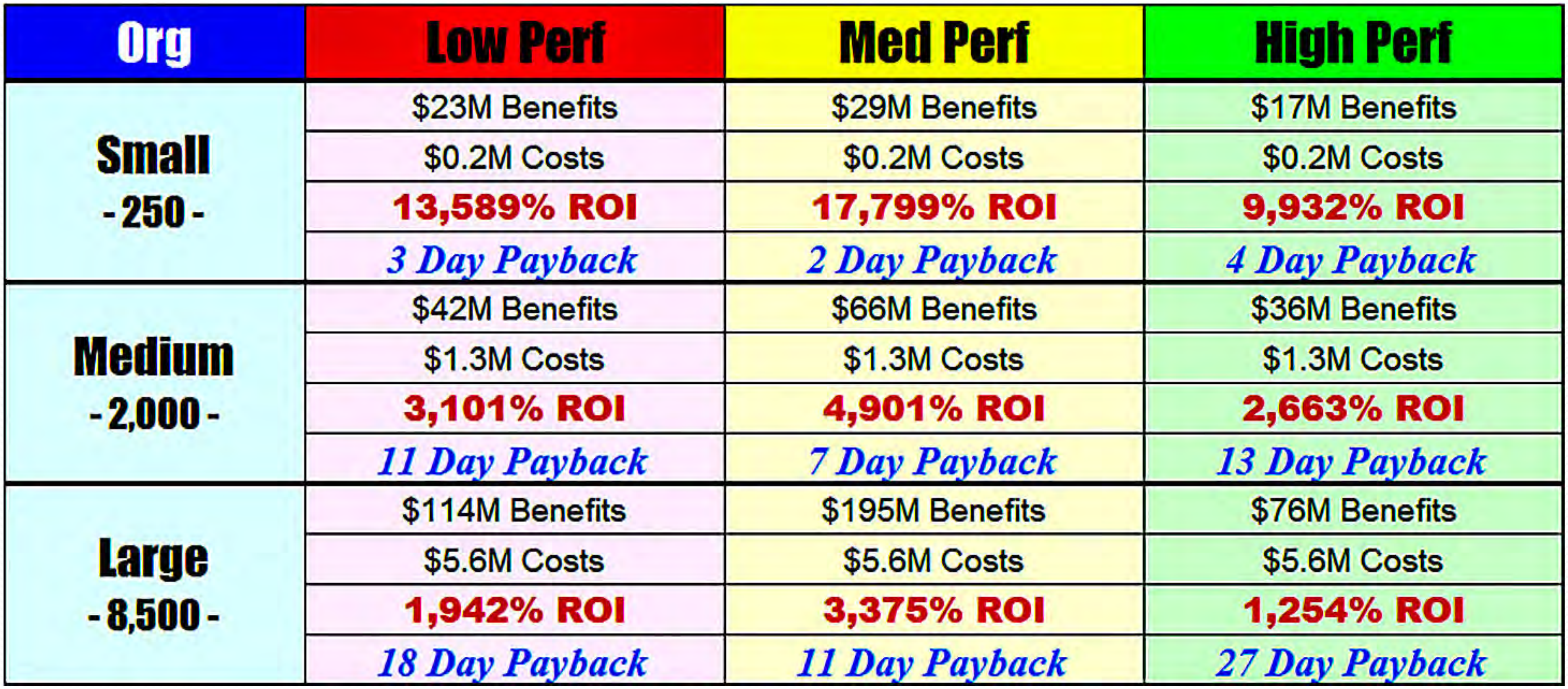
DevSecOps—Various Case Studies

WHO	RESULTS
Google	1 code repository **40,000 commits per day** 50,000 builds per day **150 million tests per day**
NETFLIX	24-day average server age 1 billion metrics per day **Self-service deploys** **Zero downtime**
amazon	Everything is monitored Code APIs for everything **136,000 deploys per day** **Very tiny two-pizza teams**
TARGET®	$1 billion annual IT budget **80 deployments per week** 17 billion API calls per month **Self-service DevOps Dojo training**
Bing	600 developers One code branch **20,000 tests per commit** **Every clean build deployed**

Rix, M. (2019). Conquering the monolith: Architecting for DevOps and release on demand. *SAFe Summit Europe, Hague, Netherlands*.

DevSecOps—Return on Investment

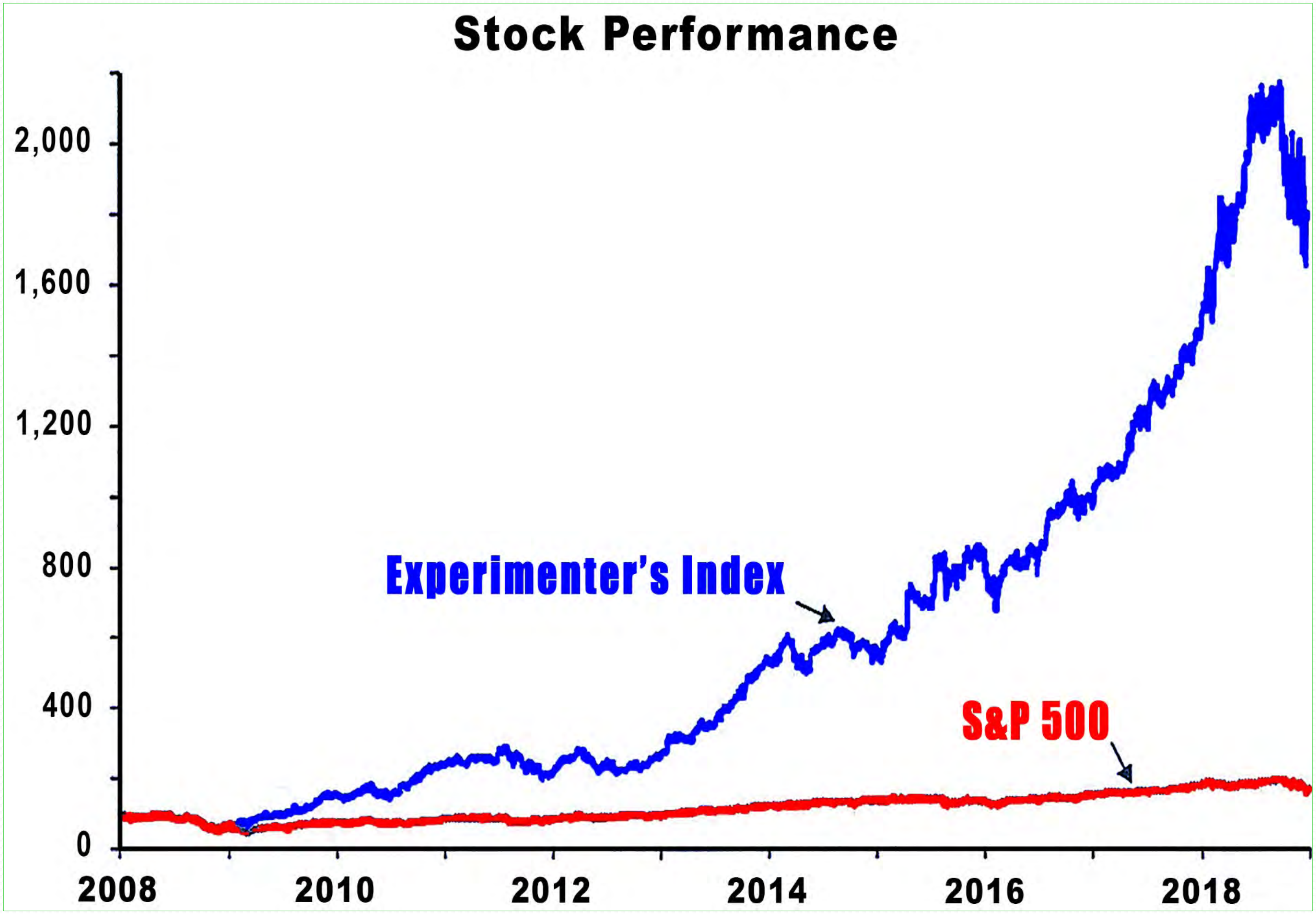
- □ Detailed DevOps economics starting to emerge
- □ ROI ranges from $17M to $195M *with minor costs*
- ☞ □ Benefits from cost savings, revenue, and availability



Org	Low Perf	Med Perf	High Perf
Small **- 250 -**	$23M Benefits	$29M Benefits	$17M Benefits
	$0.2M Costs	$0.2M Costs	$0.2M Costs
	13,589% ROI	**17,799% ROI**	**9,932% ROI**
	3 Day Payback	*2 Day Payback*	*4 Day Payback*
Medium **- 2,000 -**	$42M Benefits	$66M Benefits	$36M Benefits
	$1.3M Costs	$1.3M Costs	$1.3M Costs
	3,101% ROI	**4,901% ROI**	**2,663% ROI**
	11 Day Payback	*7 Day Payback*	*13 Day Payback*
Large **- 8,500 -**	$114M Benefits	$195M Benefits	$76M Benefits
	$5.6M Costs	$5.6M Costs	$5.6M Costs
	1,942% ROI	**3,375% ROI**	**1,254% ROI**
	18 Day Payback	*11 Day Payback*	*27 Day Payback*

Forsgren, N., Humble, J., & Kim, G. (2017). *Forecasting the value of devops transformations*: *Measuring roi of devops*. Portland, OR: DevOps Research.
Rico, D. F. (2017). *Devops return on investment (ROI) calculator*. Retrieved August 29, 2017, from http://davidfrico.com/devops-roi.xls

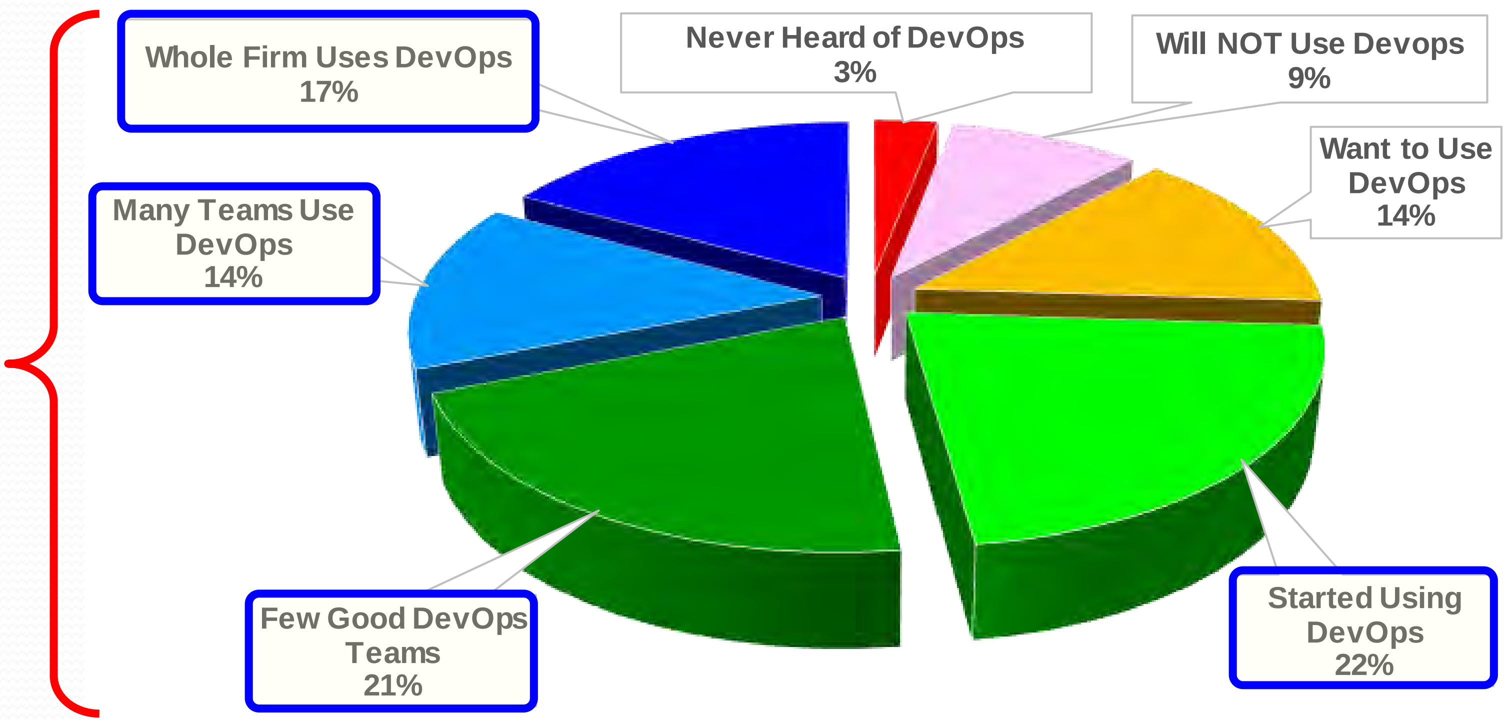
DevSecOps—Business Performance



Thomke, S. H. (2020). *Experimentation works*: *The surprising power of business experiments*. Boston, MA: Harvard Business Review Press.

DevSecOps—Adoption Statistics

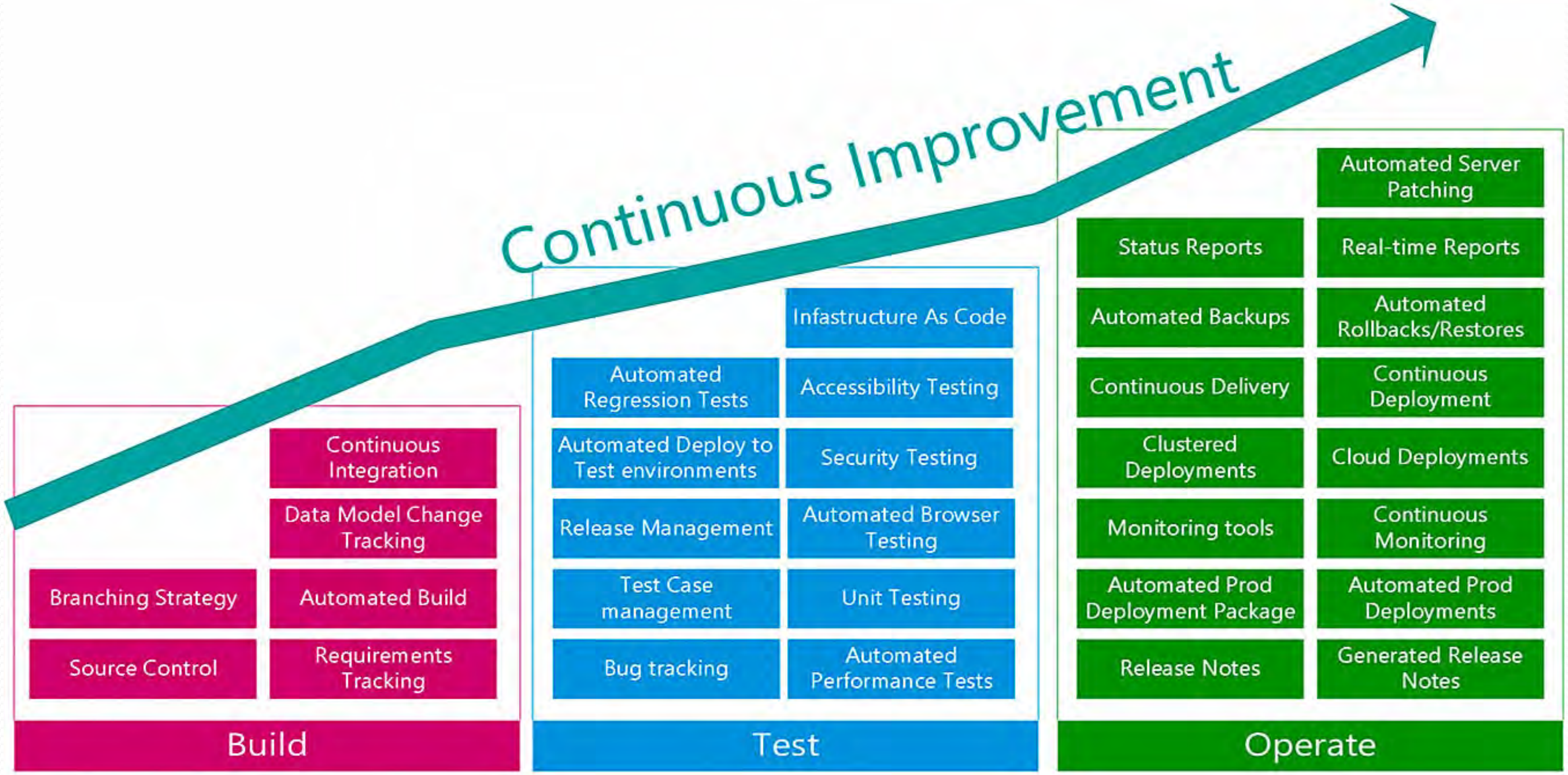
- □ DevOps adoption growing fast in-spite of slow start
- □ 74% using, 14% thinking about it, & 12% are in-dark
- ☞ □ DevOps a global industry-wide extinction-level event



Statistica. (2019). *Extent of devops adoption by software developers worldwide in 2017 and 2018*. Retrieved September 9, 2019, from https://www.statista.com/statistics/673505/worldwide-software-development-survey-devops-adoption

DevSecOps—Roadmap

- □ Having a DevOps rollout strategy is a key to success
- □ Phased, incremental, and situational implementation
- ☞ □ Includes build, testing, & IT operations, & practices



Continuous Improvement

Build

- Continuous Integration
- Data Model Change Tracking
- Branching Strategy
- Automated Build
- Source Control
- Requirements Tracking

Test

- Infastructure As Code
- Automated Regression Tests
- Accessibility Testing
- Automated Deploy to Test environments
- Security Testing
- Release Management
- Automated Browser Testing
- Test Case management
- Unit Testing
- Bug tracking
- Automated Performance Tests

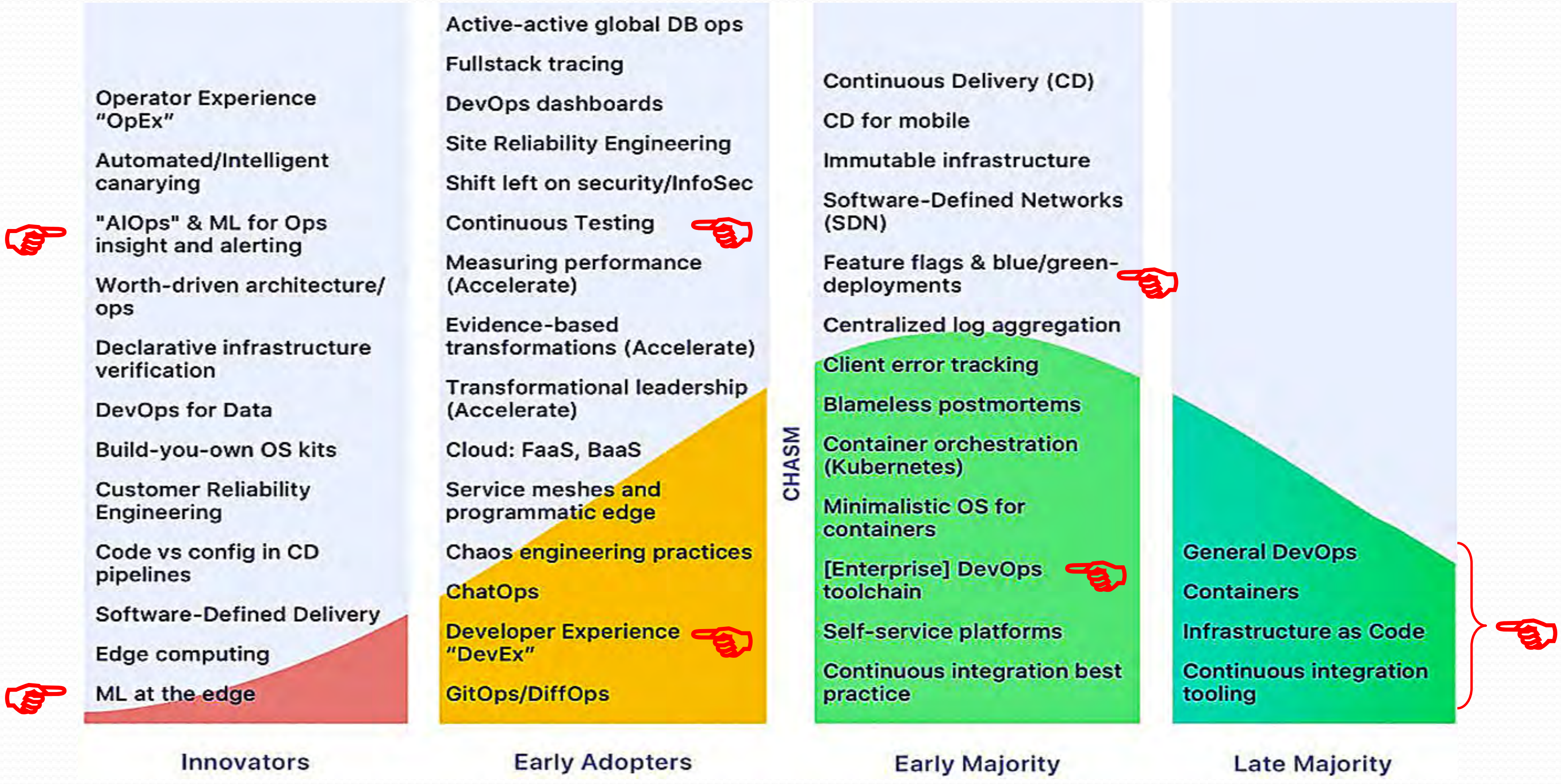
Operate

- Automated Server Patching
- Status Reports
- Real-time Reports
- Automated Backups
- Automated Rollbacks/Restores
- Continuous Delivery
- Continuous Deployment
- Clustered Deployments
- Cloud Deployments
- Monitoring tools
- Continuous Monitoring
- Automated Prod Deployment Package
- Automated Prod Deployments
- Release Notes
- Generated Release Notes

St-Cyr, J. (2015). *Evolving devops*: *Advance alm and devops practices with cont. imp. Agile Dev, Better Software, & DevOps East Conference, Orlando, Florida, USA.*

DevSecOps—Trends

- □ Containers, Ubuntu images, and pipelines are norm
- □ Fully automated testing and app security on the rise
- ☞ □ Future in DevOps Experience, BI DevOps, & AIOps



Innovators
- Operator Experience "OpEx"
- Automated/Intelligent canarying
- ☞ "AIOps" & ML for Ops insight and alerting
- Worth-driven architecture/ ops
- Declarative infrastructure verification
- DevOps for Data
- Build-you-own OS kits
- Customer Reliability Engineering
- Code vs config in CD pipelines
- Software-Defined Delivery
- Edge computing
- ☞ ML at the edge

Early Adopters
- Active-active global DB ops
- Fullstack tracing
- DevOps dashboards
- Site Reliability Engineering
- Shift left on security/InfoSec
- Continuous Testing ☞
- Measuring performance (Accelerate)
- Evidence-based transformations (Accelerate)
- Transformational leadership (Accelerate)
- Cloud: FaaS, BaaS
- Service meshes and programmatic edge
- Chaos engineering practices
- ChatOps
- Developer Experience "DevEx" ☞
- GitOps/DiffOps

CHASM

Early Majority
- Continuous Delivery (CD)
- CD for mobile
- Immutable infrastructure
- Software-Defined Networks (SDN)
- Feature flags & blue/green-deployments ☞
- Centralized log aggregation
- Client error tracking
- Blameless postmortems
- Container orchestration (Kubernetes)
- Minimalistic OS for containers
- [Enterprise] DevOps toolchain ☞
- Self-service platforms
- Continuous integration best practice

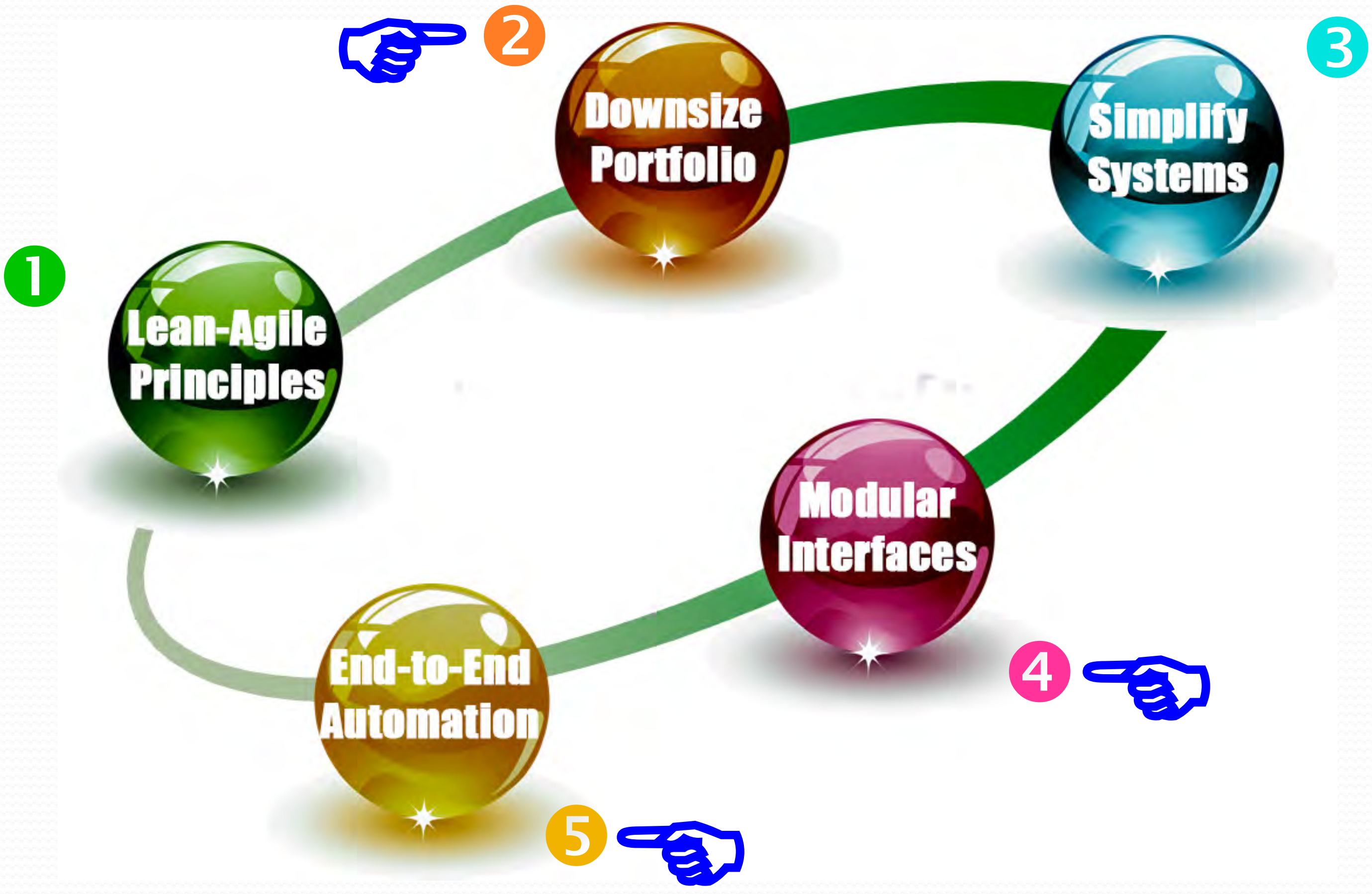
Late Majority
- General DevOps
- Containers
- Infrastructure as Code
- Continuous integration tooling

☞

Bryant, D., et al. (2019). *Devops and cloud infoq trends report*. Retrieved September 9, 2019, from http://infoq.link/devops-trends-2019

DevSecOps—Keys to Success

- □ Everything begins with lean & agile principles
- □ Next step is smaller portfolio & simpler designs
- ☞ □ Final step is modular interfaces & E2E automation



Kim, G., Debois, P., Willis, J., & Humble, J. *The devops handbook*: *How to create world-class agility, reliability, and security in technology organizations*. Portland, OR: IT Revolution Press.

DevSecOps—Summary

- □ DevOps DOES NOT mean deliver it now and fix it later
- □ Lightweight, yet disciplined approach to development
- □ Reduced cost, risk, & waste while improving quality

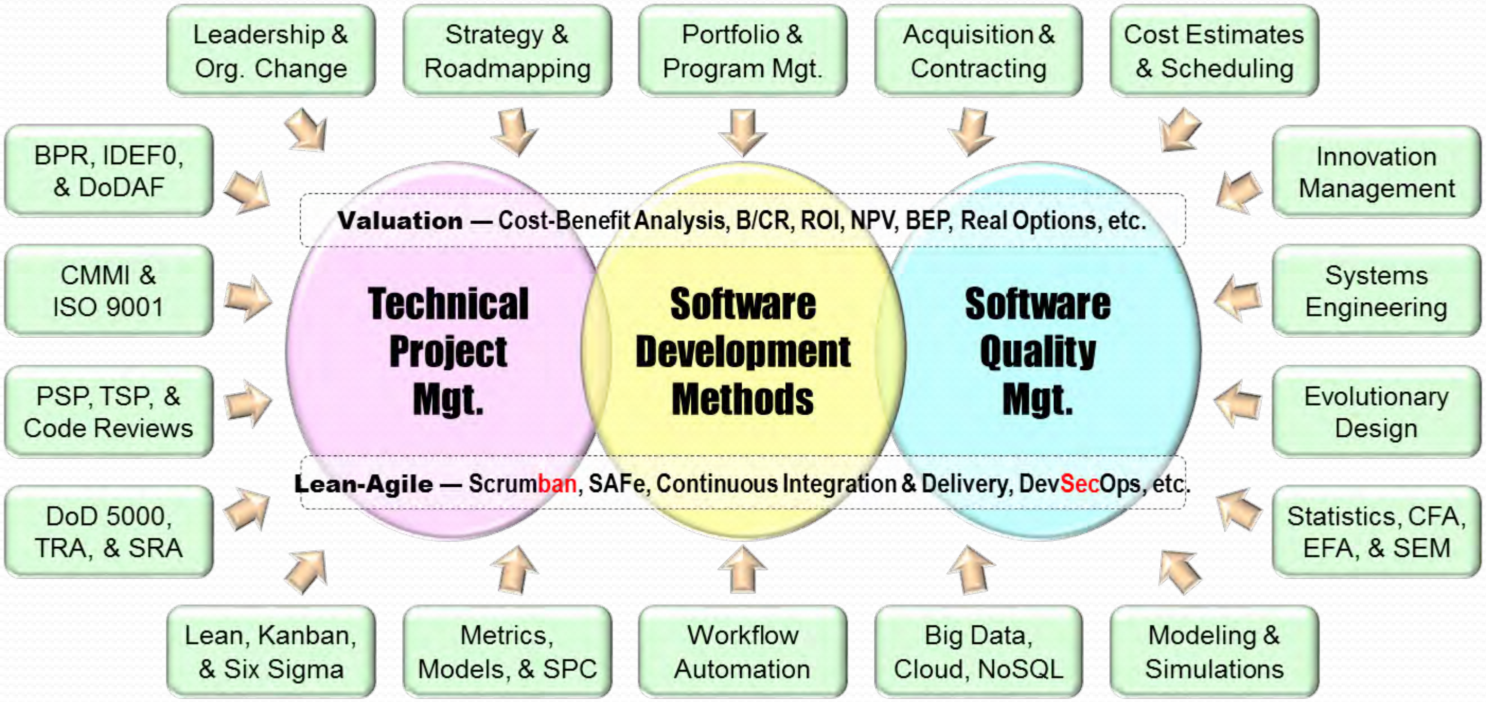
What	How	Result
Flexibility	Use lightweight, yet disciplined processes and artifacts	Low work-in-process
Customer	**Involve customers early and often throughout development**	Early feedback
Prioritize	**Identify highest-priority, value-adding business needs**	Focused Priorities
Descope	**Descope complex programs by an order of magnitude**	Vicious Simplicity
Decompose	**Divide the remaining scope into smaller batches**	Extremely Small Batches
Iterate	Implement pieces one at a time over long periods of time	Diffuse risk
Leanness	Architect and design the system one iteration at a time	JIT waste-free design
Swarm	**Implement each component in small cross-functional teams**	Radical Teamwork
Collaborate	Use frequent informal communications as often as possible	Efficient data transfer
Test Early	**Incrementally test each component as it is developed**	Early/auto Verification
Test Often	**Perform system-level regression testing every few minutes**	Early/auto Validation
Adapt	Frequently identify optimal process and product solutions	Improve performance
Security	**Bake in security and automate it throughout lifecycle**	Ironclad Security

Rico, D. F. (2019). *32 attributes of successful continuous integration, continuous delivery, and DevOps.* Retrieved September 27, 2019, from http://davidfrico.com/devops-principles.pdf

DevSecOps—Bottom Line?

DevOps ensures enterprise success by delivering large volumes of valuable, reliable, & secure IT products & services to billions of users in fractions of a second ...

Dave's Professional Capabilities



Leadership & Org. Change

Strategy & Roadmapping

Portfolio & Program Mgt.

Acquisition & Contracting

Cost Estimates & Scheduling

BPR, IDEF0, & DoDAF

CMMI & ISO 9001

PSP, TSP, & Code Reviews

DoD 5000, TRA, & SRA

Innovation Management

Systems Engineering

Evolutionary Design

Statistics, CFA, EFA, & SEM

Valuation — Cost-Benefit Analysis, B/CR, ROI, NPV, BEP, Real Options, etc.

Technical Project Mgt.

Software Development Methods

Software Quality Mgt.

Lean-Agile — Scrumban, SAFe, Continuous Integration & Delivery, DevSecOps, etc.

Lean, Kanban, & Six Sigma

Metrics, Models, & SPC

Workflow Automation

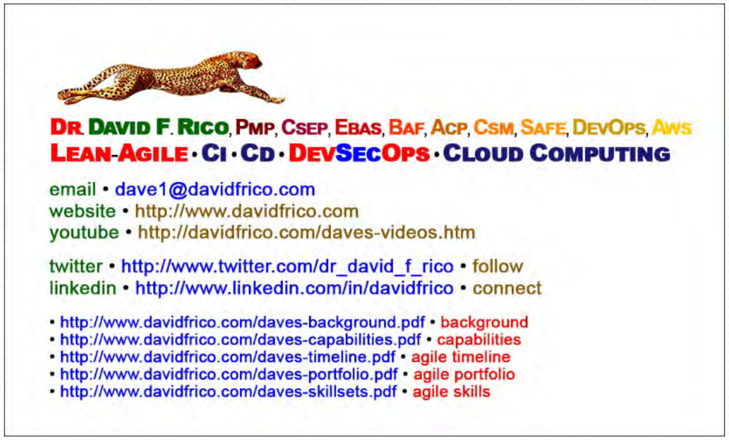
Big Data, Cloud, NoSQL

Modeling & Simulations

☞ Website: http://davidfrico.com ● LinkedIn: http://linkedin.com/in/davidfrico ● Twitter: @dr_david_f_rico ☜

STRENGTHS – Lean & Agile Thinking • Enterprise Transformation & Roadmapping • 360 Leadership Assessments • Executive & Agile Coaching • Enterprise Business Agility • Agile Acquisition Contracts • Scaled Agile Framework (SAFe) • Development Security Operations (DevSecOps) • Cloud Computing & Amazon Web Services (AWS) • Portfolio, Program, & Project Mgt. • Lean-Agile Product Management & Design Thinking • 5x5x5 Innovation & Marketing Sprints • Annual & Quarterly Strategic Planning • Technology & Product Roadmapping • Program Increment & Big Room Planning • Emergent & Evolutionary Microservices • Exploratory MVP, MVA, & MMF Experiments • Scrumban, Kanban & Lean-Agile Assessments • Performance Metrics, Measures & Dashboards • Agile lifecycle management (ALM) workflow tools ...

39+ Years in IT Industry



Dr. David F. Rico, Pmp, Csep, Ebas, Baf, Acp, Csm, Safe, DevOps, Aws
Lean-Agile · Ci · Cd · DevSecOps · Cloud Computing

email • dave1@davidfrico.com
website • http://www.davidfrico.com
youtube • http://davidfrico.com/daves-videos.htm

twitter • http://www.twitter.com/dr_david_f_rico • follow
linkedin • http://www.linkedin.com/in/davidfrico • connect

- http://www.davidfrico.com/daves-background.pdf • background
- http://www.davidfrico.com/daves-capabilities.pdf • capabilities
- http://www.davidfrico.com/daves-timeline.pdf • agile timeline
- http://www.davidfrico.com/daves-portfolio.pdf • agile portfolio
- http://www.davidfrico.com/daves-skillsets.pdf • agile skills

Pmp, Csep, Ebas, Baf, Fcp, Fct, Acp, Csm, DevOps, Aws & SAFe