Business Value of CI, CD, & DevOps{Sec}

Scaling Up to Billion User Global Systems of Systems Using End-to-End Automation & Containerized Docker Ubuntu Images

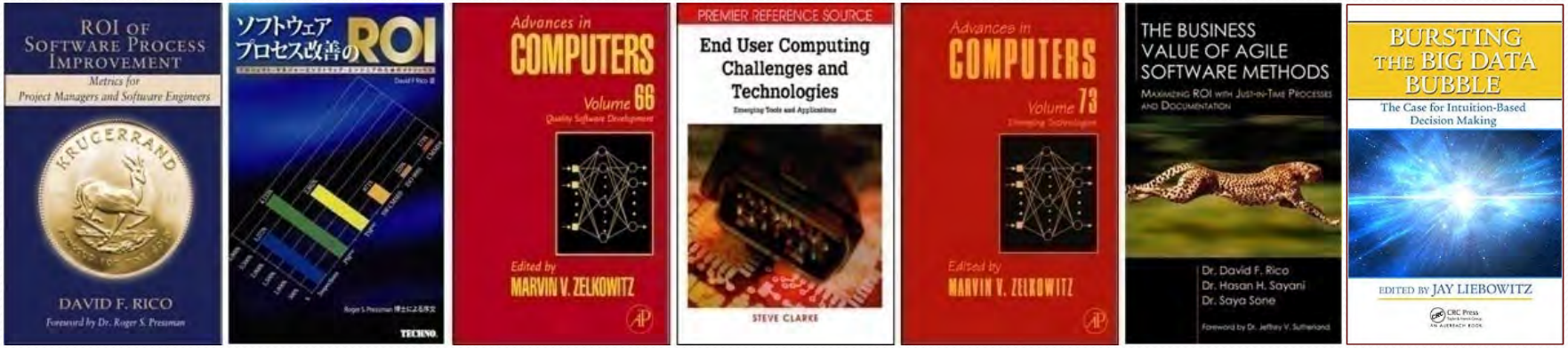
Dr. David F. Rico, PMP, CSEP, FCP, FCT, ACP, CSM, SAFe, DevOps

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

Agile Capabilities: http://davidfrico.com/rico-capability-agile.pdf
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: https://www.youtube.com/watch?v=-wTXqN-OBzA
Dave's NEWER Development Operations Security Video: https://vimeo.com/214895416
DoD Fighter Jets vs. Amazon Web Services: http://davidfrico.com/dod-agile-principles.pdf

Author Background

- Gov't contractor with 34+ years of IT experience
- B.S. Comp. Sci., M.S. Soft. Eng., & D.M. Info. Sys.
- ☞ Large gov't projects in U.S., Far/Mid-East, & Europe



→ ***Career systems & software engineering methodologist***
→ *Lean-Agile, Six Sigma, CMMI, ISO 9001, DoD 5000*
→ *NASA, USAF, Navy, Army, DISA, & DARPA projects*
→ *Published seven books & numerous journal articles*
→ *Intn'l keynote speaker, 185+ talks to 14,000 people*
→ *Specializes in metrics, models, & cost engineering*
→ *Cloud Computing, SOA, Web Services, FOSS, etc.*
→ *Professor at 7 Washington, DC-area universities*

DevOps—Dinosaur Killer

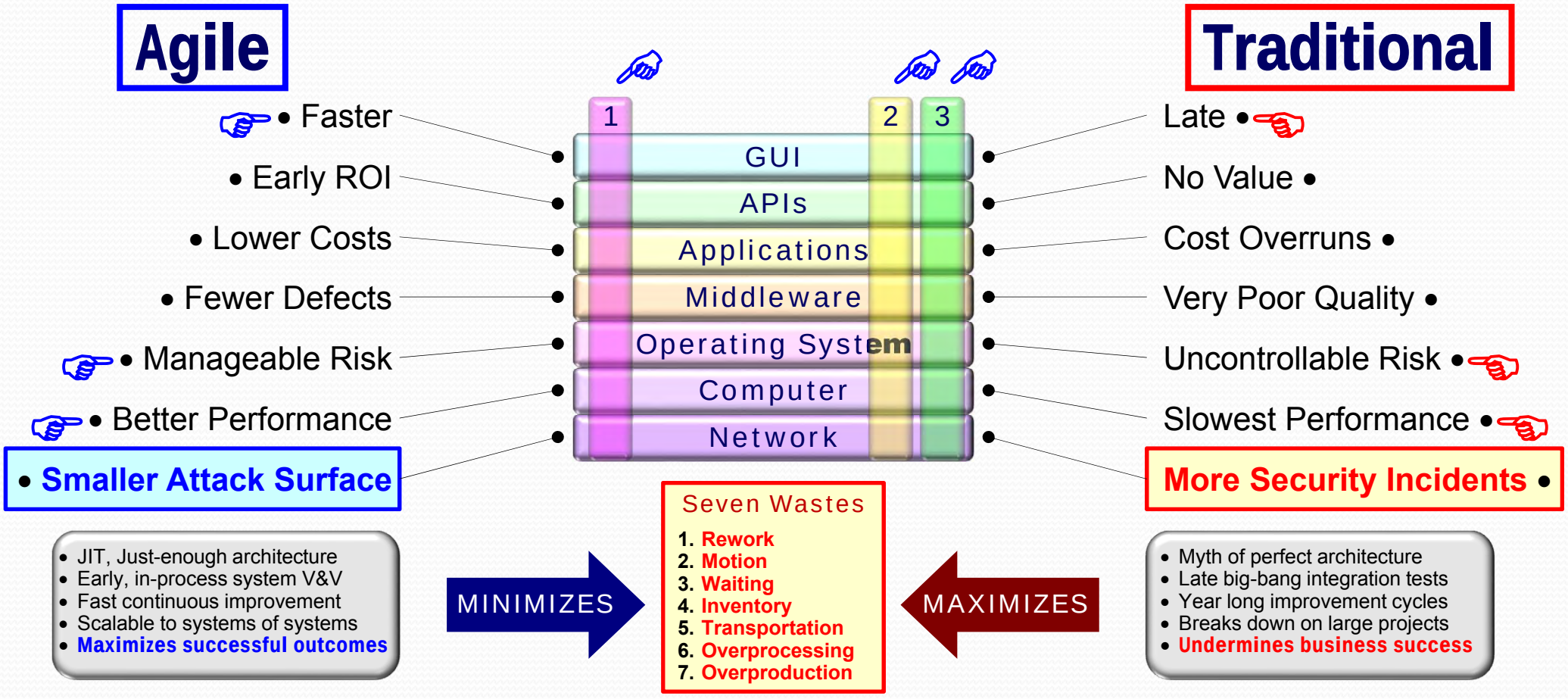
DevOps—What is it?

- 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
  - *Mult-tiered automated framework for* TDD, Continuous Integration, BDD, Continuous Delivery, & DevOps
  - *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.

DevOps—How it works?

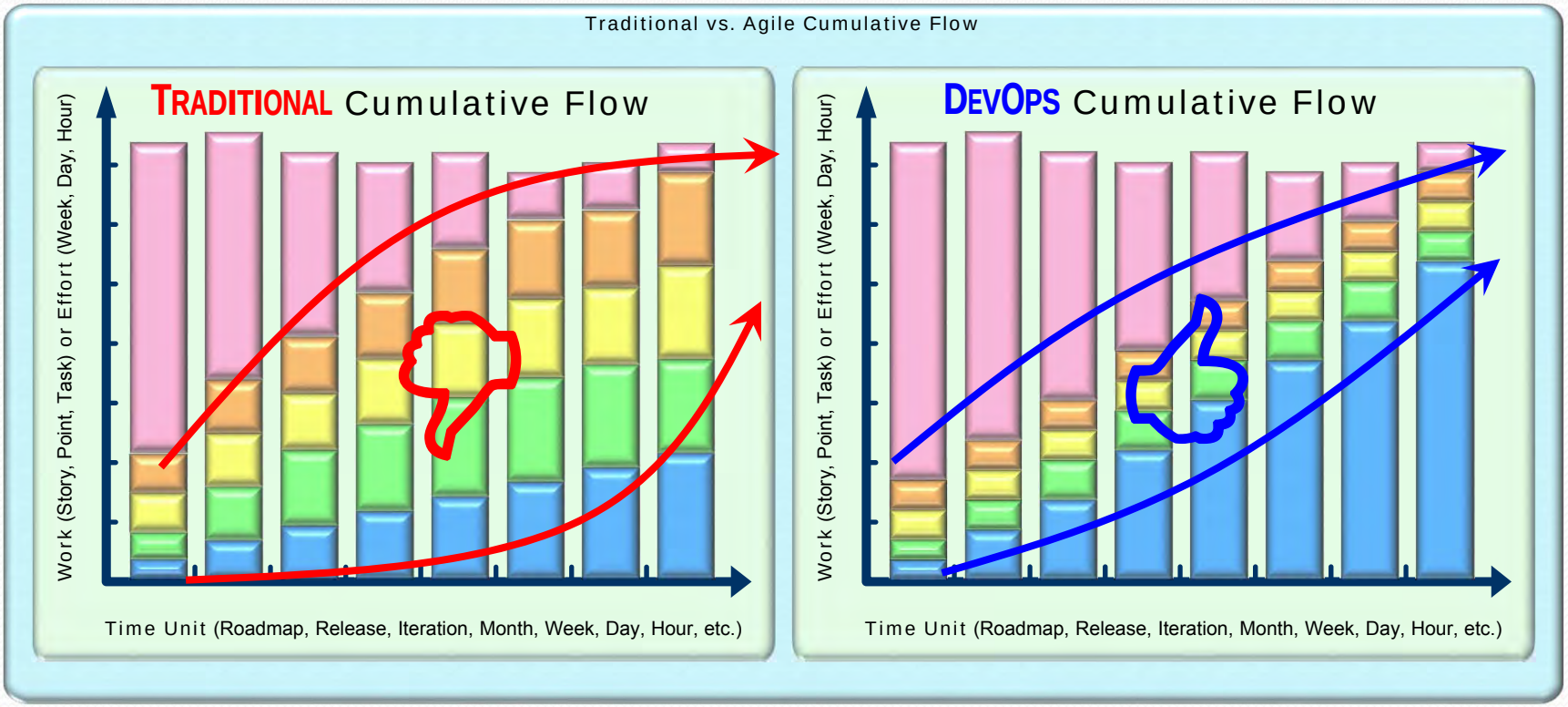
- Agile requirements implemented in slices vs. layers
- User needs with higher business value are done first
- Reduces cost & risk while increasing business success



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

DevOps—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.

DevOps—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

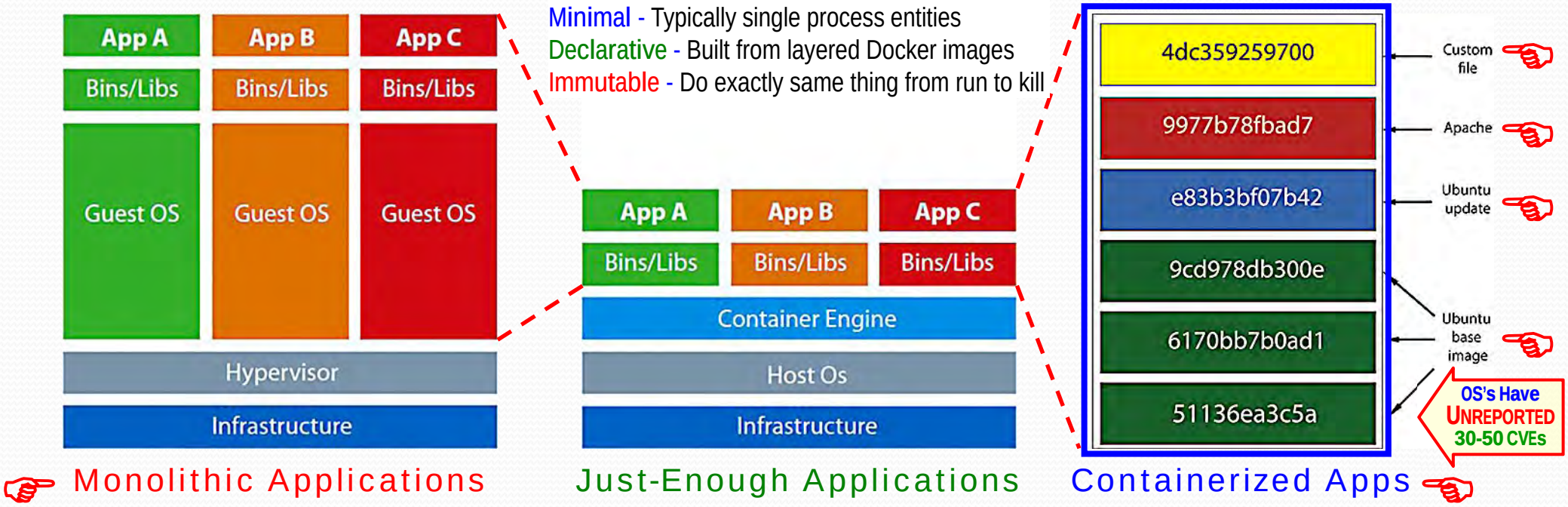
Minimum Marketable Feature - MMF -	Minimum Viable Product - MVP -	Story Map or Impact Map - SM or IM -	Vision Statement - VS -	Micro-Service - MS -
✓ Advantage	✓ Goal	✓ Goal	✓ For <*customer*>	✓ Purpose
✓ Difference	✓ Process	✓ Actors	✓ Who <*needs it*>	✓ Automated
✓ Revenue	✓ Features	✓ Impacts	✓ The <*product*>	✓ Unique
✓ Profit	✓ Priorities	✓ Deliverables	✓ Is a <*benefit*>	✓ Independent
✓ Savings	✓ Story Map	✓ Measures	✓ That <*customer*>	✓ Resilient
✓ Brand	✓ Architecture	✓ Milestones	✓ Unlike <*other*>	✓ Ecosystem
✓ Loyalty			✓ Ours <*different*>	✓ 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.

DevOps—Microservices

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

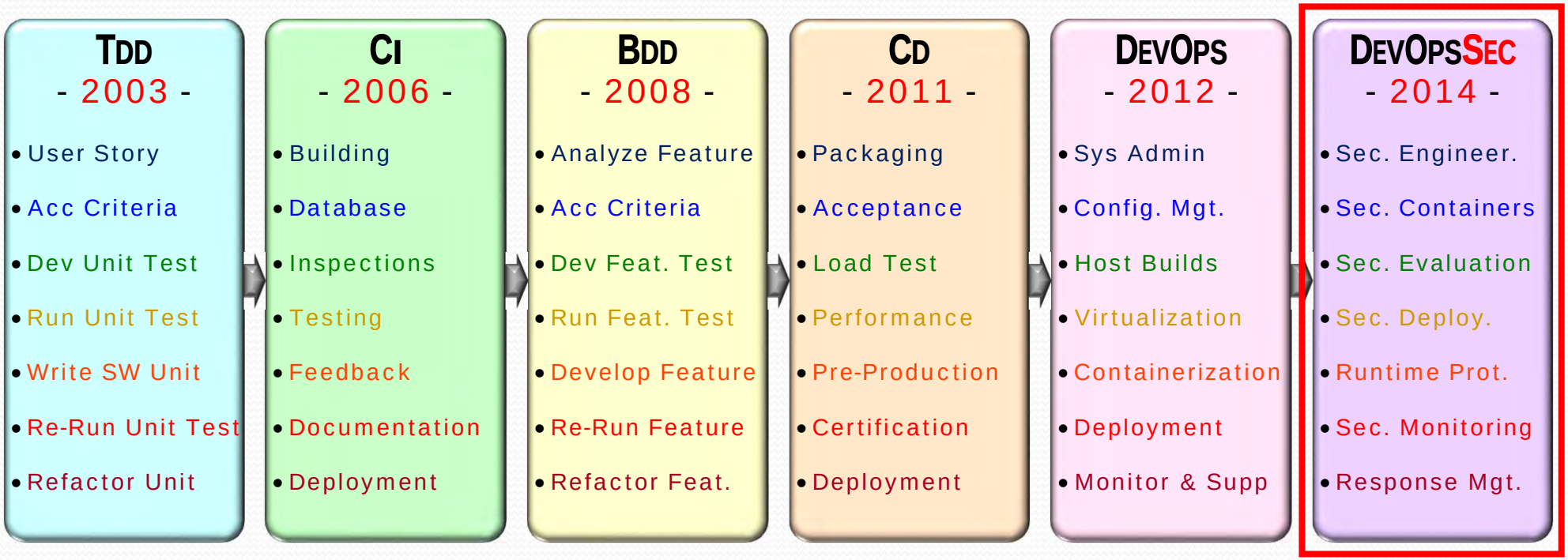


- **Small autonomous services that work together**
- **Self-contained process that provides a unique capability**
- **Loosely coupled service oriented architecture with bounded contexts**
- **Small independent processes communicating with each other using language-agnostic APIs**
- **Fined-grained independent 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.

DevOps—Evolution

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



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.

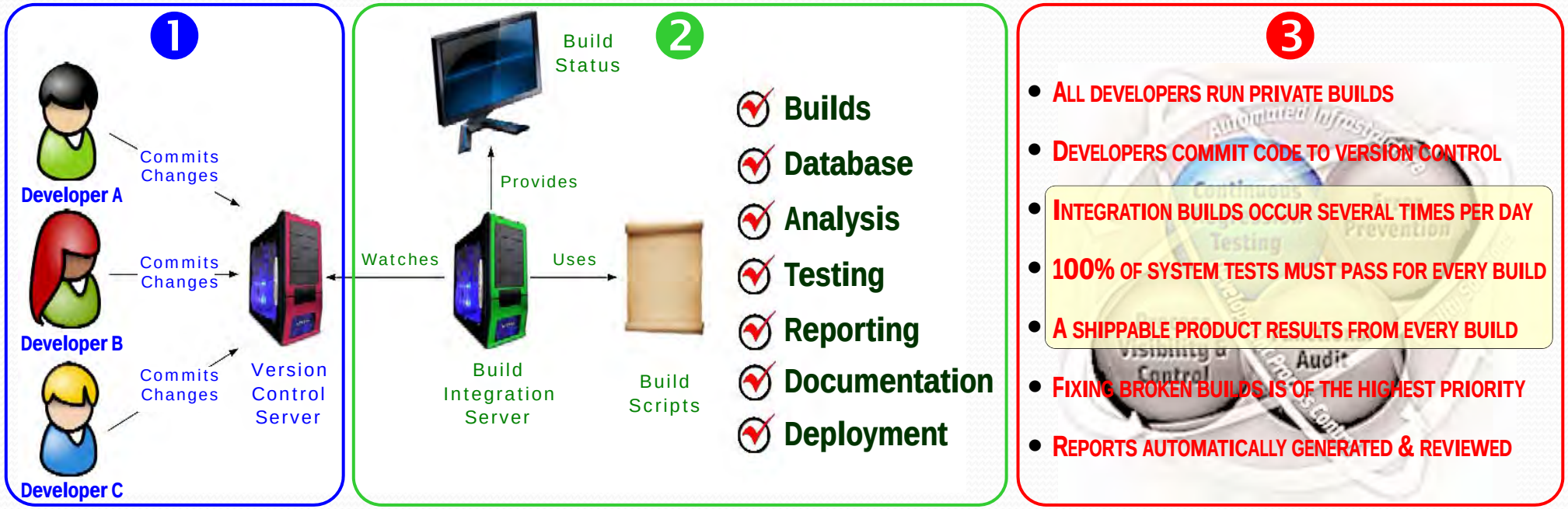
Stage 3—Continuous Integration

- 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



Constant Readiness State & CM Control

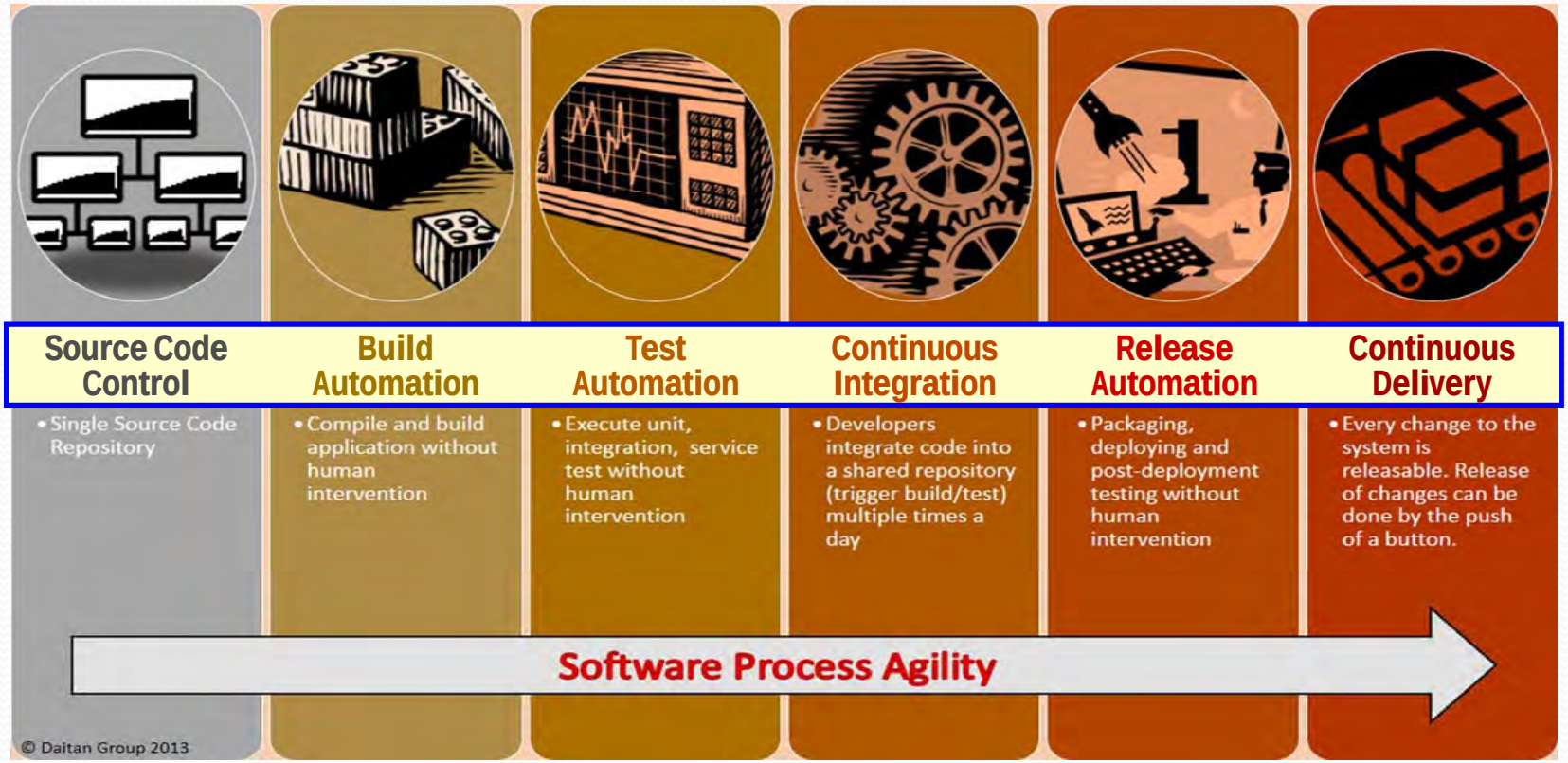
No More Late Big Bang Integration

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 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

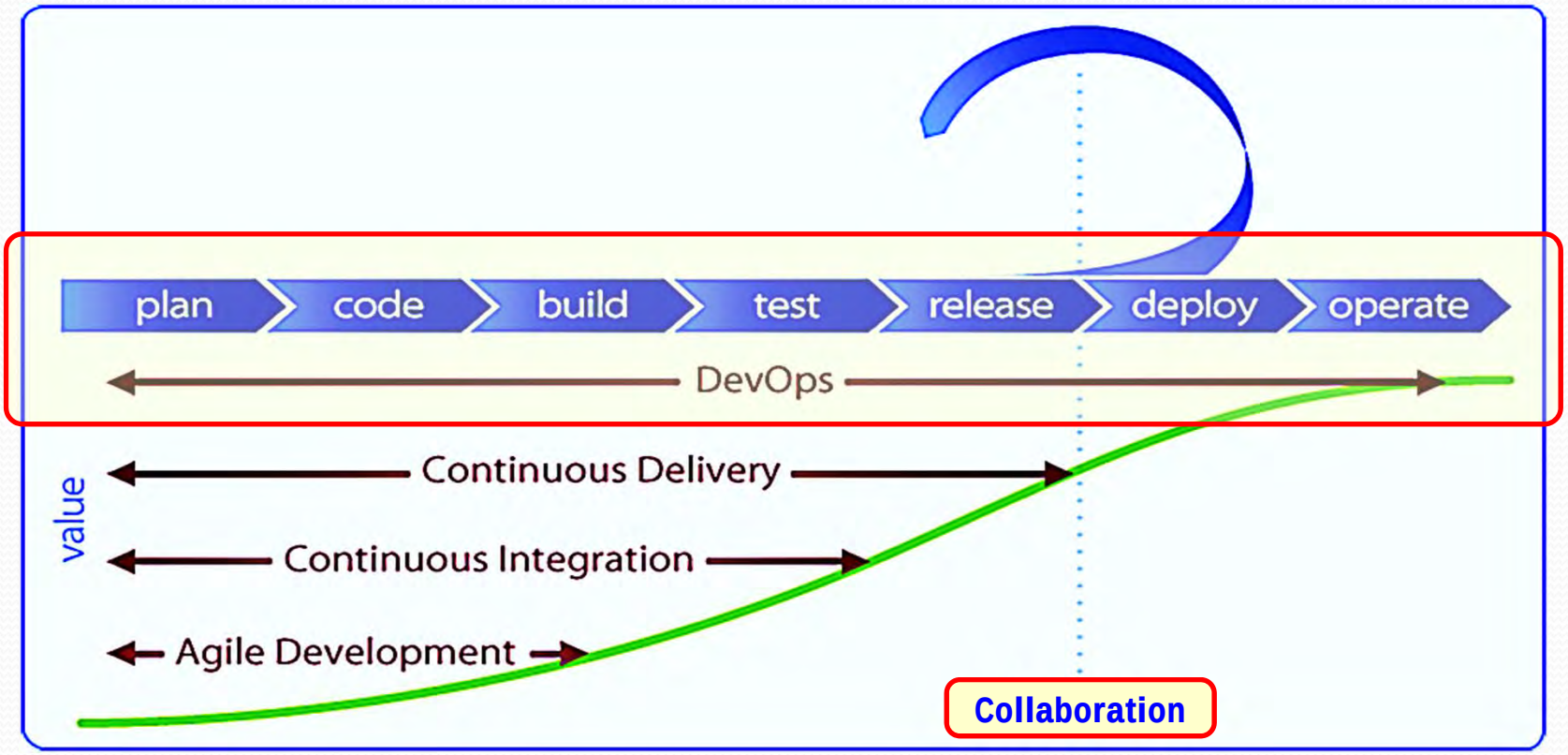


CoQ
- 80% MS Tst
- 8/10 No Val
- $24B in 90s
- Rep by CD
- Not Add MLK

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 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



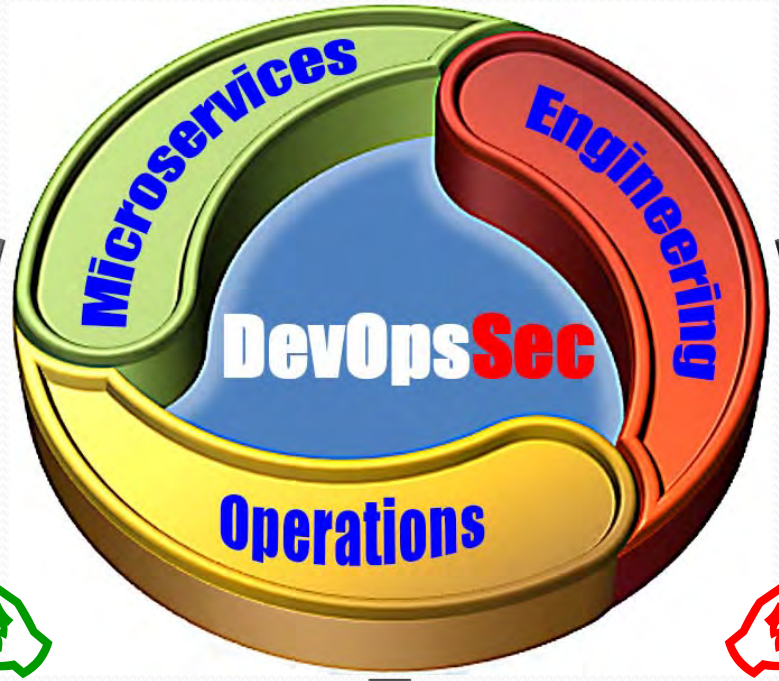
Bass, L., Weber, I., & Zhu, L. (2015). *Devops*: *A software architect's perspective*. Old Tappan, NJ: Pearson Education.
Gruver, G., & Mouser, T. (2015). *Leading the transformation*: *Applying agile and devops at scale*. Portland, OR: IT Revolution Press.
Humble, J., Molesky, J., & O'Reilly, B. (2015).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

Stage 6—Development Ops Sec

- DevOpsSec coined by Shannon Lietz in 2014
- Rugged devops, devsecops, secdevops, devopssec
- 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

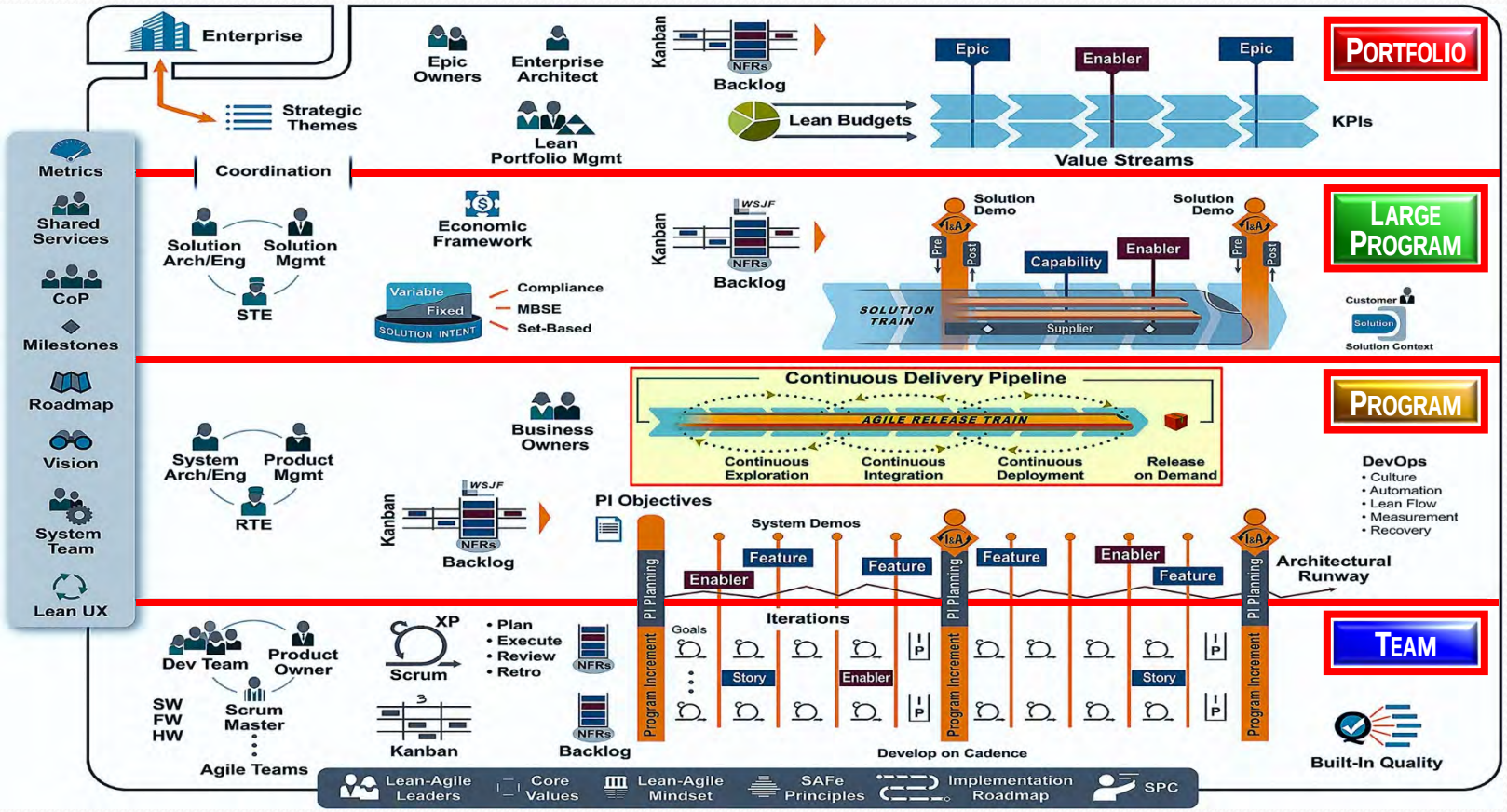
Secure Operations

- 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 7—Enterprise DevOpsSec

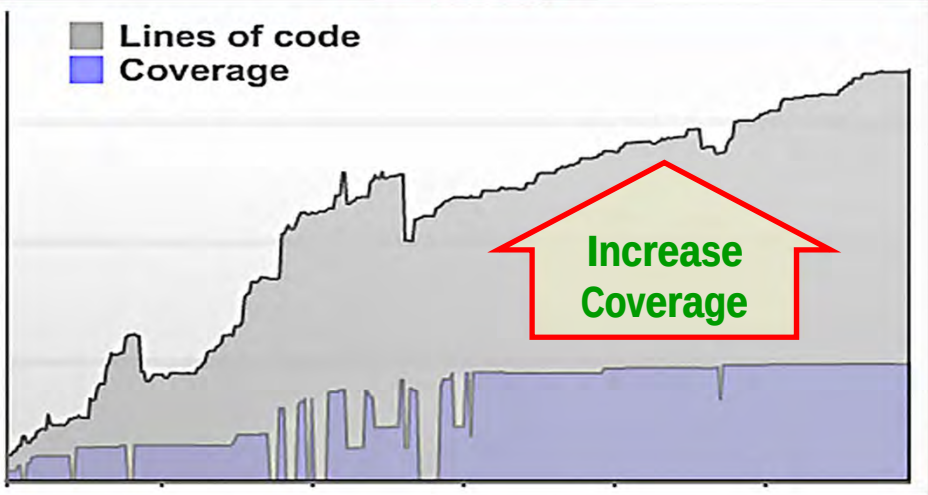
- SE framework by Dean Leffingwell of Rally in 2007
- Newest version leaner, meaner, lighter, and simpler
- Experimental bottoms-up DevOps-based innovation



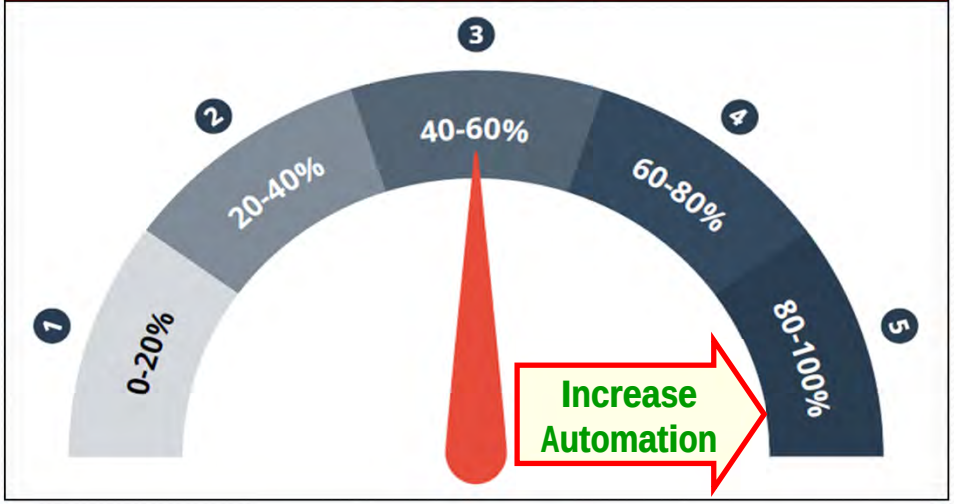
Leffingwell, D. (2007). *Scaling software agility*: *Best practices for large enterprises*. Boston, MA: Pearson Education.

DevOps—Automation Metrics

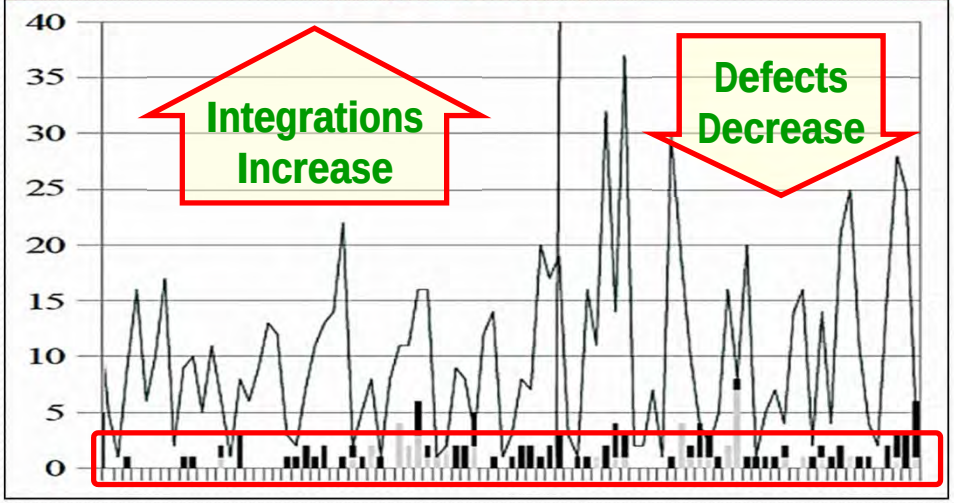
Test Coverage



Test Automation



Integration Builds

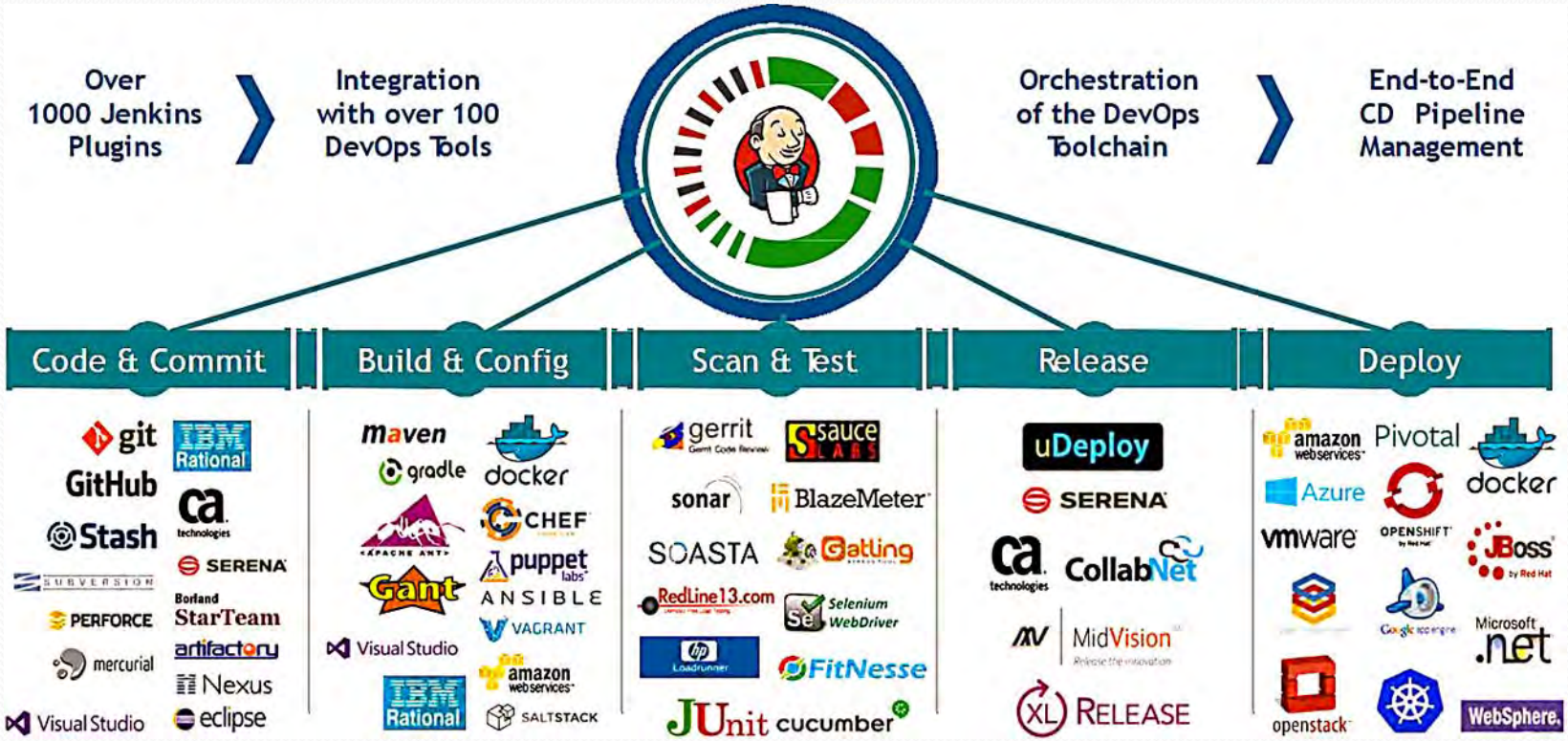


Running Tested Features

Duvall, P., Matyas, S., & Glover, A. (2006). *Continuous integration*: *Improving software quality and reducing risk*. Boston, MA: Addison-Wesley.

DevOps—Tools Ecosystem

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

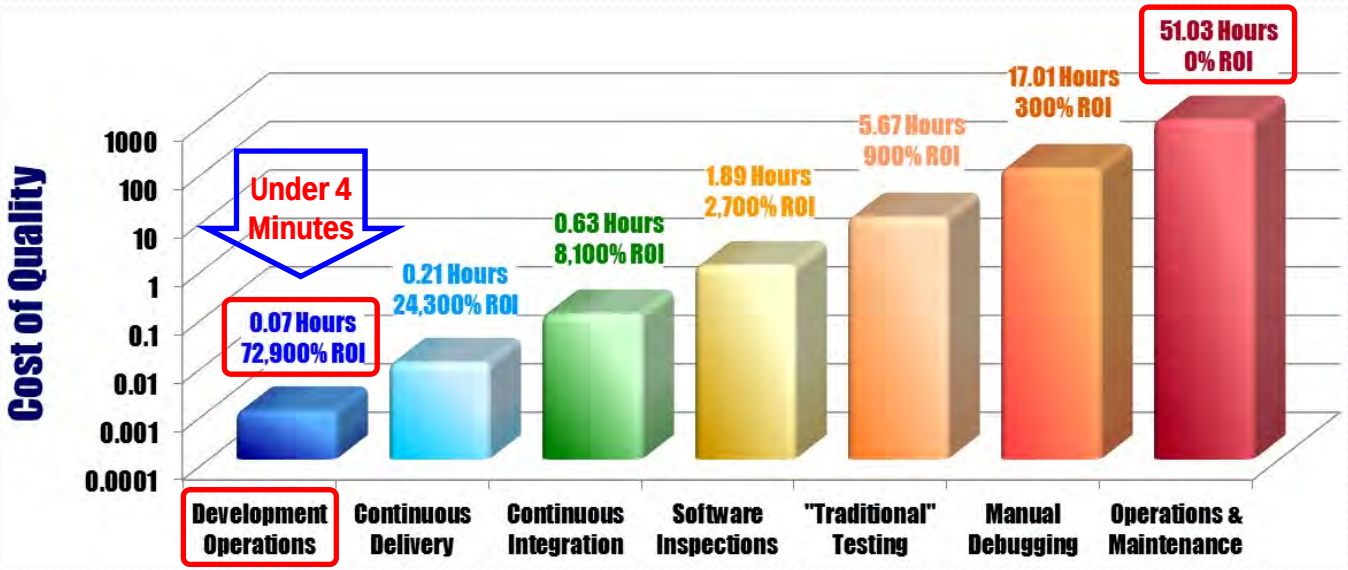


Juengst, D. (2015). *Deliver better software faster*: *With the cloudbees jenkins platform*. San Francisco, CA: CloudBees.
Weeks, D. E. (2014). *Devops and continuous delivery reference architectures* (*volume 1 & 2*). Fulton, MD: Sonatype.

DevOps—Cost of Quality

- Agile testing 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



250 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

DevOps—HP Case Study

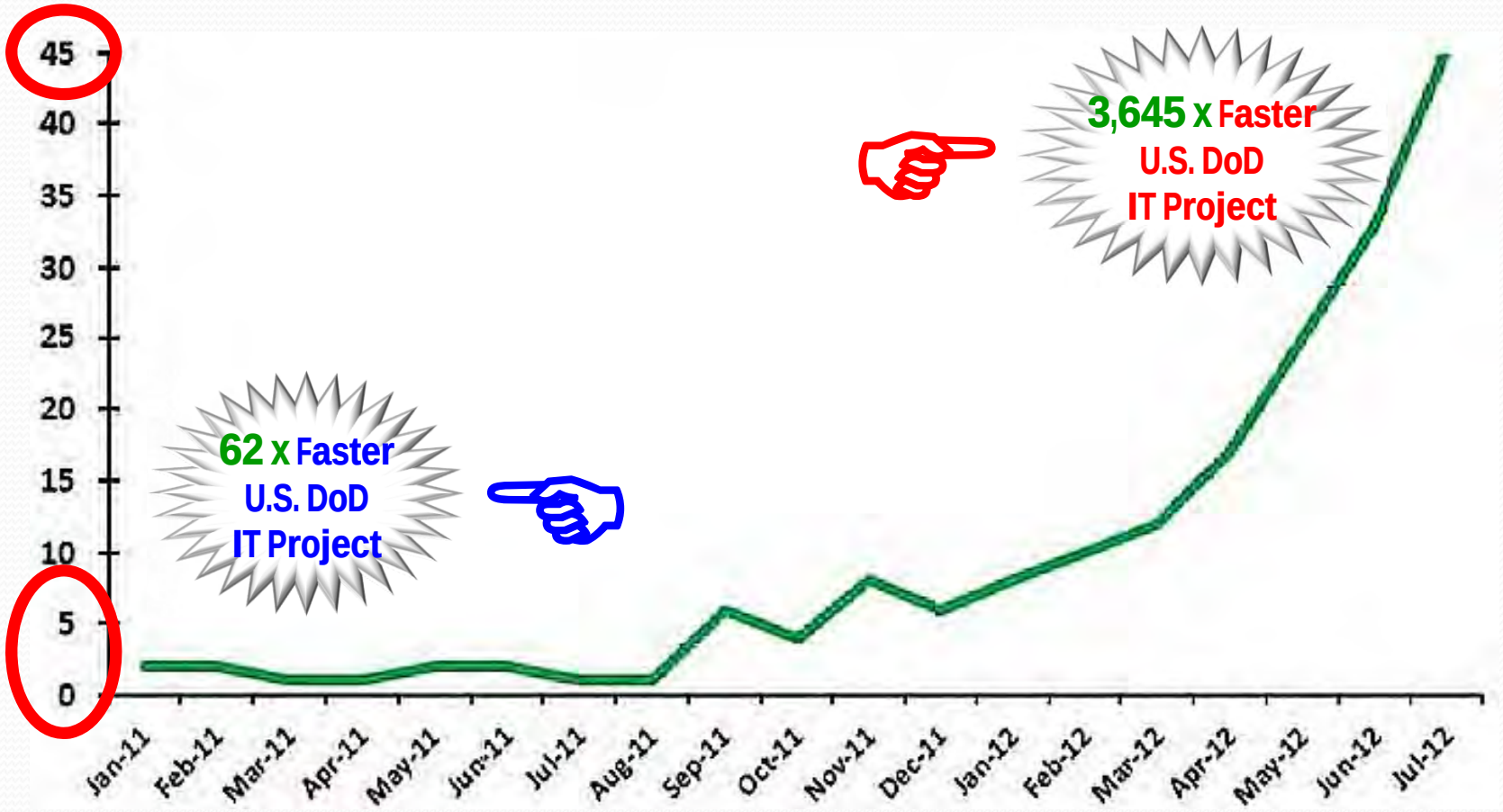
- Hewlett-Packard is a major user of CI, CD, & DevOps
- 400 engineers developed 10 million LOC in 4 years
- Major gains in testing, deployment, & innovation

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.

DevOps—Dot Com Case Study

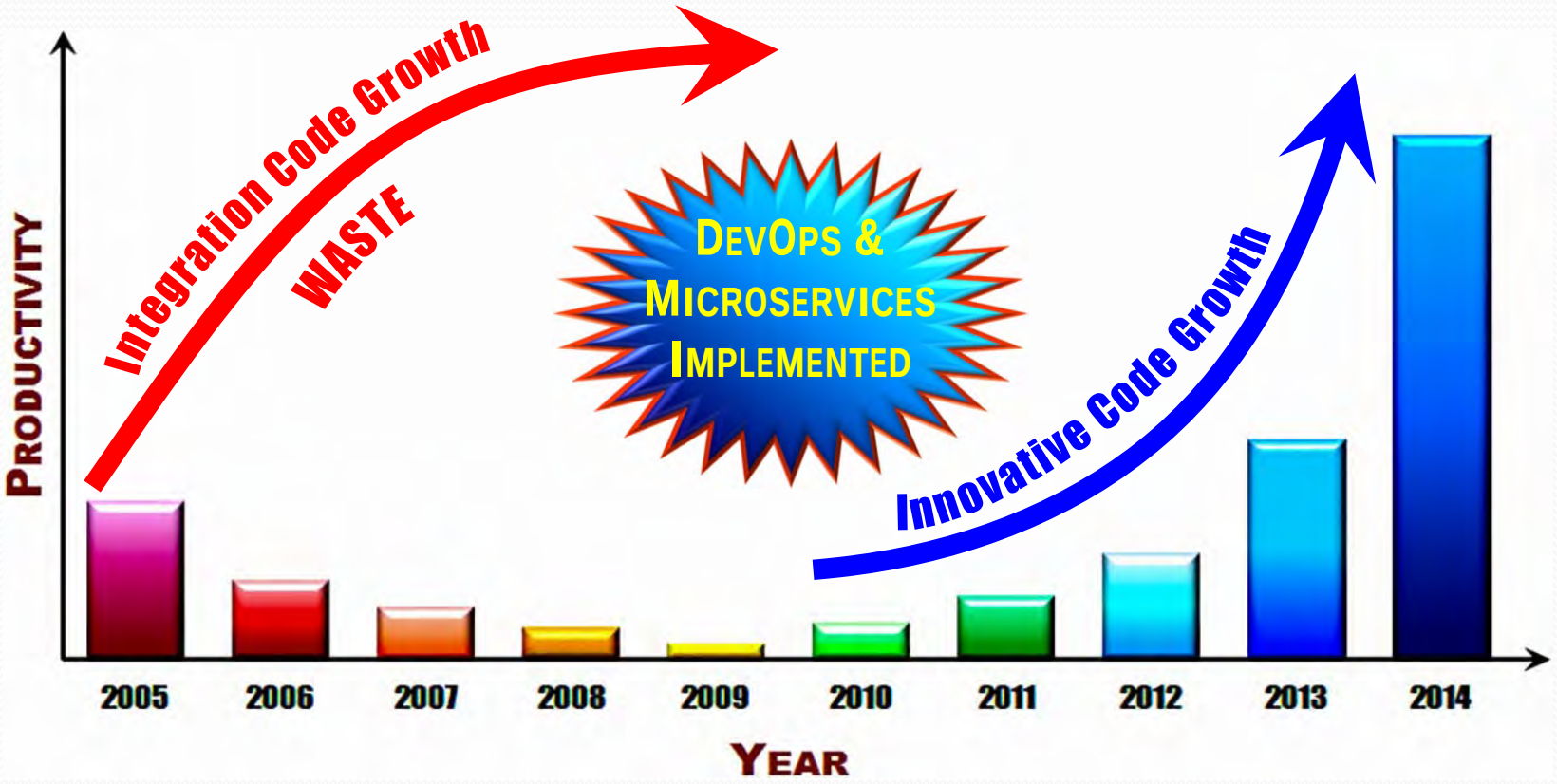
- Assembla went from 2 to 45 releases every month
- 15K Google developers run 120 million tests per day
- 30K+ Amazon developers deliver 136K releases a day



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

DevOps—Blackboard Case Study

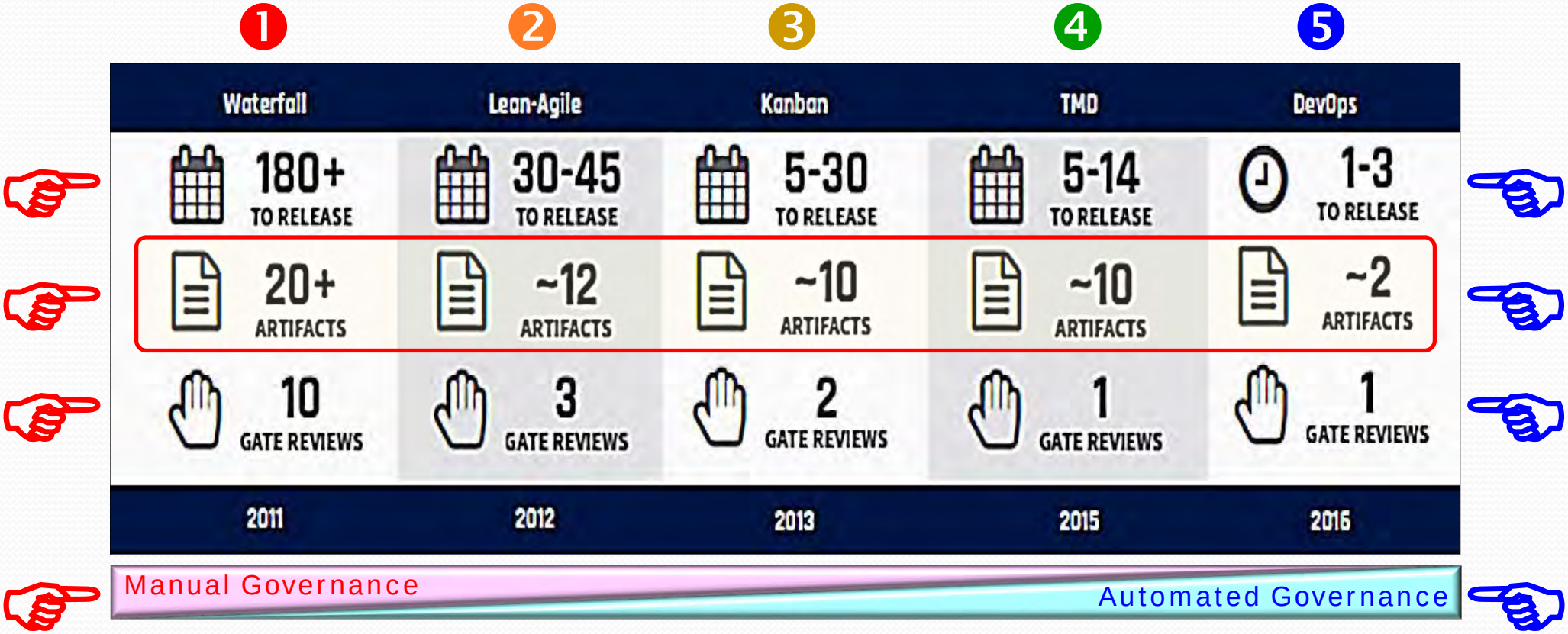
- Productivity **STOPS** due to excessive integration
- Implements DevOps & Microservices around 2010
- Waste elimination, productivity & innovation skyrocket



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

DevOps—U.S. DHS Case Study

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



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

DevOps—Return on Investment

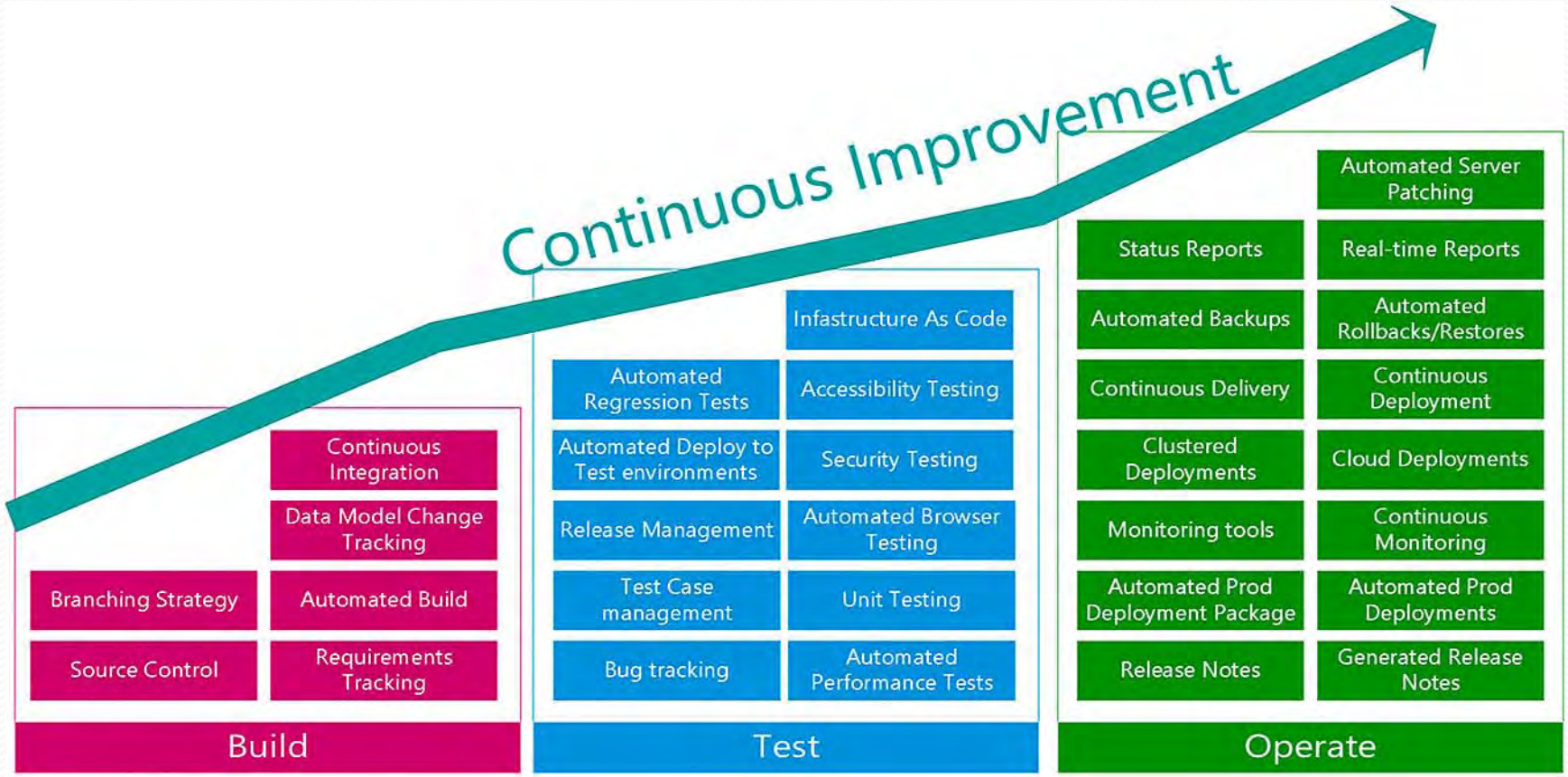
- Detailed DevOps economics starting to emerge
- ROI ranges from $17M to $195M *with minor costs*
- Benefits from cost savings, revenue, and availability

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

DevOps—Roadmap

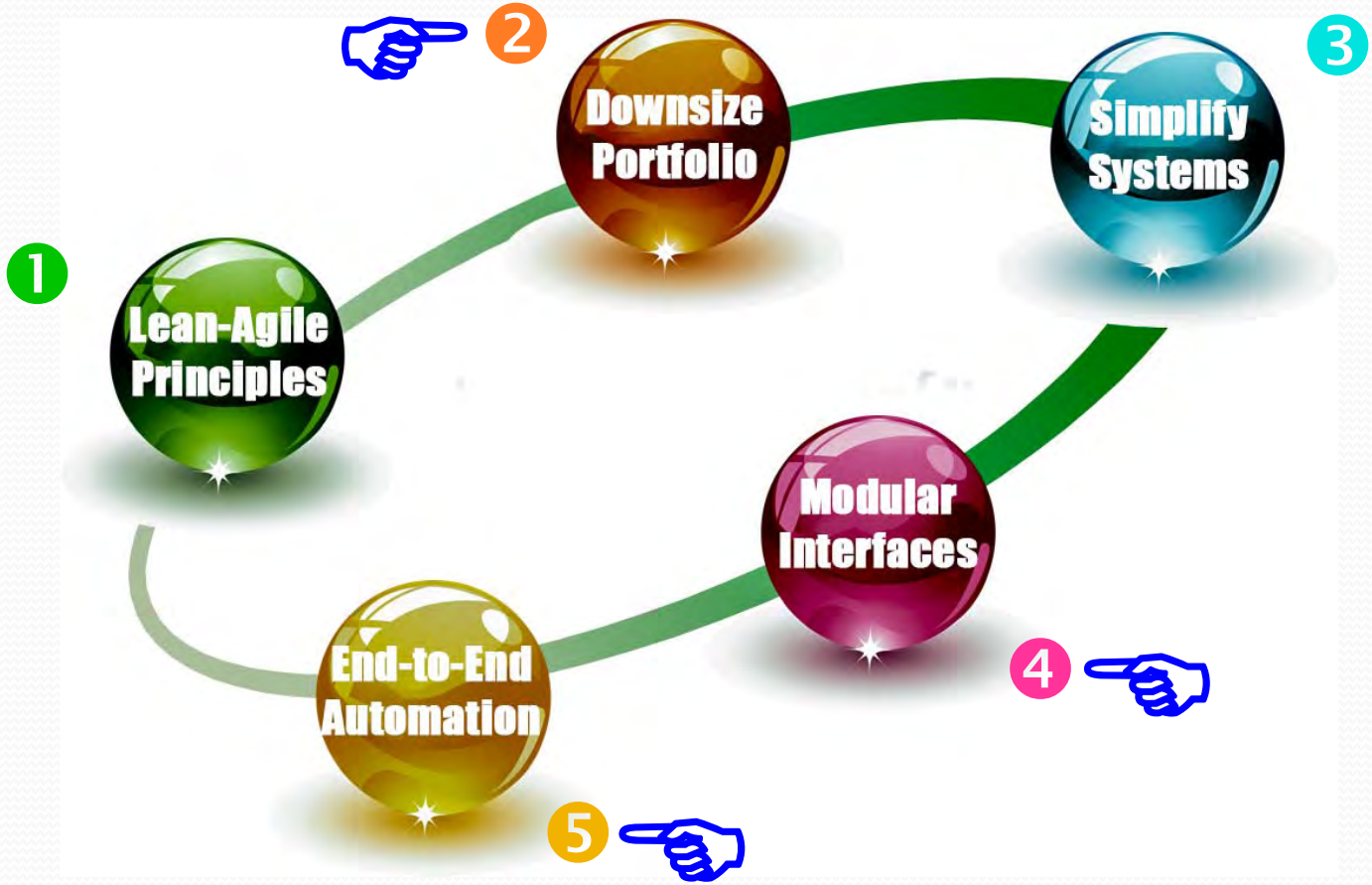
- Having a DevOps rollout strategy is a key to success
- Phased, incremental, and situational implementation
- Includes build, testing, & IT operations, & practices



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

DevOps—5 Keys to Success

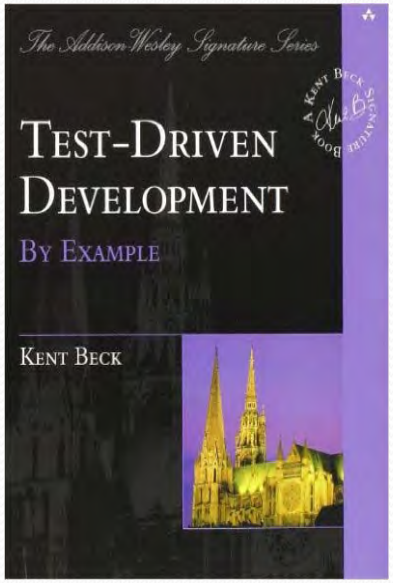
- Everything begins with lean & agile principles
- Next step is smaller portfolio & simpler designs
- Final step is modular interfaces & E2E automation

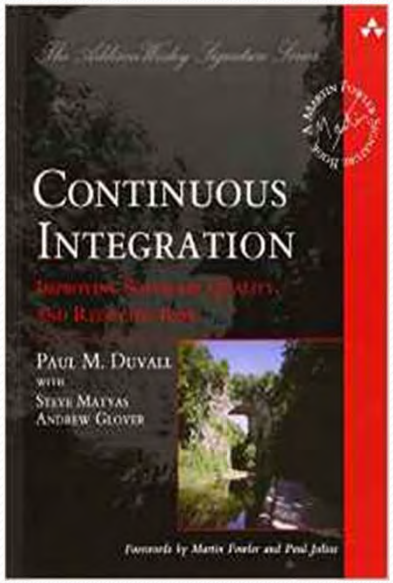


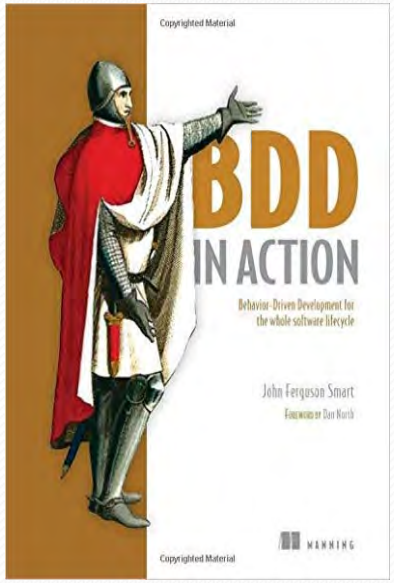
Kim, G., Debois, P., Willis, J., & Humble, J. *The devops handbook*: *How to create world-class agility, reliability, and security in technology organizations*. Portland, OR: IT Revolution Press.

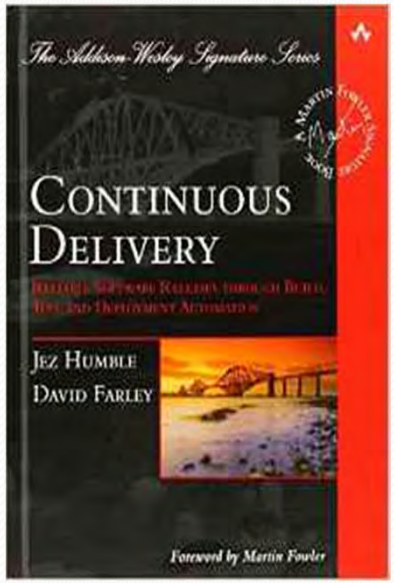
DevOps—Foundational Books

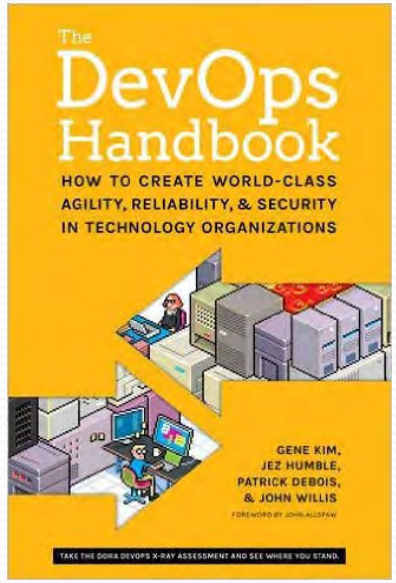
- Thousands of textbooks on agile methods
- Include requirements, design, coding, test, etc.
- ☞ Continuous Integration, Delivery, & DevOps best





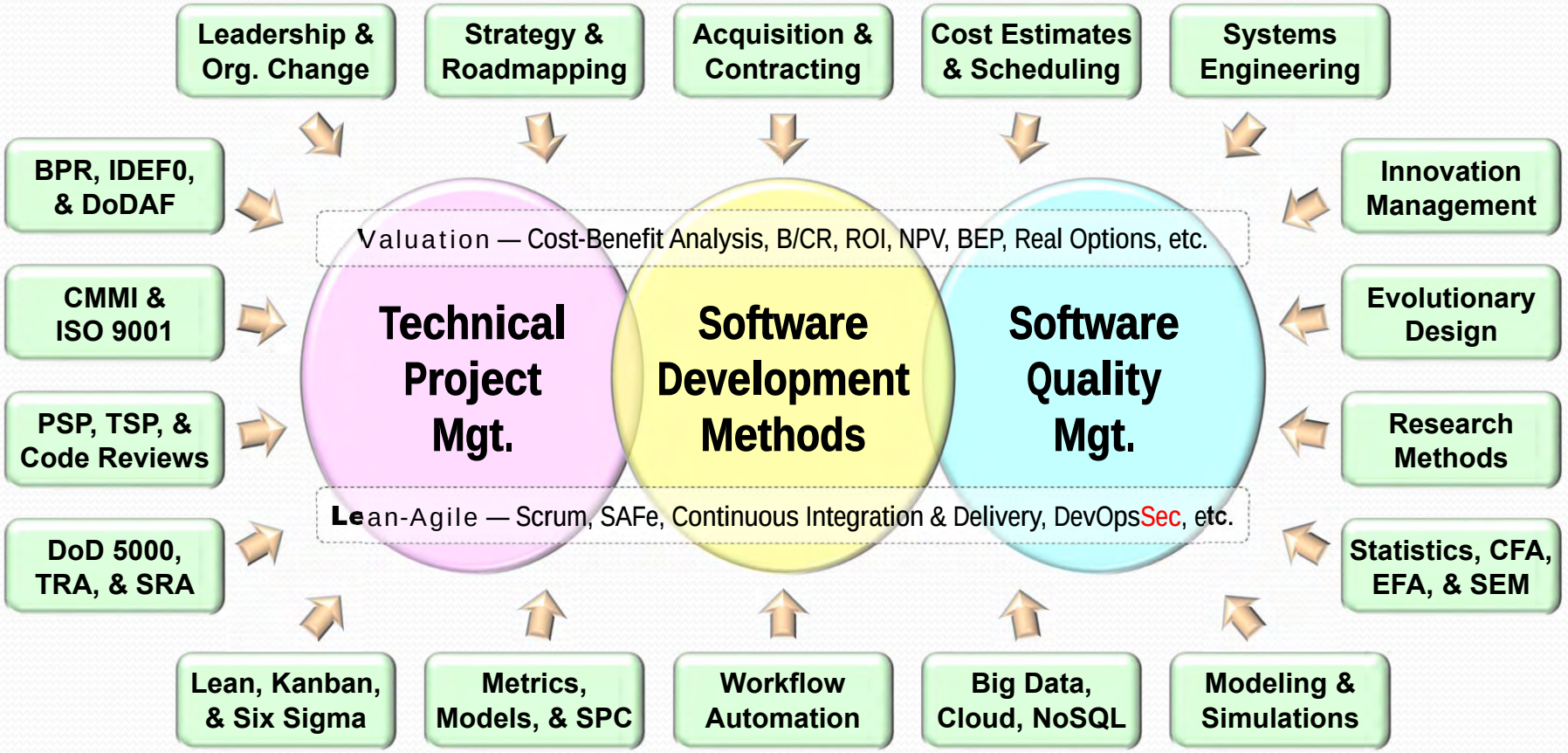






Beck, K. (2003). *Test-driven development*: *By example*. Boston, MA: Addison-Wesley.
Duvall, P., Matyas, S., & Glover, A. (2006). *Continuous integration*. Boston, MA: Addison-Wesley.
Smart, J. F. (2014). *BDD in action*: *Behavior-driven development for the whole software lifecycle*. Shelter Island, NY: Manning Publications.
Humble, J., & Farley, D. (2011). Continuous delivery. Boston, MA: Pearson Education.
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.

Dave's Professional Capabilities



STRENGTHS – Data Mining • Gathering & Reporting Performance Data • Strategic Planning • Executive & Management Briefs • Brownbags & Webinars • White Papers • Tiger-Teams • Short-Fuse Tasking • Audits & Reviews • Etc.

34+ Years in IT Industry

- **Data mining. Metrics, benchmarks, & performance.**
- **Simplification. Refactoring, refinement, & streamlining.**
- **Assessments. Audits, reviews, appraisals, & risk analysis.**
- **Coaching. Diagnosing, debugging, & restarting stalled projects.**
- **Business cases. Cost, benefit, & return-on-investment (ROI) analysis.**
- **Communications. Executive summaries, white papers, & lightning talks.**
- **Strategy & tactics. Program, project, task, & activity scoping, charters, & plans.**

PMP, CSEP, FCP, FCT, ACP, CSM, SAFe, & DevOps