# Broad Introduction

# Agile Methodologies

## For Business Executives, Technical Project Managers, and Developers

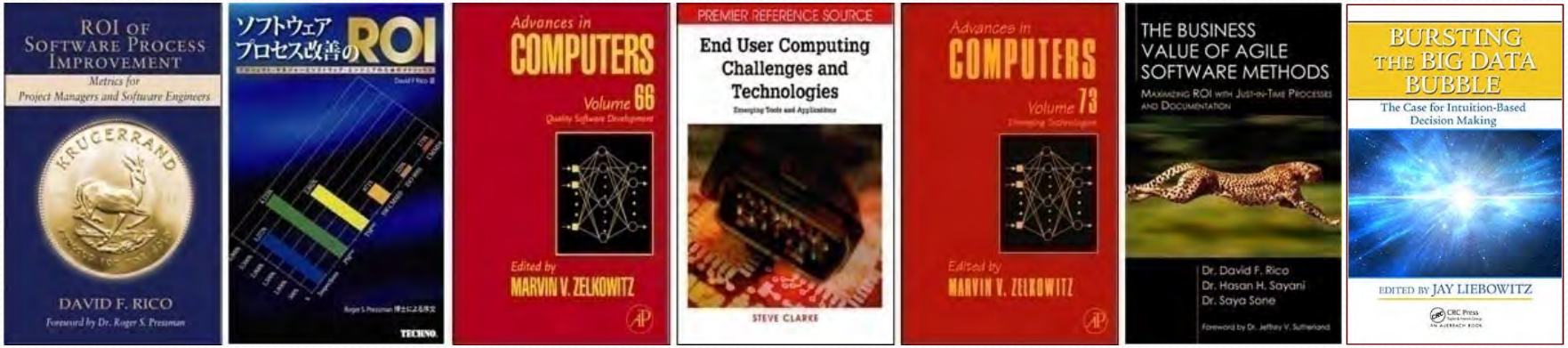
Dr. David F. Rico, PMP, CSEP, FCP, FCT, ACP, CSM, SAFE, DEVOPS

**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 Leadership Video**: http://www.youtube.com/watch?v=70LRzOk9VGY
**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/X22kJAvx44A
**DoD Fighter Jets versus Amazon Web Services**: http://davidfrico.com/dod-agile-principles.pdf

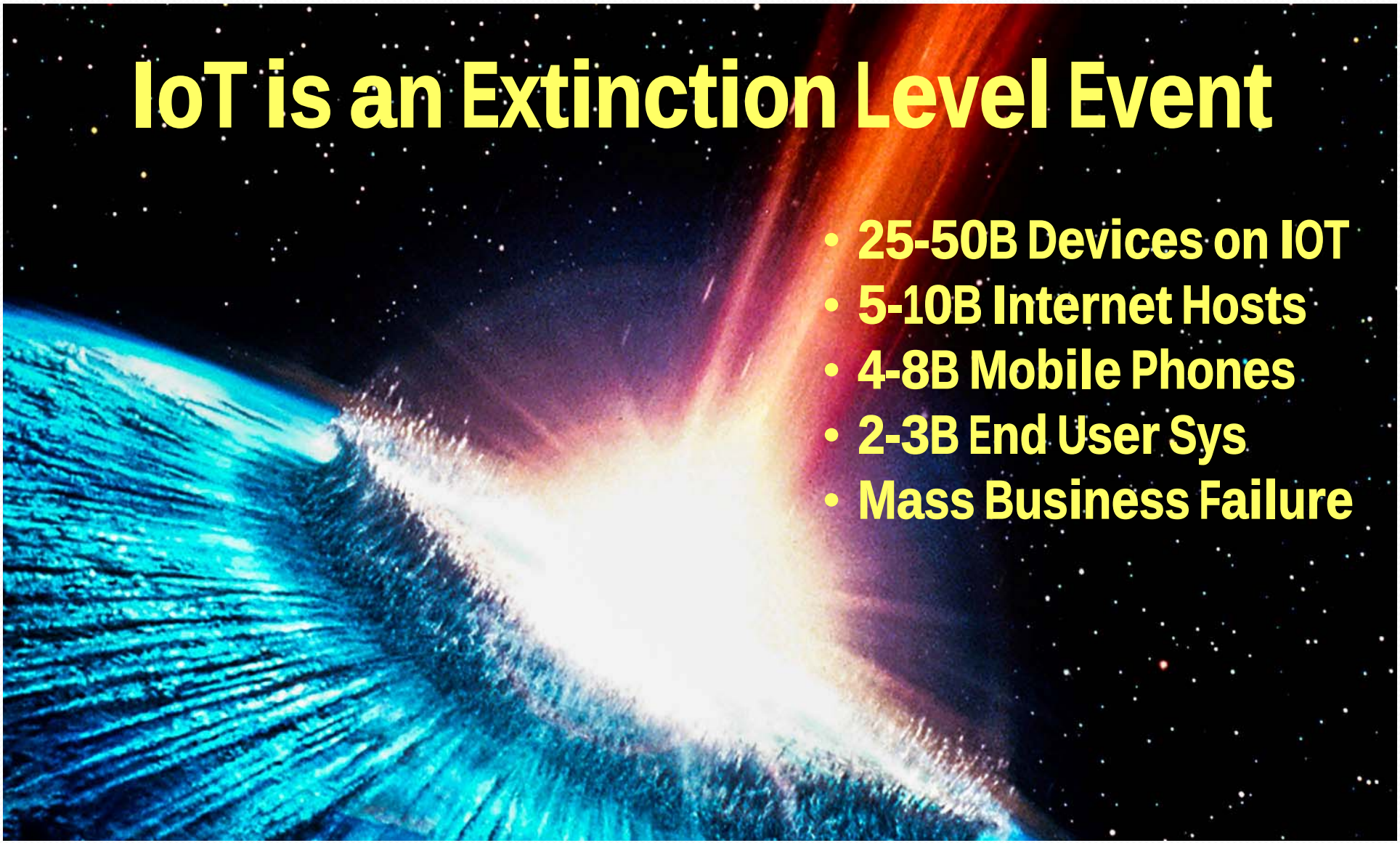
# Author Background

- Gov't contractor with 35+ 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

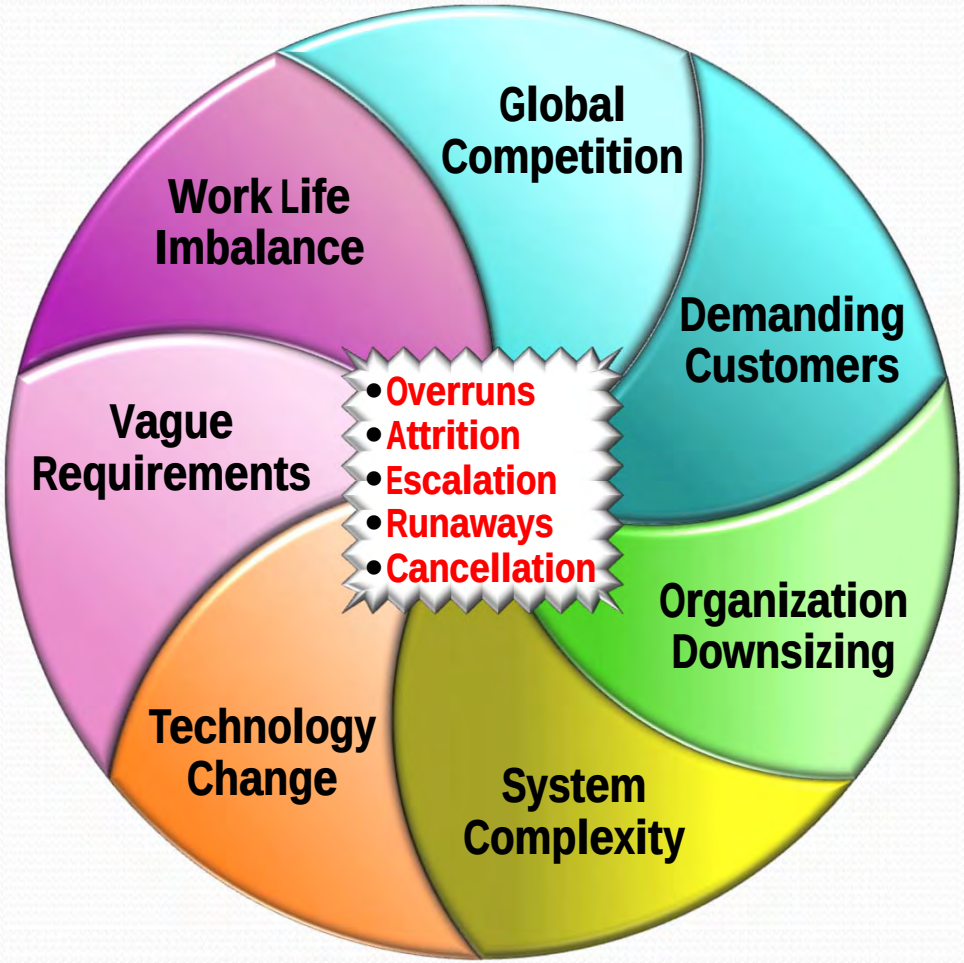


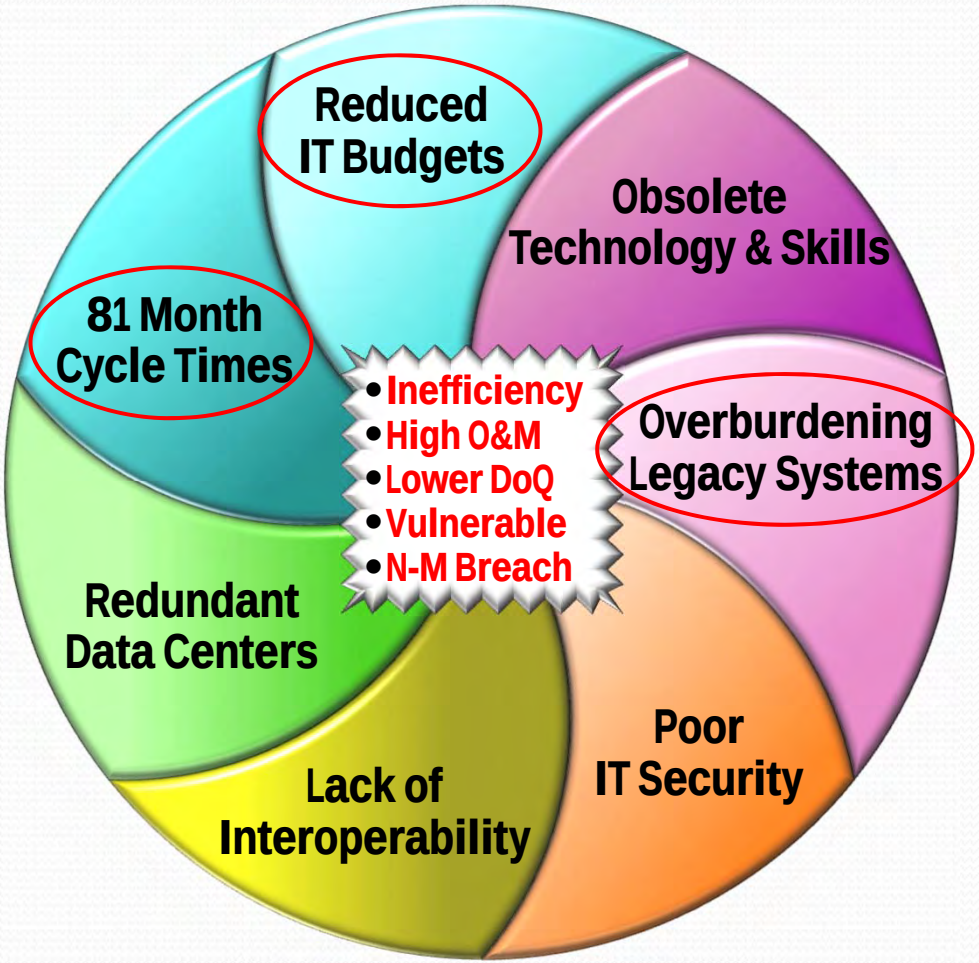

→ ***Career systems & software engineering methodologist***
→ *Lean-Agile, Six Sigma, CMMI, ISO 9001, DoD 5000*
→ *NASA, USAF, Navy, Army, DISA, & DARPA projects*
→ *Published seven books & numerous journal articles*
→ *Intn'l keynote speaker, 207+ talks to 20,000 people*
→ *Specializes in metrics, models, & cost engineering*
→ *Cloud Computing, SOA, Web Services, FOSS, etc.*
→ *Professor at 7 Washington, DC-area universities*

# Internet of Things—Dinosaur Killer



# Today's Whirlwind Environment



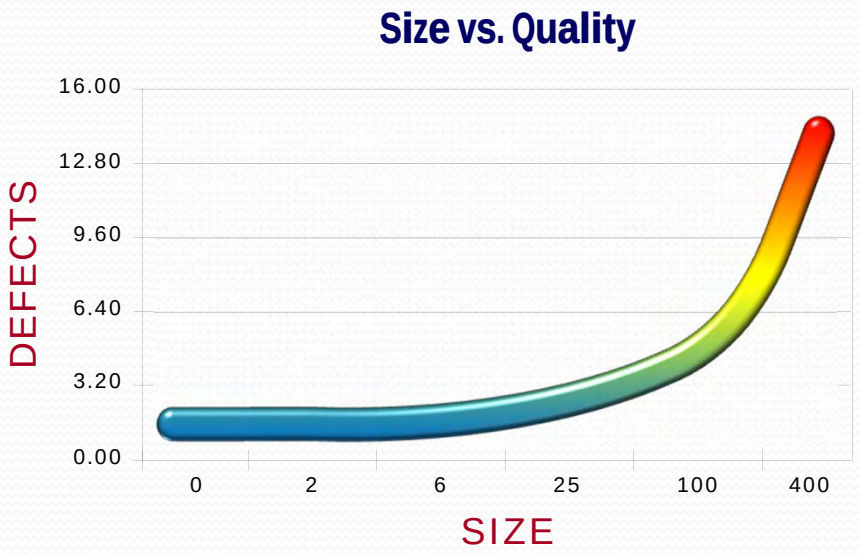





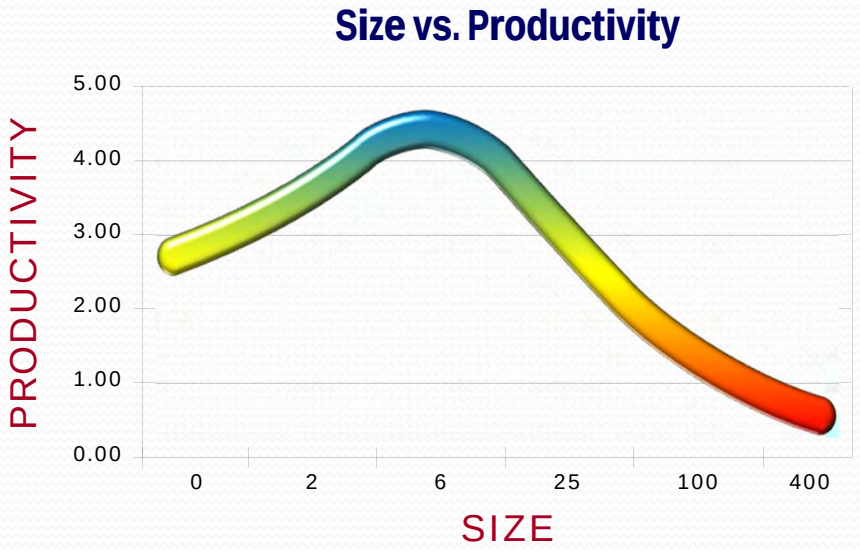
Pine, B. J. (1993). *Mass customization*: *The new frontier in business competition*. Boston, MA: Harvard Business School Press.

Pontius, R. W. (2012). Acquisition of IT: Improving efficiency and effectiveness in IT acquisition in the DoD. *Second Annual AFEI/NDIA Conference on Agile in DoD, Springfield, VA, USA.*

# Large TRADITIONAL Projects





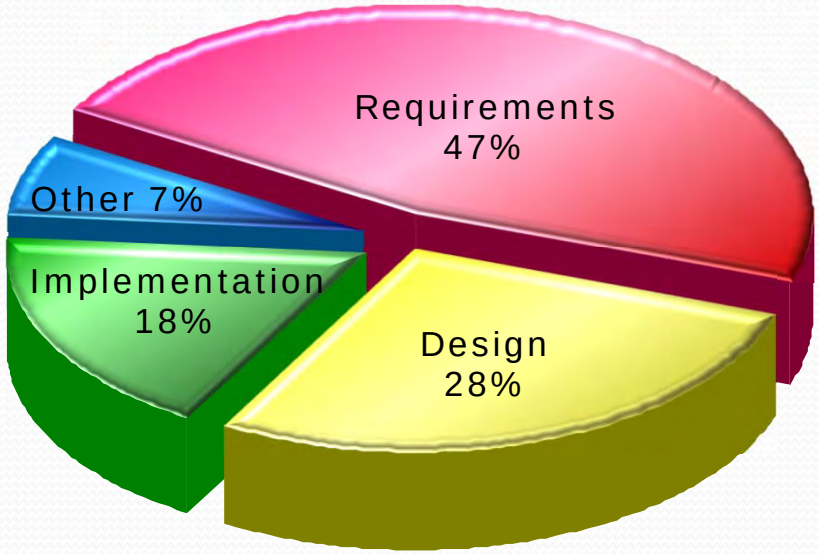




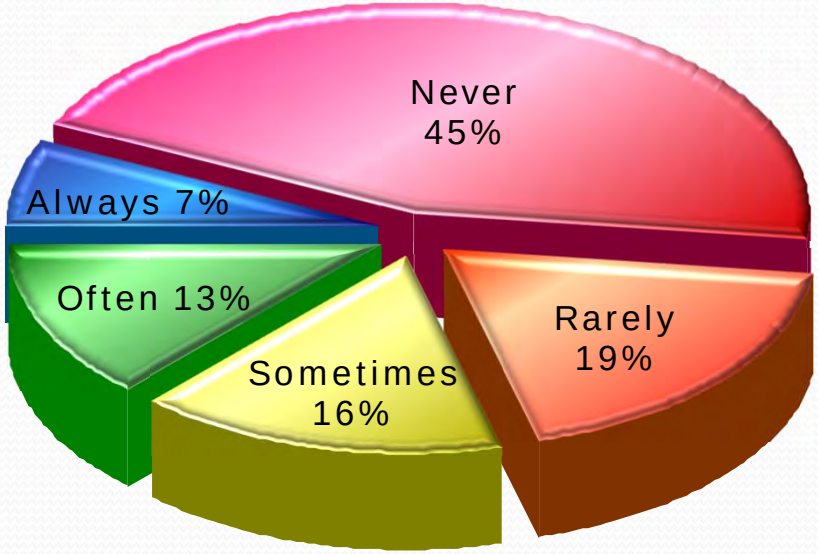
Jones, C. (1991). *Applied software measurement*: *Assuring productivity and quality*. New York, NY: McGraw-Hill.

# Large TRADITIONAL Projects—Cont'd

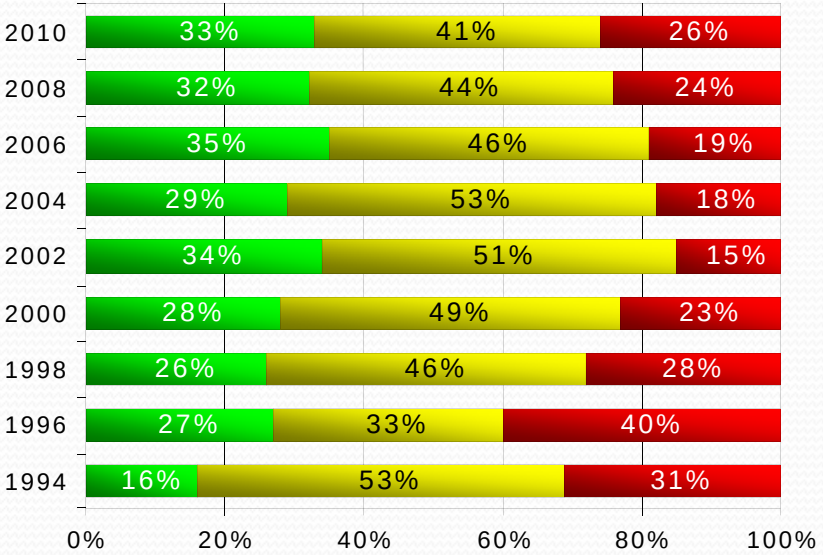
## DEFECTS



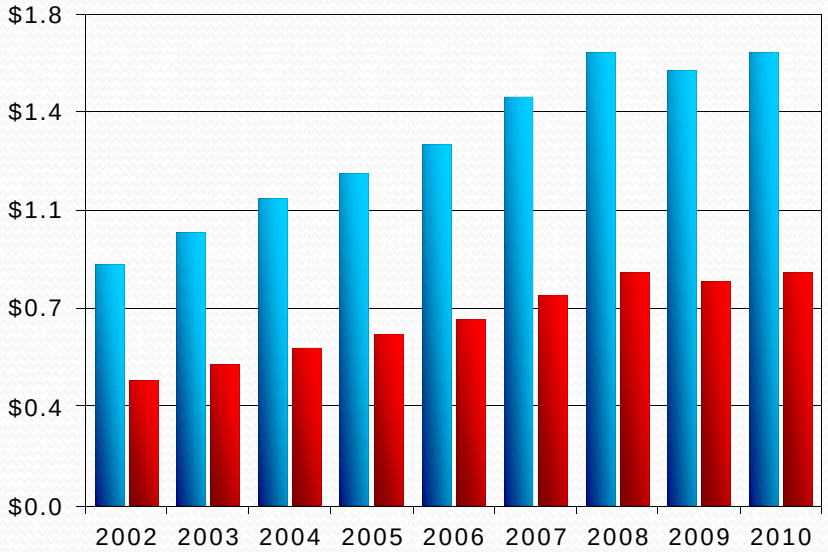

## WASTE




## IT PROJECT FAILURES




## GLOBAL IT PROJECT FAILURES



# What is Agility?

- A-gil-i-ty (ə-'ji-lə-tē) Property consisting of quickness, lightness, and ease of movement; To be very nimble
  - *The ability to create and respond to change in order to profit in a turbulent global business environment*
  - *The ability to quickly reprioritize use of resources when requirements, technology, and knowledge shift*
  - *A very fast response to sudden market changes and emerging threats by intensive customer interaction*
  - *Use of evolutionary, incremental, and iterative delivery to converge on an optimal customer solution*
  - *Maximizing* **BUSINESS VALUE** *with right sized, just-enough, and just-in-time processes and documentation*

Highsmith, J. A. (2002). *Agile software development ecosystems*. Boston, MA: Addison-Wesley.

# What are Agile Values?

- People-centric way to create innovative solutions
- Product-centric alternative to documents/process
- Market-centric model to maximize business value

**CUSTOMER COLLABORATION**
vs CONTRACT NEGOTIATION

**INDIVIDUALS & INTERACTIONS**
vs PROCESSES AND TOOLS

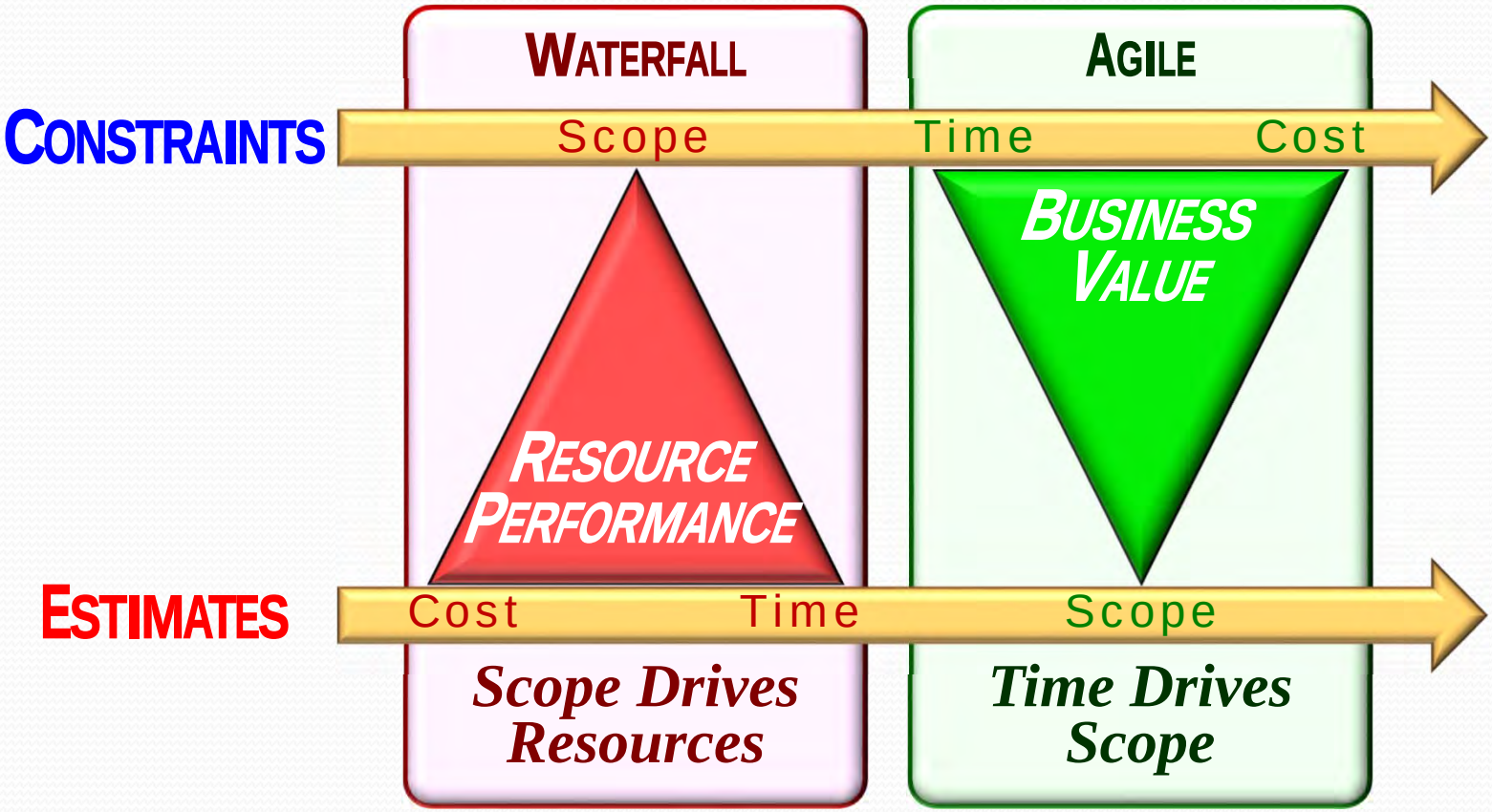
**WORKING PRODUCTS**
vs COMPREHENSIVE DOCUMENTATION

**RESPONDING TO CHANGE**
vs FOLLOWING A PLAN

Agile Manifesto. (2001). *Manifesto for agile software development*. Retrieved September 3, 2008, from http://www.agilemanifesto.org

# Agile GOLDILOCKS Zone

- Traditional project management is scope-based
- Agile project management is primarily time-based
- ☞ Early, iterative, & release of valuable features #1 job





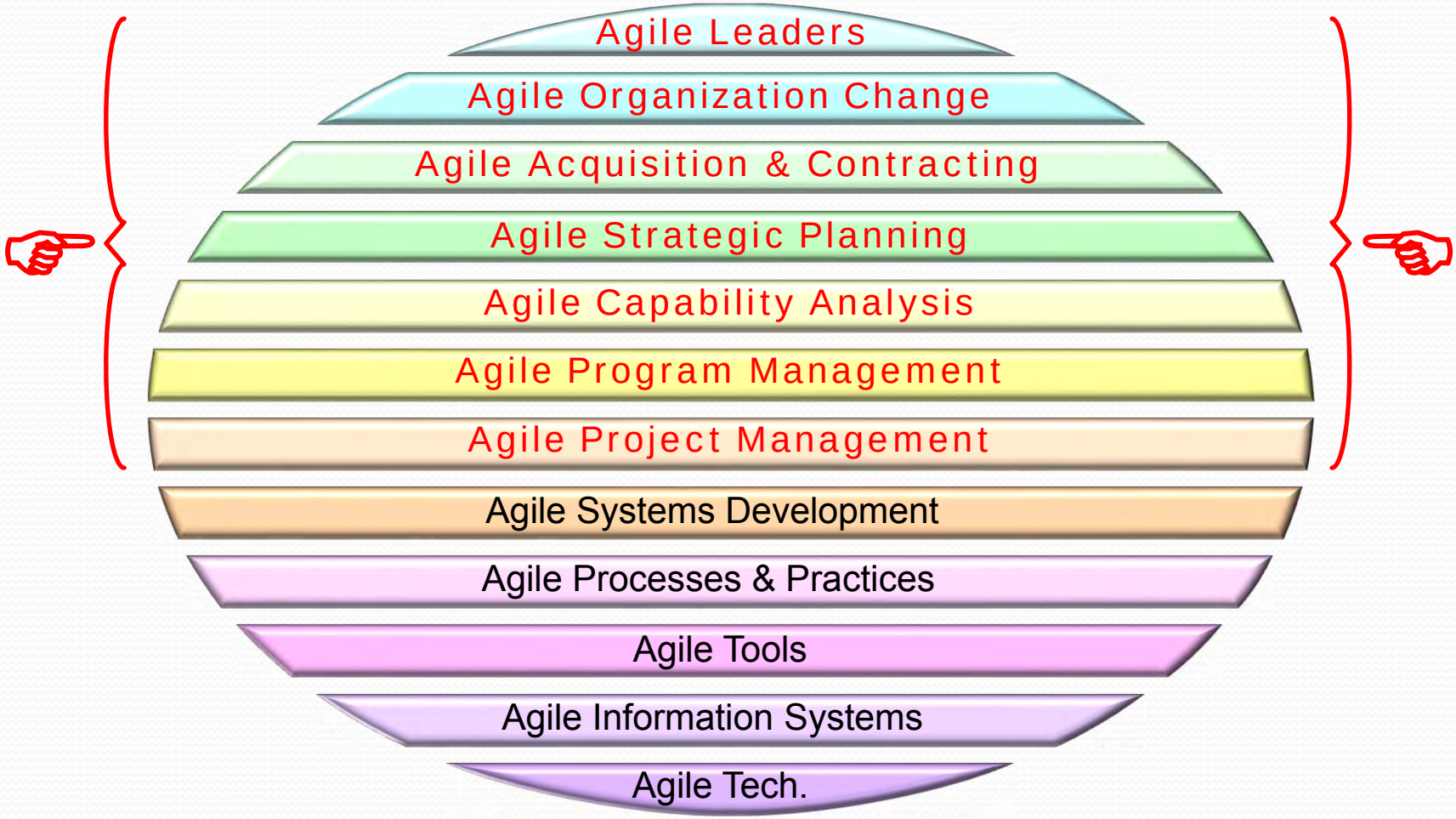
Rico, D. F. (2017). *Lean triangle: Triple constraints*. Retrieved December 17, 2017, from http://davidfrico.com/lean-triangle.pdf

Sylvester, T. (2013). *Waterfall, agile, and the triple constraint*. Retrieved December 16, 2017, from http://tom-sylvester.com/lean-agile/waterfall-agile-the-triple-constraint

Pound, E. S., Bell, J. H., Spearman, M. L. (2014). *Factory physics*: *How leaders improve performance in a post-lean six sigma world*. New York, NY: McGraw-Hill Education.

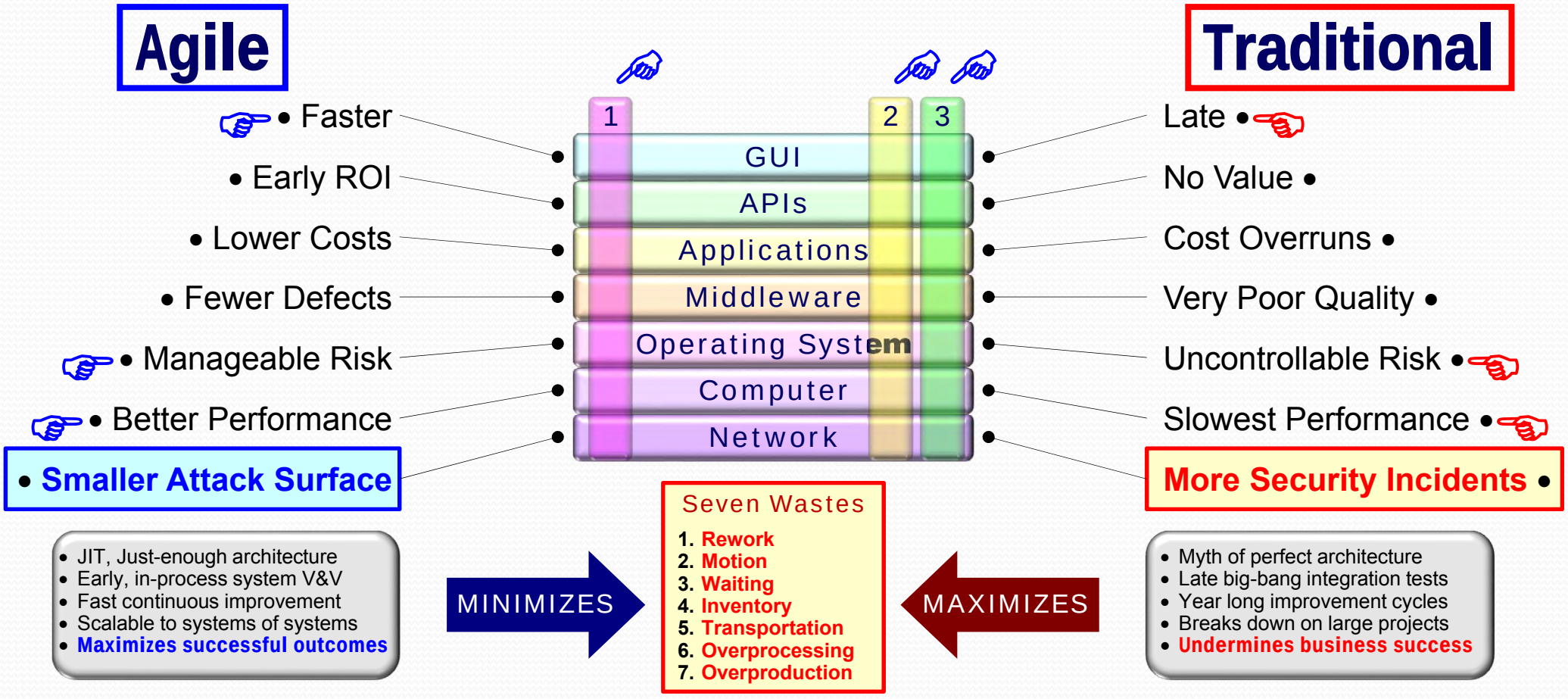
# Agile World View

- "Agility" has many dimensions other than IT
- It ranges from leadership to technological agility
- Today's focus is on organizational & enterprise agility



# Agile Methods—How they work?

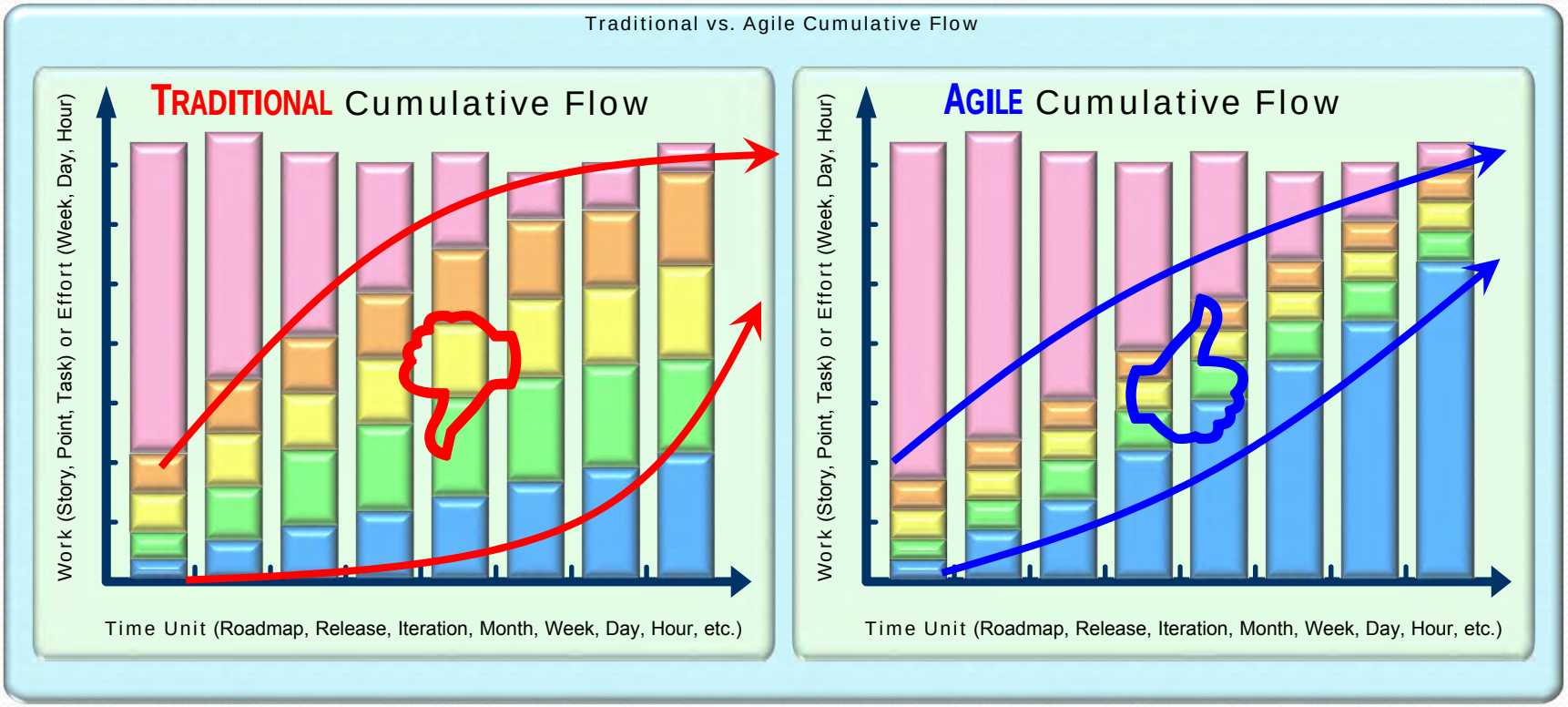
- Agile requirements implemented in slices vs. layers
- User needs with higher business value are done first
- Reduces cost & risk while increasing business success




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

# Agile Methods—Workflow Results

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

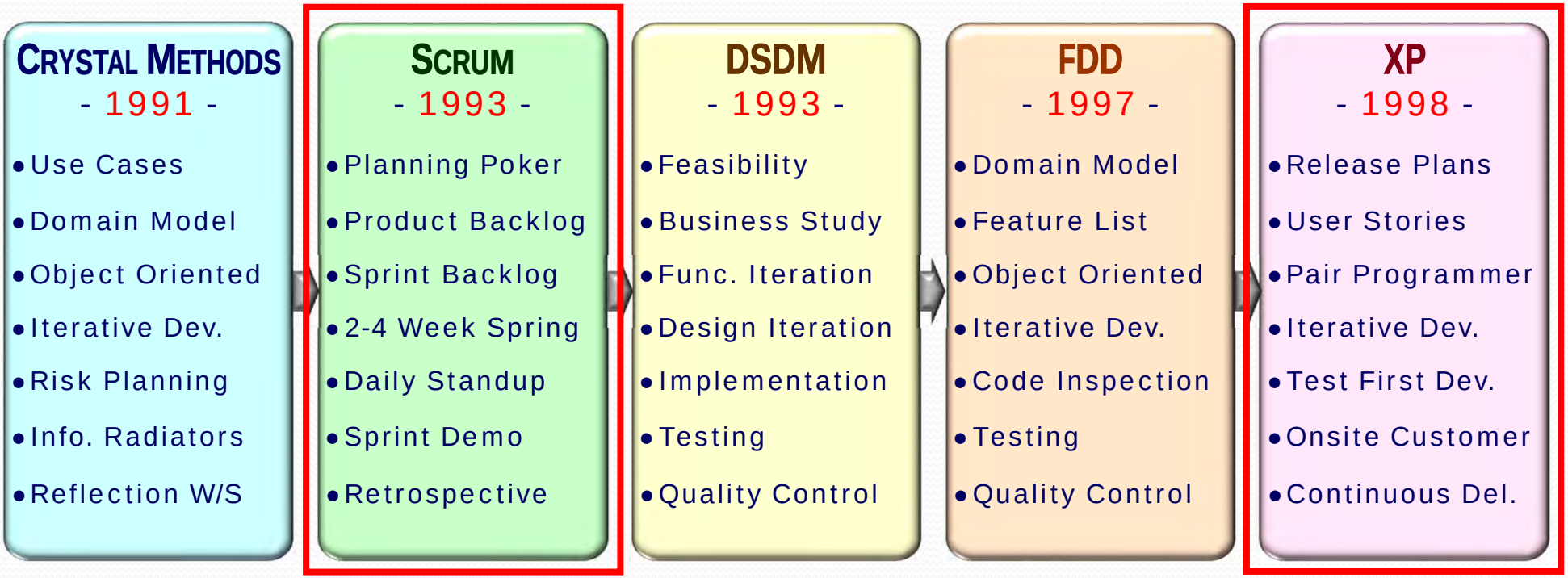




Anderson, D. J. (2004). *Agile management for software engineering*. Upper Saddle River, NJ: Pearson Education.
Anderson, D. J. (2010). *Kanban*: *Successful evolutionary change for your technology business*. Sequim, WA: Blue Hole Press.

# Models of Agile Development

- Agile methods spunoff flexible manufacturing 1990s
- Extreme Programming (XP) swept the globe by 2002
- Today, over 90% of IT projects use Scrum/XP hybrid

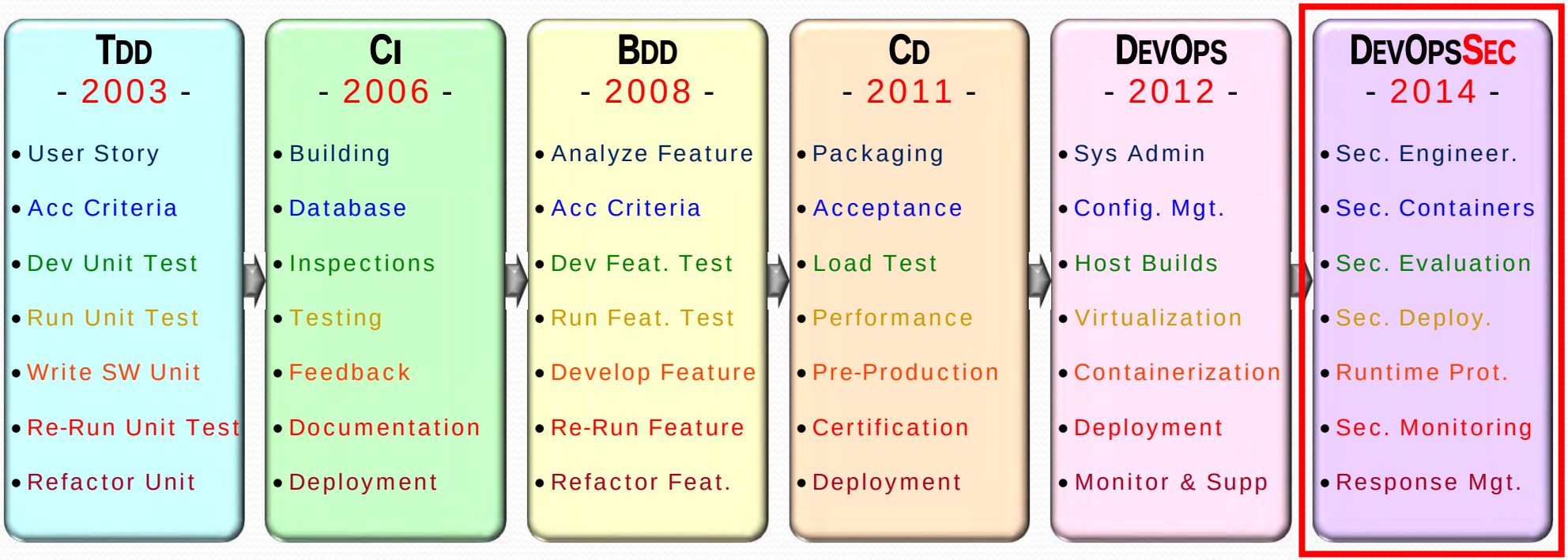




Cockburn, A. (2002). *Agile software development*. Boston, MA: Addison-Wesley.
Schwaber, K., & Beedle, M. (2001). *Agile software development with scrum*. Upper Saddle River, NJ: Prentice-Hall.
Stapleton, J. (1997). *DSDM: A framework for business centered development*. Harlow, England: Addison-Wesley.
Palmer, S. R., & Felsing, J. M. (2002). *A practical guide to feature driven development*. Upper Saddle River, NJ: Prentice-Hall.
Beck, K. (2000). *Extreme programming explained: Embrace change*. Reading, MA: Addison-Wesley.

# Models of Agile Delivery

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

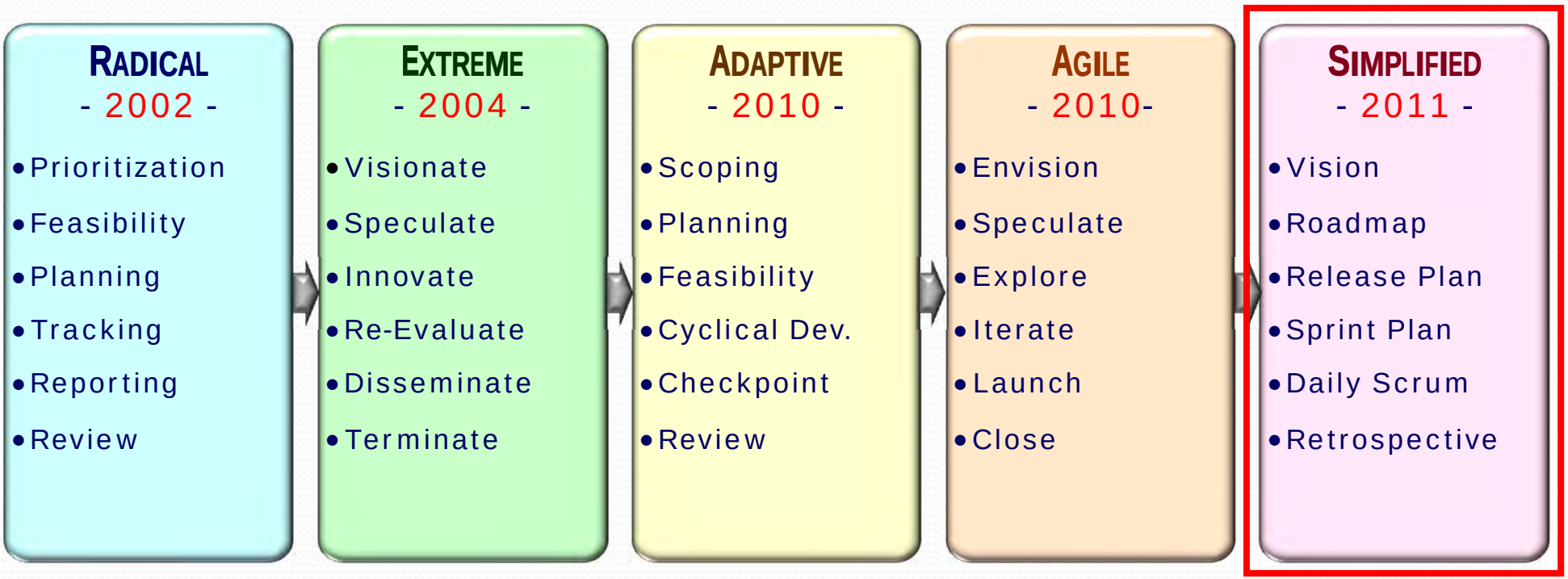




Beck, K. (2003). *Test-driven development*: *By example*. Boston, MA: Addison-Wesley.
Duvall, P., Matyas, S., & Glover, A. (2006). *Continuous integration*. Boston, MA: Addison-Wesley.
Barker, K., & Humphries, C. (2008). *Foundations of rspec*: *Behavior driven development with ruby and rails*. New York, 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.

# Models of Agile Project Mgt.

- Dozens of Agile project management models emerged
- Many stem from principles of Extreme Programming
- Vision, releases, & iterative development common

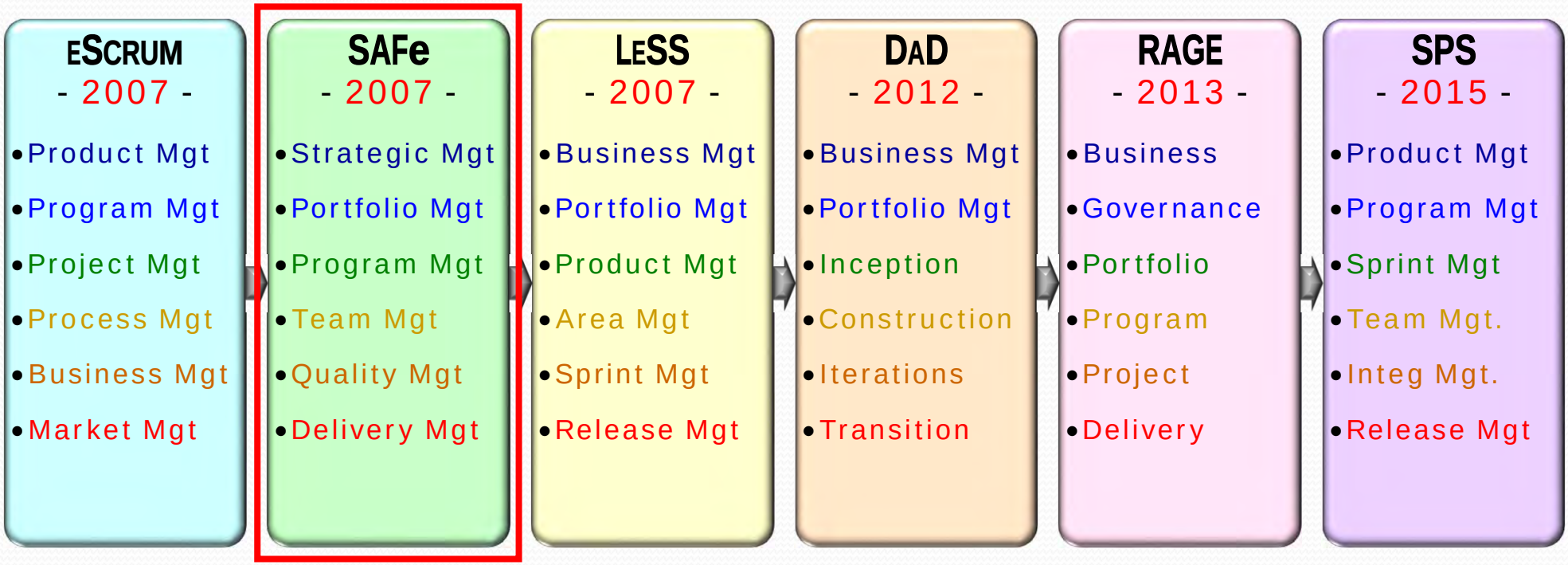




Thomsett, R. (2002). *Radical project management*. Upper Saddle River, NJ: Prentice-Hall.
DeCarlo, D. (2004). *Extreme project management*: *Using leadership, principles, and tools to deliver value in the face of volatility*. San Francisco, CA: Jossey-Bass.
Wysocki, R.F. (2010). *Adaptive project framework*: *Managing complexity in the face of uncertainty*. Boston, MA: Pearson Education.
Highsmith, J. A. (2010). *Agile project management*: *Creating innovative products*. Boston, MA: Pearson Education.
Layton, M. C., & Maurer, R. (2011). *Agile project management for dummies*. Hoboken, NJ: Wiley Publishing.

# Models of Agile Portfolio Mgt.

- Numerous models of agile portfolio mgt. emerging
- Based on lean-kanban, release planning, and Scrum
- Include organization, program, & project management

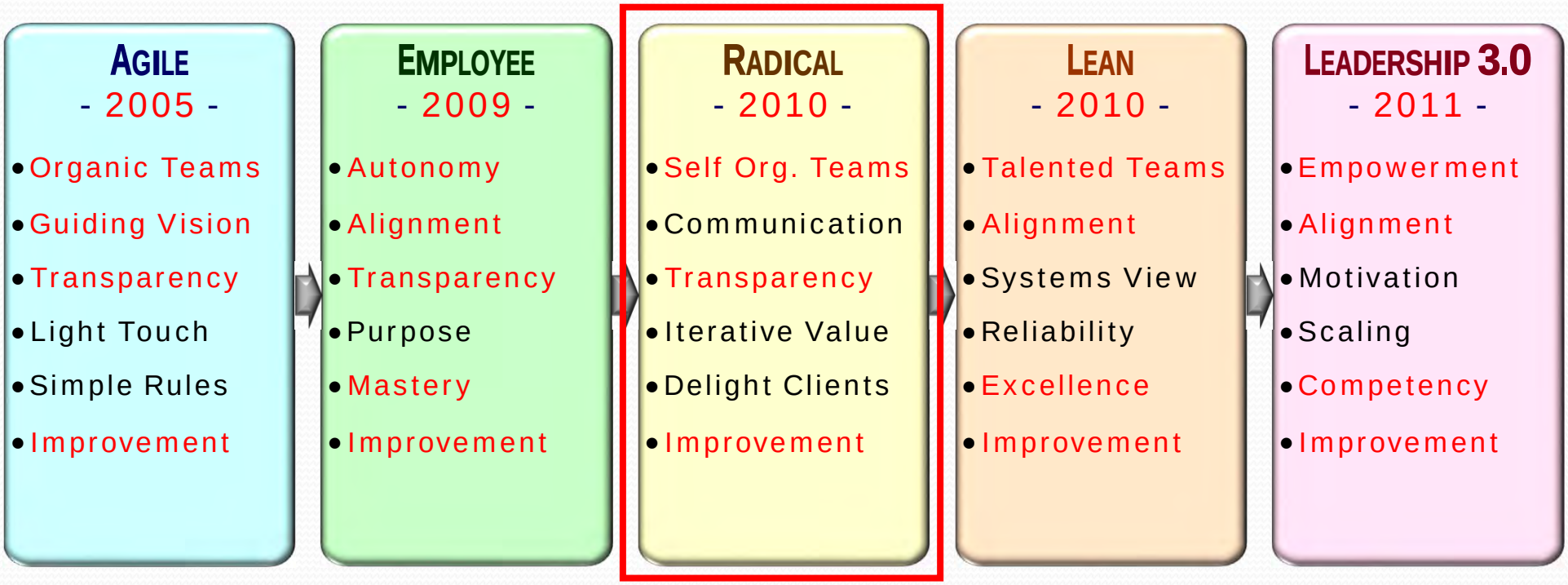




Schwaber, K. (2007). *The enterprise and scrum*. Redmond, WA: Microsoft Press.
Leffingwell, D. (2007). *Scaling software agility*: *Best practices for large enterprises*. Boston, MA: Pearson Education.
Larman, C., & Vodde, B. (2008). *Scaling lean and agile development*: *Thinking and organizational tools for large-scale scrum*. Boston, MA: Addison-Wesley.
Ambler, S. W., & Lines, M. (2012). *Disciplined agile delivery*: *A practitioner's guide to agile software delivery in the enterprise*. Boston, MA: Pearson Education.
Thompson, K. (2013). *cPrime's R.A.G.E. is unleashed*: *Agile leaders rejoice*! Retrieved March 28, 2014, from http://www.cprime.com/tag/agile-governance
Schwaber, K. (2015). *The definitive guide to nexus*: *The exoskeleton of scaled scrum development*. Lexington, MA: Scrum.Org

# Models of Agile Leadership

- Numerous theories of agile leadership have emerged
- Many have to do with delegation and empowerment
- Leaders have major roles in visioning and enabling

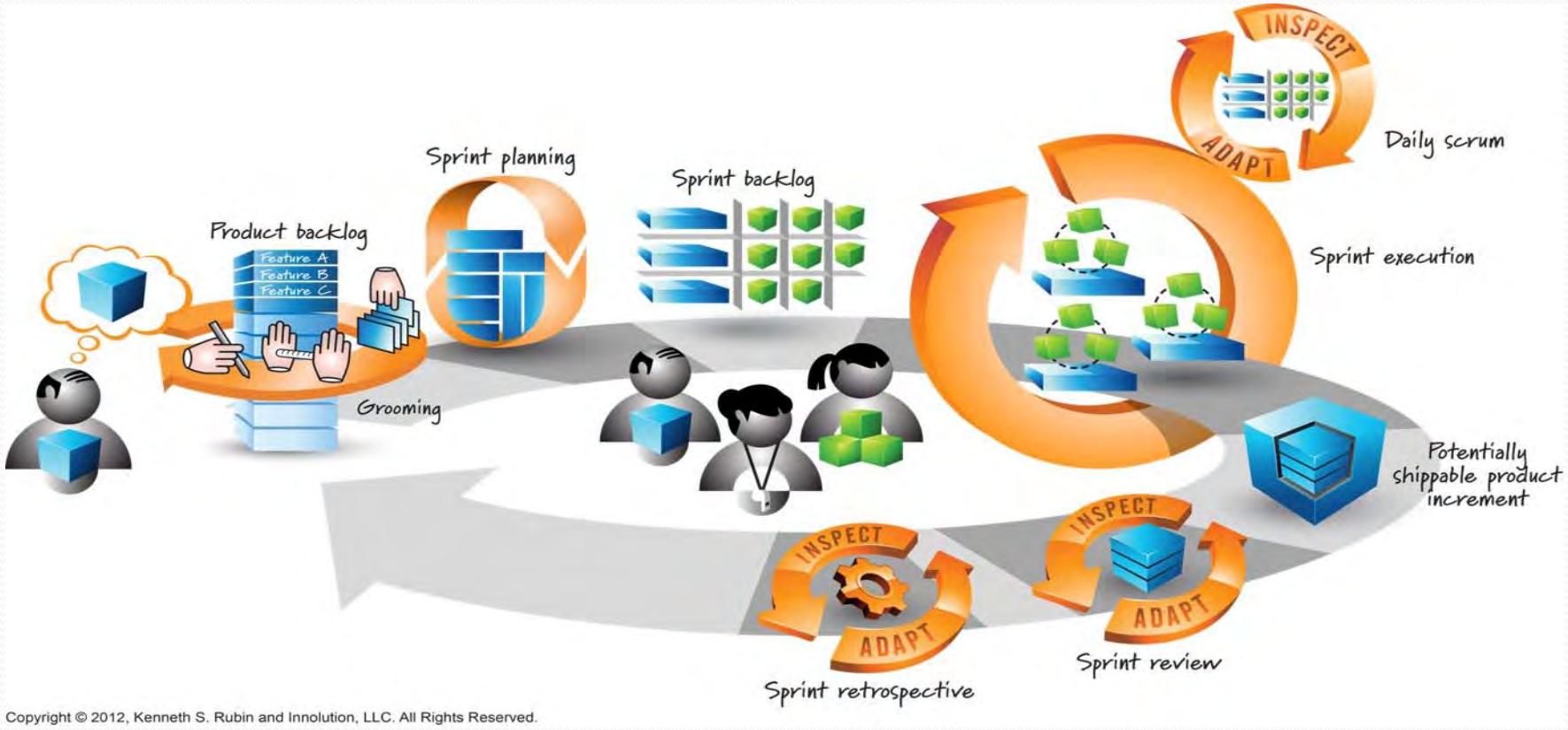




Augustine, S. (2005). *Managing agile projects*. Upper Saddle River, NJ: Pearson Education.
Pink, D. H. (2009). *Drive*: *The surprising truth about what motivates us*. New York, NY: Penguin Books.
Denning, S. (2010). *The leader's guide to radical management*: *Reinventing the workplace for the 21st century*. San Francisco, CA: John Wiley & Sons.
Poppendieck, M, & Poppendieck, T. (2010). *Leading lean software development*: *Results are not the point*. Boston, MA: Pearson Education.
Appelo, J. (2011). *Management 3.0*: *Leading agile developers and developing agile leaders*. Boston, MA: Pearson Education.

# Basic SCRUM Method

- Created by Jeff Sutherland at Easel in 1993
- Product backlog comprised of prioritized features
- Iterative sprint-to-sprint, adaptive & emergent model






Schwaber, K., & Beedle, M. (2001). *Agile software development with scrum*. Upper Saddle River, NJ: Prentice-Hall.

# Scrum—Sprint Planning

- Description. Specific iteration goals and tasks
- Owner. Product Owner and Development Team
- Frequency. At the start of each sprint [2-4 hours]

## Process Steps

1. Establish goals and choose user stories.
2. Decompose stories into tasks and create sprint backlog.

## Goals & User Stories

As a mobile banking customer, I want to create an account so I can write personal checks

- **Create account.**
- **Login to account.**
- **Setup checking account.**

## Sprint Backlog

| Task | Pri | Status | Who | App. | M | T | W | T | F |
|---|---|---|---|---|---|---|---|---|---|
| • Create account: | | | | | | | | | |
| – Setup | 1 | Done | Sue | Joe | 4 | 4 | 0 | 0 | 0 |
| – Install | 2 | Done | Sue | Joe | 4 | 4 | 0 | 0 | 0 |
| – Schema | 3 | Done | John | Joe | 0 | 0 | 8 | 0 | 0 |
| – Queries | 4 | In-work | Bob | - | 0 | 0 | 0 | 8 | 0 |
| – Forms | 5 | N/S | Patty | - | 0 | 0 | 0 | 0 | 0 |
| – Test | 6 | N/S | Sam | - | 0 | 0 | 0 | 0 | 0 |

Product owner, Scrum Master, and Developers create sprint plan. ***Sprint planning done at start of sprint***. ***Product backlog must be ready***. ***Developers select sprint goal and what can be done***.

# Scrum—Daily Standup

- Description. Establish & coordinate daily priorities
- Owner. Development Team
- Frequency. Daily [15-minutes]

**Process Steps**

1. Hold daily standup meeting.
2. Update sprint burndown chart.
3. Perform design, development, test, and evaluation.

**Daily Standup**

All Developers on Team Answer Three Questions in Round-Robin Style

- What has been done since the last meeting?
- What will be done before the next meeting?
- What obstacles are in my way?

**Sprint Burndown**


Developers hold daily standup meetings. ***Purpose is to coordinate daily priorities***. ***Identify what was done, what will be done, and impediments***. ***Task boards and Sprint burndown are updated***.

# Scrum—Sprint Review

- Description. Demonstration of working product
- Owner. Product Owner and Development Team
- Frequency. At the end of each sprint [2-4 hours]

**Process Steps**

1. Prepare sprint review meeting.
2. Hold sprint review meeting.
3. Collect feedback from stakeholders.

**Product Demonstration**

Developers Perform a Live Demo on Target Hardware and Answer Stakeholder Questions

- What was the goal of the sprint?
- What user stories were attempted?
- What user stories were implemented?

**Stakeholder Feedback**

Poll Stakeholders One-by-One in Round-Robin Style to Solicit their Feedback

- Is the product acceptable as implemented?
- Is the product acceptable with modifications?
- Is the product unacceptable as implemented?

Developers hold a sprint review. ***Sprint review performed at end of sprint***. ***Developers demo validated code to stakeholders***. ***Stakeholders vote on demo outcome. Product backlog reprioritized***.

# Scrum—Sprint Retrospective

- Description. Refine environment and processes
- Owner. Development Team
- Frequency. At the end of each sprint [1-2 hours]

**Process Steps**

1. Plan sprint retrospective meeting.
2. Hold sprint retrospective meeting.
3. Inspect and adapt.

**Sprint Retrospective**

Poll Developers on Team to Answer Three Questions to Reach Group Consensus

- **What went well in the last sprint?**
- **What could be improved in the next sprint?**
- **What people, process, and tools should change?**

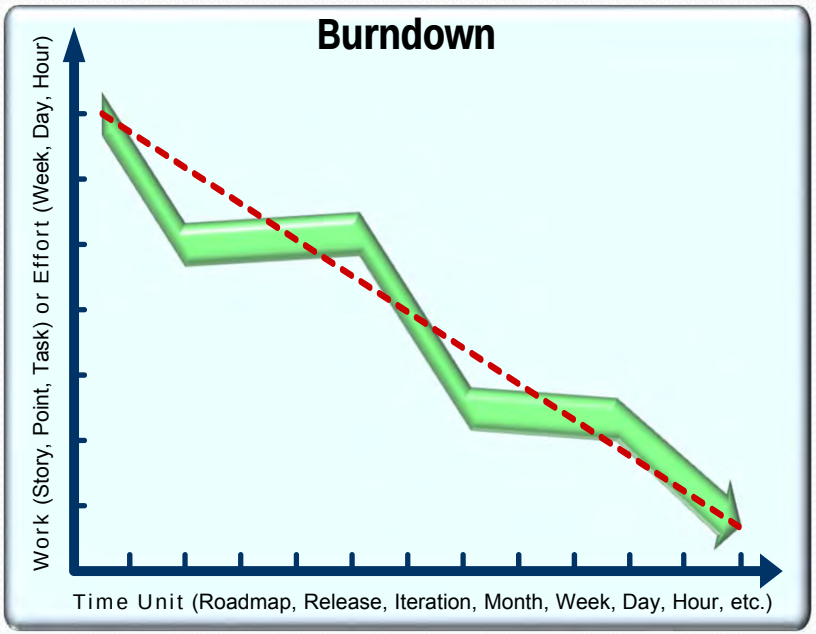
**Process Improvements**

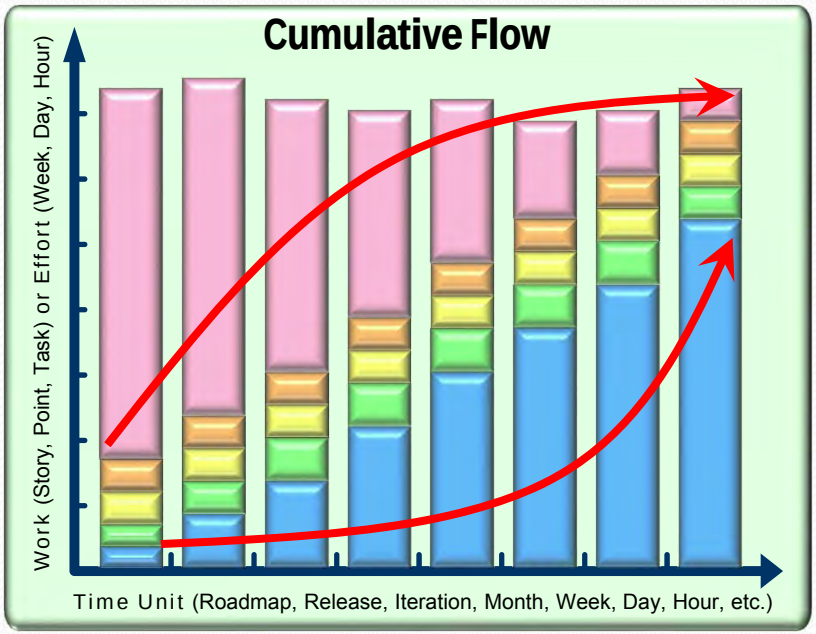
Scrum Master Records Action Items and Prepares Process Improvement Plan

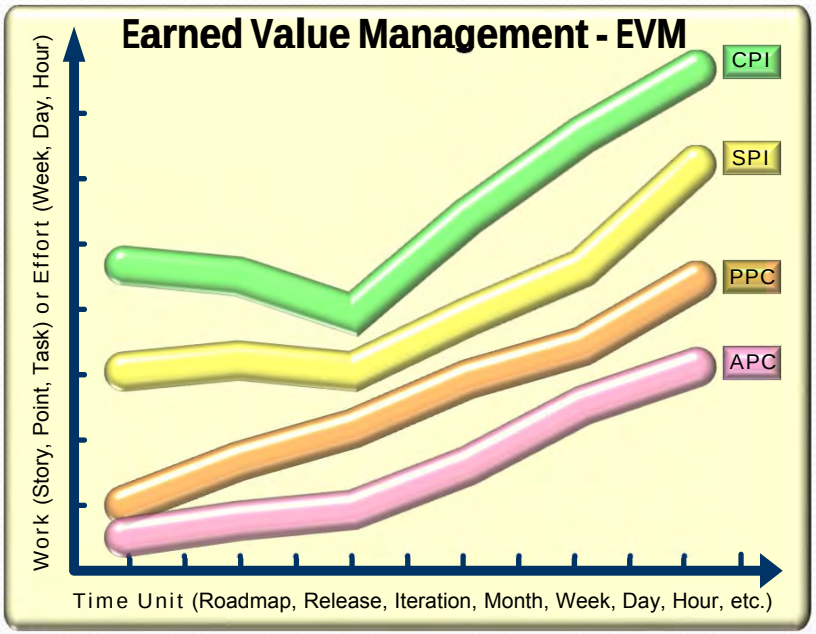
- **Scrum master records suggested improvements.**
- **Developers prioritize suggested improvements.**
- **Add high-priority non-functional items to backlog.**

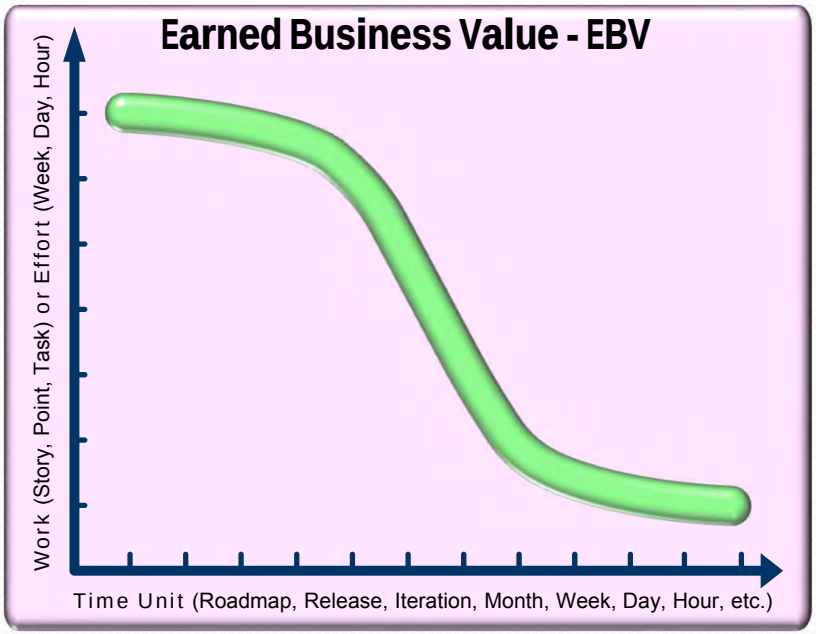
Developers hold sprint retrospective. ***Retrospective held at end of sprint. Developers identify the good and bad. Scrum master records results. Processes, tools, and backlog may be adjusted.***

# Agile Methods—Basic Metrics



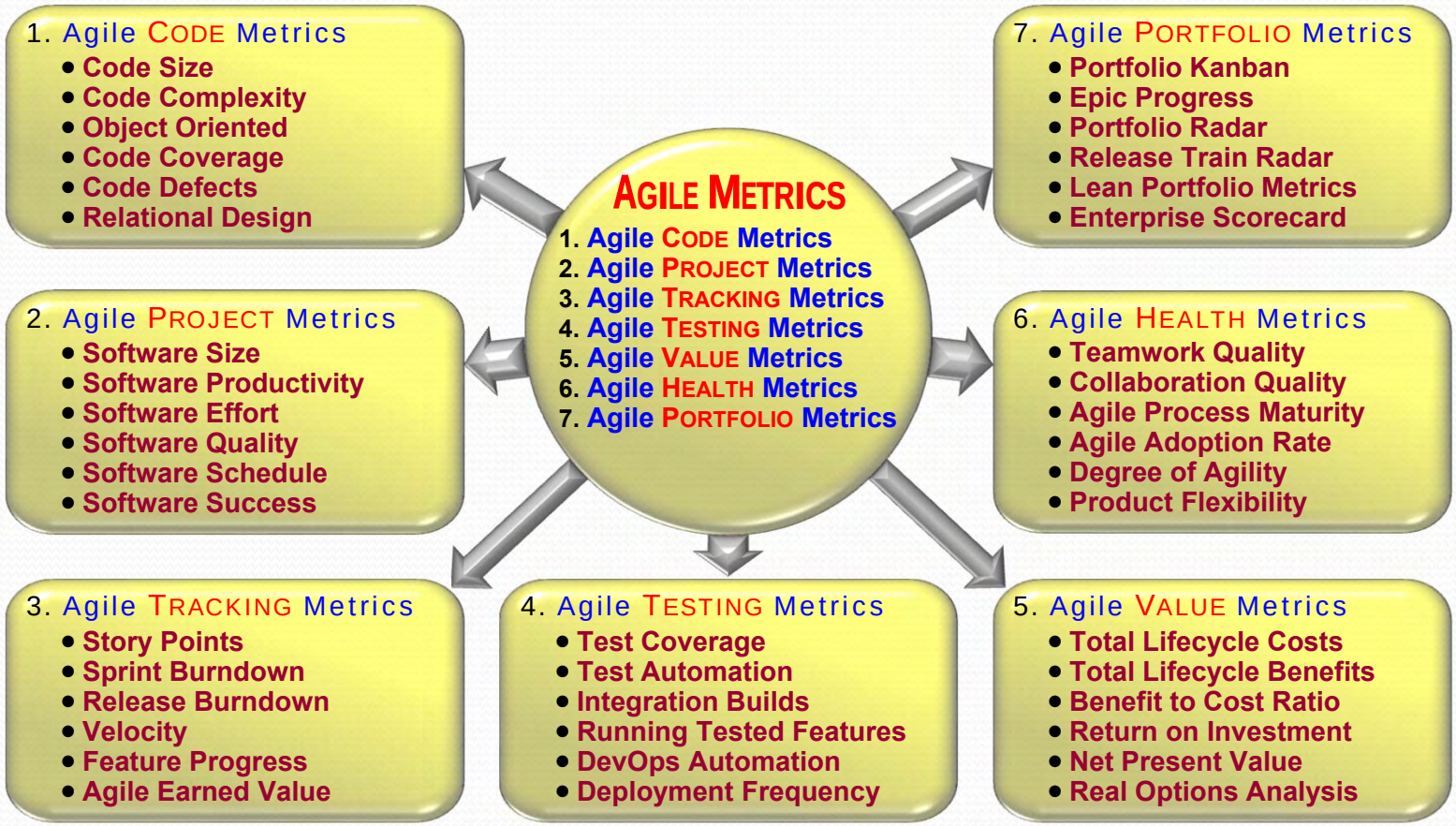



# Agile Methods—Metrics Taxonomy

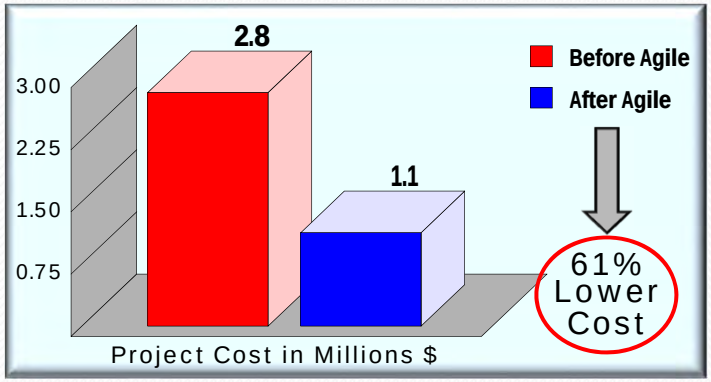
- Agile methods are based on traditional measures
- Story points, velocity, and burndown basic metrics
- Experts use Agile EVM, test, ROI & portfolio metrics

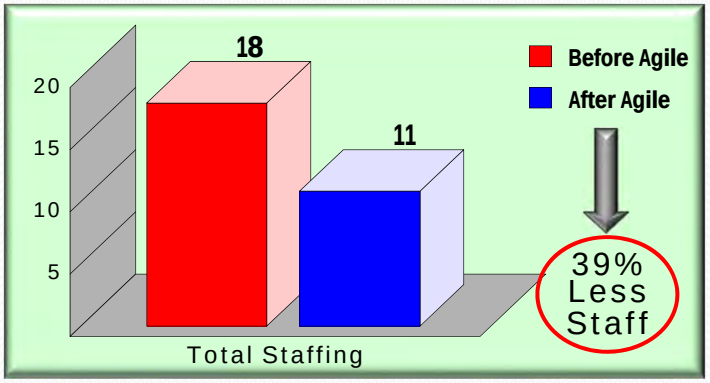


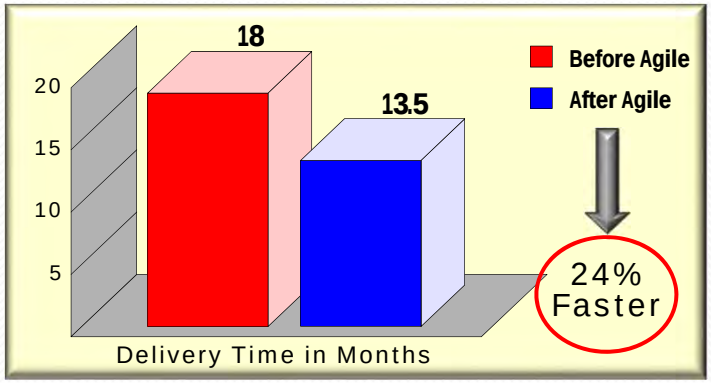

Rico, D. F., Sayani, H. H., & Sone, S. (2009). *The business value of agile software methods*. Ft. Lauderdale, FL: J. Ross Publishing.

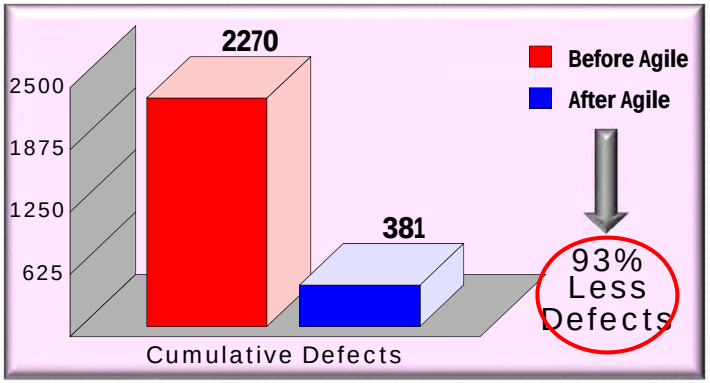
# Agile Methods—Costs & Benefits

- Analysis of 23 agile vs. 7,500 traditional projects
- Agile projects are 54% better than traditional ones
- Agile has lower costs (61%) and fewer defects (93%)




Mah, M. (2008). Measuring agile in the enterprise: *Proceedings of the Agile 2008 Conference, Toronto, Canada.*

# Agile Methods—Return on Invest.

- Costs based on avg. productivity and quality
- Productivity ranged from 4.7 to 5.9 LOC an hour
- Costs were $588,202 and benefits were $3,930,631

| Metric | Formula | Trad. Testing | Agile Testing |
|---|---|---|---|
| Costs | (10,000 ÷ **5.4436** + **3.945** × 10 × 100) × 100 | **$588,202** | **$233,152** |
| Benefits | (10,000 × 10.51 − 6,666.67 × 9) × 100 − **$588,202** | **$3,930,631** | **$4,275,681** |
| B/CR | **$3,930,631** ÷ **$588,202** | **7:1** | **18:1** |
| ROI | (**$3,930,631** − **$588,202**) ÷ **$588,202** × 100% | **567%** | **1,734%** |
| NPV | $(\sum_{i=1}^{5}$ (**$3,930,631** ÷ 5) ÷ $1.05^5$) − **$588,202** | $2,806,654 | $3,469,140 |
| BEP | **$588,202** ÷ ($4,509,997 ÷ **$588,202** − 1) | $88,220 | $12,710 |
| ROA | NORMSDIST(**2.24**) × **$3,930,631** − NORMSDIST(**0.85**) × **$588,202** × EXP(−5% × 5) | $3,504,292 | $4,098,159 |

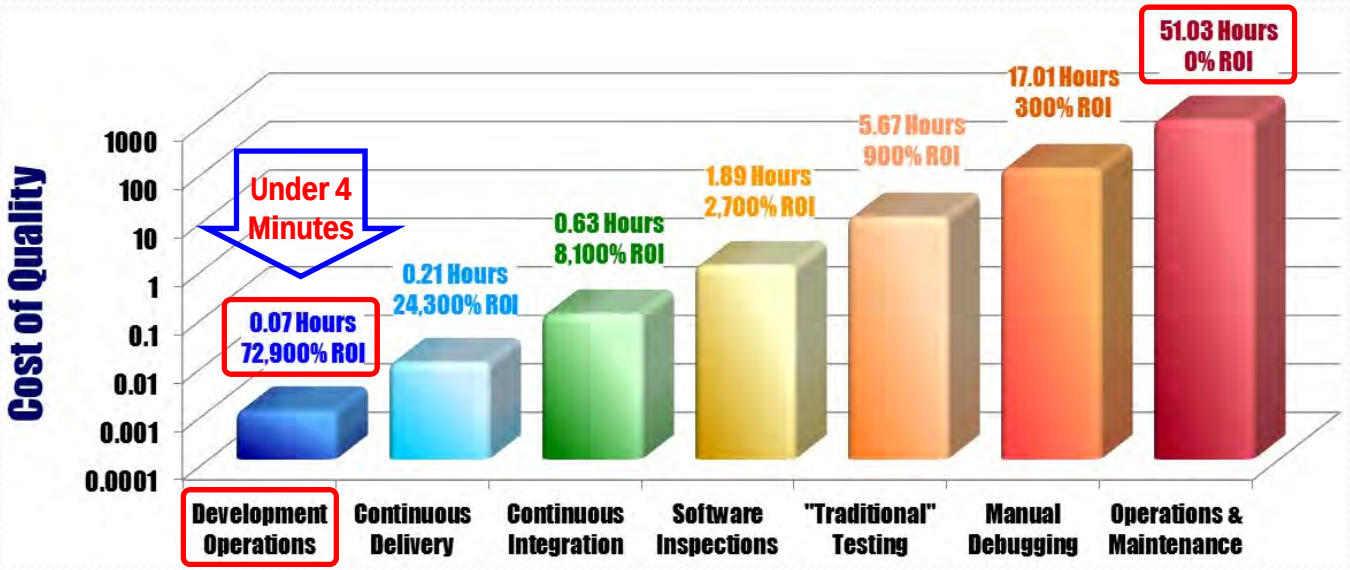
$d1 = [\ln(Benefits \div Costs) + (Rate + 0.5 \times Risk^2) \times Years] \div Risk \times \sqrt{Years}$, $d2 = d1 - Risk \times \sqrt{Years}$

Rico, D. F., Sayani, H. H., & Sone, S. (2009). *The business value of agile software methods: Maximizing ROI with just-in-time processes and documentation*. Ft. Lauderdale, FL: J. Ross Publishing.

# Agile Methods—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 |




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

# Agile Methods—HP Case Study

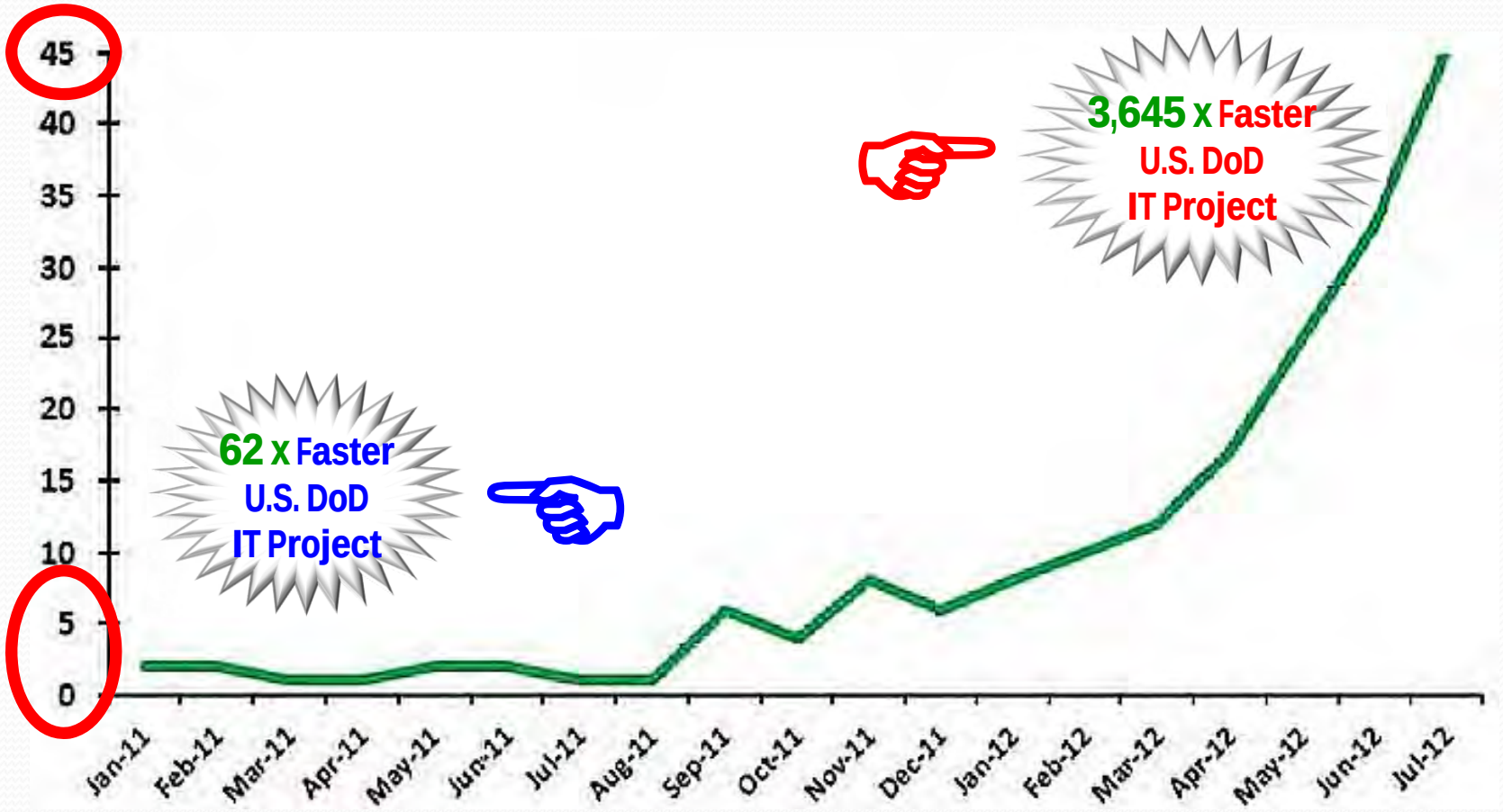
- Hewlett-Packard is a major user of CI, CD, & DevOps
- 400 engineers developed 10 million LOC in 4 years
- Major gains in testing, deployment, & innovation

| 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.

# Agile Methods—Dot Com Cases

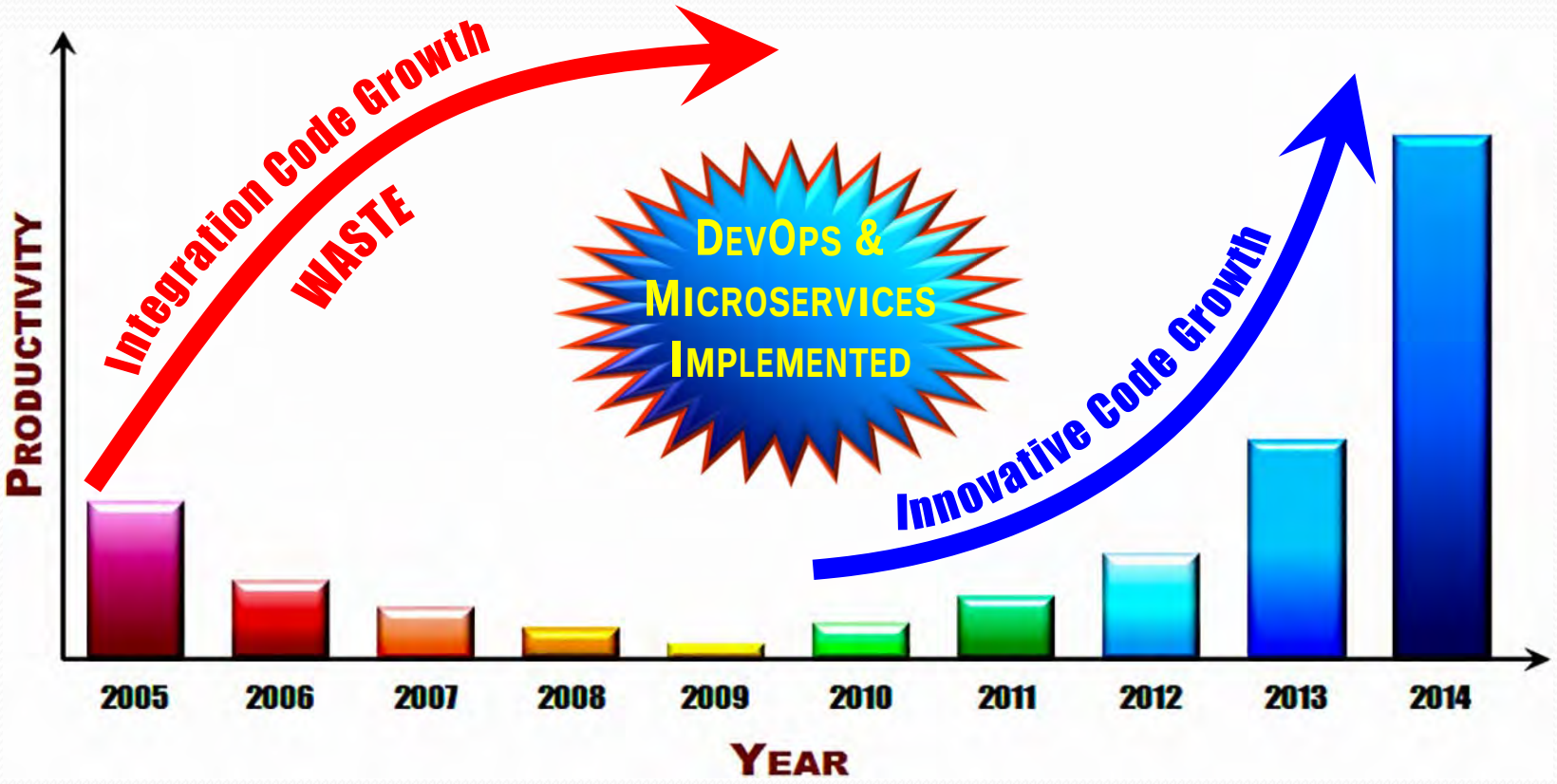
- Assembla went from 2 to 45 releases every month
- 15K Google developers run 120 million tests per day
- 30K+ Amazon developers deliver 136K releases a day




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

# Agile Methods—Blackboard Case

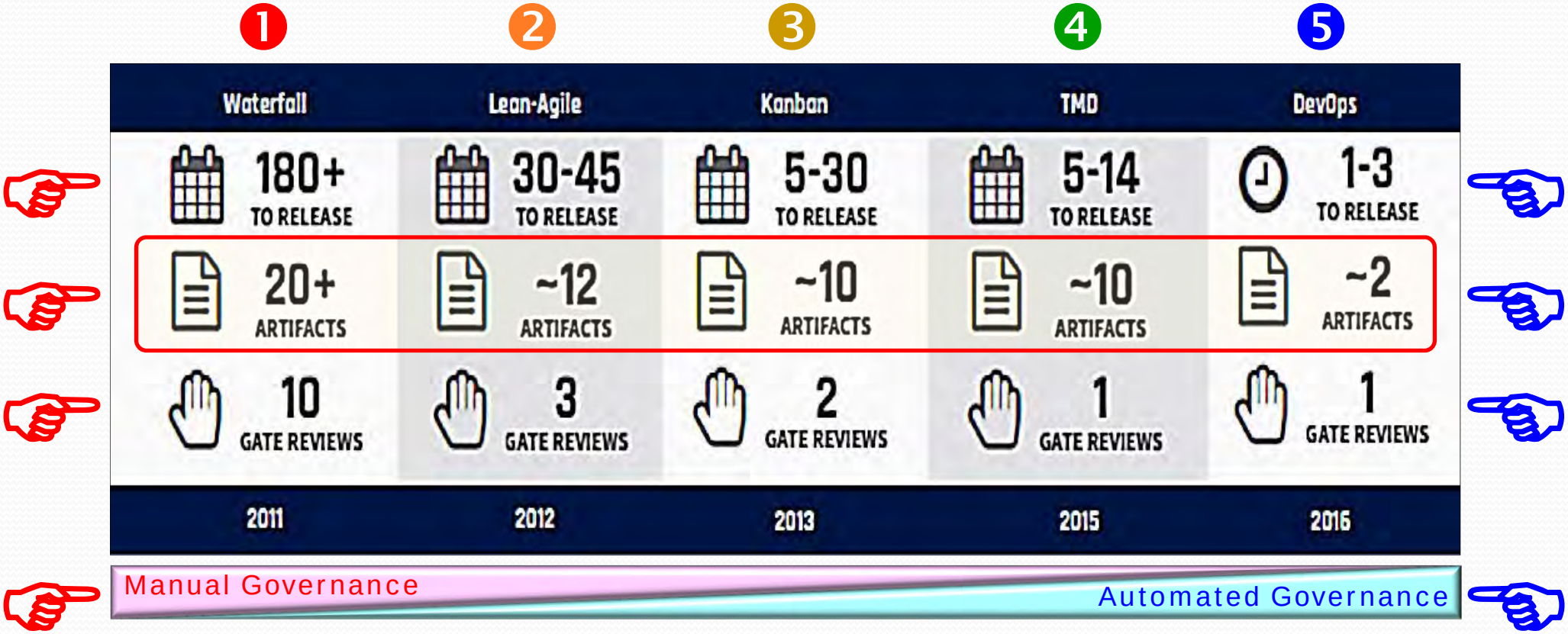
- Productivity **STOPS** due to excessive integration
- Implements DevOps & Microservices around 2010
- ☞ Waste elimination, productivity & innovation skyrocket




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

# Agile Methods—U.S. DHS Case

- 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.

# Agile Methods—Enterprise ROI

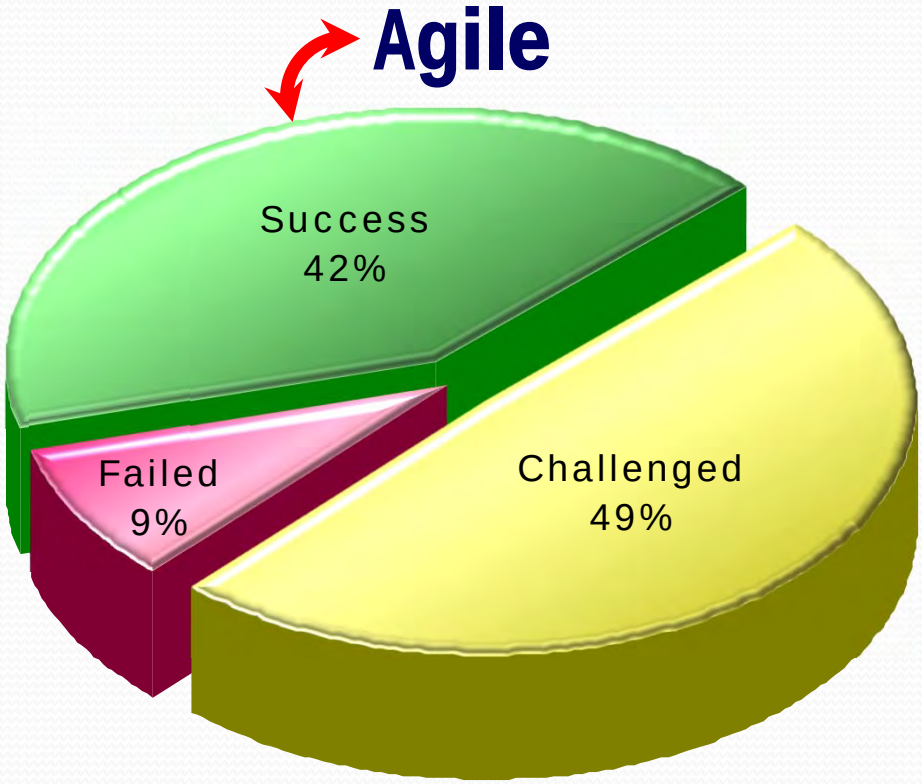
- Detailed DevOps economics starting to emerge
- ROI ranges from $17M to $195M *with minor costs*
- Benefits from cost savings, revenue, and availability

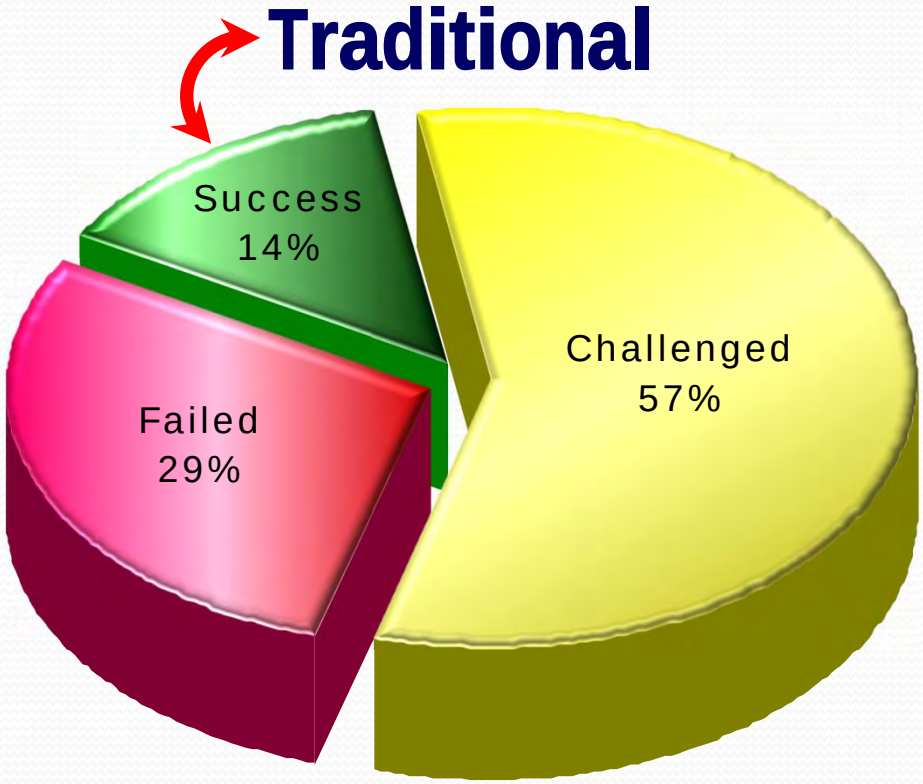
| Org | Low Perf | Med Perf | High Perf |
|---|---|---|---|
| **Small - 250 -** | $23M Benefits | $29M Benefits | $17M Benefits |
| | $0.2M Costs | $0.2M Costs | $0.2M Costs |
| | **13,589% ROI** | **17,799% ROI** | **9,932% ROI** |
| | *3 Day Payback* | *2 Day Payback* | *4 Day Payback* |
| **Medium - 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

# Agile Methods—Success Rate

- Traditional projects succeed at 50% industry avg.
- Traditional projects are challenged 20% more often
- Agile projects succeed 3x more and fail 3x less often

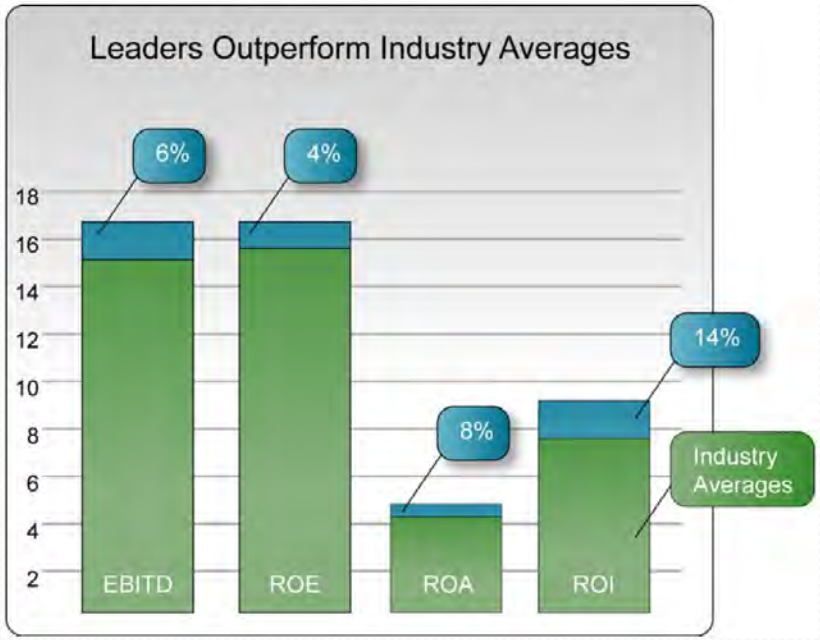



Standish Group. (2012). *Chaos manifesto*. Boston, MA: Author.

# Agile Methods—Business Benefits

- Study of 15 agile vs. non-agile Fortune 500 firms
- Based on models to measure organizational agility
- ☞ Agile firms out perform non agile firms by up to 36%

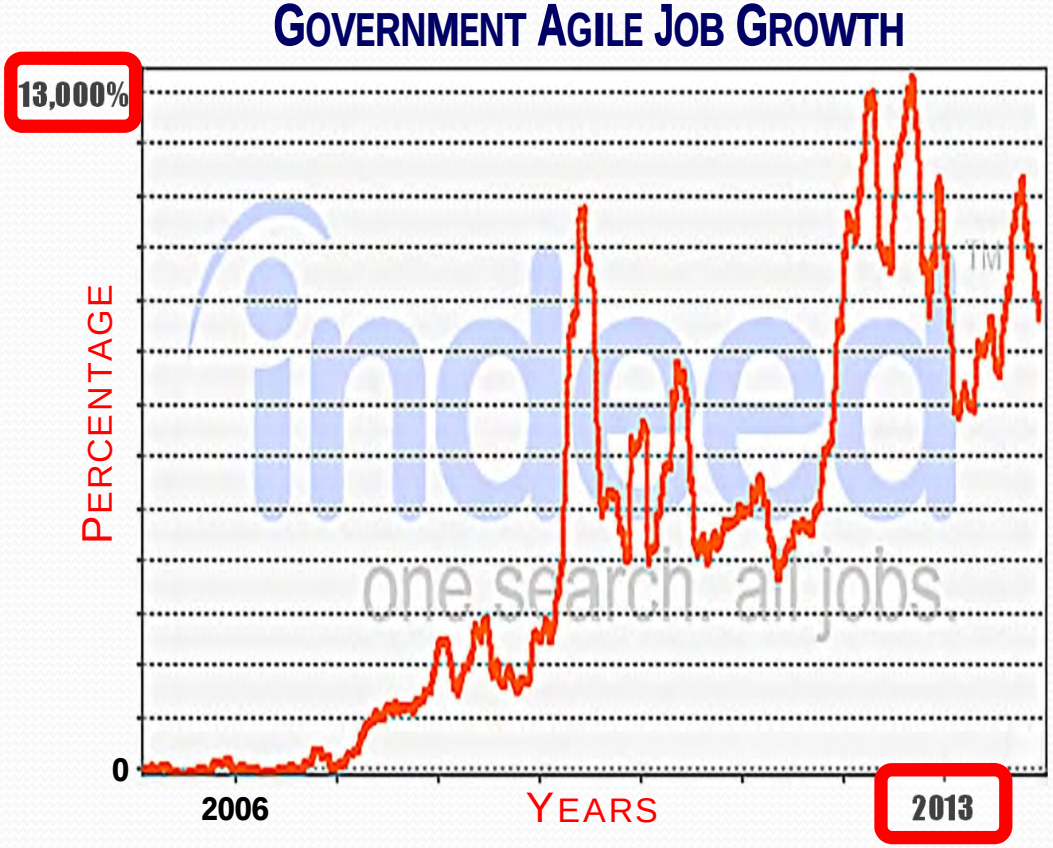


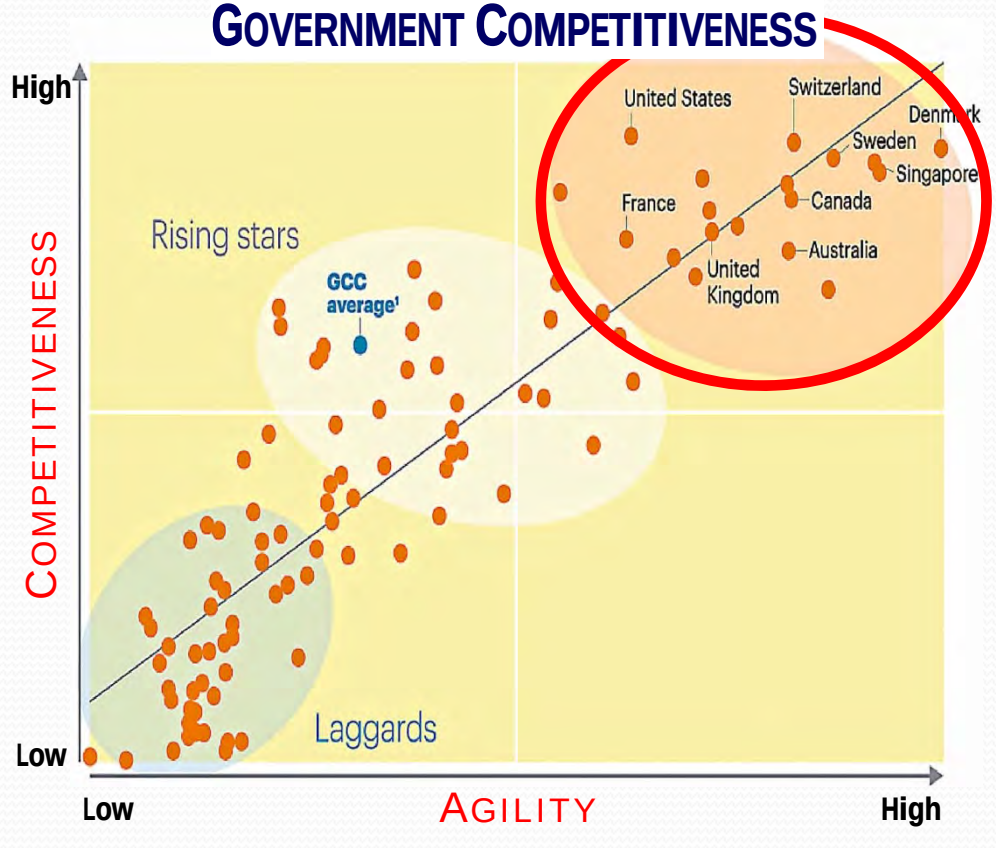


Hoque, F., et al. (2007). *Business technology convergence*. The role of business technology convergence in innovation and adaptability and its effect on financial performance. Stamford, CT: BTM Institute.

# Agile Methods—National Benefits

- U.S. gov't agile jobs grew by 13,000% from 2006-2013
- Adoption is higher in U.S. DoD than Civilian Agencies
- GDP of countries with high adoption rates is greater



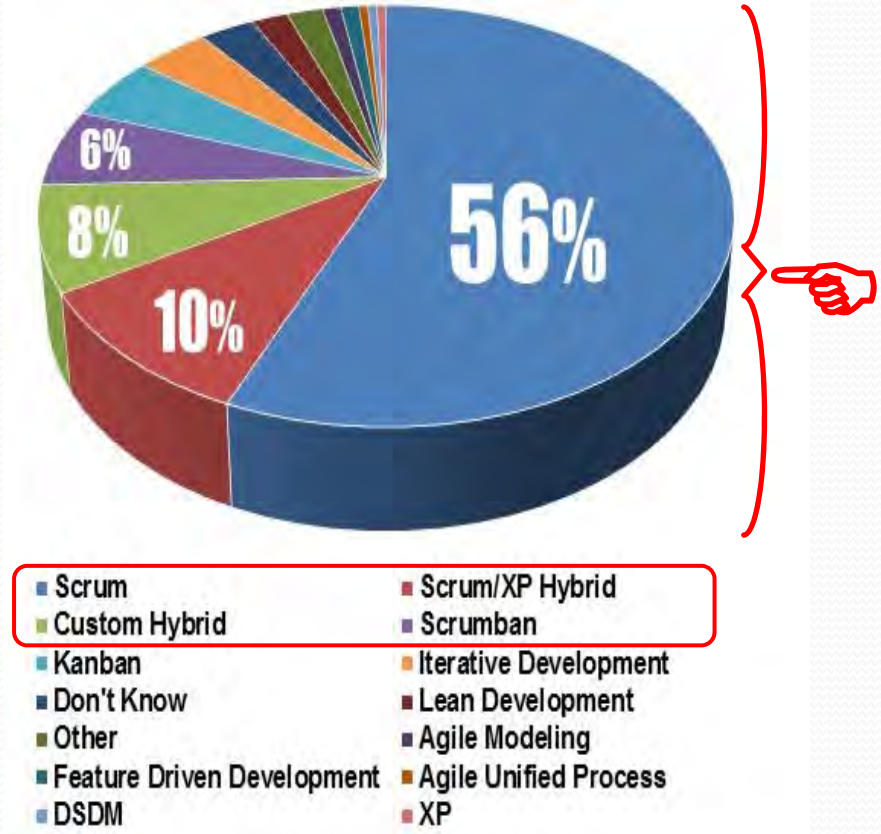


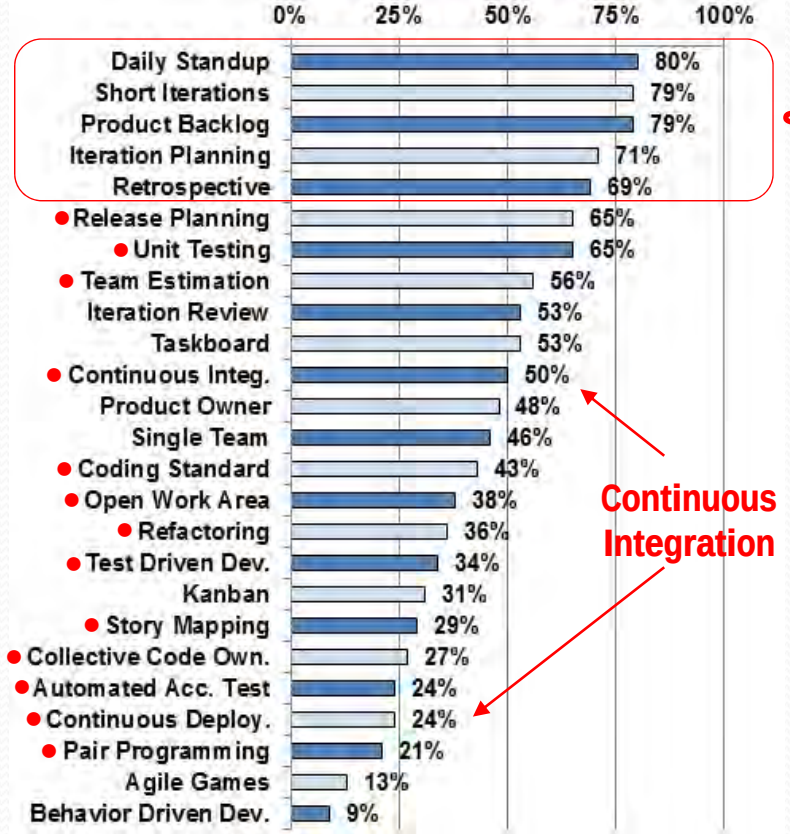
Suhy, S. (2014). *Has the U.S. government moved to agile without telling anyone*? Retrieved April 24, 2015, from http://agileingov.com

Porter, M. E., & Schwab, K. (2008). *The global competitiveness report*: *2008 to 2009*. Geneva, Switzerland: World Economic Forum.

# Agile Methods—Adoption Statistics

- VersionOne found 94% using agile methods today
- Most are using Scrum with several key XP practices
- Lean-Kanban is a rising practice with a 31% adoption

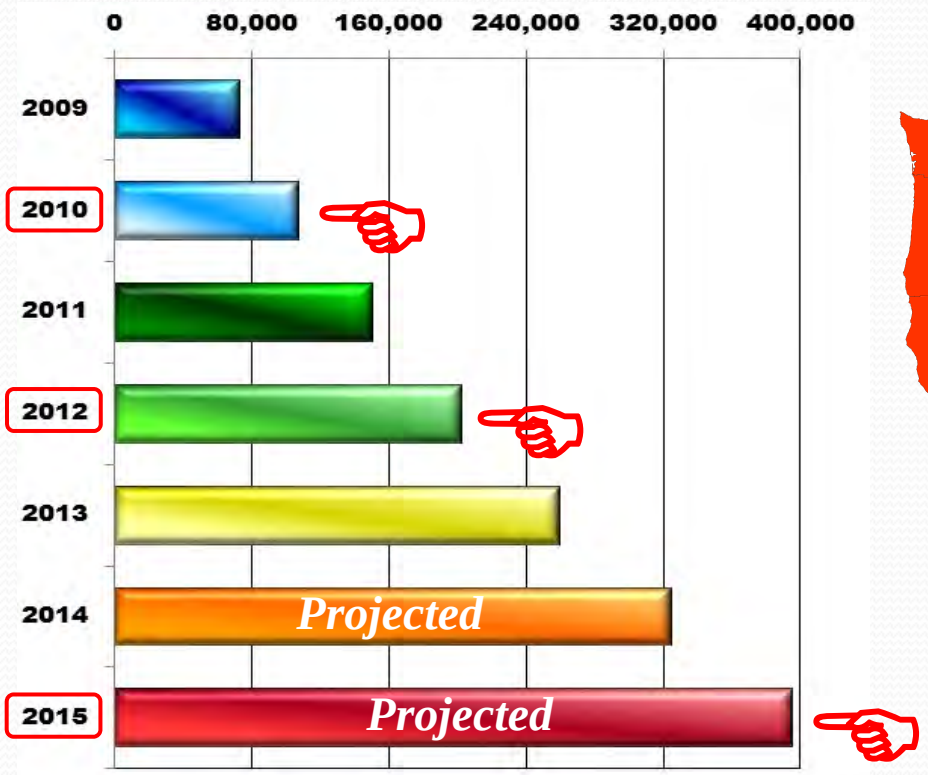


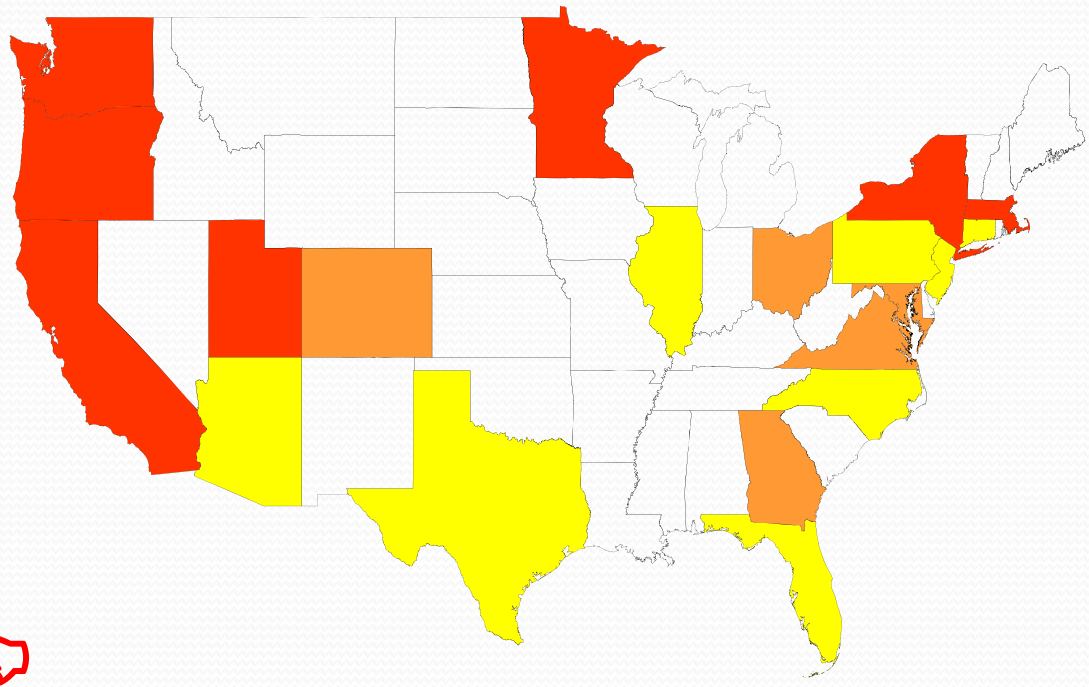




Holler, R. (2015). *Ninth annual state of agile survey*: *State of agile development*. Atlanta, GA: VersionOne.

# Agile Methods—National Adoption

- Number of CSMs have doubled to 400,000 in 4 years
- 558,918 agile jobs for only 121,876 qualified people
- 4.59 jobs available for every agile candidate (5:1)



** PMI-PMPs grew from 552,977 to 625,346 in 2014 (i.e., added 72,369)*


Scrum Alliance. (2013). *Scrum certification statistics*. Retrieved April 28, 2015, from http://www.scrumalliance.org

Taft, D. K. (2012). *Agile developers needed*: *Demand outpaces supply*. Foster City, CA: eWeek.

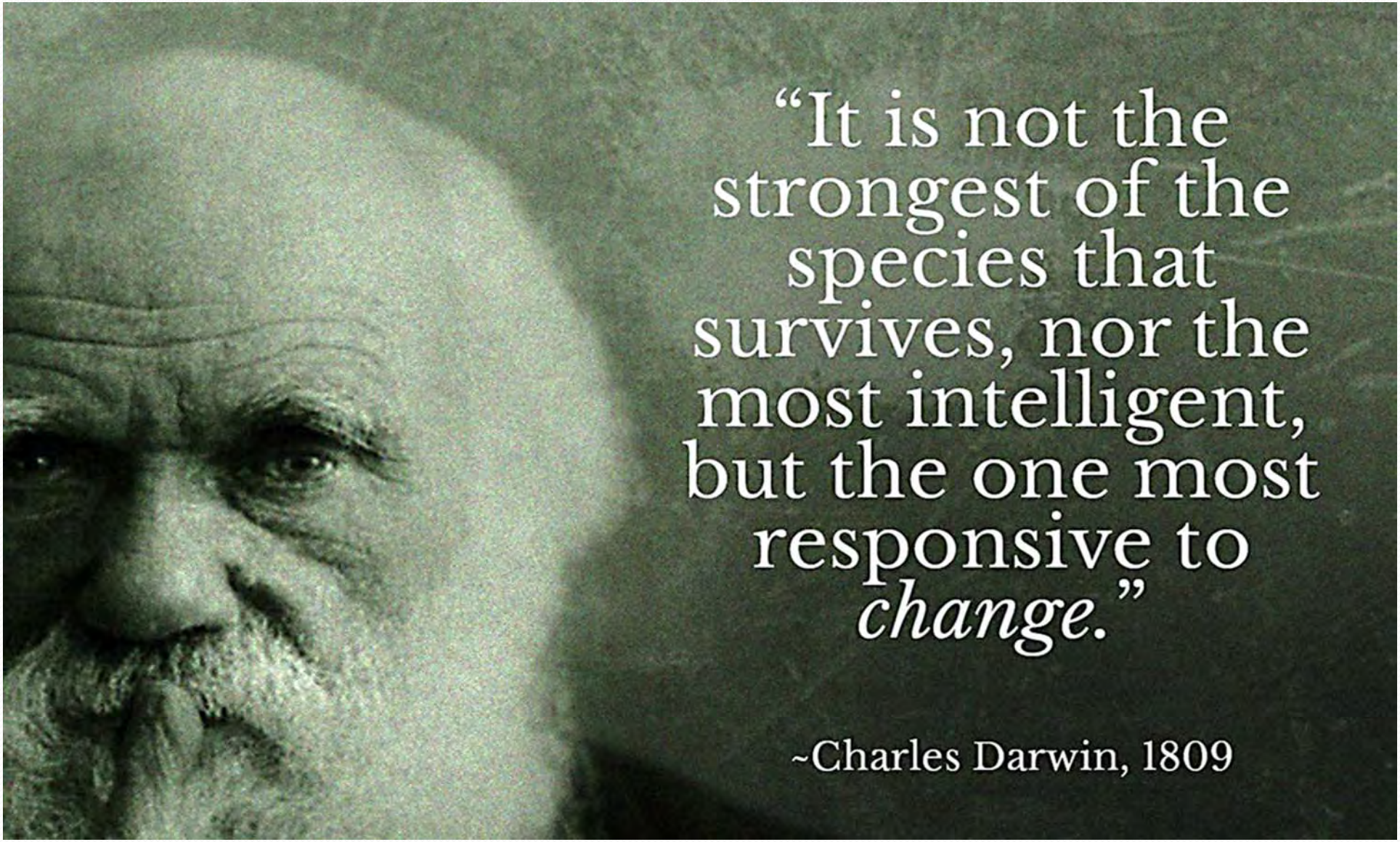
# Agile Methods—Summary

- Agile methods DON'T mean deliver it now & fix it later
- Lightweight, yet disciplined approach to development
- Reduced cost, risk, & waste while improving quality

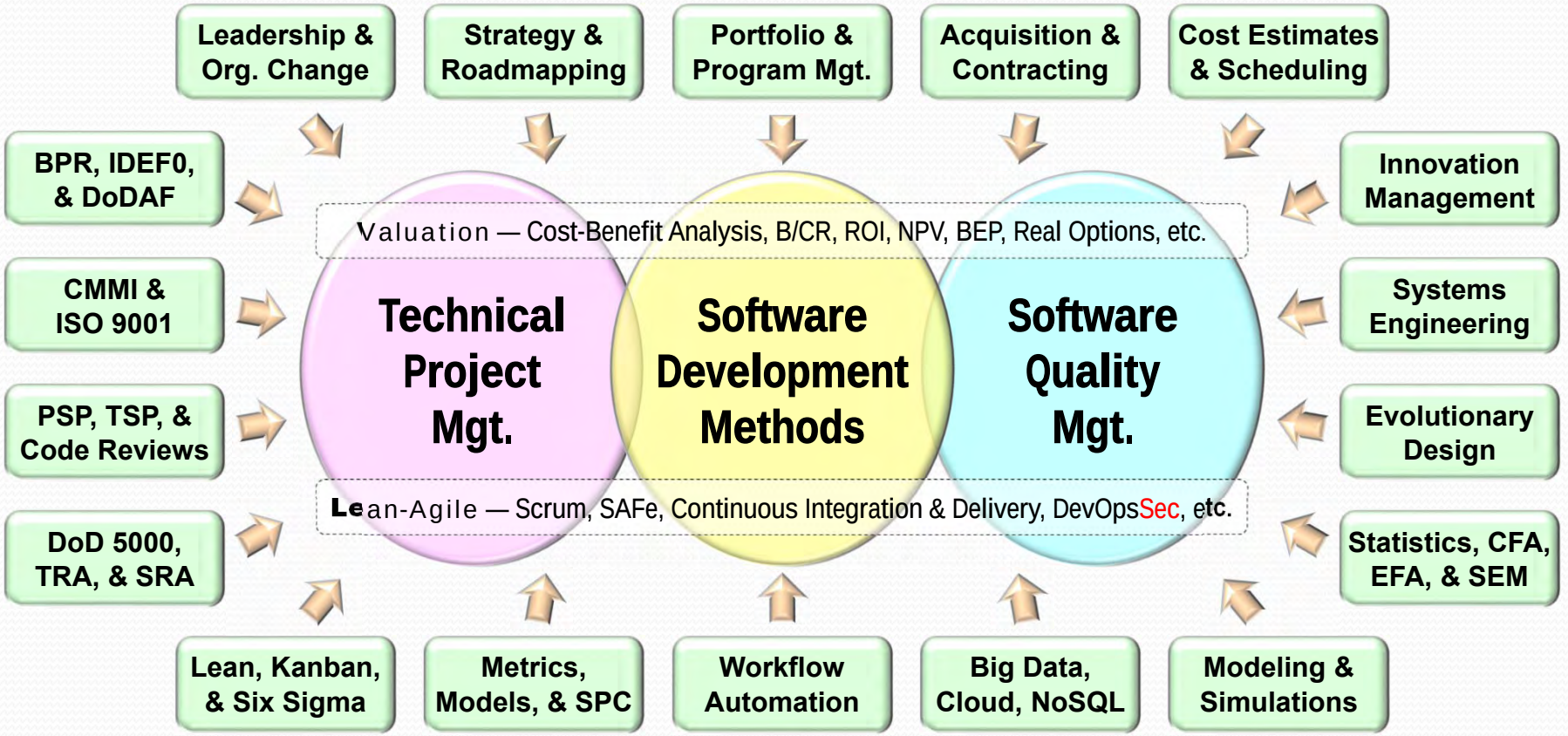
| What | How | Result |
|---|---|---|
| Flexibility | Use lightweight, yet disciplined processes and artifacts | Low work-in-process |
| Customer | Involve customers early and often throughout development | Early feedback |
| Prioritize | Identify highest-priority, value-adding business needs | Focus resources |
| Descope | Descope complex programs by an order of magnitude | Simplify problem |
| Decompose | Divide the remaining scope into smaller batches | Manageable pieces |
| 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 | Knowledge transfer |
| Collaborate | Use frequent informal communications as often as possible | Efficient data transfer |
| Test Early | Incrementally test each component as it is developed | Early verification |
| Test Often | Perform system-level regression testing every few minutes | Early validation |
| Adapt | Frequently identify optimal process and product solutions | Improve performance |


Rico, D. F. (2012). *What's really happening in agile methods: Its principles revisited*? Retrieved June 6, 2012, from http://davidfrico.com/agile-principles.pdf
Rico, D. F. (2012). *The promises and pitfalls of agile methods*. Retrieved February 6, 2013 from, http://davidfrico.com/agile-pros-cons.pdf
Rico, D. F. (2012). *How do lean & agile intersect*? Retrieved February 6, 2013, from http://davidfrico.com/agile-concept-model-3.pdf

# On Adaptibility — Charles Darwin



# Dave's Professional Capabilities




STRENGTHS – Communicating Complex Ideas • Brownbags & Webinars • Datasheets & Whitepapers • Reviews & Audits • Comparisons & Tradeoffs • Brainstorming & Ideation • Data Mining & Business Cases • Metrics & Models • Tiger Teams & Shortfuse Tasks • Strategy, Roadmaps, & Plans • Concept Frameworks & Multi-Attribute Models • Etc.

35+ 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