Lean & Agile

Performance Measurement

Metrics, Models, and Measures for Managing Programs & Projects

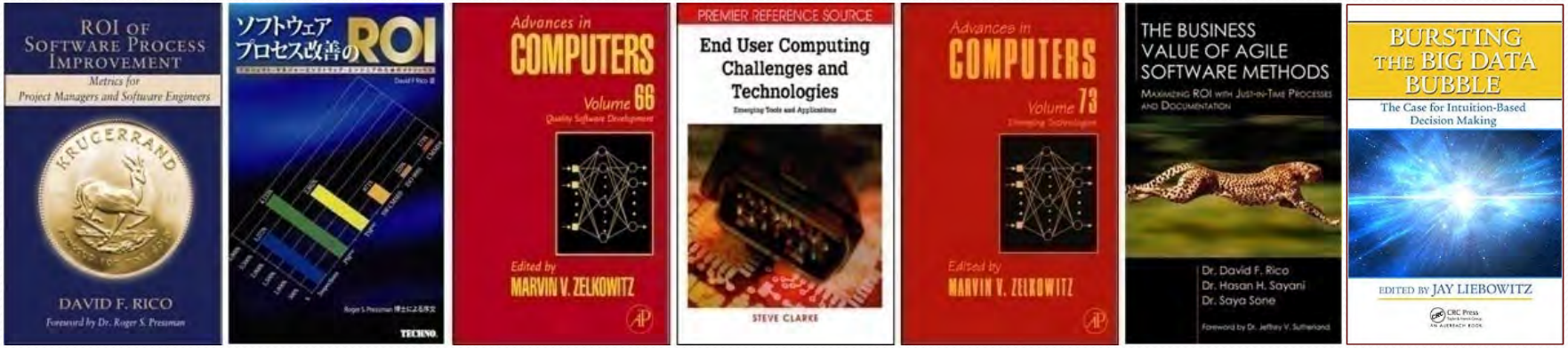
Dr. David F. Rico, PMP, CSEP, FCP, FCT, ACP, CSM, SAFE, DEVOPS

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

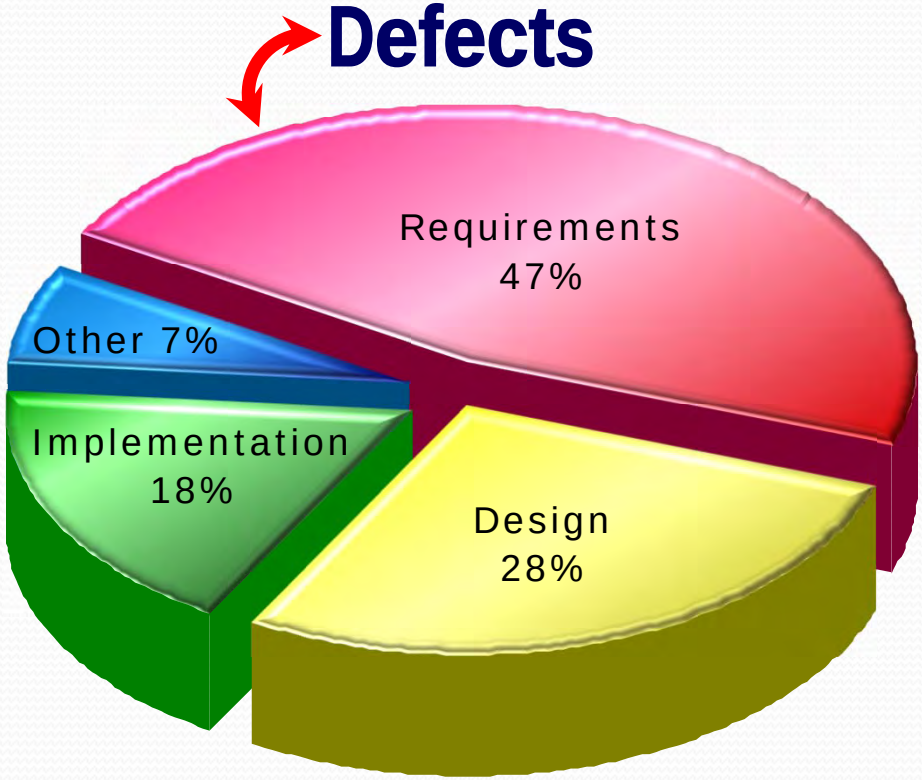


→ ***Career systems & software engineering methodologist***
→ *Lean-Agile, Six Sigma, CMMI, ISO 9001, DoD 5000*
→ *NASA, USAF, Navy, Army, DISA, & DARPA projects*
→ *Published seven books & numerous journal articles*
→ *Intn'l keynote speaker, 185+ talks to 14,000 people*
→ *Specializes in metrics, models,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

On Metrics—Peter Drucker

Requirements Defects & Waste

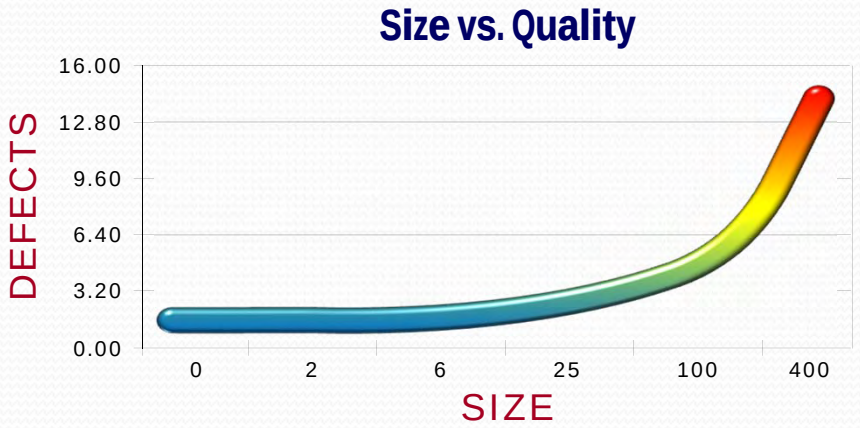
- Requirements defects are #1 reason projects fail
- 80% of requirements exist only as tacit knowledge
- 65% to 95% of explicit requirements are never used

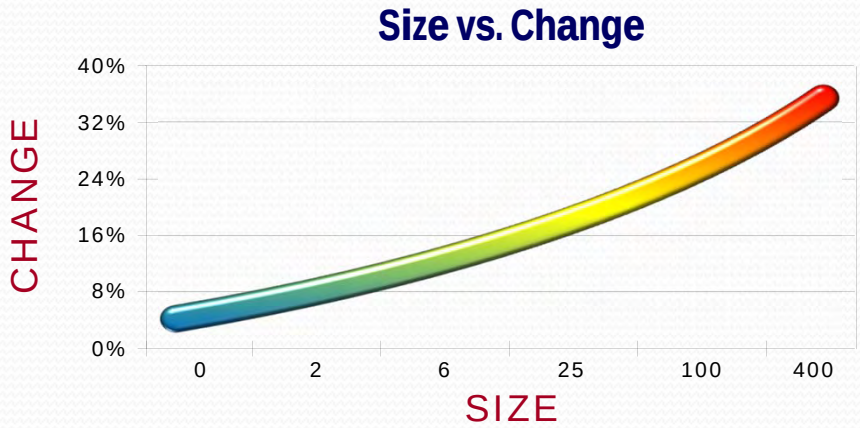


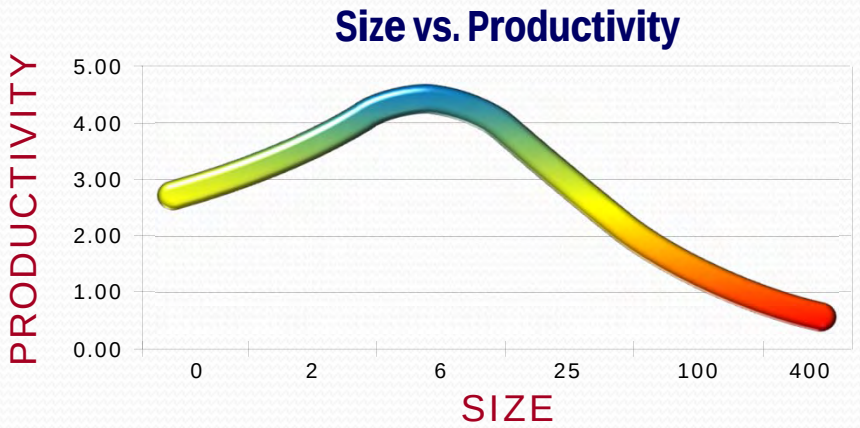
Sheldon, F. T. (1992). Reliability measurement: From theory to practice. *IEEE Software, 9*(4), 13-20.
Johnson, J. (2002). ROI: It's your job. *Extreme Programming 2002 Conference, Alghero, Sardinia, Italy*.
Goffin, K., & Mitchell, R. (2005). *Innovation management: Strategy and implementation*. London, UK: Palgrave-Macmillan.
Chedalawada, A. (2012). Lean-agile overview. *Second Annual AFEI/NDIA Conference on Agile in DoD, Springfield, VA, USA*.

Large Traditional Projects

- Big projects result in poor quality and scope changes
- Productivity declines with long queues/wait times
- Large projects are unsuccessful or canceled







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

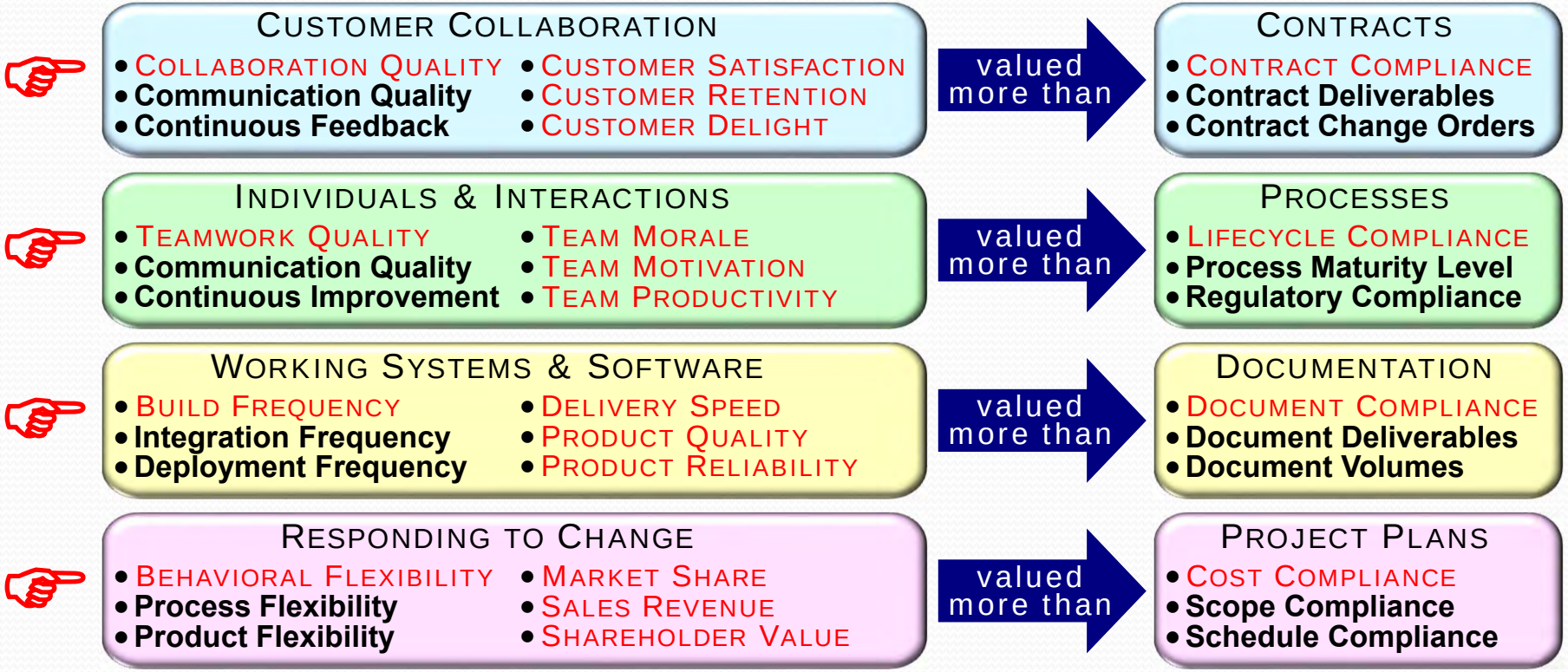
What are Agile Metrics?

- Met-ric (mĕt′rĭk) A standard of measurement; system of related measures; quantification of a characteristic
  - Quantitative measure *of a degree to which agile project* processes *or resulting* systems *possess some property*
  - Numerical ratings *to measure the* size, cost, complexity, *or* quality *of software produced using agile methods*
  - Measurement *of a particular characteristic of an agile project's* scope, time, cost, progress, *or* technical perf.
  - *Measure of the degree of* customer collaboration, team-work, iterative development, *or* adaptability *to change*
  - *Ensuring* **BUSINESS VALUE** *by measuring operational and team performance, customer satisfaction, and ROI*

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

What are Some Agile Metrics?

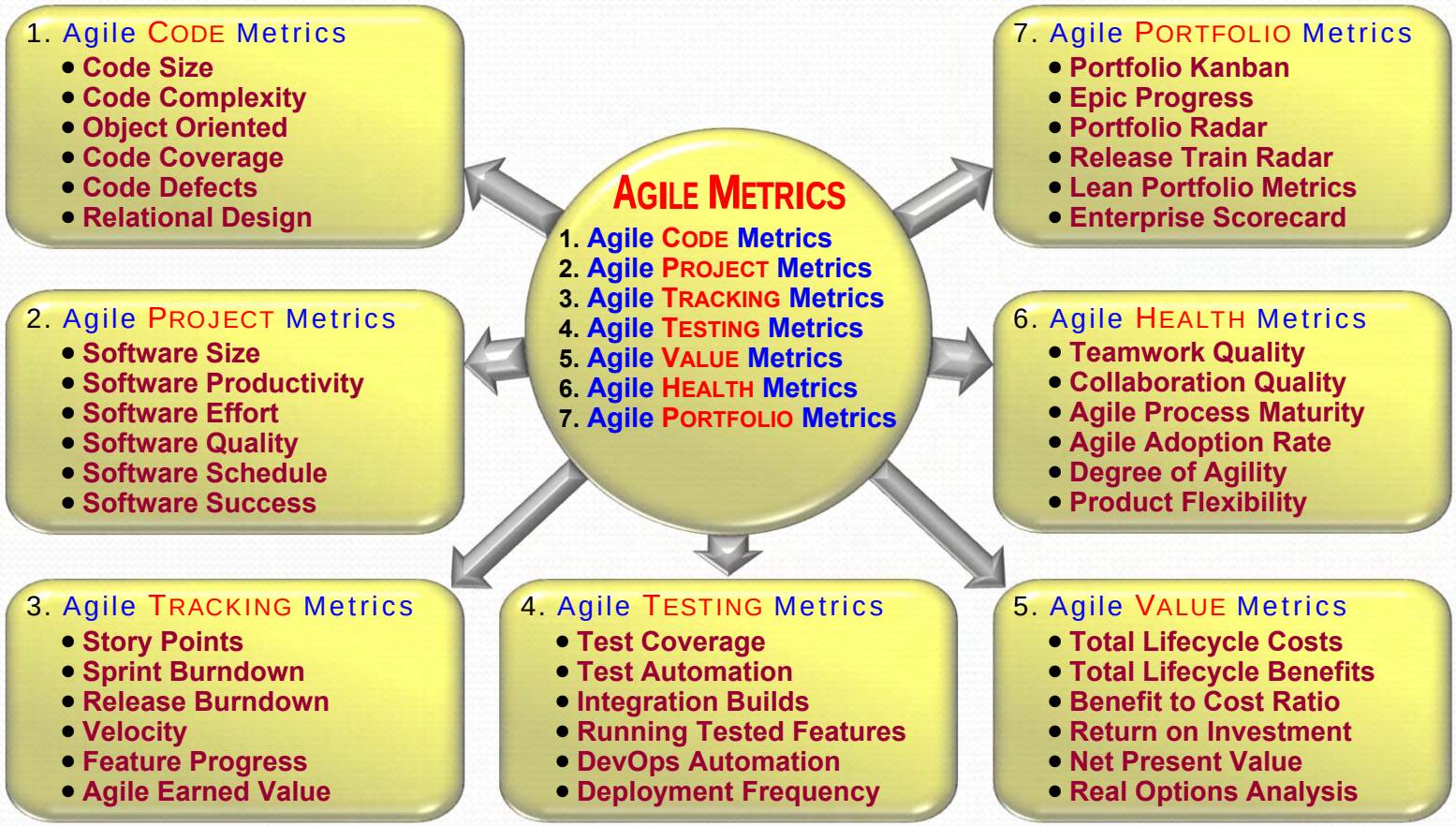
- Collaboration maximizes customer satisfaction
- Iteration maximizes speed, quality, and feedback
- Adaptability maximizes continuous improvements



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

Agile 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 Code Metrics

- Software source metrics created in the 1960s/1970s
- Halstead software science & complexity very popular
- Complexity, OO, and defect metrics most widely used

Metric	Description
Code Size	Volume or amount of software source code
Code Complexity	Intricacy, difficulty, or complication of software source code
Object Oriented	Cohesion, coupling, or modularity of software source code
Code Coverage	Executable, reachable, or testable software source code
Code Defects	Flawed, imperfect, or non-conformant software source code
Relational Design	Normalized, non-redundant, or anomaly-free data schema

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

Agile Code Metrics—Example

Lines of Code	
Minimum	6,493
Maximum	5,050,450
Mean	425,179

Cyclomatic Complexity	
Minimum	158
Maximum	816,066
Mean	53,035

Avg. Defect Density	
Minimum	0.00
Maximum	1.22
Mean	0.25

Number of Functions	
Minimum	47
Maximum	215,925
Mean	12,880

Halstead Effort	
Minimum	2,276
Maximum	71,949,783
Mean	6,399,178

Average Function Length	
Minimum	13.97
Maximum	345.72
Mean	66

Avg. Number of Defects	
Minimum	1
Maximum	4,967
Mean	283.49

Defect Type	Defects	%
NULL Pointer Dereference	6,448	27.95%
Resource Leak	5,852	25.73%
Unintentional Ignored Expressions	2,252	9.76%
Use Before Test (NULL)	1,867	8.09%
Buffer Overrun (statically allocated)	1,417	6.14%
Use After Free	1,491	6.46%
Unsafe use of Returned NULL	1,349	5.85%
Uninitialized Values Read	1,268	5.50%
Unsafe use of Returned Negative	859	3.72%
Type and Allocation Size Mismatch	144	0.62%
Buffer Overrun (dynamically allocated)	72	0.31%
Use Before Test (negative)	49	0.21%

Chou, A. (2008). *Coverity scan report*: *Open source software*. San Francisco, CA: Coverity, Inc.

Agile Project Metrics

- Core software project metrics created in 1960s/1970s
- Software size, productivity, & effort were very popular
- Software productivity & quality metrics still relevant

Metric	Description
Software Size	Estimate of conceptual, logical, or physical software volume
Software Productivity	Relative rate or speed at which software is produced
Software Effort	Estimate of time needed for software development project
Software Quality	Degree to which software conforms to its requirements
Software Schedule	Software timeline in milestones, activities, or deliverables
Software Success	Average probability of on-time software schedule delivery

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

Agile Project Metrics—Example

Software Size (Lines of Code)							Effort	Schedule
FP	HTML	Java	Ruby	Python	C#	SQL	Hours	Months
1	91	53	46	46	40	13	4	0.03
10	914	533	457	457	400	128	61	0.59
100	9,143	5,333	4,571	4,571	4,000	1,280	809	4.50
1,000	91,430	53,330	45,710	45,710	40,000	12,800	10,418	13.29
10,000	914,300	533,300	457,100	457,100	400,000	128,000	352,000	42.86
100,000	9,143,000	5,333,000	4,571,000	4,571,000	4,000,000	1,280,000	5,038,168	60.00
1,000,000	91,430,000	53,330,000	45,710,000	45,710,000	40,000,000	12,800,000	61,395,349	72.43

Productivity (Lines of Code per Hour)							Quality	Success
FP	HTML	Java	Ruby	Python	C#	SQL	Defects/LOC	On-Time%
1	23.44	13.67	11.72	11.72	10.25	3.28	0.0012	83.16%
10	14.93	8.71	7.47	7.47	6.53	2.09	0.0031	81.25%
100	11.30	6.59	5.65	5.65	4.94	1.58	0.0057	74.77%
1,000	8.78	5.12	4.39	4.39	3.84	1.23	0.0134	60.76%
10,000	2.60	1.52	1.30	1.30	1.14	0.36	0.0238	28.03%
100,000	1.81	1.06	0.91	0.91	0.79	0.25	0.0386	13.67%
1,000,000	1.49	0.87	0.74	0.74	0.65	0.21	0.0498	7.18%

Jones, C. (2007). *Estimating software costs*: *Bringing realism to estimating*. New York, NY: McGraw-Hill.

Agile Tracking Metrics

- Basic agile metrics confluence of XP-Scrum practices
- XP release planning formed basis of Scrum planning
- Today's basic agile metrics were tailored for Scrum

Metric	Description
Story Points	Degree of size, difficulty, or complexity of a user story
Sprint Burndown	Estimated hours completed on a daily basis each iteration
Release Burndown	Estimated story points completed each iteration on a project
Velocity	Software productivity expressed in story points per iteration
Feature Progress	Number, degree, or percent of planned features completed
Agile Earned Value	Simplified set of earned value measures for agile projects

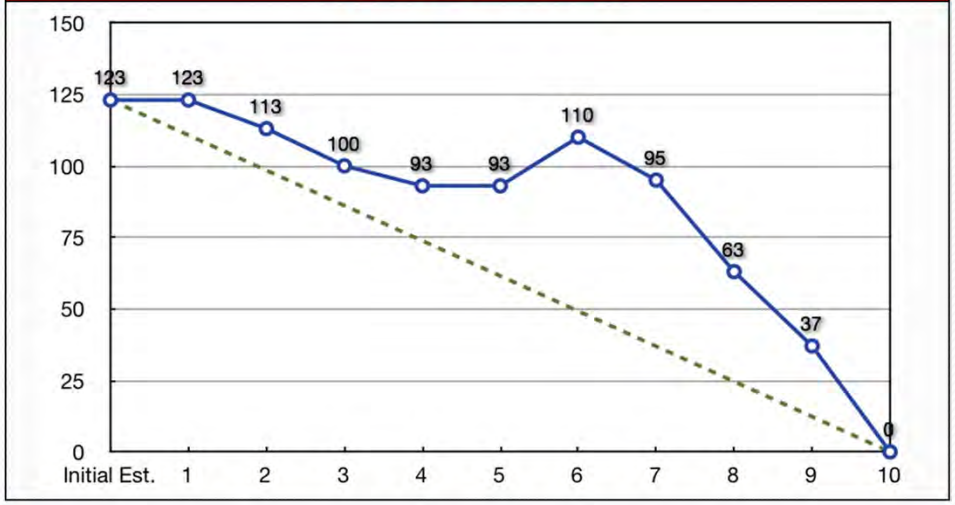
Cohn, M. (2006). *Agile estimating and planning*. Upper Saddle River, NJ: Pearson Education.

Agile Tracking Metrics—Example

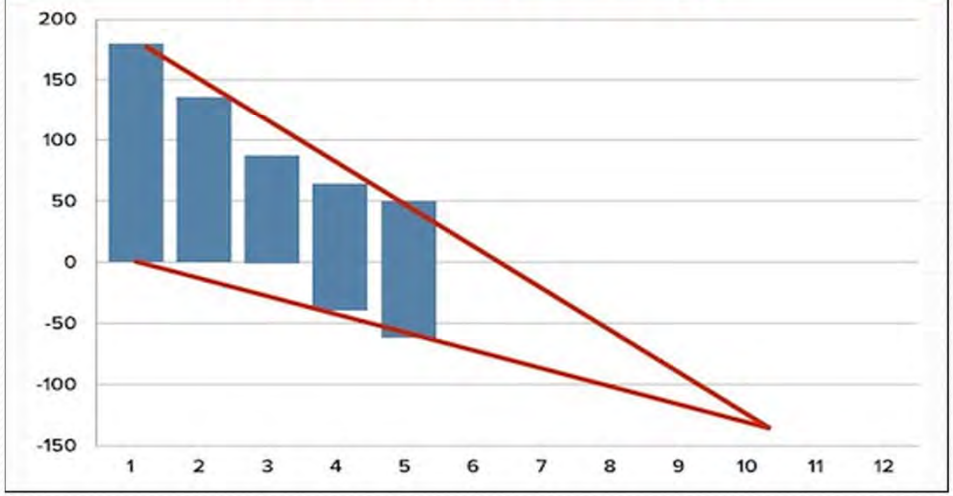
Story Points

Relative Size	Story Points	Staff Hours	Staff Days	Staff Month	Staff Years	2-Week Sprints	3-Sprint Releases
User Story	1	22	3	0.1	0.0	0.1	0.0
	2	44	6	0.3	0.0	0.1	0.0
	3	67	8	0.4	0.0	0.2	0.1
	5	111	14	0.6	0.1	0.3	0.1
Feature	8	178	22	1.0	0.1	0.4	0.1
	13	289	36	1.7	0.1	0.7	0.2
	21	467	58	2.7	0.2	1.2	0.4
	34	755	94	4.4	0.4	1.9	0.6
Epic	55	1,222	153	7.0	0.6	3.1	1.0
	89	1,977	247	11.4	1.0	4.9	1.6
	144	3,199	400	18.5	1.5	8.0	2.7
	233	5,177	647	29.9	2.5	12.9	4.3

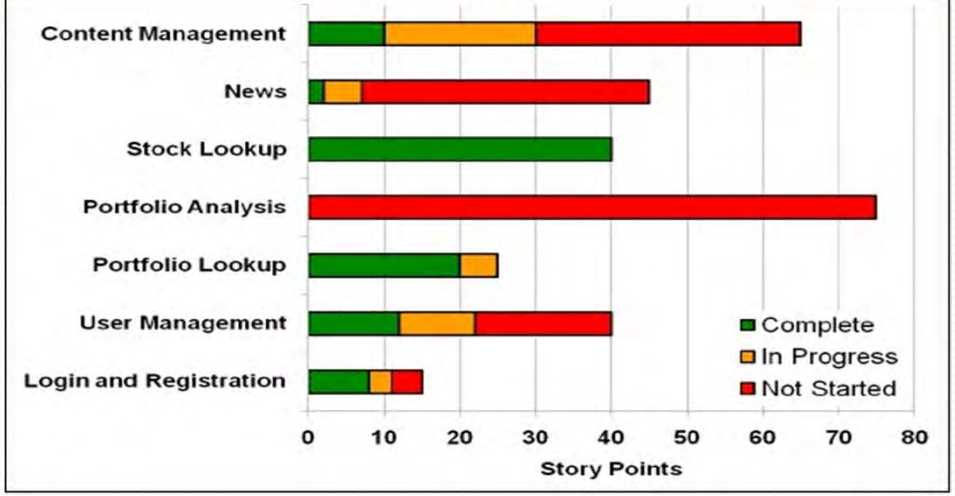
Sprint Burndown



Release Burndown (+Velocity)



Feature Progress



Cohn, M. (2006). *Agile estimating and planning*. Upper Saddle River, NJ: Pearson Education.

Agile Testing Metrics

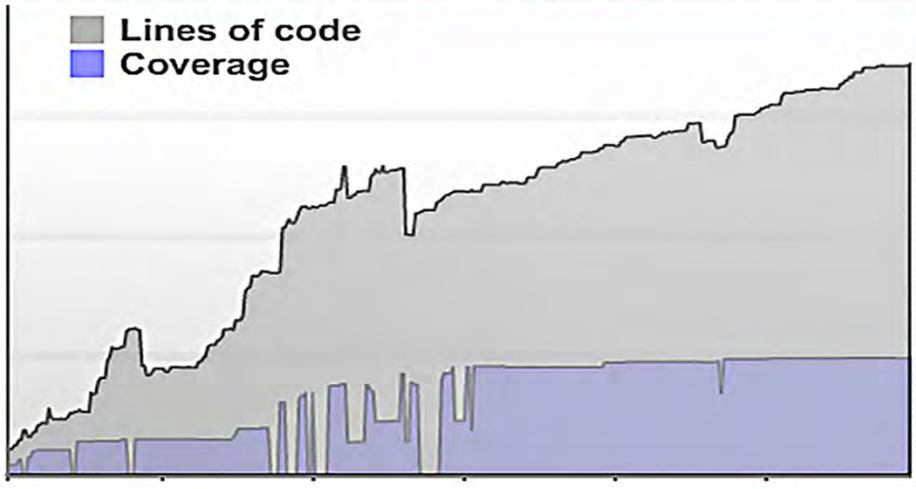
- Software test automation emerged during the 1970s
- Reached their height in personal computer (PC) era
- Most are FOSS and used by successful agile teams

Metric	Description
Test Coverage	Percent or degree to which software source code is tested
Test Automation	Ratio or degree to which software tests are automated
Integration Builds	Frequency of automated software builds and integrations
Running Tested Features	Number of completed and tested features or user stories
DevOps Automation	Ratio or degree to which deployments are automated
Deployment Frequency	Frequency of automated software deployments or deliveries

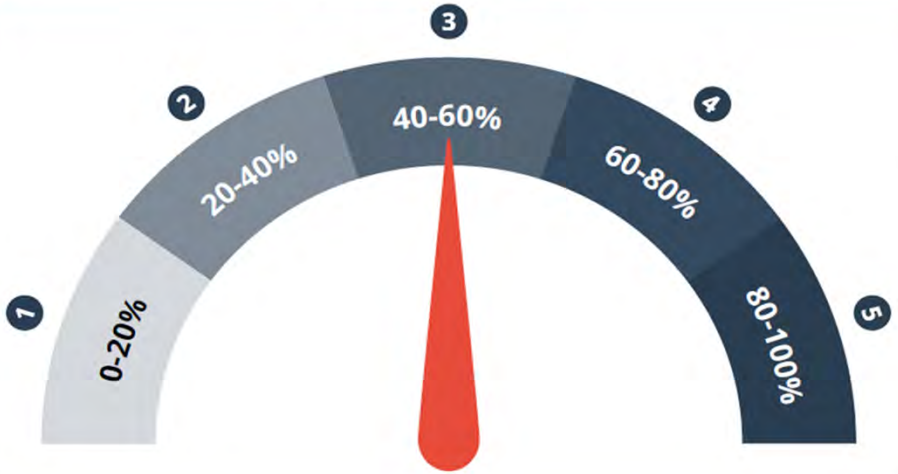
Duvall, P., Matyas, S., & Glover, A. (2006). *Continuous integration*: *Improving software quality and reducing risk*. Boston, MA: Addison-Wesley.

Agile Testing Metrics—Example

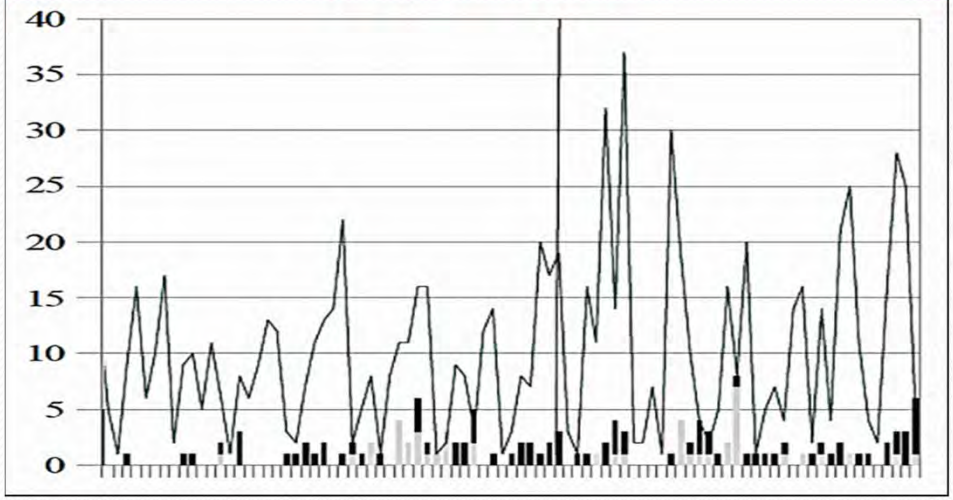
Test Coverage



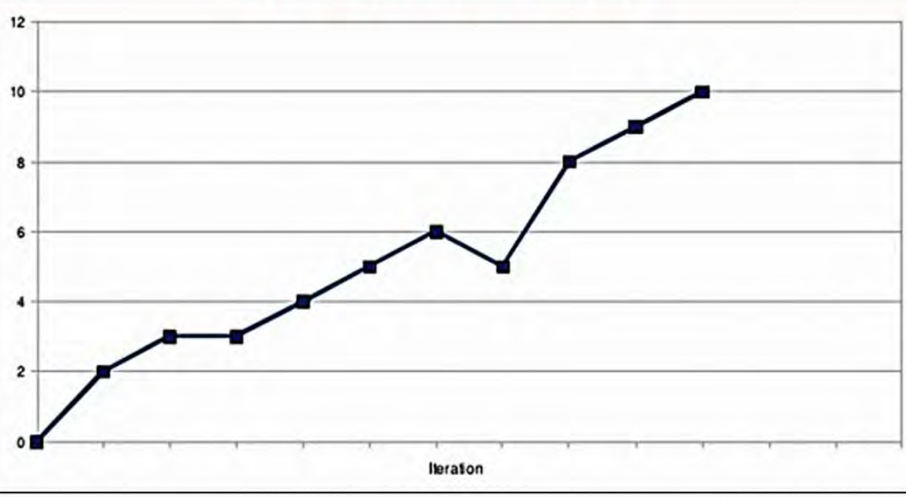
Test Automation



Integration Builds



Running Tested Features



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

Agile Value Metrics

- Business value metrics form basis of agile methods
- Most measures used throughout the 20th century
- Most useful at the portfolio and program levels

Metric	Description
Total Lifecycle Costs	Sum of all software development and maintenance costs
Total Lifecycle Benefits	Sum of all software development and maintenance benefits
Benefit to Cost Ratio	Ratio of total lifecycle benefits to costs
Return on Investment	Ratio of adjusted total lifecycle benefits to costs
Net Present Value	Discounted value of adjusted total lifecycle benefits
Real Options Analysis	Risk-adjusted value of total lifecycle benefits to costs

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

Agile Value Metrics—Example

Business Value Metrics

Costs Sum of Costs	Total amount of money spent	$\sum_{i=1}^{n} Cost_i$
Benefits Sum of Benefits	Total amount of money gained	$\sum_{i=1}^{n} Benefit_i$
B/CR Benefit to Cost Ratio	Ratio of benefits to costs	$\frac{Benefits}{Costs}$
ROI Return on Investment	Ratio of adjusted benefits to costs	$\frac{Benefits - Costs}{Costs} \times 100\%$
NPV Net Present Value	Discounted cash flows	$\sum_{i=1}^{Years} \frac{Benefits_i}{(1+Discount\ Rate)^{Years}} - Costs_0$
BEP Breakeven Point	Point when benefits exceed costs	$\frac{New\ Costs}{Old\ Costs / New\ Costs - 1}$
ROA Real Options Analysis	Value gained from strategic delay	$N(d_1) \times Benefits - N(d_2) \times Costs \times e^{-Rate \times Years}$

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

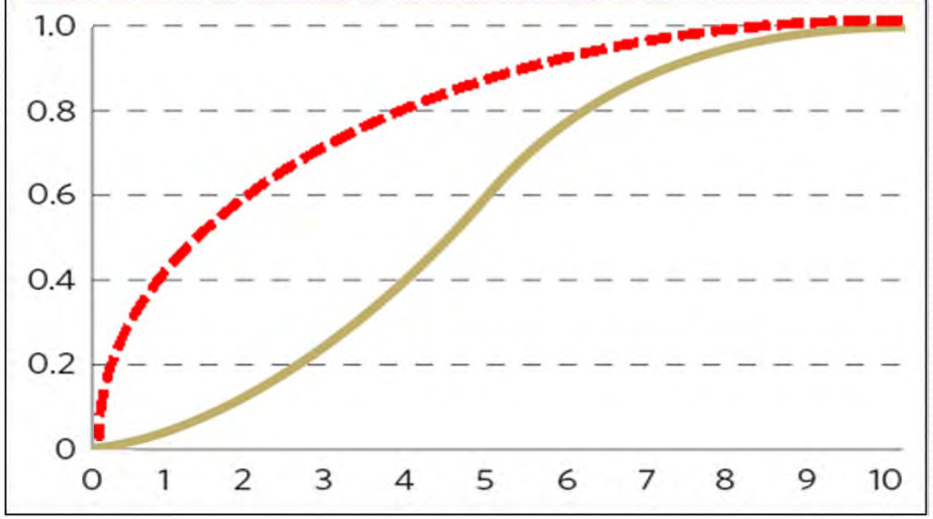
Business Value Formulas

Costs	(10,000 ÷ **5.4436** + **3.945** × 10 × 100) × 100
Benefits	(10,000 × 10.51 − 6,666.67 × 9) × 100 − **$588,202**
B/CR	**$3,930,631** ÷ **$588,202**
ROI	(**$3,930,631** − **$588,202**) ÷ **$588,202** × 100%
NPV	($\sum_{i=1}^{5}$ (**$3,930,631** ÷ 5) ÷ 1.05^5) − **$588,202**
BEP	**$588,202** ÷ ($4,509,997 ÷ **$588,202** − 1)
ROA	NORMSDIST(**2.24**) × **$3,930,631** − NORMSDIST(**0.85**) × **$588,202** × EXP(−5% × 5)

Business Value Measures

Metric	Scrum	Contin. Integ.	DevOps
Costs	$588,202	$233,152	$32,315
Benefits	$3,920,631	$4,275,681	$4,476,517
Benefit-Cost	7:1	18:1	139:1
ROI%	567%	1,734%	13,753%
NPV	$2,806,654	$3,469,140	$3,843,880
Breakeven	$88,220	$12,710	$233
Real Options	$3,504,292	$4,098,159	$4,451,359

Earned Business Value



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

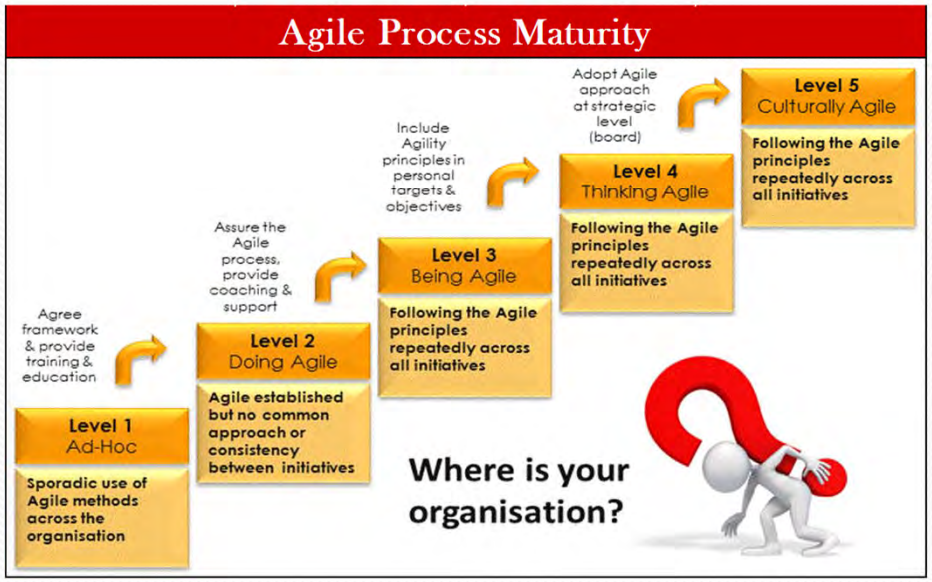
Agile Health Metrics

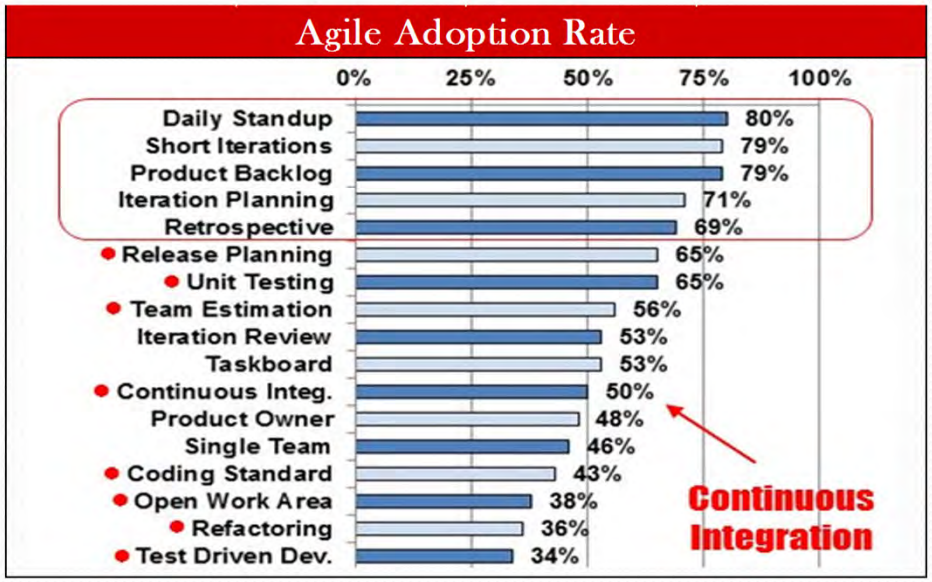
- Agile health metrics emerged in mid-2000s
- Designed to measure agile process compliance
- Best ones assess teamwork & collaboration quality

Metric	Description
Teamwork Quality	Degree to which teamwork results in project success
Collaboration Quality	Degree to which collaboration results in project success
Agile Process Maturity	Degree to which agile processes are consistently applied
Agile Adoption Rate	Degree to which agile processes are widely used
Degree of Agility	Degree to which agile behaviors are consistently applied
Product Flexibility	Degree to which agile products are technologies are utilized

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

Agile Health Metrics—Example





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

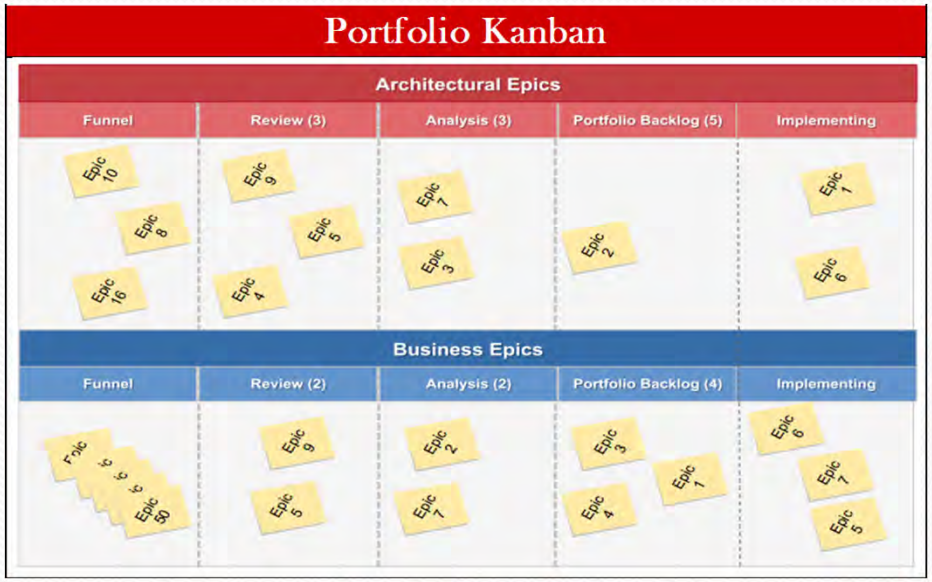
Agile Portfolio Metrics

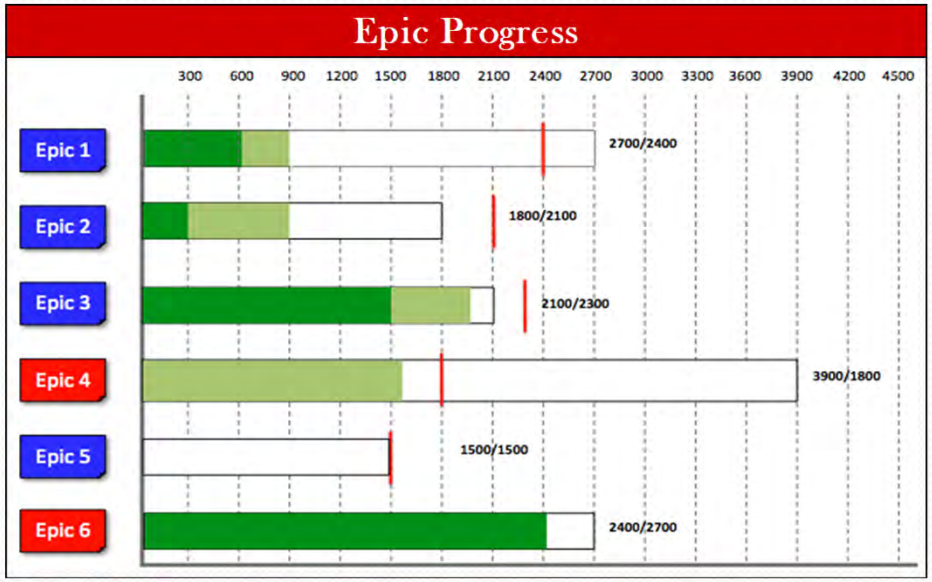
- Business value metrics traditionally used for portfolios
- Processes now emerging for portfolio management
- Lean-Kanban practices & measures most popular

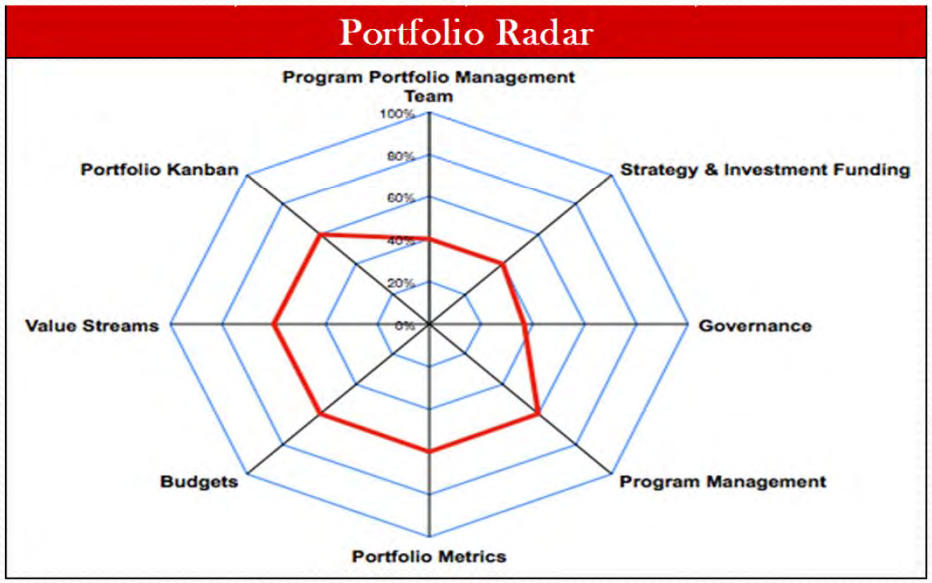
Metric	Description
Portfolio Kanban	Information display to optimize flow of portfolio epics
Epic Progress	Number, degree, or percent of planned epics completed
Portfolio Radar	Degree to which portfolio practices and behaviors are used
Release Train Radar	Degree to which agile release train practices are utilized
Lean Portfolio Metrics	Degree to which lean measures are utilized
Enterprise Scorecard	Degree to which an agile enterprise scorecard is used

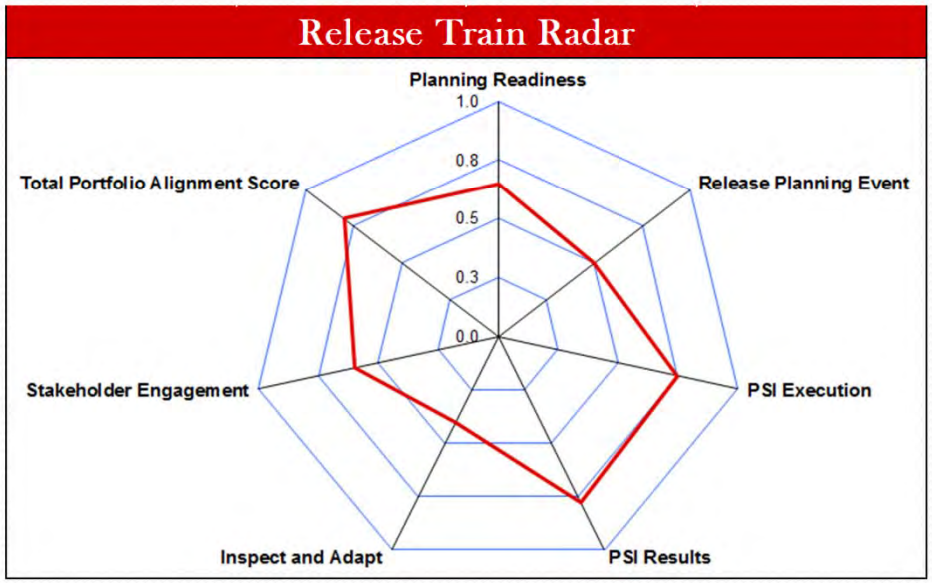
Leffingwell, D. (2015). *Scaled agile framework* (*SAFe*). Retrieved June 12, 2015 from http://www.scaledagileframework.com

Agile Portfolio Metrics—Example









Leffingwell, D. (2015). *Scaled agile framework* (*SAFe*). Retrieved June 12, 2015 from http://www.scaledagileframework.com

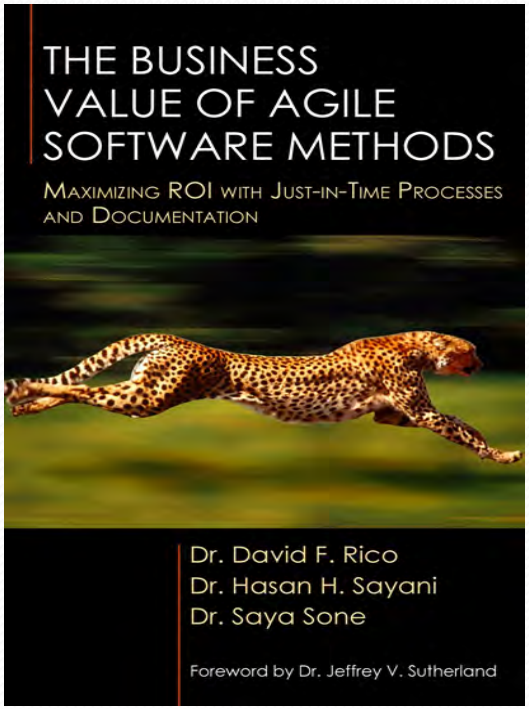
Lean & Agile Metrics Summary

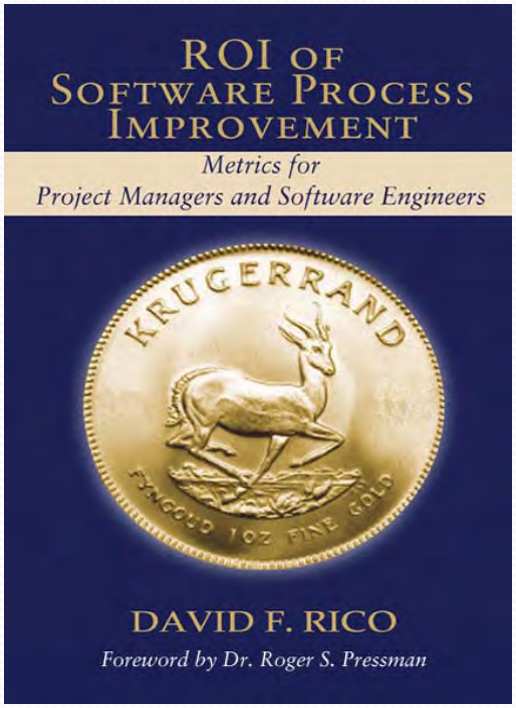
- Traditional metrics and principles apply to lean & agile
- Metrics range from source code up to portfolio levels
- Metrics apply to teams, projects, and organizations

- **Measure** - *You can't manage what you don't measure*.
- **Early & Often** - *Don't hesitate to measure early and often*.
- **Traditional Metrics** - *Don't throw the baby out with the bathwater*.
- **Alignment** - *Align metrics and measures with lean-agile principles*.
- **Resistance** - *Expect resistance to change with respect to metrics*.
- **Hierarchy** - *Use metric hierarchy ranging from code to portfolios*.
- **Basic** - *Remember to use basic metrics such as burndown charts*.
- **Testing** - *Testing metrics may be the single most important metrics*.
- **Health** - *Use health metrics to assess team, project, and org. perf.*
- **Portfolio** - *Portfolio metrics used to track organizational projects*.
- **Easy** - *Collecting and analyzing metrics is easier than you think*.
- **FOSS** - *Don't break the bank on multi-million dollar metric tools*.

Books on ROI of SW Methods

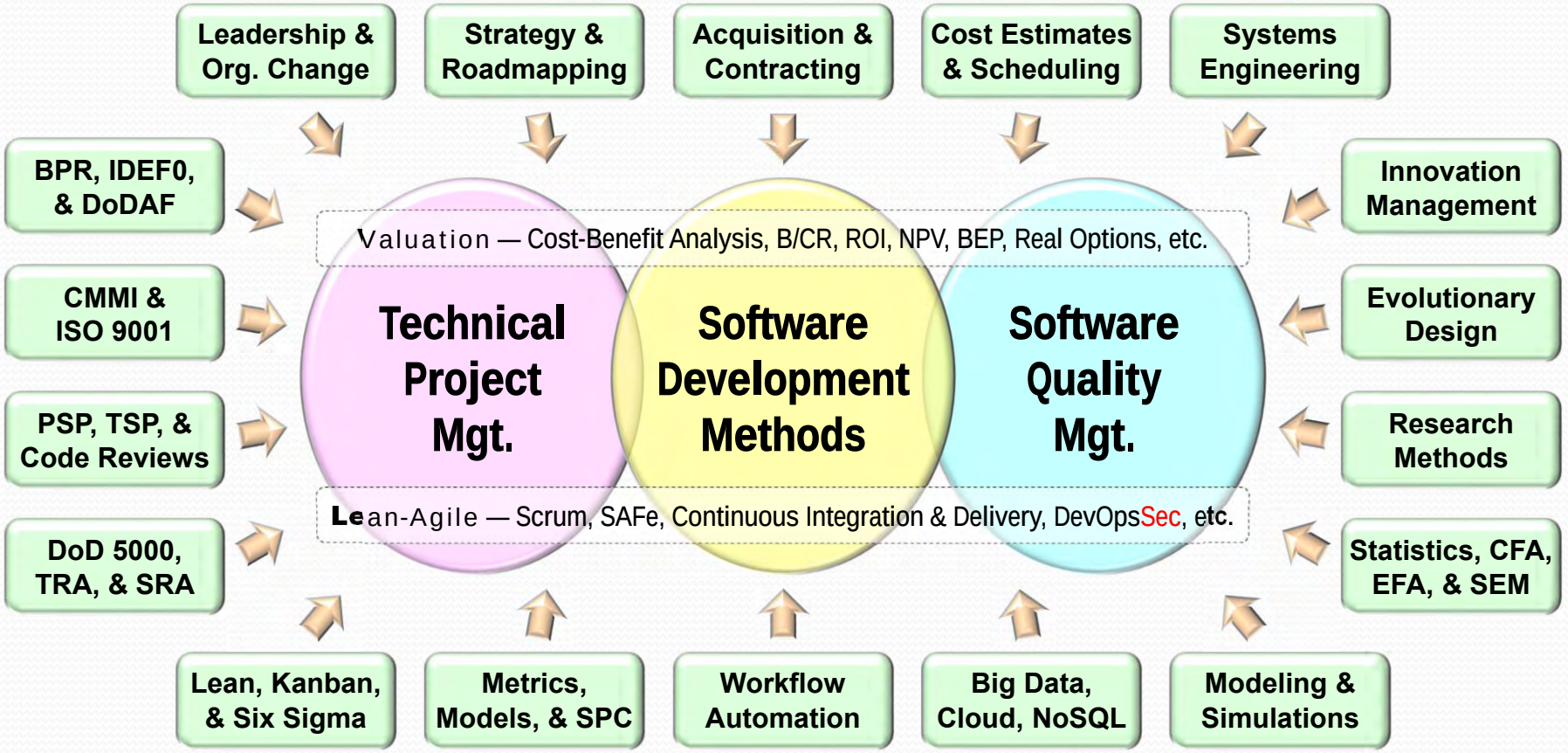
- Guides to software methods for business leaders
- Communicates the business value of IT approaches
- Rosetta stones to unlocking ROI of software methods





- http://davidfrico.com/agile-book.htm (***Description***)
- http://davidfrico.com/roi-book.htm (***Description***)

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

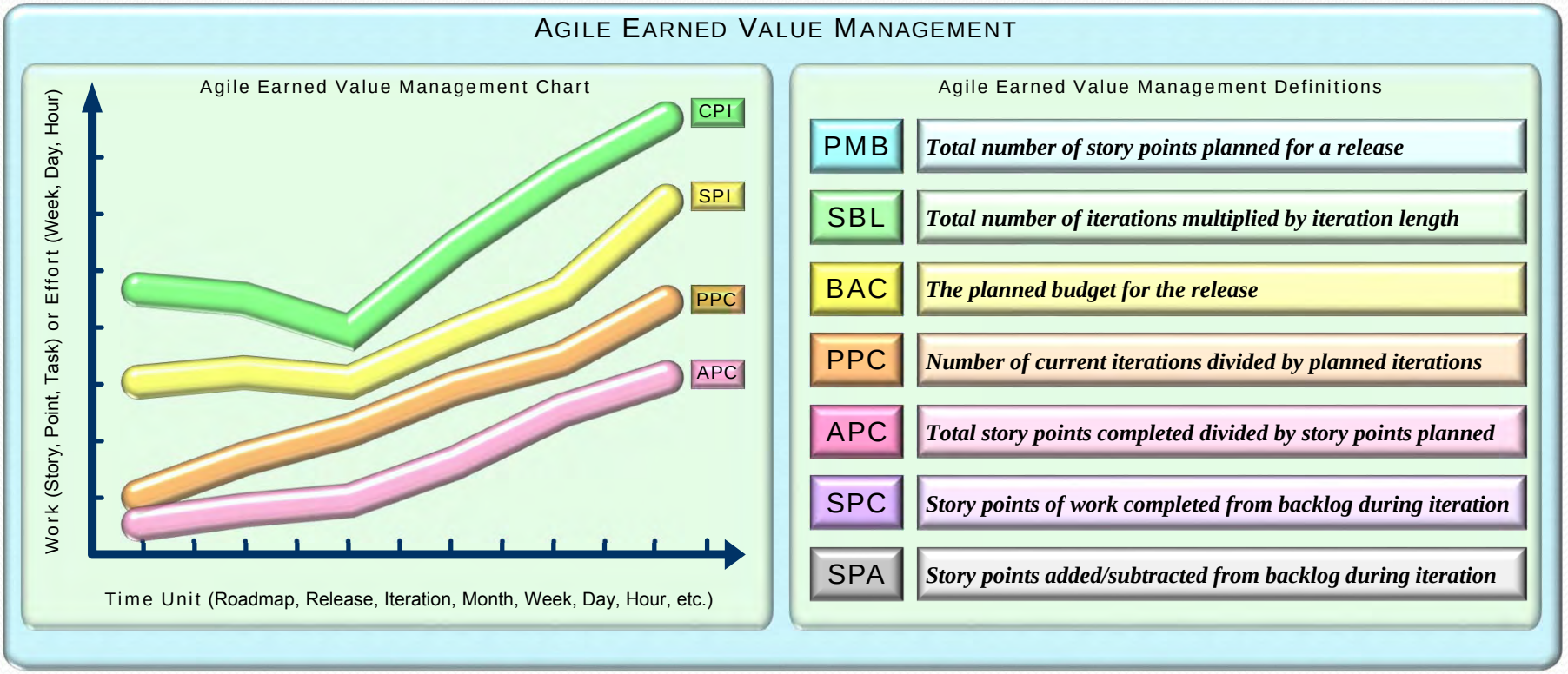
- **Data mining. Metrics, benchmarks, & performance.**
- **Simplification. Refactoring, refinement, & streamlining.**
- **Assessments. Audits, reviews, appraisals, & risk analysis.**
- **Coaching. Diagnosing, debugging, & restarting stalled projects.**
- **Business cases. Cost, benefit, & return-on-investment (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

Backup Slides

Agile Earned Value Metrics

- Adaptation of earned value mgt. for agile projects
- Value accrues with completed sprints and releases
- ☞ Better measure of value due to agile DoD, RTF, & CI

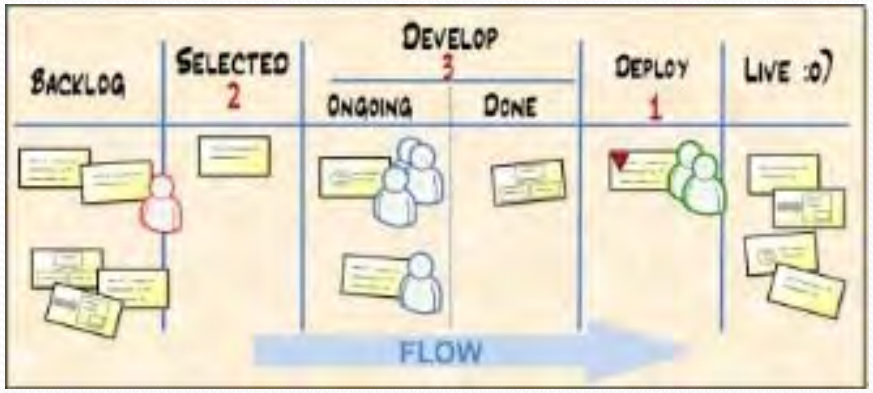


Sulaiman, T. (2010). *AgileEVM*: *Information for good decision making*. San Francisco, CA: CollabNet, Inc.
Sulaiman, T., & Smits, H. (2007). Measuring integrated progress on agile software development projects. *Methods & Tools, 5*(3), 2-9.
Sulaiman, T., Barton, B., & Blackburn, T. (2006). Agile EVM: Earned value management in scrum projects. *Agile 2006 Conference, Minneapolis, Minnesota, USA*, 7-16.
Rico, D. F. (2015). *Lean & agile earned value management*: *How to use EVM to manage projects, programs*, & portfolios, Retrieved from, http://davidfrico.com/rico15v.pdf

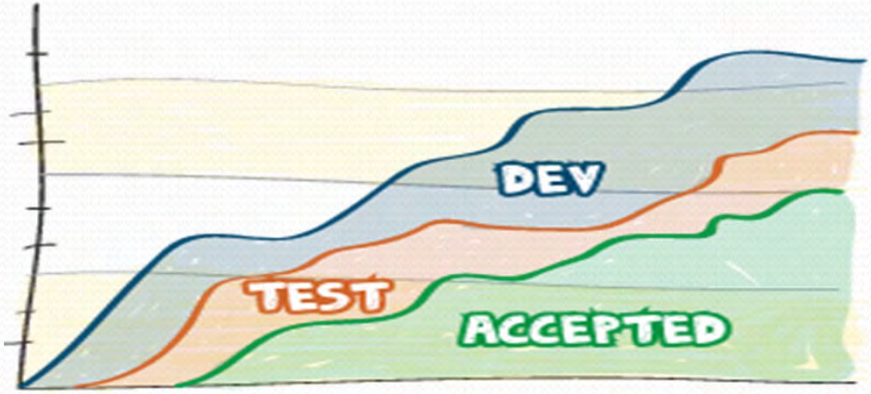
Agile Lean Metrics

- Late big bang integration increases WIP backlog
- Agile testing early and often reduces WIP backlog
- ☞ CI/CD/DevOps lower WIP, Cycle Time, & Lead Time

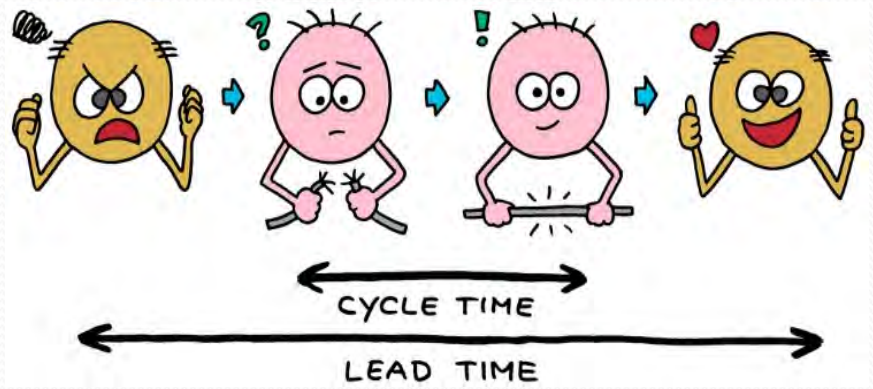
Kanban Board



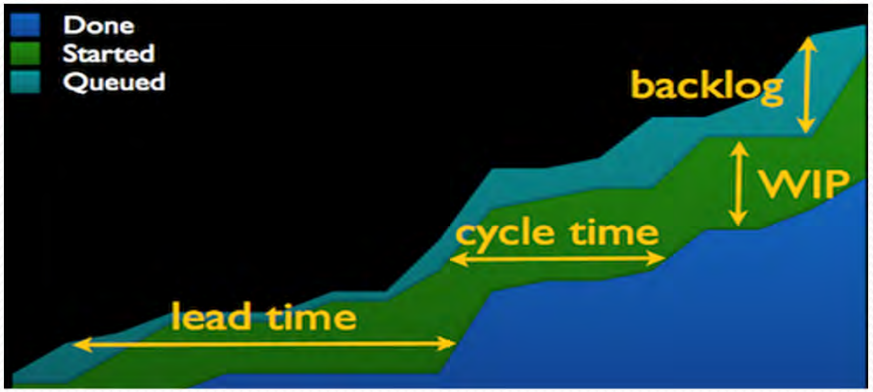
Cumulative Flow Diagram



Lead Time & Cycle Time



Putting it All Together



Nightingale, C. (2015). *Seven lean metrics to improve flow*. Franklin, TN: LeanKit.

Agile SAFe Metrics

- Basic SAFe metrics & assessments at all levels
- Many are rollups of burndown, velocity, & bus. value
- Multi-level kanbans, backlogs, & performance tracking

Portfolio	Lean Portfolio Metrics	Comprehensive but Lean set of metrics that can be used to assess internal and external progress for an entire portfolio.
	Portfolio Kanban	Ensures Epics and Enablers are reasoned and analyzed prior to a PI boundary, prioritized, and have acceptance criteria.
	Epic Burn-up Chart	Tracks progress toward epic completion, i.e., Initial estimate, Work completed, and Cumulative work completed.
	Epic Progress Measure	At-a-glance view of the status of all epics in a portfolio, i.e., Epic X, progress, and current vs. initial est. story points.
	Enterprise Scorecard	Four perspectives to measure performance for each portfolio, i.e., Efficiency, Value delivery, Quality, and Agility.
	LPM Self Assessment	Structured, periodic self-assessment to continuously measure and improve portfolio processes.
	Value Stream KPIs	Set of criteria or KPIs to evaluate value stream investments, i.e., revenues, innovation, intangibles, and lean metrics.
Large Solution	Solution Kanban Board	Ensures Capabilities and Enablers are reasoned and analyzed prior to PI boundary, prioritized, and have acc. criteria.
	Solution Predictability	Aggregation of individual predictability measures for ARTs to assess the overall predictability of Solution Trains.
	Solution Performance	Aggregation of individual performance measures for ARTs to assess the overall performance of Solution Trains.
	Economic Framework	Decision rules to align work to financial objectives of Solution and guide economic decision-making process.
	WSJF	Prioritization model used to sequence jobs (e.g., Features, Capabilities, and Epics) to maximize economic benefit.
	Cost of Delay	A way of communicating the impact of time on the outcomes we hope to achieve, i.e., combining urgency and value.
	Duration (Job Size)	Length of time required to complete an epic, enabler, capability, or feature, i.e., size or complexity in story points.
Program	Feature Progress	Tracks feature and enabler status during PI and indicates which features are on track or behind, i.e., plan vs. actual.
	Program Kanban	Ensures Features are reasoned and analyzed prior to a PI boundary, prioritized, and have acceptance criteria.
	Program Predictability	Aggregation of Team PI Performance Reports to assess the predictability of ART, i.e., planned vs. actual business value.
	Program Performance	Aggregation of team metrics collected at end of PI, i.e., functionality (velocity, etc.) and quality (tests, defects, etc.).
	PI Burn-down Chart	Shows progress toward PI timebox to track work planned for PI against work accepted, i.e., iterations vs. story points.
	Cumulative Flow	Graph to visualize amount of work waiting to be done (backlog), work in progress (started), and completed (validated).
	Art Self Assessment	Structured, periodic self-assessment to continuously measure and improve program processes.
Team	CD Pipeline Efficiency	Measures efficiency of steps in terms of touch and wait time, i.e., analysis, backlog, build, validate, deploy, release, etc.
	Deployments and Releases	Deployment and release frequency progress as a ratio of deployment to production vs. product release frequency.
	Recovery over time	How often physical or logical rollbacks performed by overlaying points in time for deployment, release, and rollbacks.
	Innovation Indicators	Hypothesis measures of MMF and MVP business outcomes based upon actionable innovation accounting measures.
	Hypotheses Tested	Number of successful vs. unsuccessful hypothesis tests (with goal of increasing the number, frequency, and success).
	Team Performance	Individual team metrics collected at end of PI, i.e., functionality (velocity, etc.) and quality (tests, defects, etc.).
	Team Kanban	Ensures Stories and tasks are reasoned and analyzed prior to a PI boundary, prioritized, and have acceptance criteria.
	Team Business Value	Estimate of actual business value achieved for each team's PI objectives during a PI demo by customer and agile team.
	Team Self-Assessment	Structured, periodic self-assessment to continuously measure and improve team processes.

Rico, D. F. (2017). *Scaled agile framework (SAFe). SAFe 4.5 metrics* (*short version*). Retrieved August 29, 2017 from http://www.davidfrico.com

Agile DevOps ROI Metric

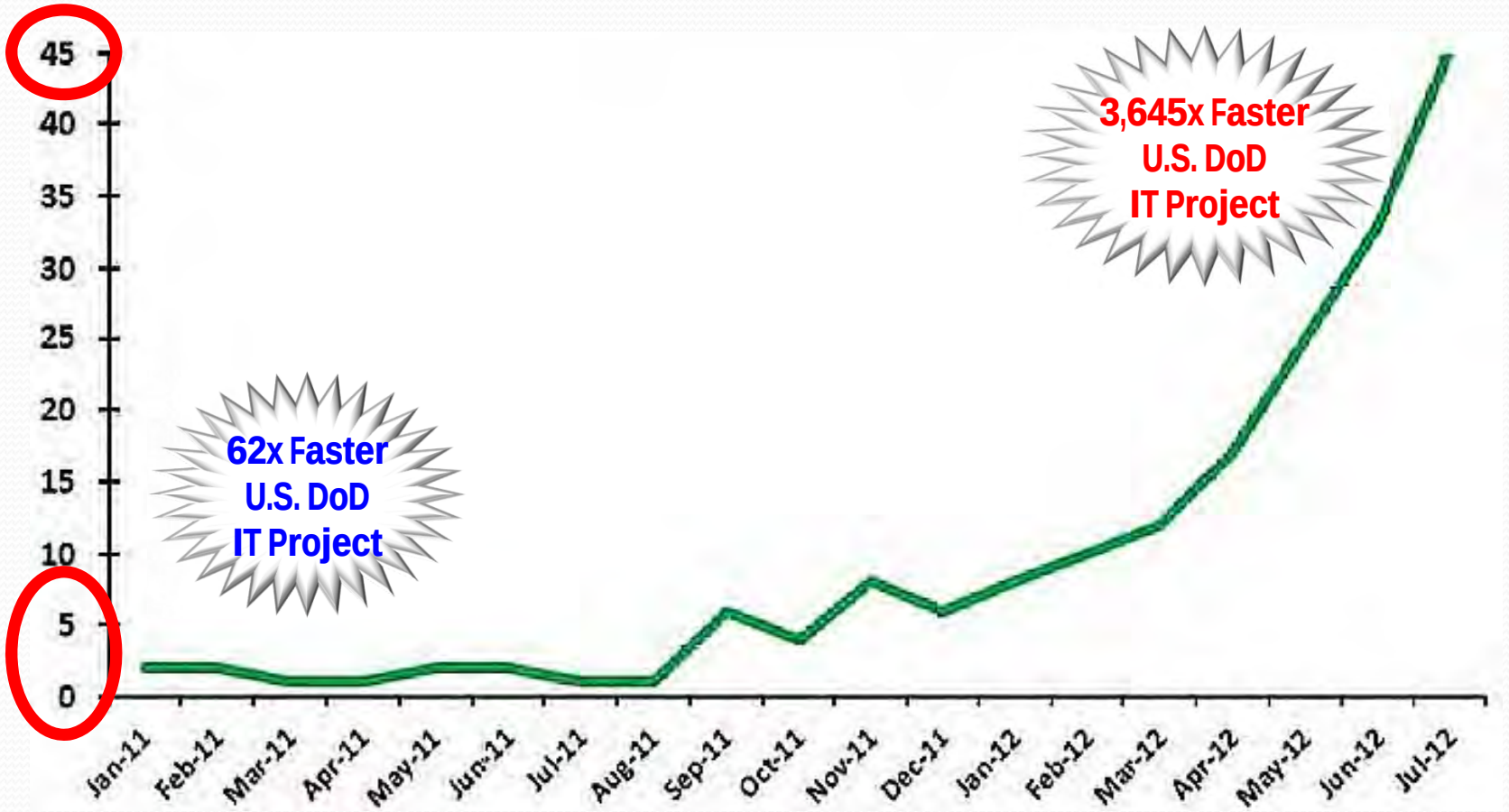
- Detailed agility economics starting to emerge
- ROI ranges from $17M to $195M *with minor costs*
- Benefits from cost savings, revenue, and availability

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 Deployment Metric

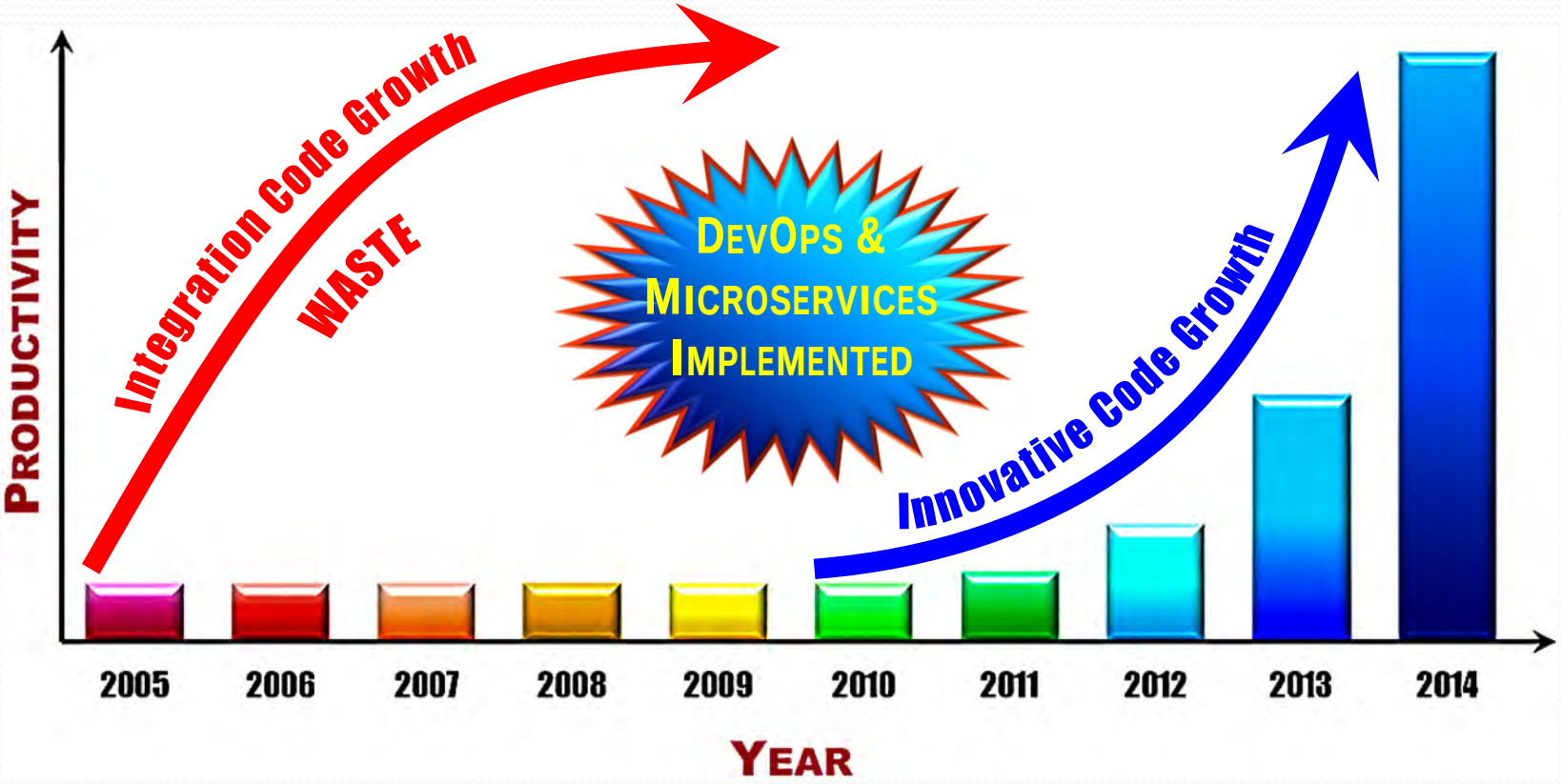
- Assembla went from 2 to 45 monthly releases w/CD
- 15K Google developers run 120 million tests per day
- 30K+ Amazon developers deliver 8,600 releases a day



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

Agile Microservices Metric

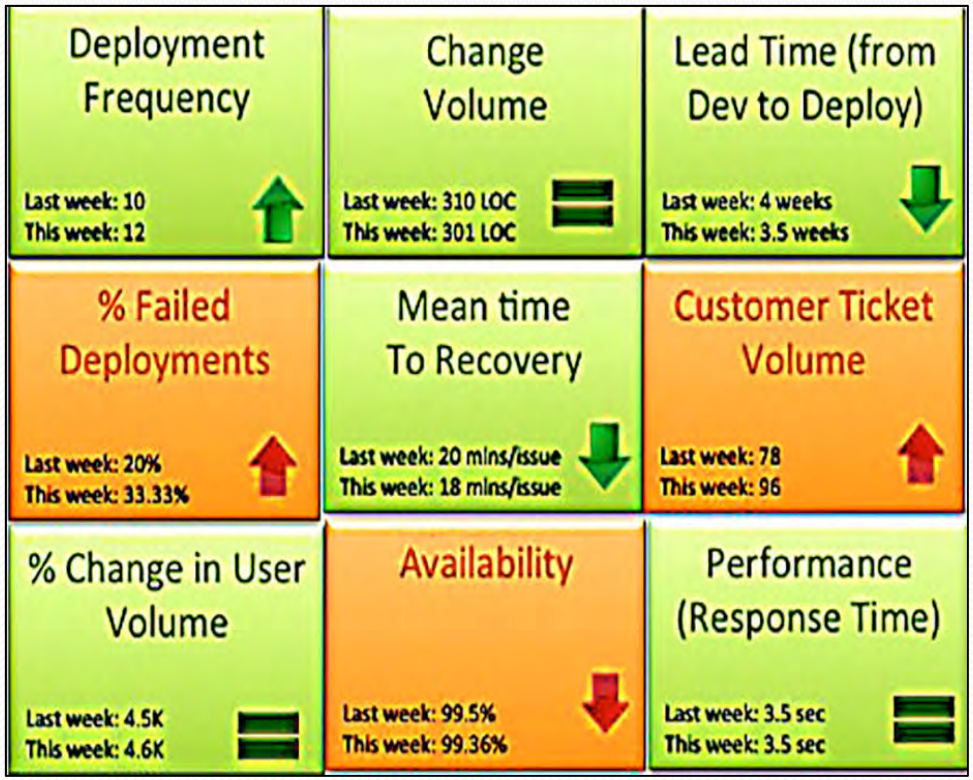
- Productivity **STOPS** due to excessive integration
- Implements DevOps & Microservices around 2010
- Waste elimination, productivity & innovation skyrocket

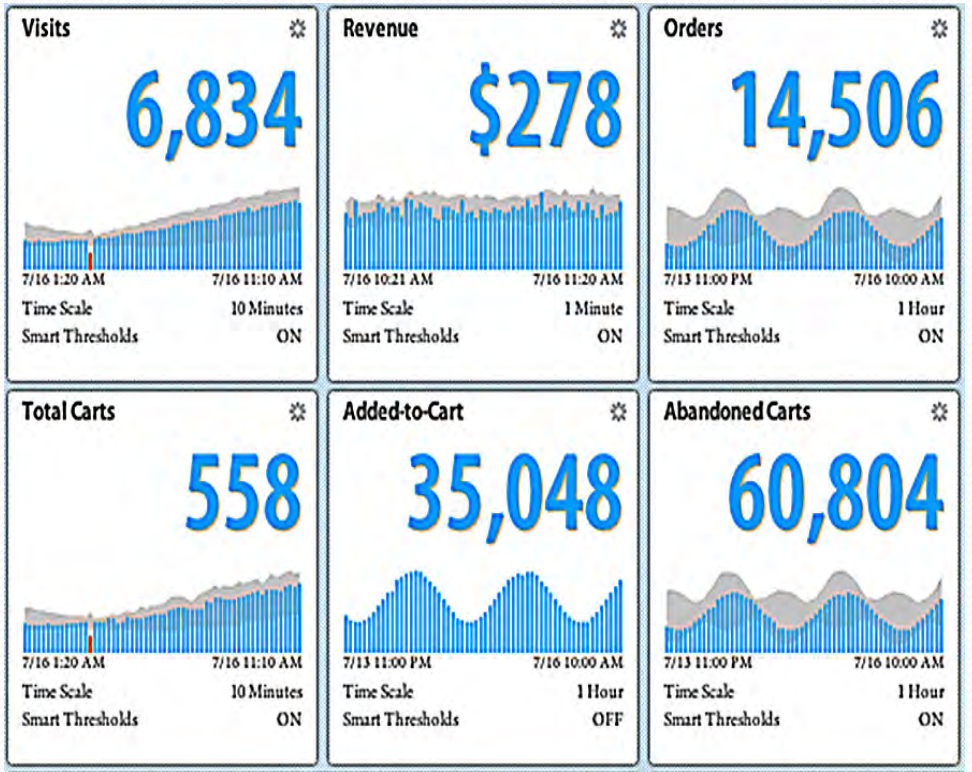


Ashman, D. (2014). Blackboard: Keep your head in the clouds. *Proceedings of the 2014 Enterprise DevOps Summit, San Francisco, California, USA.*

Agile DevOps Metrics

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





Velasquez, N. F. (2014). *State of devops report*. Portland, OR: Puppet Labs, Inc.

Agile DevOps Metrics—Example

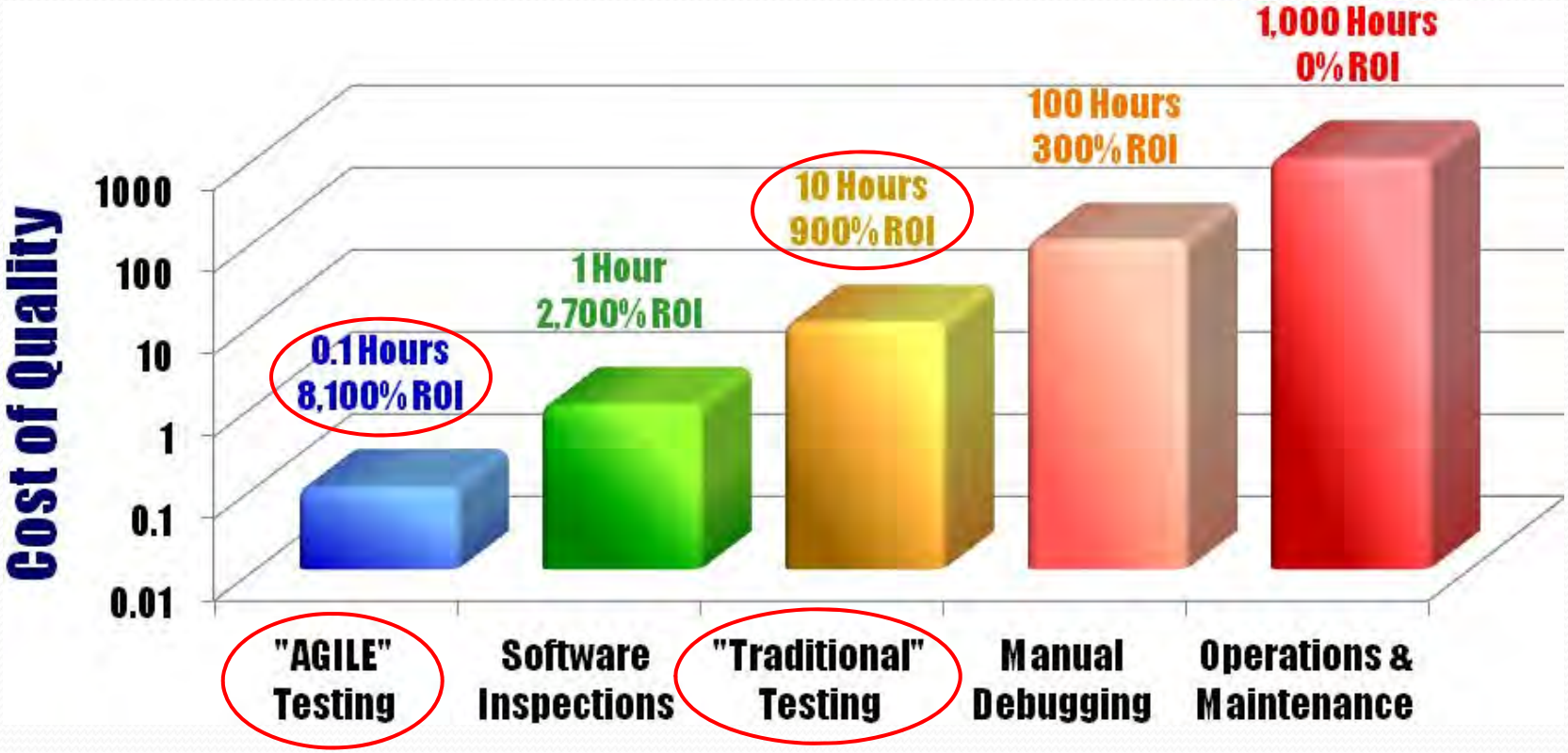
- Hewlett-Packard is a major user of CI, CD, & DevOps
- 400 engineers developed 10 million LOC in 4 years
- Major gains in testing, deployment, & innovation

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 Cost of Quality Metric

- Agile testing is 10x better than code inspections
- Agile testing is 100x better than traditional testing
- Agile testing is done earlier "and" 1,000x more often

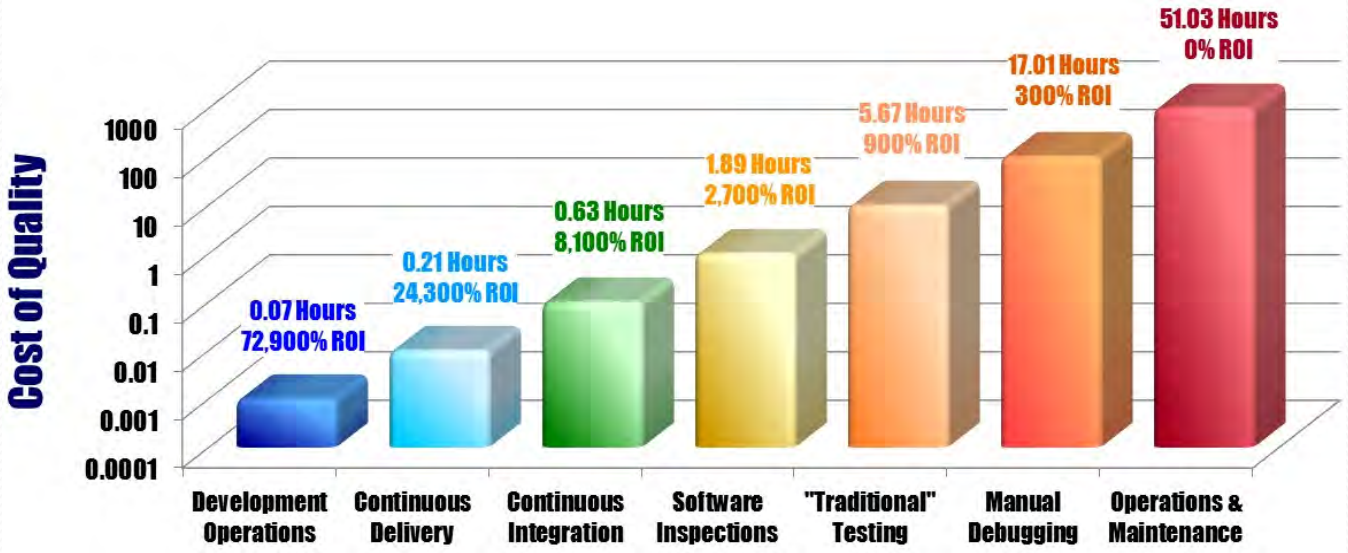


Rico, D. F. (2012). *The Cost of Quality (CoQ) for Agile vs. Traditional Project Management.* Fairfax, VA: Gantthead.Com.

Agile DevOps CoQ Metric

- 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



Rico, D. F. (2016). *Business value, ROI, and cost of quality (CoQ) for DevOps*. Retrieved May 10, 2016, from http://davidfrico.com

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

Agile Business/Enterprise Metrics

8. Capital Infrastructure Agility

1. Strategic Agility

7. Organization Design Agility

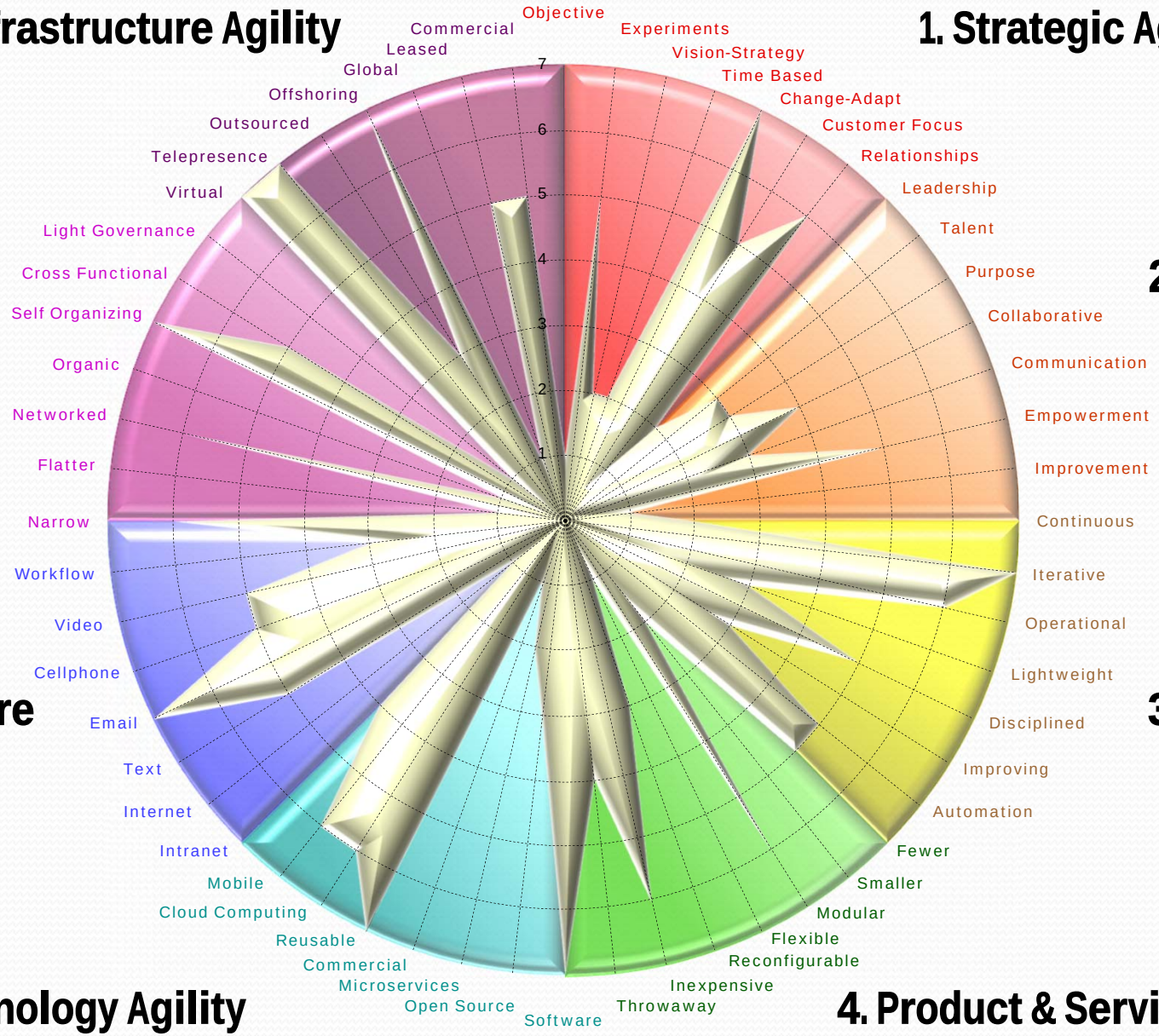
2. Cultural Agility

6. IT Infrastructure Agility

3. Process Agility

5. Technology Agility

4. Product & Service Agility



Rico, D. F. (2016). *Agile businesses*: *A metamodel of lean and agile organizational strategies*. Retrieved March 1, 2016, from http://davidfrico.com

Generic Gov't/Commercial Metrics

Strategic

Increase (Commercial)

- Products & Services
- Product Safety & Reliability
- Reputation, Image, & Brand Equity
- Customers
- Marketshare
- Sales
- Revenues
- Profits
- Return on Investment

Increase (Government)

- Mission Efficiency & Effectiveness
- National Security & Safety Posture
- Identification of High-Value Targets
- Actionable Intelligence
- Intelligence Value Estimate
- Exploit Multiple Signal Sources
- Exploit Emerging Signal Sources
- Exploit Emerging Missions & Threats
- Strategic & Tactical Military Readiness

Operational

Reduce (Commercial & Gov't)

- Technical Complexity, Scale, & Size
- Development, Test, & Evaluation Costs
- Cycle Time & Delivery Speed
- Rework, Defects, Faults, & Failures
- Cost, Schedule, & Budget Overruns
- Turnover, Attrition, & Knowledge Loss
- Programmatic & Technical Risks
- Tech Obsolescence & Legacy Sys. Cost
- Hardware & Software Purchasing Time
- Integration & Interoperability Costs

Increase (Commercial & Gov't)

- Efficiency & Effectiveness
- Delivery Order Quantity (DoQ)
- Speed, Productivity, & Competitiveness
- Innovation, New Ideas, & Technology
- Morale, Retention, & Emp. Satisfaction
- Communication & Knowledge Sharing
- Cust. Satisfaction, Loyalty, & Retention
- Faster Tech. & Infrastructure Refresh
- Decisionmaking & Governance Speed
- Certification & Accreditation Speed

Generic Gov't/Project Mgt. Metrics

- Gov't projects require broad range of measurements
- Many gov't measures are qualitative vs. quantitative
- Data gathered by meetings, phone, & conversations

- **Schedule Performance** - *Cost & schedule performance indices (using Earned Value Mgt).*
- **Top 10 Risks** - *Critical cost, schedule, & technical risks (using Probability Estimates).*
- **Staffing Levels** - *Is project fully staffed (especially with critical technical resources)?*
- **Budget Performance** - *Is project achieving total budget parameters (at agreed rates)?*
- **Top Accomplishments** - *Is project achieving early, value-adding performance objectives?*
- **CDRL Performance** - *Is project delivering minimal contractual deliverables & reports?*
- **Emerging Needs** - *Are there any unforeseen changes in scope (new value propositions)?*
- **Performance Delays** - *Are there significant performance delays (internal dependencies)?*
- **Training Performance** - *Have required & ongoing training requirements been satisfied?*
- **Contract Performance** - *Have contracts & subcontracts been awarded (& executing)?*
- **External Dependencies** - *Are there major performance delays (external dependencies)?*
- **Facilities/Equipment Issues** - *Have all facility & equipment requirements been satisfied?*
- **Travel Issues** - *Are travel resources, policies, & requirements completely satisfied?*
- **Special Activities** - *What's the status of research, prototyping, & exploratory "what-ifs"?*