

High performance. Delivered.

Metrics That Matter

in the context of software testing



consulting | technology | outsourcing

Discussion Outline

- Do Metrics Matter?
- The Problem & The Need
- Metrics and Room For Misleading Interpretation
- Test Metrics Framework Vision
- Just-Enough Metrics
- How To Get Started
- Conclusion

The Problem

- Management has little patience for detailed status reports.
- Management doesn't understand testing.
 - Testing is confused with improving.
 - Testing is considered a linear, independent task.
 - Testing is assumed to be exhaustive.
 - Testing is assumed to be continuous.
 - Test results are assumed to stay valid over time.
 - Impact of regression testing is not appreciated.
 - Test metrics are hard to interpret.

The Need – Somebody Needs Some Information

- “What’s the status of testing?”
- “What are you doing today?”
- “When will you be finished?”
- “Why is it taking so long?”
- “Have you tested _____, yet?”
- “What’s the quality of _____?”
- “Why did you not find all development defects?”

Need for Control

Consider two very different projects:

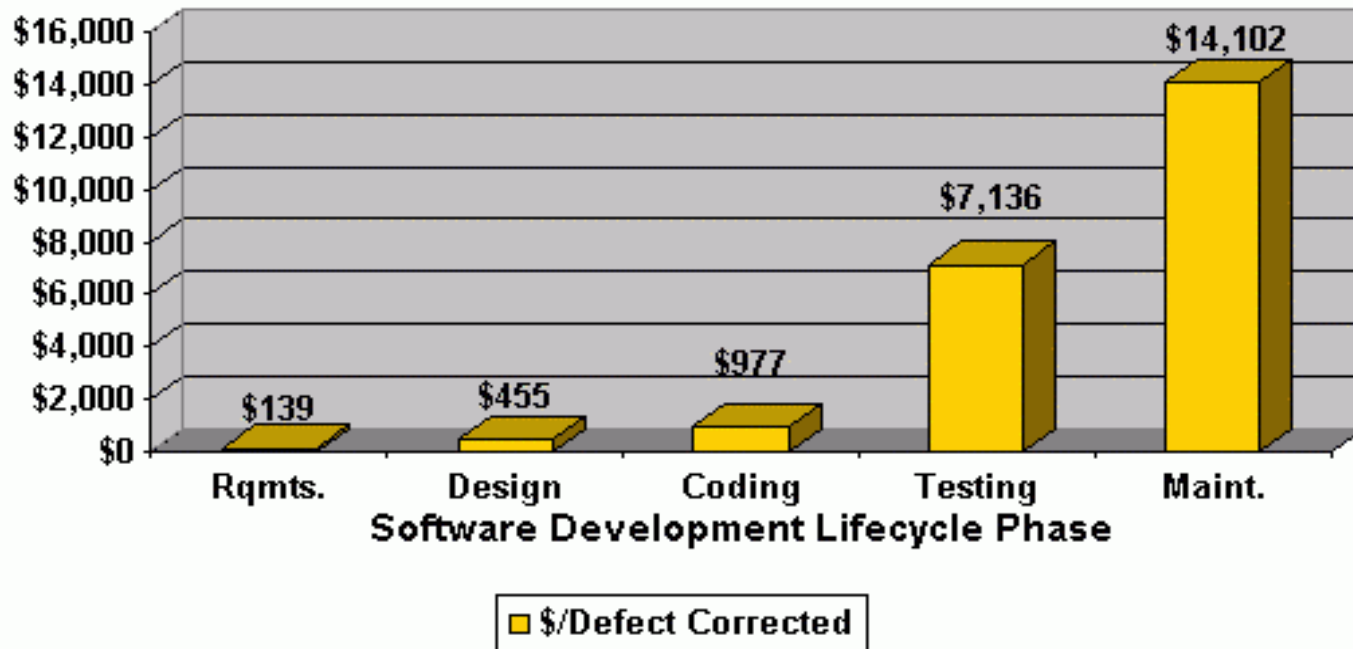
Project	Project A	Project B
Cost:	\$ 1.0M	\$ 1.0M
Value delivered:	\$ 1.1M	\$ 50.0M

Which project needs control?

Cost per Defect (Boehm & Basili)

Costs of Correcting Defects

Source: B. Boehm and V. Basili, "Software Defect Reduction Top 10 List," *IEEE Computer*



This industry average is used as a baseline for arriving at cost avoidance

Industry References: 3 B. Boehm and V. Basili, "Software Defect Reduction Top 10 List," *IEEE Computer*, IEEE Computer Society, Vol. 34, No. 1, January 2001, pp. 135-137.

Cost Per Defect (Laurent Bossavit)

But “average cost to fix one defect” is a stupid metric, as Capers Jones argues in a paper on “A Short History of the Cost Per Defect Metric” (see Jones²⁷). It makes bad projects look good, and good projects look bad.

How? By failing to divide the costs of fixing into two categories: *fixed* costs of detecting and fixing defects - costs which are the same no matter how buggy or how good the product is - and *variable* costs, those which you pay for each defect.

The more defects you have, the more your fixed costs get spread around, and the *lower* your “average cost per defect”. The better your quality, the fewer defects you have, the *higher* the average will be.

“The Leprechauns of Software Engineering - How folklore turns into fact and what to do about it”; Laurent Bossavit, 2013

Test Metrics Framework Vision

“Lord Kelvin, a renowned British physicist, is reputed to have said:
‘When you can measure what you are speaking about, and express it
in numbers, you know something about it ... [otherwise] your
knowledge is of a meager and unsatisfactory kind; it may be the
beginning of knowledge, but you have scarcely in thought advanced
to the stage of science.’”

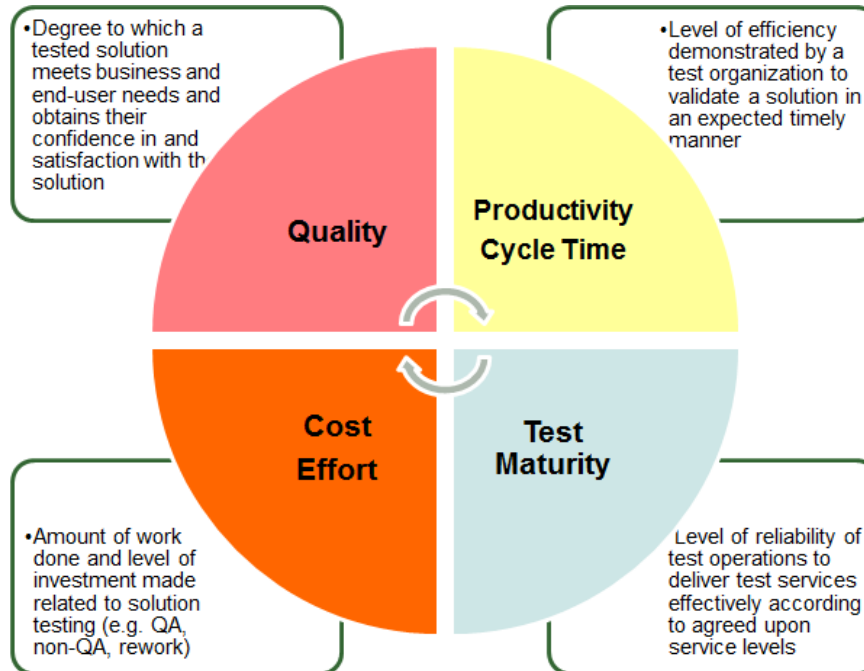
- Measuring Business Performance, A. Neely, The Economist Books, 1998



Test Metrics Framework Vision

- Making fact-based decisions and eliminate emotions
- Influence the behavior of project stakeholders
- Gauge the quality of delivered solutions
- Measure the return on investments (ROI) made to optimize delivery of IT solutions
- Improve efficiency of IT organizations and corresponding processes
- Improve maturity of test operations
- Get a grip around cost

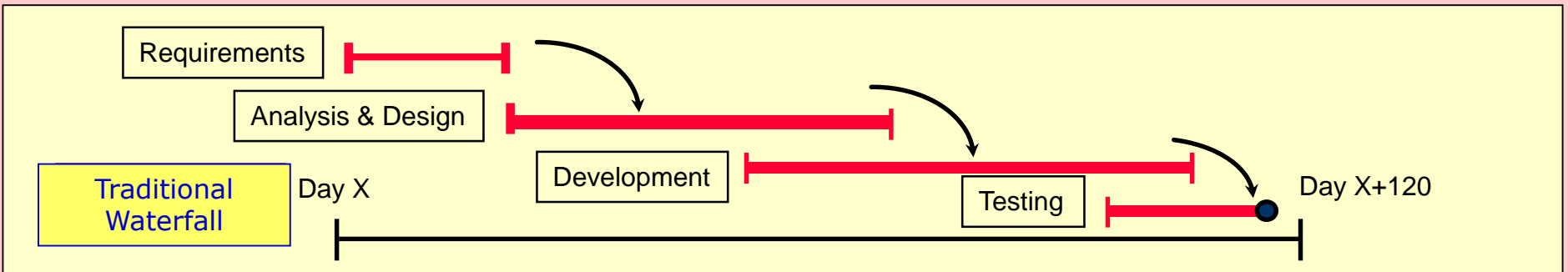
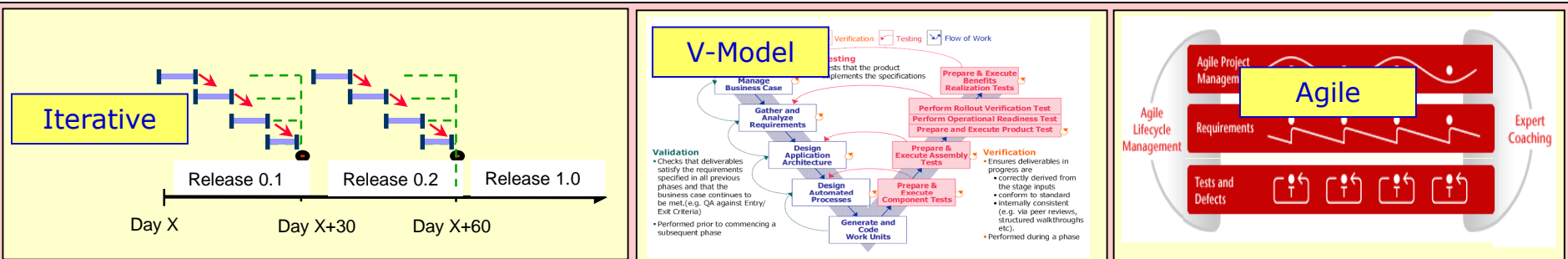
Accenture Test Metrics Framework



- *Agile⁺*
- *Analytics*
- *Mobility*
- *Digital*
- *Automation⁺*



Test Metrics – By Software Development Life Cycle

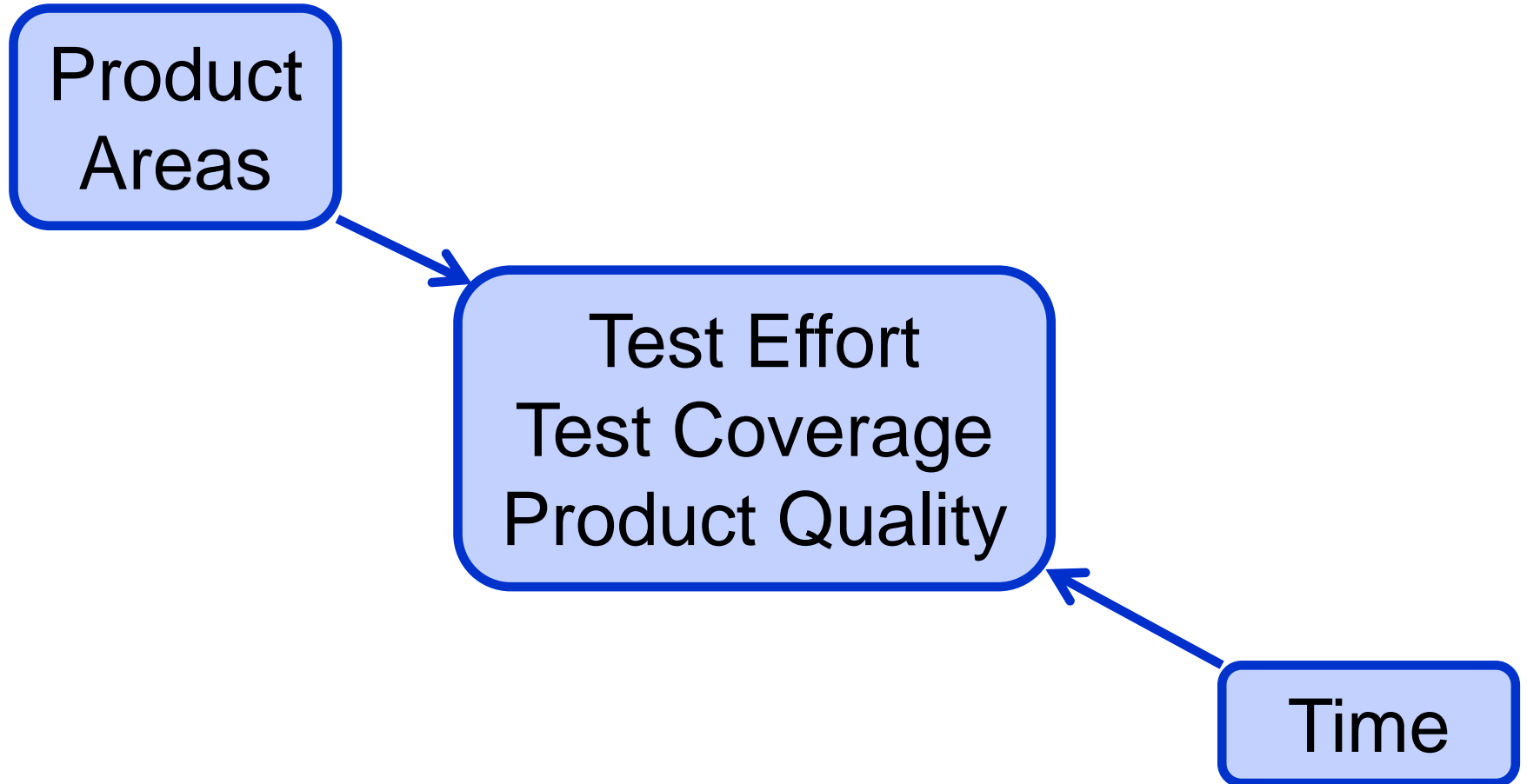


Performance Test & Validation					Production / Operational Cost & System Availability
Test Automation & Regression Test					
Component Test	Assembly Test	Product Test	UAT	Operational Readiness	
Test Planning, Preparation, Execution & Validation					
Test Strategy, Test Approach, Test Plan, Test Cycles, Test Closure Memo					
Test Metrics					

- Test Methods vary in scope, delivery cycles and stakeholder involvement
- All test methods require a standard set of test artifacts created and maintained throughout the life cycle at defined milestones
- **Test Metrics** support the entire SDLC from requirements to production

Just-Enough Metrics

What do we need in a testing dashboard?









Low-Tech Dashboard

Product Area	Effort	Coverage	Quality	Comments
Inventory management				
Order entry				
Fulfillment				
Customer service				
Accounts receivable				
Finance				
Warehouse management				




Low-Tech Dashboard – Test Effort

None	Not testing; not planning to test.
Start	No testing yet; expect to start soon
Low	Regression or smoke testing only; maintaining coverage
High	Focused testing effort; increasing coverage.
Paused	Temporarily ceased testing, though area is testable.
Blocked	Can't effectively test, due to blocking problem.
Final	Going through final tests and signoff procedure















Low-Tech Dashboard – Test Coverage

0		We have no good information about this area.
1		Smoke tests: major functions & simple data
1+		More than sanity, but many functions not tested.
2		Common cases: all functions touched; common & critical tests executed.
2+		Some data, state, or error coverage beyond level 2.
3		Corner cases: strong data, state, error, or stress testing.

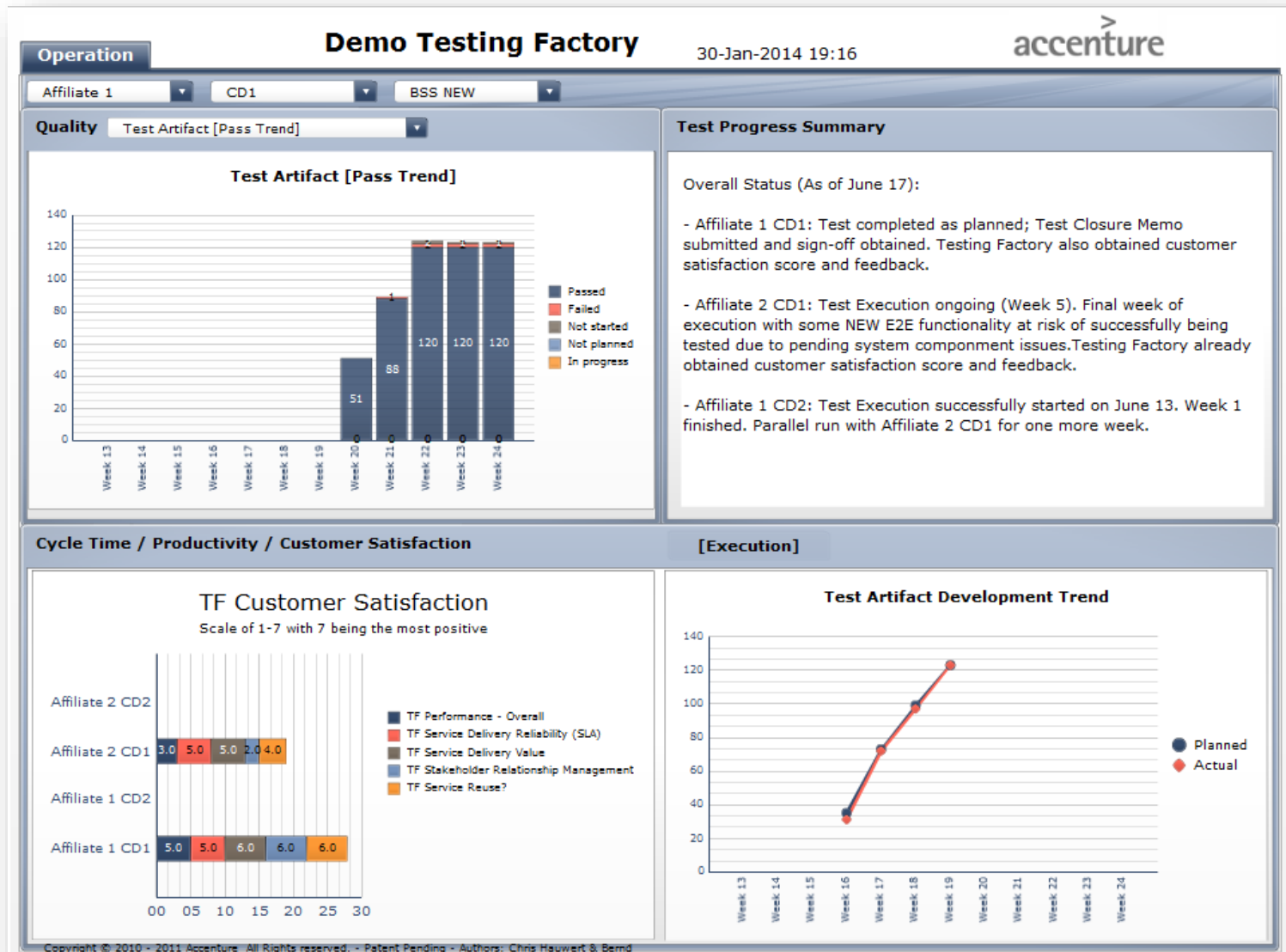
Low-Tech Dashboard – Product Quality

Green		“We know of no problems in this area that threaten to stop deployment or interrupt testing, nor do we have any definite suspicions about any.”
Yellow		“We know of problems that are possible showstoppers, or we suspect that there are important problems not yet discovered.”
Red		“We know of problems in this area that definitely stop deployment or interrupt testing.”

Low-Tech Dashboard

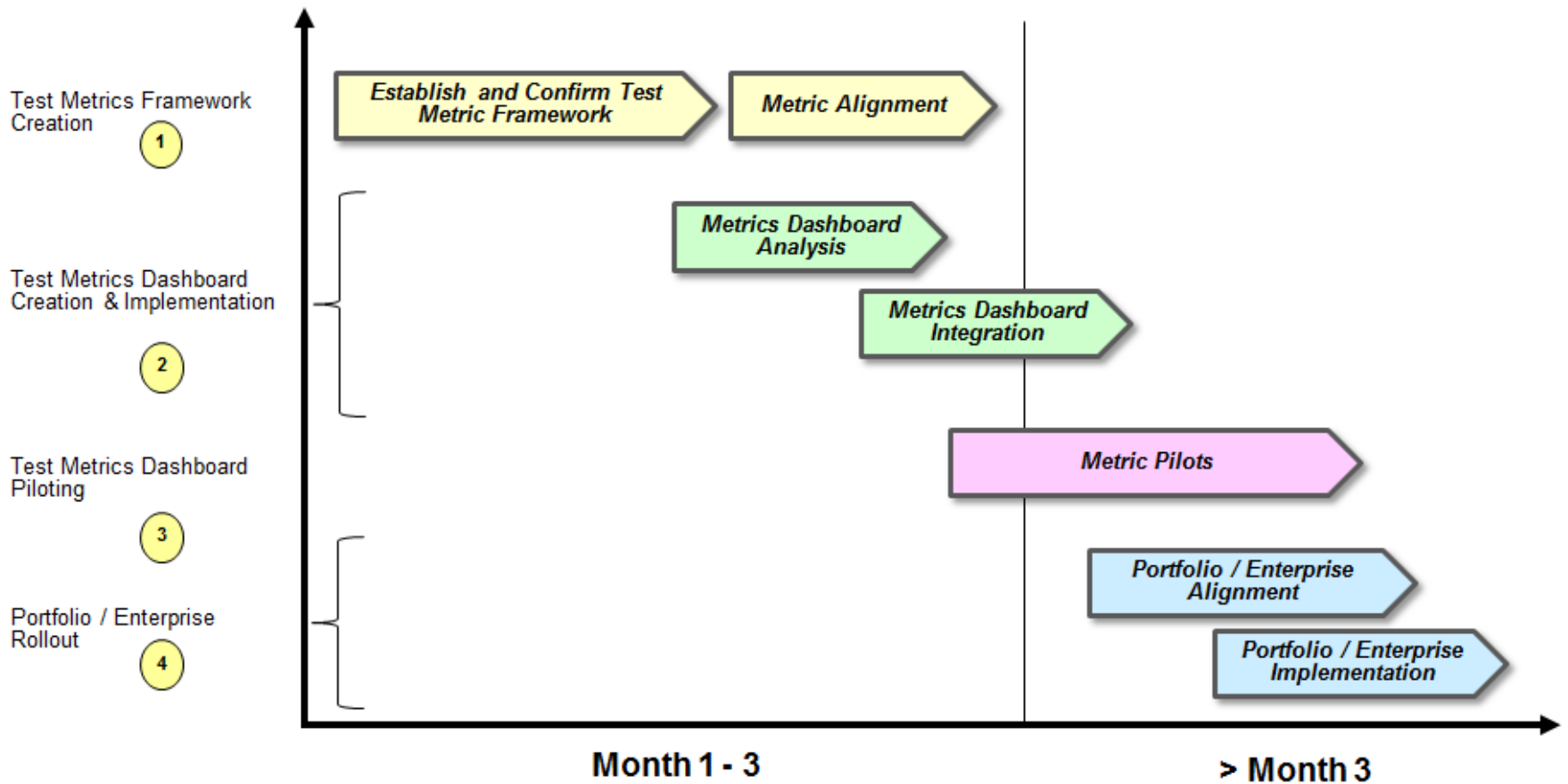
Product Area	Effort	Coverage	Quality	Comments
Inventory management	Start			Loading master data
Order entry	High			
Fulfillment	High			Automation broken
Customer service	Final			
Accounts receivable	Low			Need help to test
Finance	Blocked			Crashes – see #1406, #1407
Warehouse management	None			Data not migrated

Light-Weight Dashboard – Crystal Dashboard



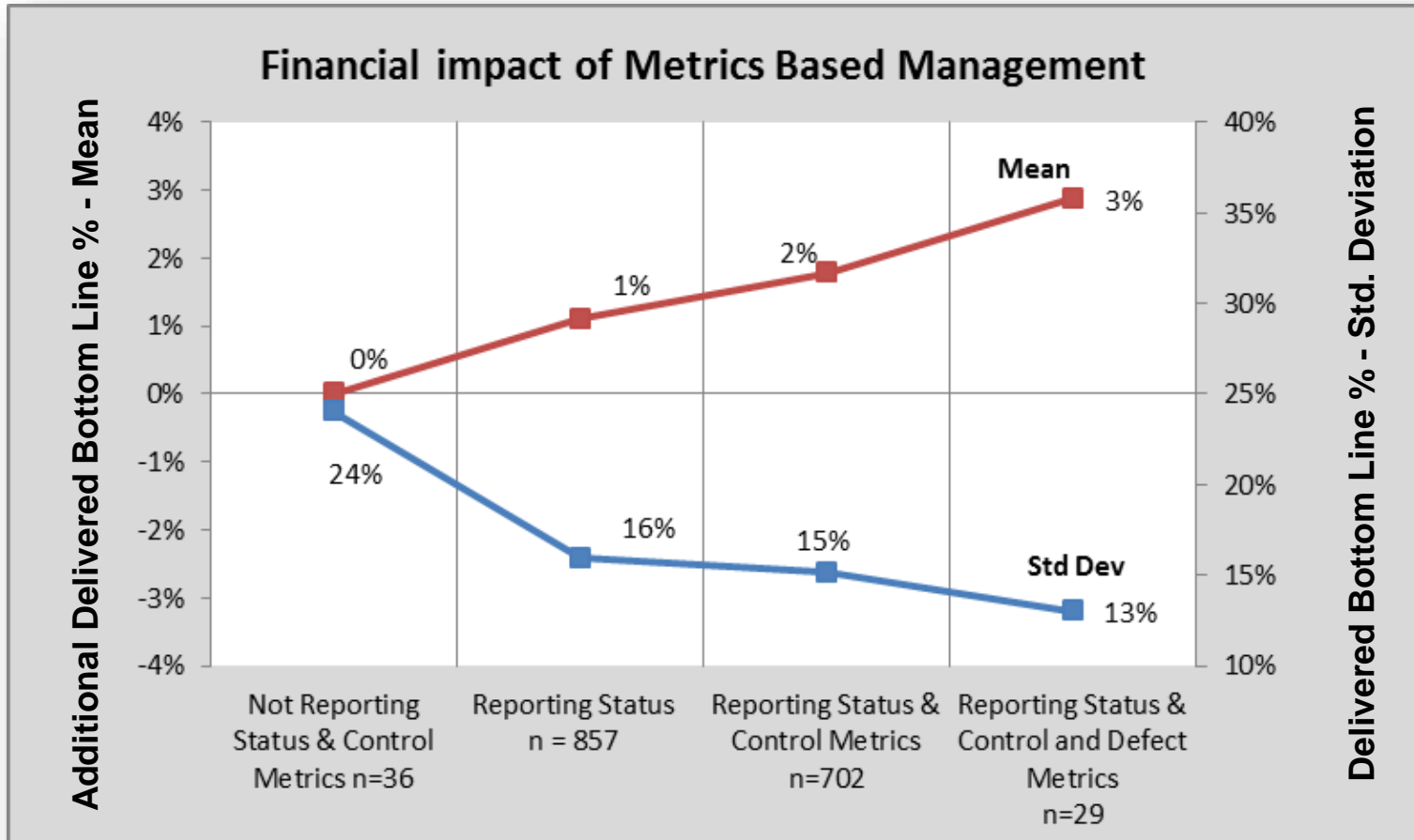
Test Metrics Program – How To Get Started

Setting up a standard test metric reporting capability requires a project approach that includes various phases, such as analysis, build, piloting and rollout. Key is to align with stakeholders, such as project managers or IT delivery teams on the benefits of having metric standardization.



Conclusion

Metrics Matter in A Proper Context



Financial bottom line analysis of top 1,000 IT engagements demonstrates a pronounced positive differentiation for engagements reporting both status and metrics in comparison to those doing neither or simply doing status reporting. Financial impact is further enhanced for engagements focusing on quality management processes (track and report defect metrics).