Performance Measurement in CMMI: A Focus on Variation Among Otherwise Similar Programs

Dennis R. Goldenson
Software Engineering Institute

11th Annual Practical Software and Systems Measurement Users' Group Conference: Defining the Future of Measurement Practice

Golden, Colorado – 26 July 2007

© 2007 Carnegie Mellon University

Acknowledgments

Special thanks are due to Sasha Babkin, Ruth Buys, Bob Ferguson, Wolf Goethert, Rick Hefner, Mark Kasunic, Angel Liu, Natalia Matvienko, Larry McCarthy, Pete McLoone, Lynn Penn, Jim Perry, Jennie Qi, Rolf Reitzig, Debra Roy, Diego Rubio, V A Santhakumar, Millee Sapp, Bob Stoddard & Dave Zubrow.

© 2007 Carnegie Mellon University

Today’s Talk

Setting the stage

Our approach & Some Example results

How can you help?

© 2007 Carnegie Mellon University

The talk in a nutshell

A two-fold problem:

- Need for credible evidence of CMMI effect
- Need for better measurement & analysis in our field

A solution ... that addresses both

- Analyze variation & covariation properly
- With a focus on modeling cause & effect statistically
- & with data initially from high maturity organizations

© 2007 Carnegie Mellon University

Why is this Work Important?

Substantial evidence exists

- CMMI-based process improvement can & has led to concomitant improvement in performance outcomes
- Predictably faster, better, cheaper product development & maintenance

Skepticism exists about the value of disciplined adherence to well defined processes ... & CMMI in particular

- As opposed to solutions *de jour*
- That are ***not*** necessarily at odds with processes that satisfy the goals of CMMI best practices – e.g., Agile or Six Sigma methods

© 2007 Carnegie Mellon University

Current State

Case studies

- With quantitative evidence of process improvement & concomitant performance gains
- Typically showing total results over time ... often rolled up over multiple projects

Often accompanied by qualitative affirmations

- Based on experience of those doing the work
- That process improvement is the major source of change

But little explicit discussion of competing explanations

- Whether or how they were considered
- Leading to accusations of spurious correlation

© 2007 Carnegie Mellon University

What's Missing?

Generalizable comparative analyses are needed

- On performance outcomes of ***differences*** in process enactment
- Under ***varying*** organizational circumstances & product characteristics
  - ***that may affect variation in both successful process enactment & performance outcomes***

With more attention to:

- Issues of data integrity & validity of comparisons across organizational units
- Choice of appropriate multivariate statistical techniques
  - To instantiate process performance models
  - Looking at common cause of process effects mediated by project context

© 2007 Carnegie Mellon University

Understanding Variation & Causal Thinking

There almost ***always*** is variation

- In product quality, project & organizational performance
- How processes are enacted
  - The existence of defined processes
  - Adherence/compliance with them
  - & how well the processes are enacted ... the “goodness” issue
- & the other factors that may effect both process & performance outcomes

Looking only at total results

- May mask important differences at the project level ... where most of the development work actually takes place

© 2007 Carnegie Mellon University

Today's Talk

Setting the stage

<u>Our approach & Some Example results</u>

How can you help?

© 2007 Carnegie Mellon University

Overview

Work with larger enterprises

- Comprised of multiple projects, contractual programs, & other constituent organizational units
- That already have &/or are evolving common measures of performance
- Along with complimentary measures of process enactment, organizational & product attributes

An important way to provide:

- More generalizable, comparative analyses explaining variation
- In both successful process enactment & the performance outcomes that the processes are meant to achieve

Especially important *in lieu* of shared measures that enable wider state of the practice & benchmarking analyses

© 2007 Carnegie Mellon University

Analyzing Covariation

Analyses of joint distributions among two or more variables can be very helpful

- Better insight into reasons for anomalies& inconsistencies in process enactment and performance outcomes
- More informed conjectures about opportunities for improvement
- Enhancing root cause analysis ... benchmarking if you will, e.g.,
  - BAE Minneapolis examines details of inspection process enactment, e.g., code size per review, team skill makeup, frequency of inspections
  - Proactively before refining process definitions as well as to monitor compliance with existing processes
  - Similar monitoring at Motorola Software Group (MSG)

© 2007 Carnegie Mellon University

Work with Motorola Software Group

MSG performance & project context data definitions are well defined.

- However, process capability needs to be measured more consistently & analyzed more explicitly

Process compliance/adherence ***is*** measured & monitored at MSG.

- Used regularly in quantitative project management & causal analysis
- & used regularly in piloting & deploying new processes & technologies

However, ***covariation*** with performance & other factors could be analyzed more proactively in explicit process performance models.

- Shared “how-to" process measures are crucial for using benchmarks to identify promising benchmarking opportunities
- Data as tightly coupled as possible to CMMI-based "what" process measures also important for internal Motorola purposes

© 2007 Carnegie Mellon University

Cost of Quality at Motorola Software Group (MSG)

Cost of Quality (COQ) at MSG includes effort spent on...

- Review / Inspection
- Test development & execution
- Quality auditing, training, other process Improvement & problem prevention

Cost of Poor Quality (COPQ) includes...

- Rework & related failure correction throughout the life cycle

Both expressed as percentages of overall effort spent for product development

Results presented here examine ***test development & execution***

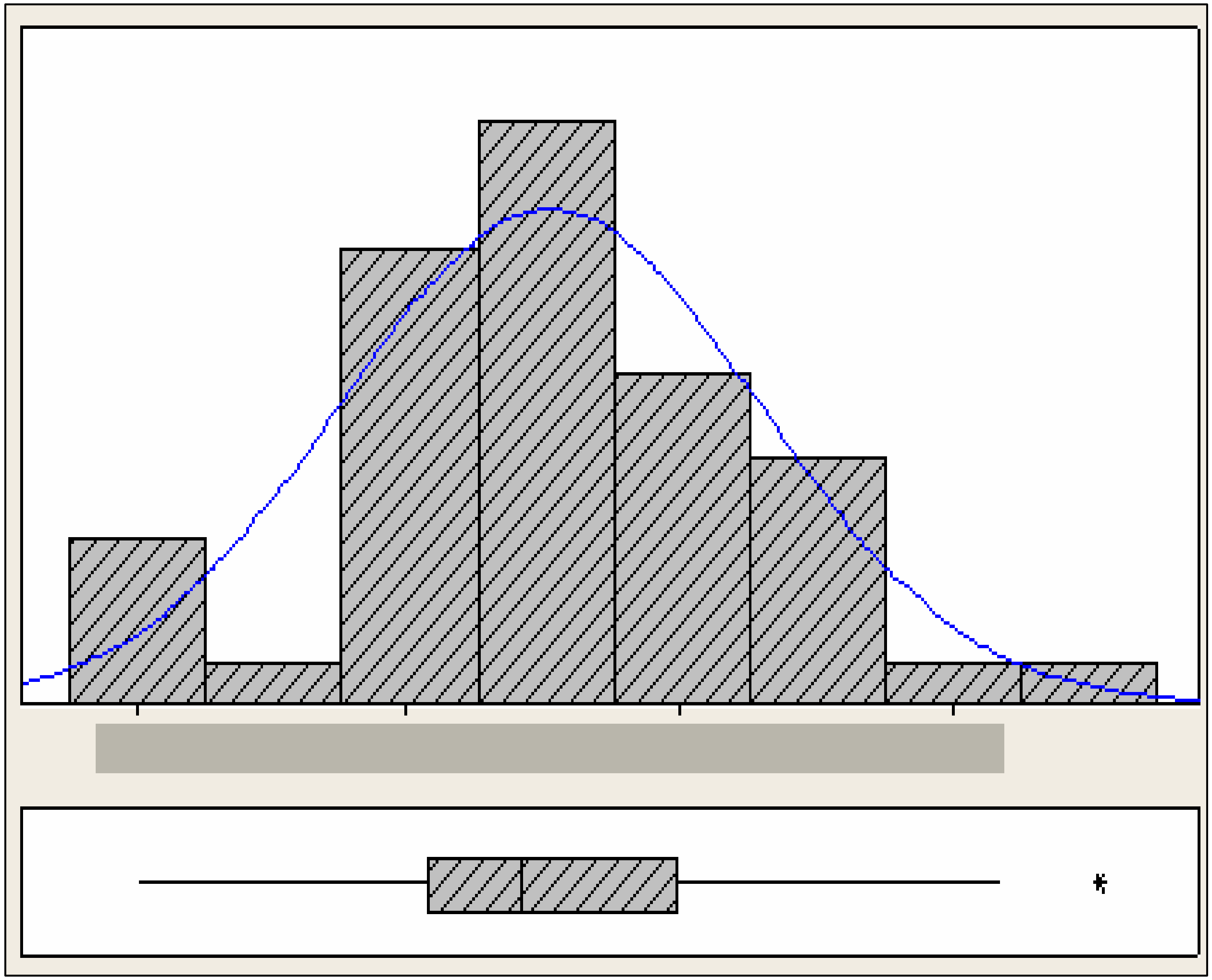
- Proactive drill-downs
- Initial results from 2002 ... N = 46 projects

© 2007 Carnegie Mellon University

Variation in Cost of Quality for Test Appraisal

Results from MSG China Center

- N = 46
- Actual values of data distribution (not shown here) are quite good by our sense of "industry standards"
- Still, there are variations that MSG wishes to reduce further
- **A critical factor for COQ reduction**



© 2007 Carnegie Mellon University

Differences by Domain

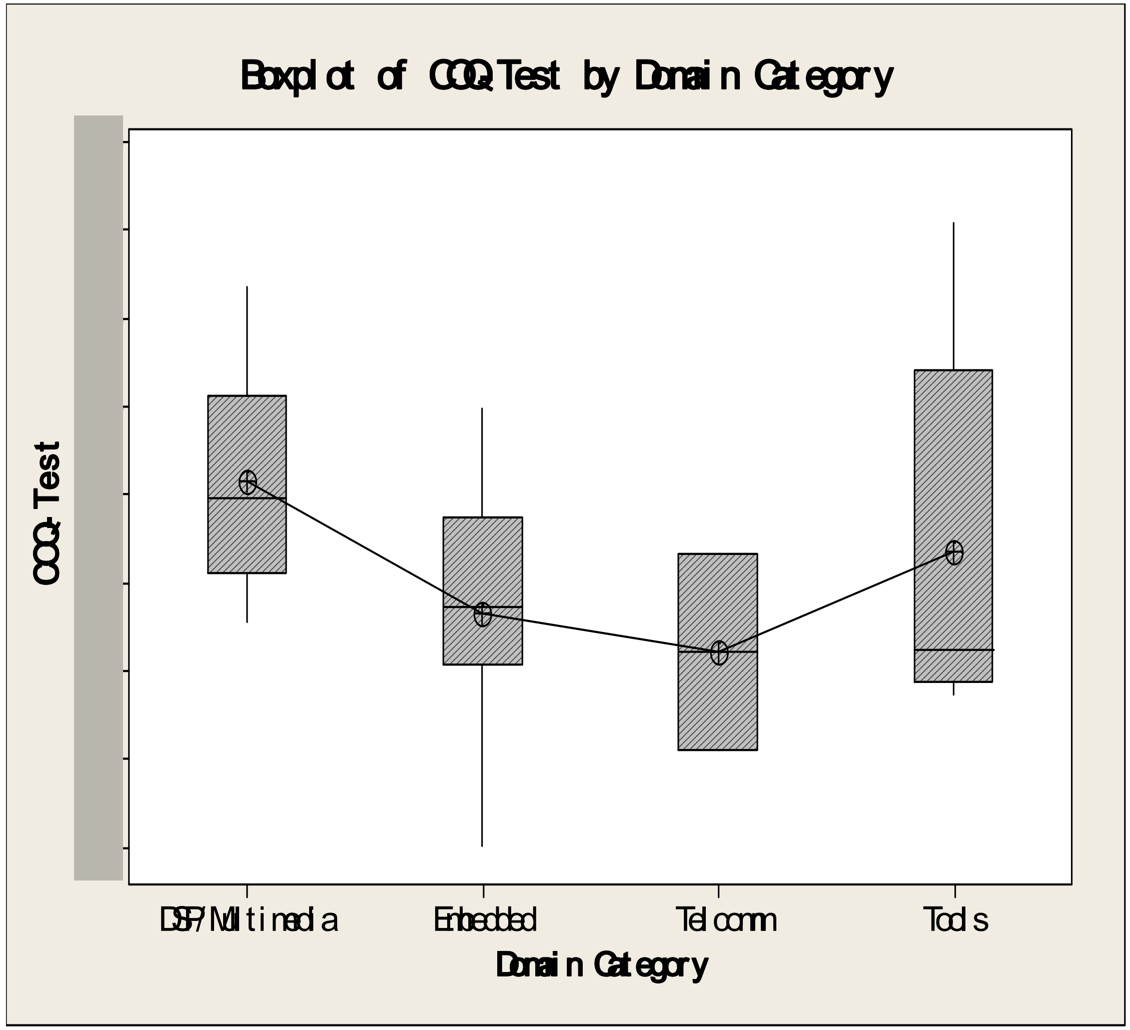
DSP/Multimedia projects have higher COQ-test

- Most Assembly projects fall here
- Some porting & optimization-oriented development projects, with high performance requirements
- & some product-oriented projects

Embedded vary more

Ns:

- 10 DSP Multimedia
- 29 Embedded
- 2 Telecom; 5 Tools



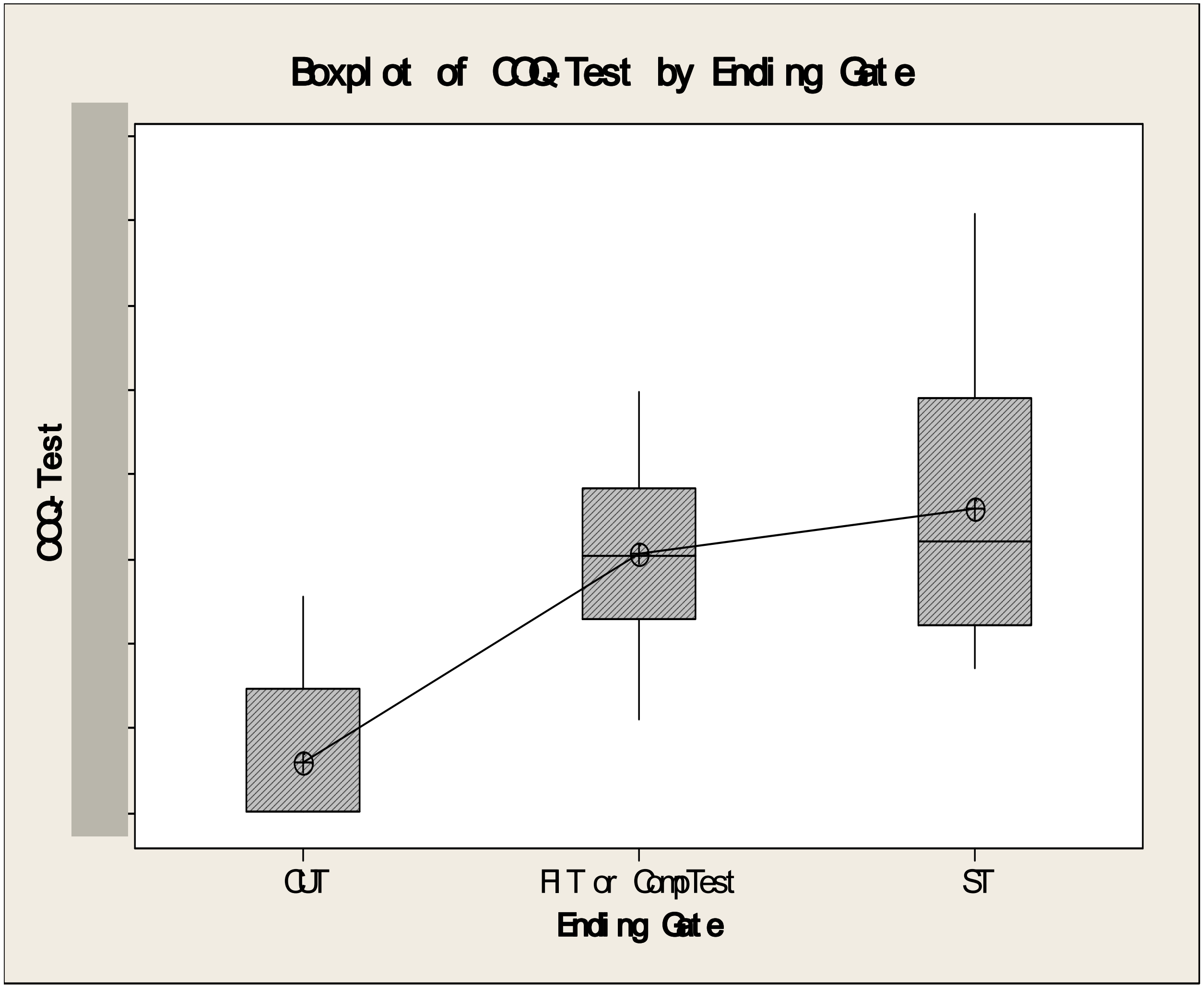
© 2007 Carnegie Mellon University

Differences by Motorola "End Gates"

Projects ending at System Test often have higher COQ-Test

Ns:

- 5 Code & Unit Test (CUT)
- 20 Component Test & Feature Integration Test (FIT)
- 21 System Test (ST)



© 2007 Carnegie Mellon University

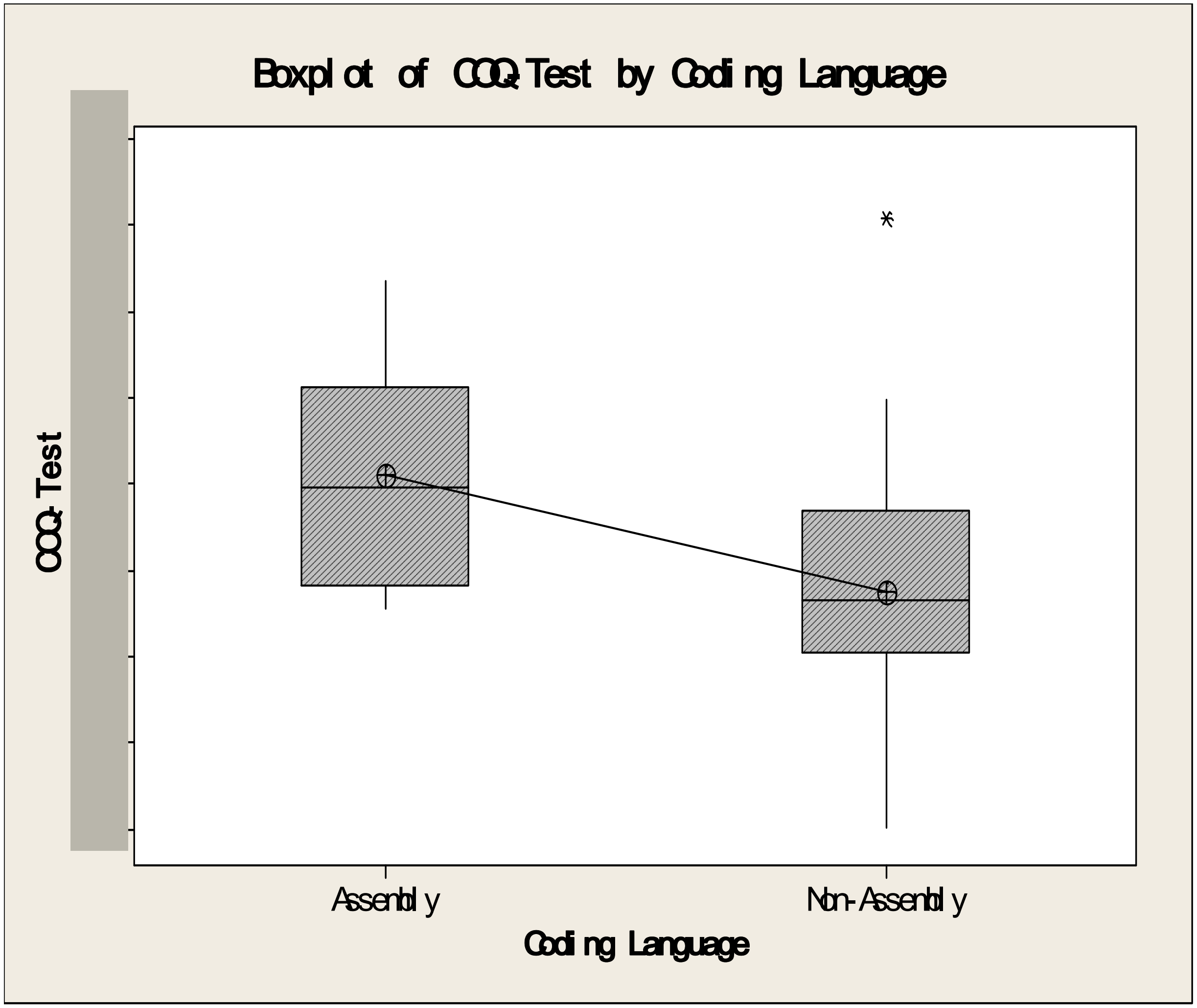
Differences by Coding Language

Major effects on test effort & COQ-Test

- Projects coded in assembly significantly more costly to test

Ns:

- 10 Assembly
- 26 Non assembly



© 2007 Carnegie Mellon University

Explaining the Variation

Factors that vary jointly with COQ-test – Adjusted $R^2 = .67; p < .0001$

- Motorola end gates (Code & unit test; feature & component test; or system test)
- Delta Code Size in KLOC
- Domain (Multimedia, embedded, tools & telecom)
- Coding language (assembly versus non-assembly)
- Base Code size in thousand assembly-equivalent lines of code (KAELOC)

Other factors examined – weak relationships, not significant

- Project Lifecycle (full or partial)
- In Process Faults (IPF) ... Post Release Defects (PRD)
- COQ for review/inspection ... Total Document Size (pages)

© 2007 Carnegie Mellon University

Some Actions Taken

1. Encourage test automation
   - Especially for product-oriented projects
2. Develop organizational integration & system test guidelines
   - To reduce test development effort
3. Encourage project test case reuse & automation
   - Especially for long term projects
4. Enhance analysis of escaped defects
   - Develop causal analysis guidelines
   - Introduce causal analysis methods such as ODC
5. Optimize regression test strategy
   - Introduce fault prediction tool
6. Better sharing of practices & lessons learned among projects.

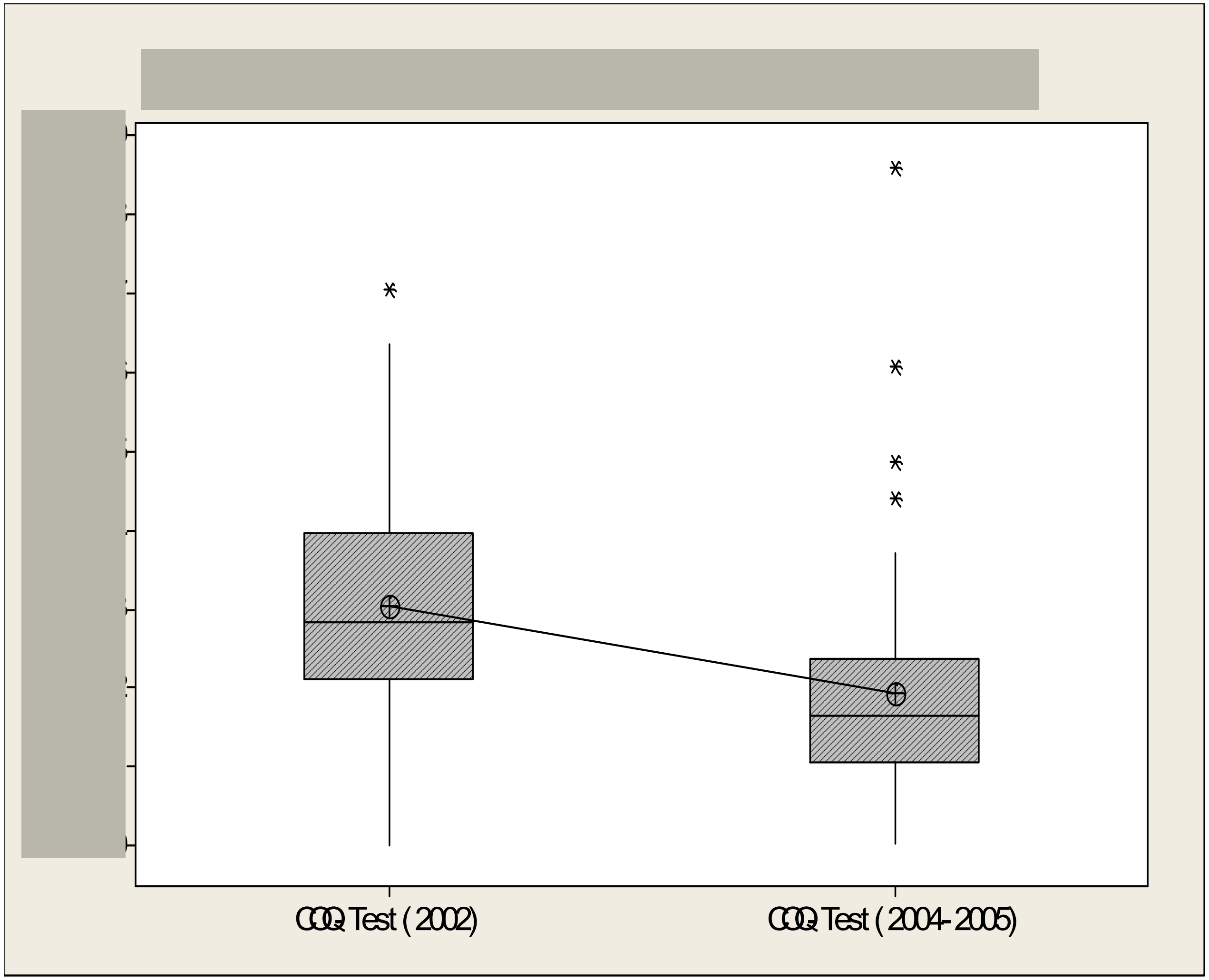
© 2007 Carnegie Mellon University

Effects of Process Change

Improvement actions reduced COQ-test cost

Ns:

- 56 in 2002
- 75 in 2004-2005
- P = .002



© 2007 Carnegie Mellon University

Other Standard MSG Performance Measures

Phase Containment Effectiveness

- The proportion of faults that are found at the first check point after they were introduced
- Updated incrementally by phase to monitor & control pertinent processes

In-Process Faults

- Number of faults found before completion of the project’s final phase prior to release Customer Satisfaction

First Estimation Accuracy

Cycle Time Reduction Rate

Post-Release Defect Rate

Customer Satisfaction

© 2007 Carnegie Mellon University

Today's Talk

Setting the stage

Our approach & Some Example results

How can you help?

© 2007 Carnegie Mellon University

Participate in Our Work & Contribute Your Own

We are always on the lookout for serious collaborators for:

- Enterprise Wide Performance Benchmarking studies
- The other work described today
- Any other ideas you may wish to pursue

Share your experiences with us & the wider systems & software community

- Submit a brief article for *DACS Software Tech News*
  - Next issue ~September 2007
  - Quantitative case studies welcome ... comparisons of variations among projects/programs preferred
- Also plan to edit an issue (perhaps issues) of *Software Process Improvement and Practice*

Please see me here in Golden about these & other opportunities

© 2007 Carnegie Mellon University

The Value Proposition

What's in it for our collaborators?

- Additional insight into their performance & its drivers
- Continuous improvement of their measurement processes
- Improvement of their internal benchmarking capabilities

What's in it for us?

- Better understanding & evidence regarding the effects of process improvement on project performance
- An enhanced ability to provide useful results for the larger software & systems engineering communities
- Learning through collaborative work with capable organizations & individuals who are not us

© 2007 Carnegie Mellon University

Thank You for Your Attention!

Dennis R. Goldenson
dg@sei.cmu.edu

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890
USA

© 2007 Carnegie Mellon University