Minimum Metrics for Meaningful Management

Robin F. Goldsmith, JD

GO PRO MANAGEMENT, INC.

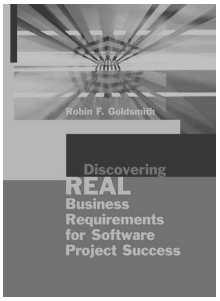
SYSTEM ACQUISITION & DEVELOPMENT

QUALITY/TESTING

PRODUCTIVITY

BUSINESS ENGINEERING

TRAINING



22 CYNTHIA ROAD
NEEDHAM, MA 02494-1412
INFO@GOPROMANAGEMENT.COM
WWW.GOPROMANAGEMENT.COM
(781) 444-5753 *VOICE/FAX*

Objectives

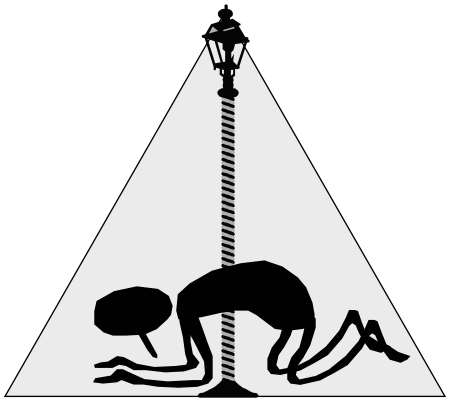
- Describe an appropriate structure for a measurement program that managers can use
- Suggest a set of core measurements
  - General and specific to your organization
  - Ways to collect
  - Ways to analyze, present, and use
- Identify issues and strategies for successful program implementation

*If You Owned and Managed Your Software Organization, What **Should** You Know to Run It Right?*

What Would You Measure?	*What Resistance Would You Get?*

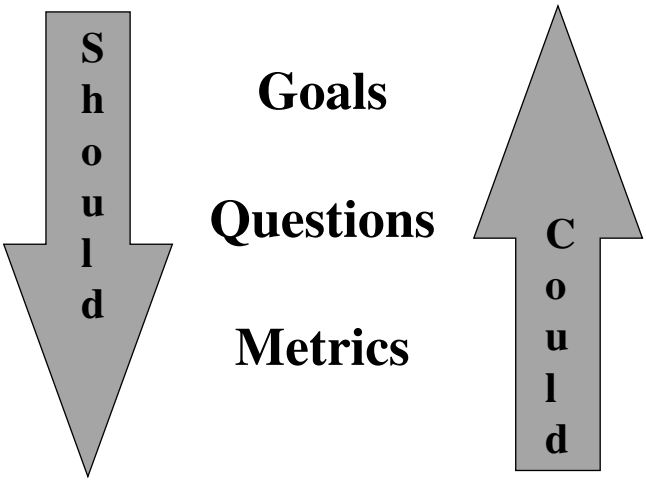
*What **Could** You Expect Your People to Measure Willingly?*

-
-
-
-
-
-

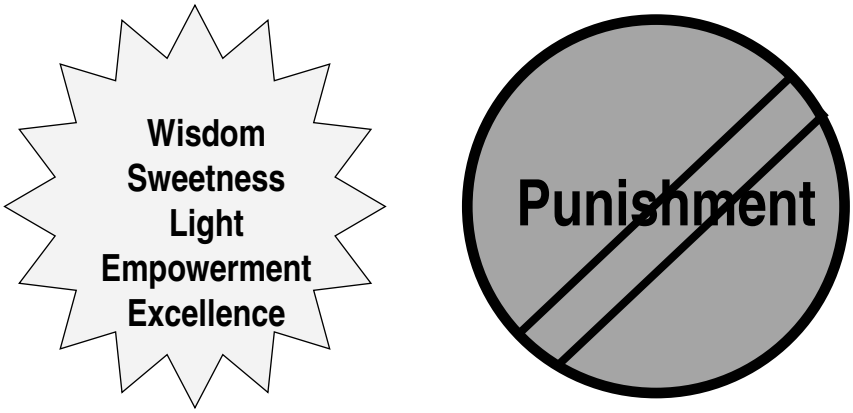


I'm hunting for my keys.

Top-Down vs. Bottom-Up



Limiting Parameter: How Measures Will Be Used



Two Types of Measures: (1) Results and (2) What's Causing the Results

1. Results measures include
   - How to tell whether the results are good/desirable
   - Need a **baseline** in order to show effects of changes
   - Indicators of status, positive and negative (warnings)
2. Causal measures must validly and reliably relate to the results
   - Based on the REAL process creating the results—procedural action steps *and* beliefs/customs
   - Help suggest possible ways to improve
   - Can be intermediate status indicators

Project Variables to Measure

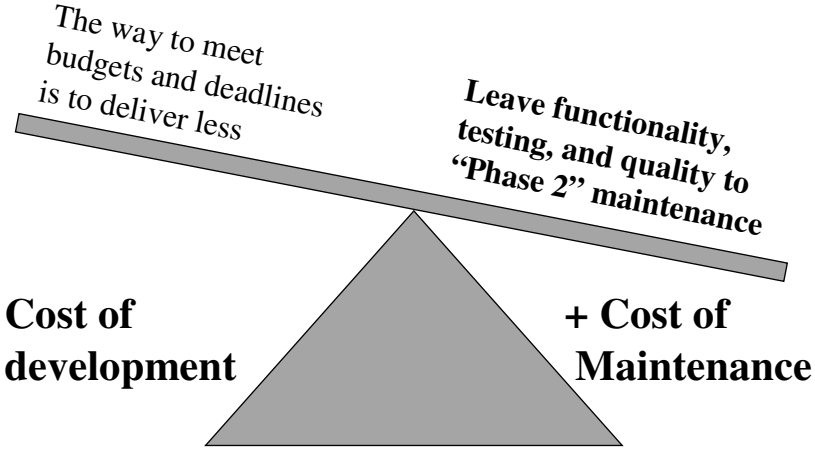
Product (Functionality and Quality)

Cost (Resources)

Time (Schedule)

Pick Two: Clever but bad advice
In fact, higher quality often is quicker & cheaper

Full Life-Cycle Perspective Is Essential for Meaningful Context



Quality Assurance Economics

- Maintenance is 66-90% of system cost
- Maintenance is mainly completion/ correction of development
- 2/3 of finished system errors are requirements and design errors
- Fixing a requirements error will cost
  - 10X+ during programming
  - 75X-1000X+ after installation (maintenance)

Does your organization routinely and reliably know these measures?

Secret: Don't Delay Waiting to Create Full Life-Cycle Project Baseline

- ➲ Pick three typical projects that have been in production for 6-18 months
- ➲ Find time sheets from the start through the present for people on the projects
- ➲ Relate to specific phases/activities by coordinating with contemporaneous descriptive materials, e.g., schedules, status reports, and problem reports

✪ *Core Measure:* **Size**

Project	***Product***
• Budget, person-days	• Small, medium, large modules
• Users, work load	• High, medium, low complexity
• Transactions,screens	• Memory size
• Database volume	• Processing capacity
• Network traffic	• Test cases
• Modules, objects	• Lines of code
• Requirements, design, instructions items/pages	• Function points
• Sum of products	

Lines of Code (LOC)

- Objectively measurable
  - Can automate, e.g., with check-in
  - Can baseline measure existing modules
- The activity which translates into time
- Relate to language and tools used, new/modified/reused/package, complexity, person's experience with language, application, environment

Other Product Size Measure Considerations

- Function Points is widely accepted
  - Language independent, correlates
  - Can be calculated from design
- Consistency is most important
- Regardless, size measure needs to calibrate for own organization with
  - Engineering elements that take the effort
  - Characteristics affecting production

✪ *Core Measure:* **Cost** *(Effort) Actual Resource Use by Task*

- By project
  - By phase (or RAD iteration)
    - » By task, module, component, build
      - By activity
        - Overhead (illness, vacation, misc. non-project)
        - Project administration and management
        - Project-related production (analysis, design, test planning, coding, unit test, integration test, system test, special test, acceptance test, train, document, install, operate, debug, enhance, etc.)

What are obstacles to collection and accuracy?

✪ *Core Measure:* **Schedule**

Task	Resp	Budg	Wk0 1 2 3 4 5 6 7 8 9
Unit 1	Joe	40	
		36	24 12
Unit 2	Sue	48	
		64	24 16 24
Unit 3	Joe	56	
		60	24 24 12
Unit 4	Sue	40	
		40	20 20
Integ	TM	16	

Projects fall behind one day at a time

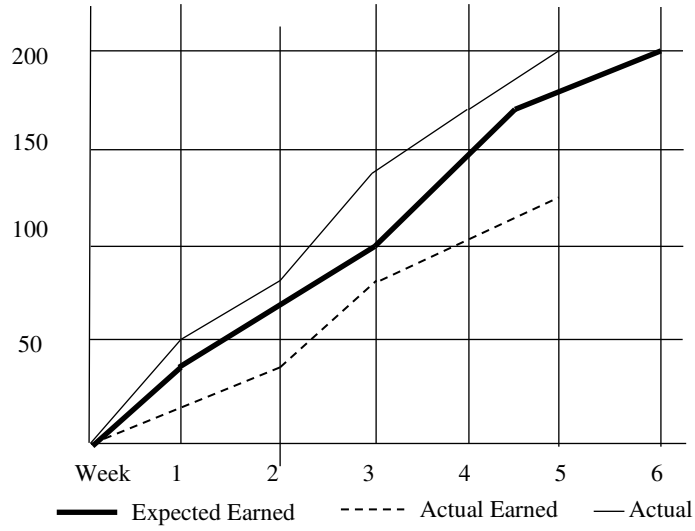
What Percentage Completed?

Unit 3? **90%** $\frac{60}{56}$ **= 107%** **?**

Project? **90%**

			Accountant's Actual/Budget		**Earned Value** Earned/Budget	
Unit 1	Joe		40		40	40
		36				
Unit 2	Sue		48		48	48
		64				
Unit 3	Joe		56		0	56
		60				
Unit 4	Sue		40		40	40
		40				
Integ	TM		16		0	16
			200/200=100%		128/200	=64%

Expected Rate of Earning Value



Good project status measure for dash-board

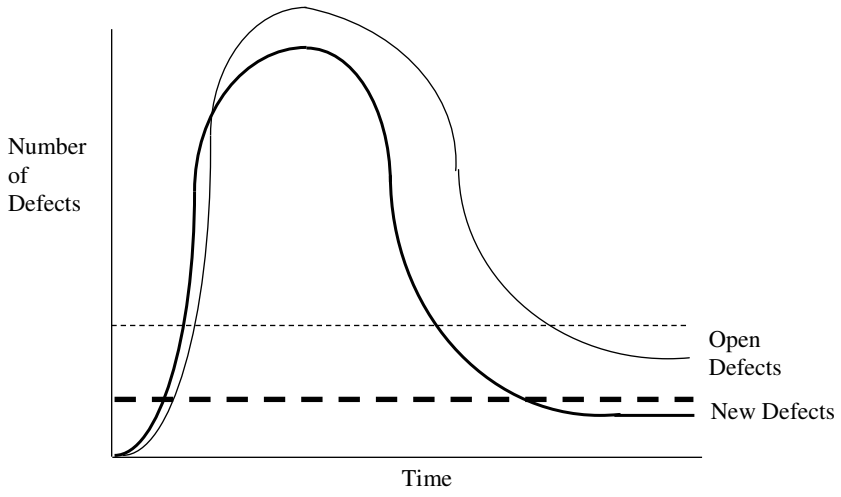
How would you measure actual value?

✪ *Core Measure:* **Defects** *Categorization and Counts*

- Severity
- When detected
- When injected (& age)
  - Requirements
  - Design
  - Coding
  - Fixing new code
  - Maintaining/ enhancing old code
- Module, Build
- Module characteristics
- Type of error, e.g.,
  - I-O
  - Logic
  - Table usage
  - Interface
  - Communications
  - Performance level

Defect profiles are base for estimating future defects

Projecting Ship Date from Defects



Measuring Test Effectiveness

			Defects Found	Defects Missed	Detection Percent	WIIFM Proactive
$30	Unit Test	$1	45 30	15 30	75% 50%	$45
$30	Integration Test	$2	12 15	3 15	80% 50%	$24
$20	System Test	$4	1 5	2 10	33% 33%	$4
$40	Acceptance Test	$8	1 5	1 5	50% 50%	$8
$50	Production	$10	1 5	0 0		$10
$170 hours			60			$91 hours

Metrics to Put Defects in Context

- Cost/time to find indicated number of defects
  - Method and tools employed, e.g., test vs. formal technical review
  - Degree of coverage
    - » Functional
    - » Structural (executable statements, branches)
- Cost/time to fix (rework)
  - Relative to time of detection
  - Relative to age of defect

Are defects the only quality indicator?

✪ *Core Measure:* **Performance**

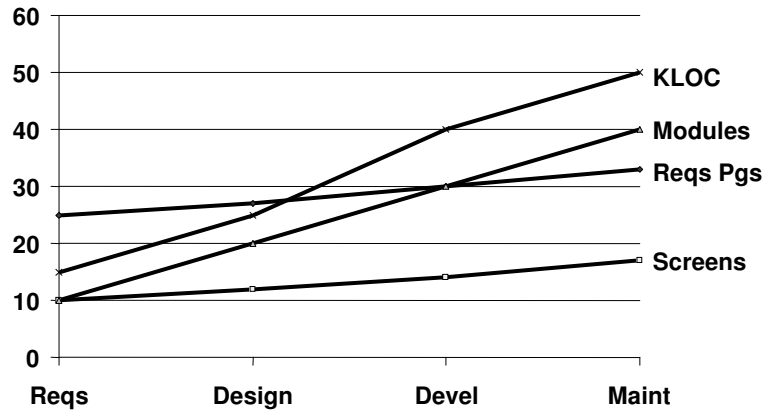
Product

- Uptime, downtime
- Throughput
- Response time
- Operator error rate
- Problem backlog
- Mean time between failures
- Work arounds
- Actual use

Process

- Support response
  - Answer phone
  - Immediate solution
  - Escalation steps
  - Implement solution
  - Solution adequacy
  - Customer experience
- Anticipation of need
- Provides benefits

Project Stability and Growth



Key Management Metrics

$$\text{Productivity} = \frac{\text{Size}}{\text{Effort}}$$

What gets measured gets done
What gets rewarded gets repeated

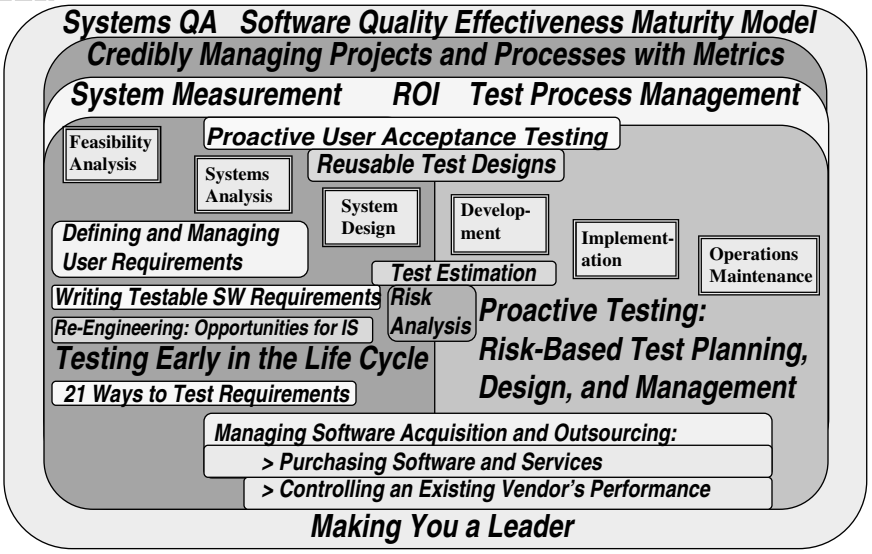
$$\text{Effectiveness} = \frac{\text{Value}}{\text{Effort}}$$

$$\text{Value} = \frac{\text{How Many, How Well Business Needs Met}}{\text{Cost}}$$

Summary

- Measure projects over the full life cycle, start with a baseline of completed projects
- Core measures include resource costs per activity, progress relative to schedules, and product size and quality/performance
- Integrate measurement and reward structures nonpunitively, focusing on delivering business value

Go Pro Management, Inc. Seminars--Relation to Life Cycle



Robin F. Goldsmith, JD

robin@gopromanagement.com (781) 444-5753
www.gopromanagement.com

- President of Go Pro Management, Inc. consultancy since 1982, working directly with and training professionals in business engineering, requirements analysis, software acquisition, project management, quality and testing.
- Previously a developer, systems programmer/DBA/QA, and project leader with the City of Cleveland, leading financial institutions, and a "Big 4" consulting firm.
- Degrees: Kenyon College, A.B.; Pennsylvania State University, M.S. in Psychology; Suffolk University, J.D.; Boston University, LL.M. in Tax Law.
- Published author and frequent speaker at leading professional conferences.
- Formerly International Vice President of the Association for Systems Management and Executive Editor of the *Journal of Systems Management*.
- Founding Chairman of the New England Center for Organizational Effectiveness.
- Member of the Boston SPIN and SEPG'95 Planning and Program Committees.
- Chair of BOSCON 2000 and 2001, ASQ Boston Section's Annual Quality Conferences.
- Member ASQ Software Division Methods Committee.
- Member IEEE Std. 829 for Software Test Documentation Standard Revision Committee
- Admitted to the Massachusetts Bar and licensed to practice law in Massachusetts.
- Author of book: ***Discovering REAL Business Requirements for Software Project Success***