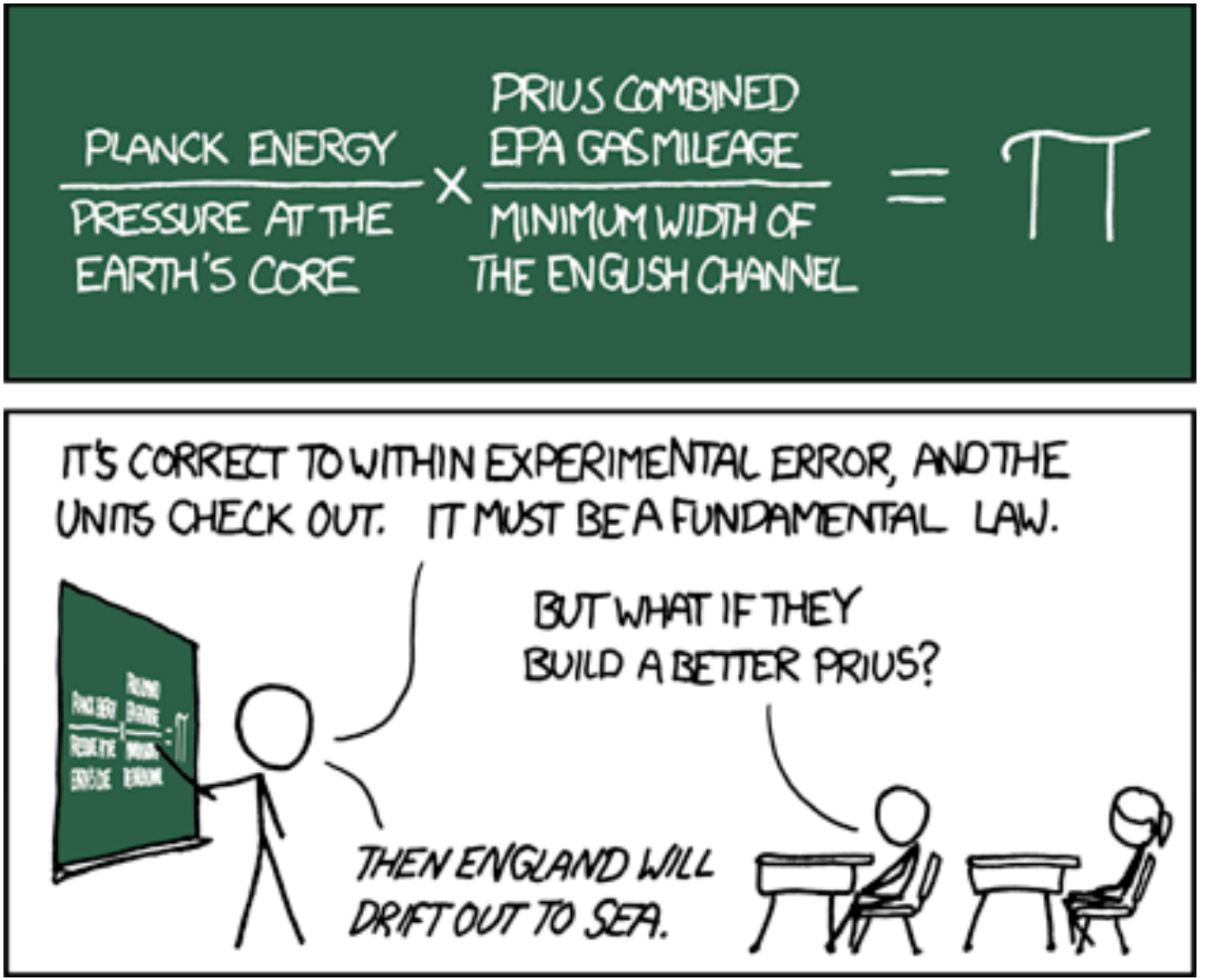
CSSE 372 Software Project Management:
Managing Software Projects with Measures

Shawn Bohner
Office: Moench Room F212
Phone: (812) 877-8685
Email: bohner@rose-hulman.edu

ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Dimensional Analysis Abuse



Learning Outcomes: Plan (verb)

Create a plan for an intermediate size software project and manage to the plan. Maintain a software project schedule.

- Introduce Measures and Metrics
- Describe Software Project Measures
- Apply Software Metrics

Why Measure Software Projects?

- You can't manage what you can't measure
- Chinese Proverb: If you don't know where you are going, any road will do
  - Watts Humphrey's Corollary: If you don't know where you are, a map won't help

We use measurement a lot in Engineering.

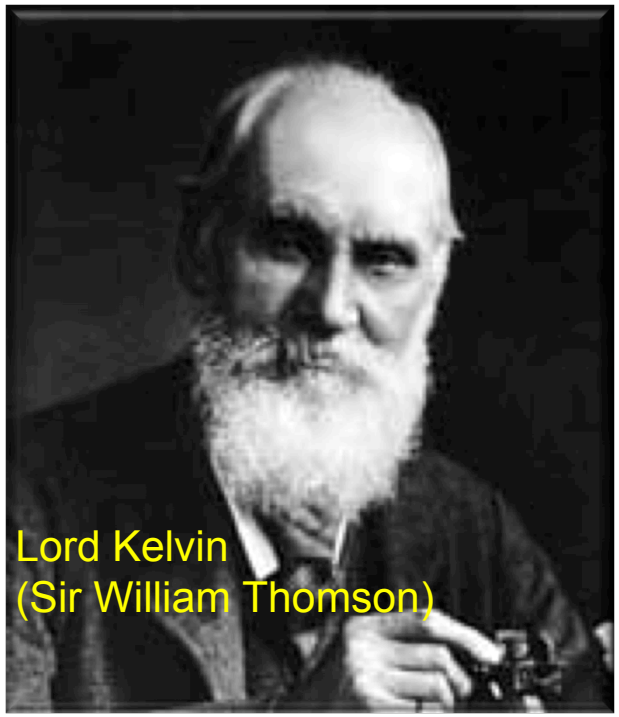
What is the difference between a quantitative & qualitative measure?

- Think for 15 seconds…
- Turn to a neighbor and discuss it for a minute

Measurement Basics

Measurement is the process of objectively assigning values to entities to characterize specific attributes

- An **Entity** is an object (product) or event (process)
- An **Attribute** is a feature or property of an entity
- An **Objective** is a well-defined rule for assigning the numbers


Lord Kelvin
(Sir William Thomson)

Entity = human
Attribute = height
Measurement = 6 feet
Objective =>
bigger number is "taller height"

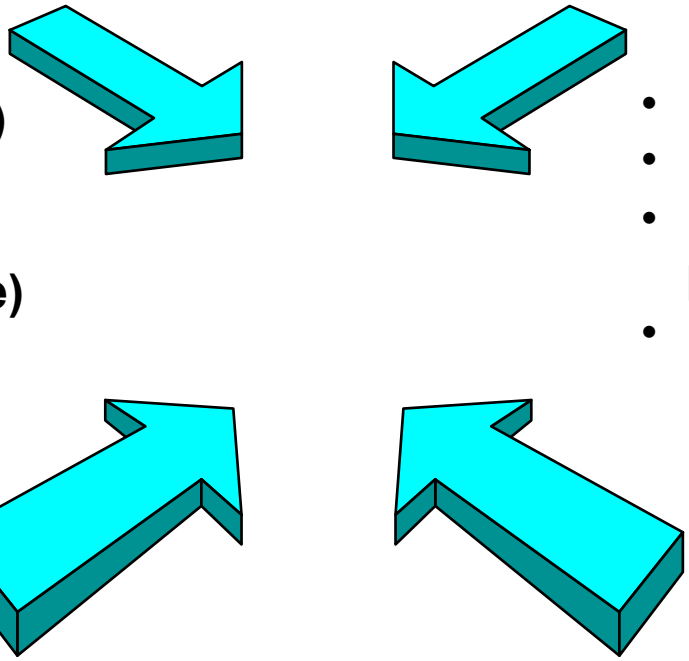
Measures, Metrics and Indicators

- A *measure* provides a quantitative indication of the extent, amount, dimension, capacity, or size of some attribute of a product or process

- The IEEE glossary defines a *metric* as "a quantitative measure of the degree to which a system, component, or process possesses a given attribute."

- *Indicator* is a metric or combination of metrics that provide insight into the software process, a software project, or the product itself

Software Management Measures



Project Management

- Estimates vs. Actuals
  - –Size (change and build)
  - –Cost/Budget
  - –Effort/Schedule
- Risks (impact & exposure)
  - –Resource Availability
  - –Technology
  - –Delivery

Business Management

- Business Value of IT
- Business Risks
- Financial breakeven point
- ROI/ROA/ROE

Product Management

- Size in KLOCs or FPs
  - –Maintenance and Development
- Quality/Reliability
  - –Pre and Post-Delivery Defects
  - –Severity
- Change actions
  - –Type (fix, enhancement, update)
  - –Impact of change (small -> large)
  - –Priority (emergency -> routine)

Process Management

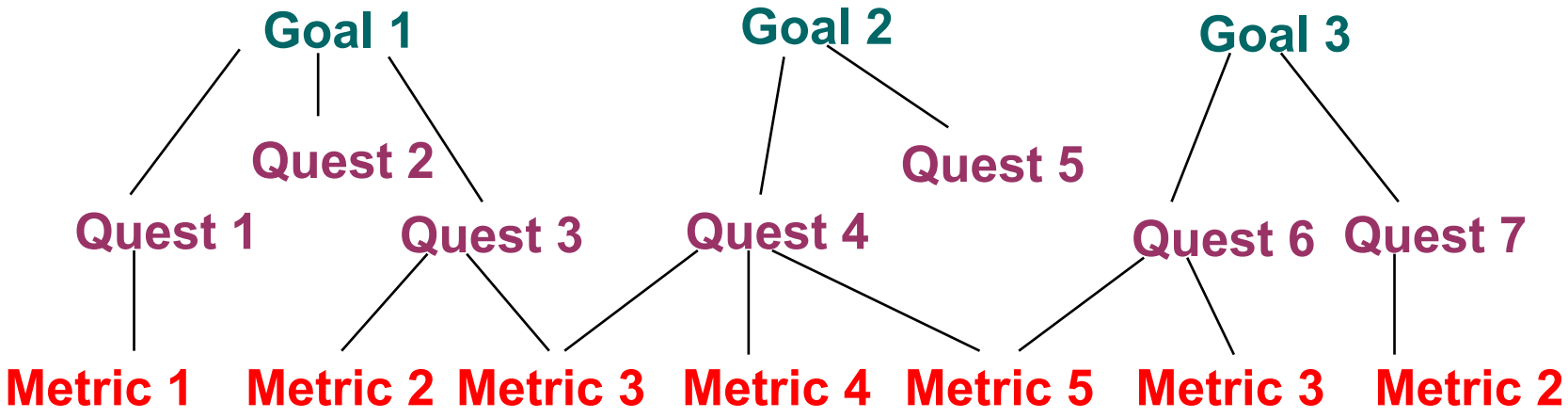
- Throughput - rate of changes delivered
- Cost of operations
- Inventory
- # of concurrent changes
- Testing Efficiency

Measurement is about Visibility

Project Management measures must be balanced for best visibility of the right decisions

Goal/Question/Metric (GQM) Paradigm

1. Generate set of goals for a given software project
2. Derive a set of questions which quantify those goals
3. Develop a set of metrics which provide information to answer those questions



Exercise: Using GQM, define a metric for determining the rate of production in a software project?

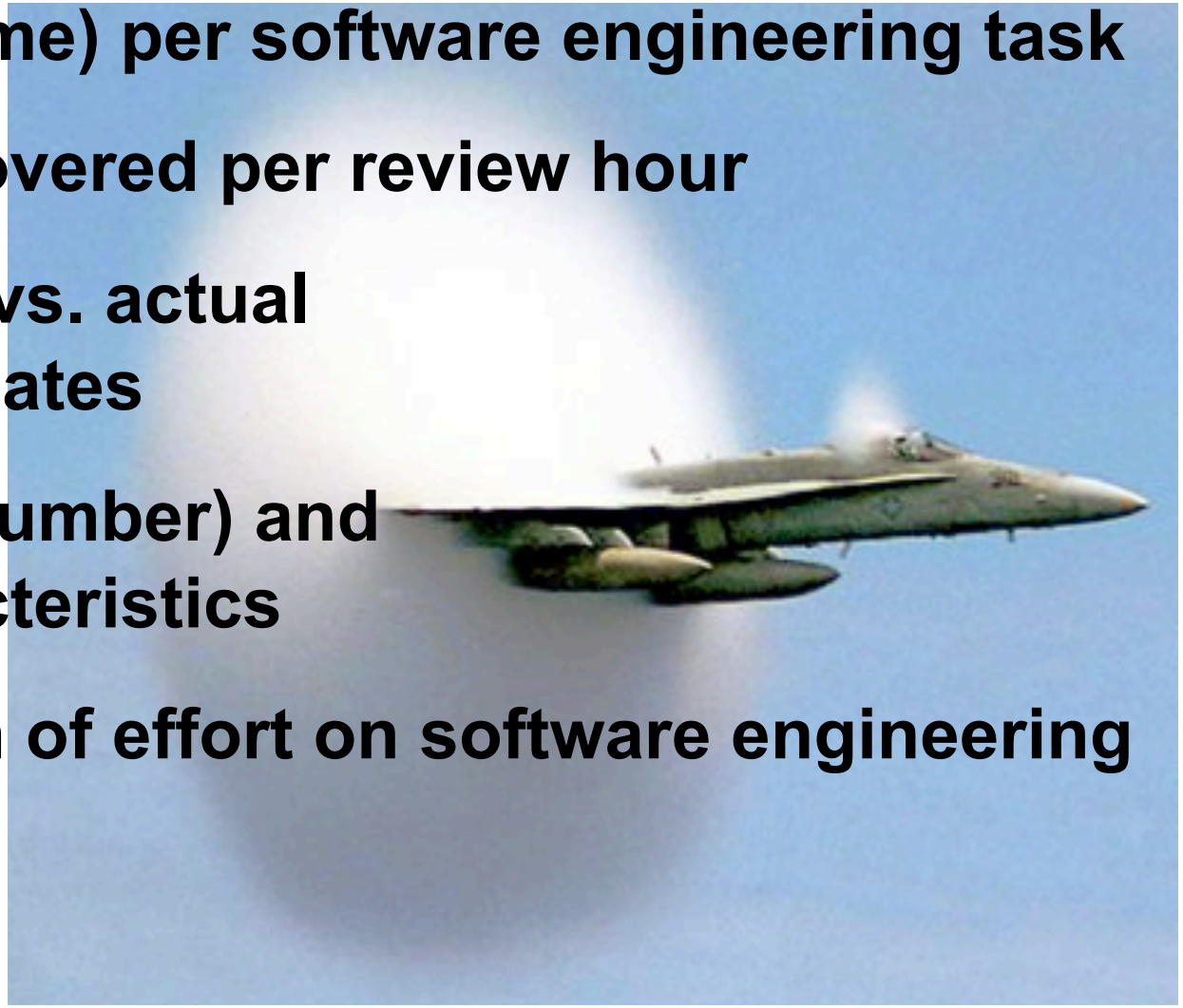
Hint: Productivity is usually output/input

Consider quality too…

- **What is the goal?**
- **What are the question(s)?**
- **What are the metric(s)**

Typical Project Metrics

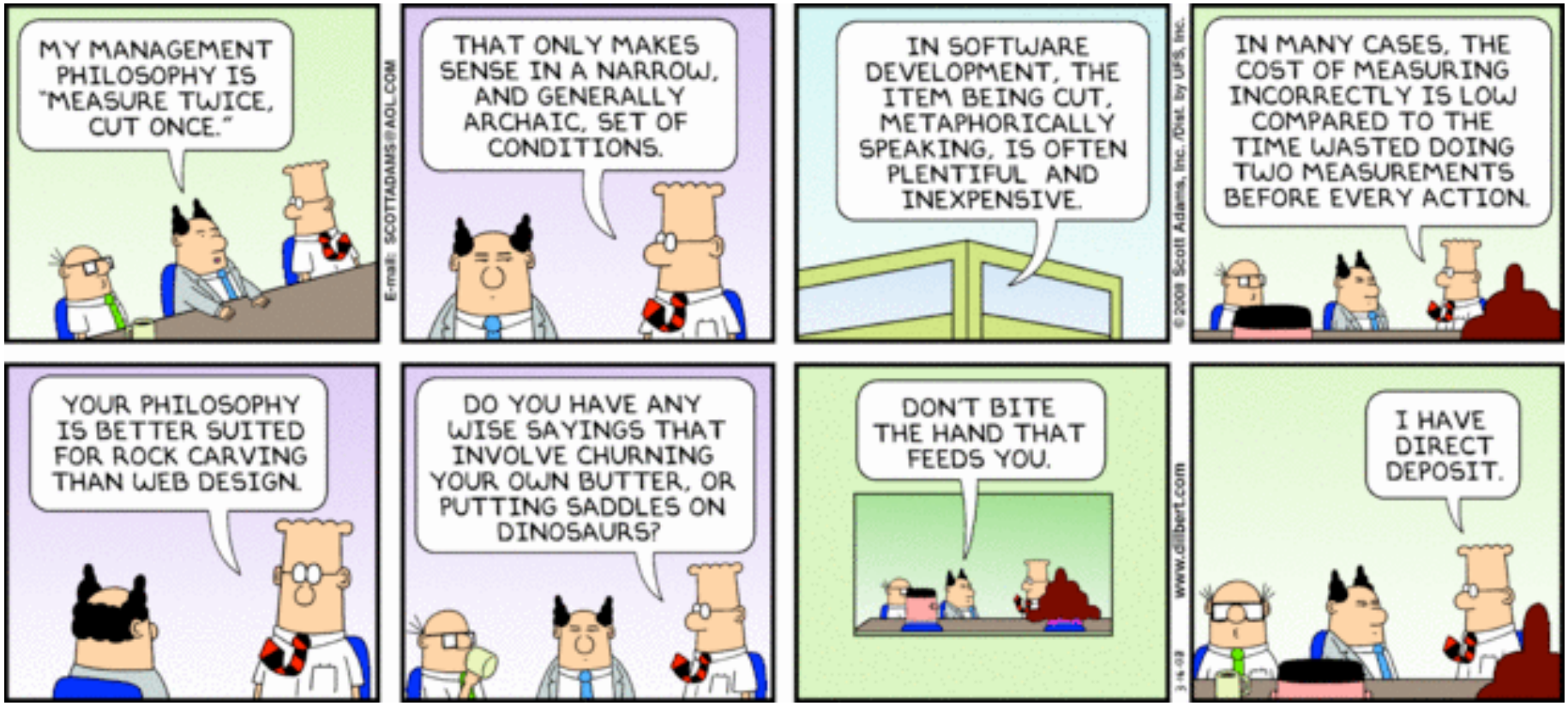
- Effort (or time) per software engineering task
- Errors uncovered per review hour
- Scheduled vs. actual milestone dates
- Changes (number) and their characteristics
- Distribution of effort on software engineering tasks



NASA SW Development Metric Set

Metric	Description
Software Size	# of source lines of code at delivery
Staffing	Engineering hours per project, subsystem, and DR/SR closed
Requirements Stability	Total # of requirements to be implemented by project
Development Progress	Planned vs actual Number of units designed, coded, and tested
CRU Performance	Percent of CPU, disk, memory, and I/O channel utilization
Test Case Completion	Planned and actual # of test cases successfully completed
Test Focus	Fraction of DRs Closed with a software fix by system
Facility Utilization	# of facility hours requested, scheduled, and used in test
Fault Density	# of DRs per 1000 lines of code over time per system
DR Open Duration	Time from DR written to closure
Software Reliability	Probability that the software "works" for a specified time in a specified environment
Software Complexity	# of control paths in the software per module

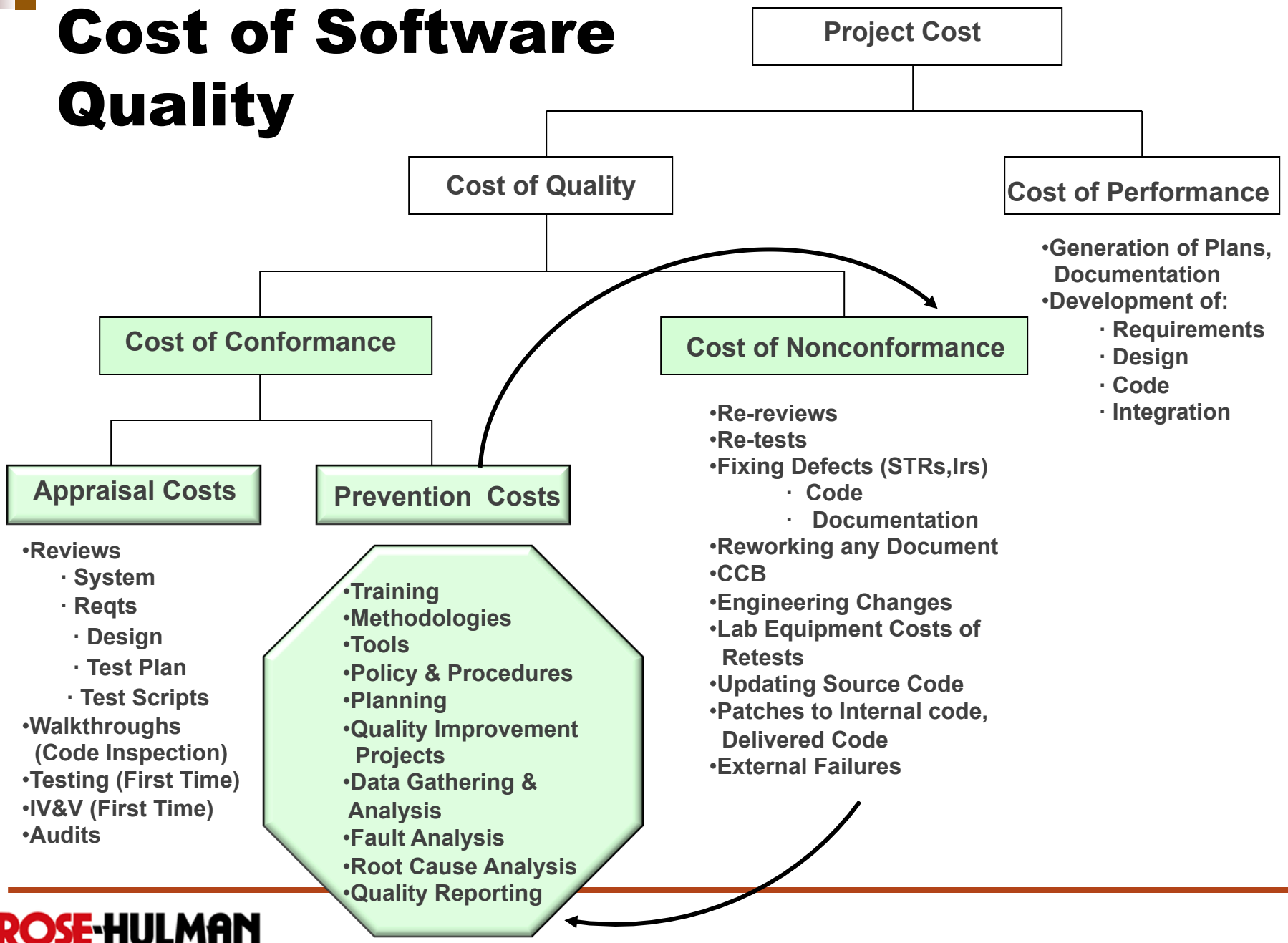
Dilbert's take on Software Measures...



What is Software Complexity?

- Complexity - degree to which a system or component is difficult to analyze, understand, or explain. Key factors include:
  - the # and intricacy of interfaces and branches
  - the degree of nesting
  - the types of data structures
- Provides a means of relating the complexity of modules produced by the same process, in the same language, or within some other aspect of interest

Cost of Software Quality



Key Principles for Analyzing Metrics

- Clearly defined metrics, consistently applied
- Metrics are only indicators, use them accordingly

- Focus on leading indicators over lagging ones
- Recognize indicators of problems
  - Lack of change
  - Frequent change
  - Slow, steady deviation from plans

Software metrics are navigational instruments giving position, direction, and rate of change

How are Measures/Metrics Used?

Solving problems —

Which choice or improvement should be made?

- Benchmarking for performance improvement

Getting attention —

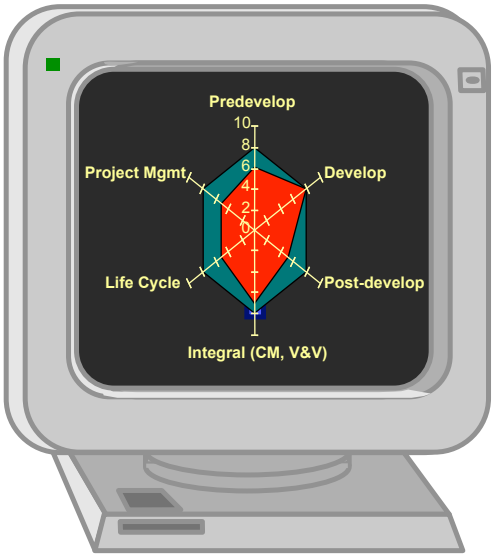
What situations need to be addressed?

- Dashboard of indicators

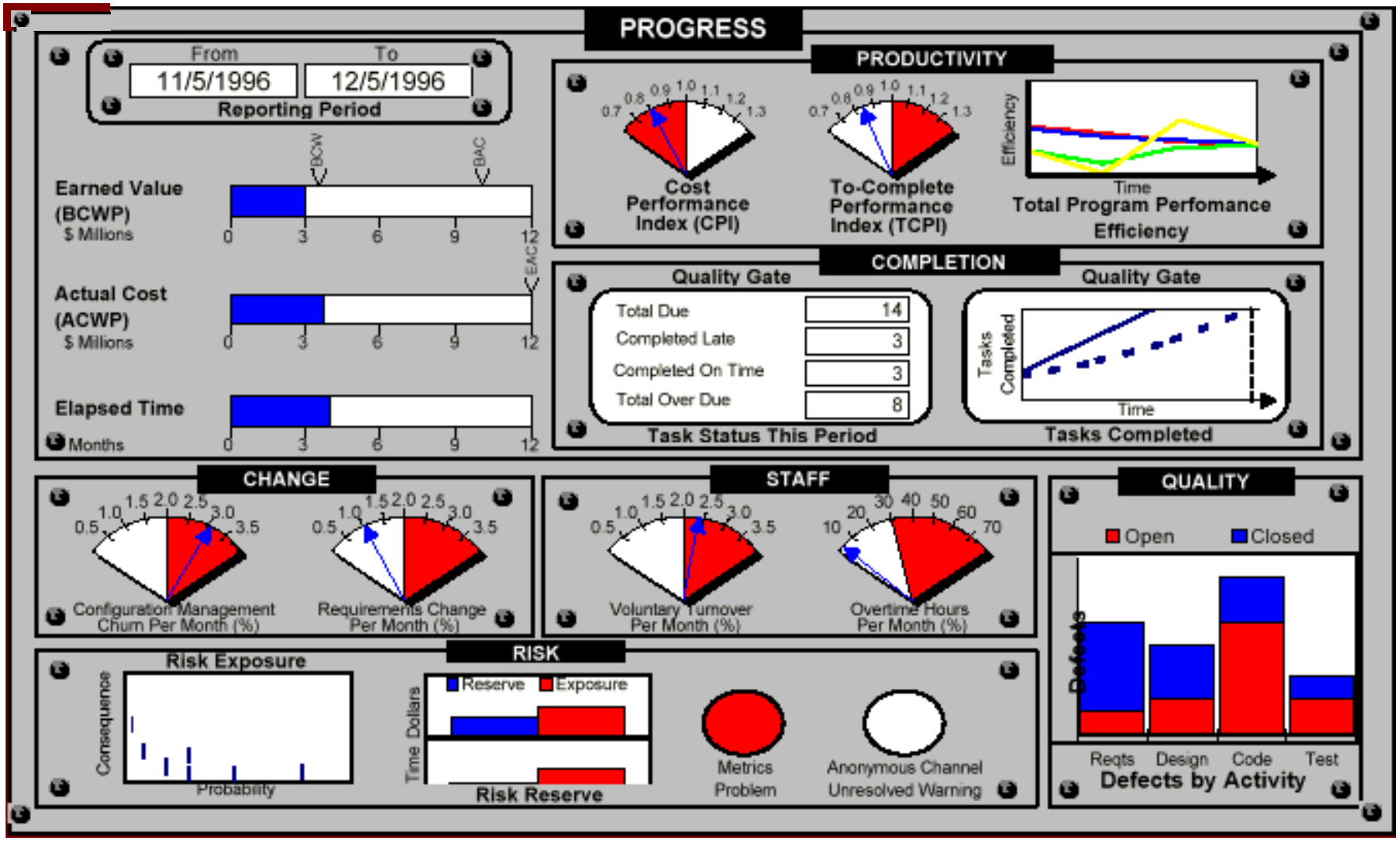
Keeping score —

How well is it doing?

- Scorecard on goals



Management Dashboard



Source: Software Project Manager's Network

COMMON SENSE

Just because you can, doesn't mean you should.

Measurement Common Sense

- Collect what contributes to key decisions
  - And cheap to acquire
  - And does not create undue stress (this means you may have to sell it to those measured)
  - And does not measure people directly
- Apples to Apples Comparisons
  - Granny Smith and Macintosh apples
- When you stop making decisions on the measures, stop collecting them…
  - "We may need this someday" is not a good reason to keep measuring …the payoff must be high

Homework and Reading Reminders

- Read Chapter 9 of text for tomorrow
- Final Project – SW Proj. Mgt. Plan (SPMP)
  - Completed by team…
  - Due by 11:55pm, Friday, November 2nd, 2012.
  - No late days –review swap with another team
- Complete Homework 6 – Earned Value Analysis
  - Due by 11:55pm, Tuesday, October 16th, 2012