1 Metrics for Metrics

2 What We're Talking About

Measures: values representing characteristics of interest
Metrics: computations using values to express characteristics more clearly

$ are costs, income, profits

3 Software Models

Why models? Because we need to understand what the characteristics represent [mean]. The model provides us with a framework to hang things on and discuss abstract concepts. [Especially important with software- not much tangible going on]

(With metrics^2, we are creating a model for costs/ROI on metrics on software…) Models presented later in presentation

4 Setting Up Metrics^2

Models presented later in presentation

(Chances are, you've got the measures you need from the regular metrics programs)

5 Data Mining

Ways to make cost-effective use of metrics

The trick here is knowing what you've got and selecting measures/metrics that represent reasonable costs and returns

6 IT Metrics Tool Kit

Know what you've got in the way of measures and metrics

Use measures and metrics you've got, integrate and display values

Be sure your models are acceptable to people who use them (management)

Remember to broadcast results; silence can be deadly (people assume the worst)

7 Psychology of Metrics

8 What Goes Wrong?

Seat-of-the-pants: "unsupportable or esoteric arithmetic manipulations made at a moment's notice with less than a moment's thought"

Consistent: e.g., LOC not counted the same way for all COBOL programs, or not counted the same way each time

Accurate: e.g., "time to fix" values programmers put into defect tracking system

Cost focus does not allow for shifting (or causes shifting without regard for effects)

9 Potential Problems

Cost shifting from late Testing to early Inspection

Early investments may require training, added resources; net savings may be time-shifted

10 Analyzing Metrics Results

Look at how well the model fits; compare parts of the organization, one part over time (do the metrics fit the facts/events)

Are the measures too expensive to make? (not just $)

11 Refining Metrics

Look for cost savings: is less accurate good enough? Does the same data exist already? Is there another way to get the data?

Do the measures really mean what we think?

Have we hit diminishing returns?

12 IT Economics

Watch out! Traditional IT economics pulls us away from best path

Deming example of saving on air-fare: Person takes red-eye flights but can't function for meeting there or when they return

"Save by cutting metrics programs" this means improvement is by guesswork

13 Economics of Metrics

14 Relating $ to Metrics

15 Metrics and the IT Budget

16 ROI For Metrics

17 Value of Information

Metrics related values are really hard to quantify or justify. Usually best to ignore them.

18 Cost of Acquisition

One time costs amortized over period of use

Interpretation cost in comparison to current cost of management decisions on best methods (usually this is a wash or improvement over current methods of gathering and interpreting process effectiveness)

Fuzz costs because 1) we aren't clear about what to do, so it takes longer
2) we don't do the right thing (at least at first)
3) we are slow to recognize how to improve

Fuzzier data is easier to acquire (rough guesses are good enough), removing uncertainty has costs associated [improvement in defect densities by focusing on inspecting high complexity modules] Lots of values, may be other causal factors, costs & returns hard to compute

19 Short Term Justification

Quick computations to see what user perceives as valuable, how valuable

1st order estimate: LOC rather than function points
average reduction in defect density over all modules rather than sorting out the modules based upon multiple classifications and multiple changes

20 Keys for Success
21 Conclusions