The Metrics Minefield

Michael Bolton
DevelopSense
http://www.developsense.com

KWSQA, September 2006

It All Started With A Question

I was a only a kid, but I had a question.

Four out of five dentists surveyed recommend sugarless gum for their patients who chew gum.

What did the other one recommend?

More Than One Question, Really

Actually, I had a *lot* of questions…

- *Only* four out of *only* five?
- Or did they mean 8000 out of 10000?
- Who *didn't* participate in the survey?
- How was the survey taken?
- What does "recommend" mean?
- What did they recommend to their patients who *didn't* chew gum?
- What choices were offered besides sugarless gum?
- *Who was asking?*

The Metrics Minefield

We have determined that there are large numbers of mines buried in the metrics minefield.

The Metrics Minefield

This is a report on what some of our best minesweepers have discovered so far, with a few suggestions on how we might avoid or clear some of the mines.

The Metrics Minefield

The intention is not to suggest that metrics are useless, but to identify some examples of gross dysfunction, and to try to avoid it.

Mine #1

Our field sometimes seems obsessed with metrics, but doesn't seem to pay much attention to *measurement validity*.

What Do Metrics Measure?

- "Software Engineering Metrics: What Do They Measure and How Do We Know?"
  - co-authored by Cem Kaner and Walter P. Bond
  - provides several definitions of measurement, most of which seem to amount to "using numbers to describe something" or "putting a number on some attribute"
  - Kaner and Bond's synthesized definition:

"Measurement is the empirical, objective assignment of numbers, according to a rule derived from a model or theory, to attributes of objects or events with the intent of describing them."

How Do We Measure?

- Measurement always has some model lurking in the background
- Metric models are based on some comparison, which may be explicit or implicit
- We can
  - count things
  - compare individual things with each other
  - compare individual things with a reference
  - compare individual things with elements in a group
  - compare groups
  - count things over time (rates)
  - create derivative metrics by performing multiple measurements and comparisons over time

Why Do We Measure?

Implicit in our motive for measurement is some model of *assessment*, based on some *comparison*, made in accordance with some *observation*.

Why Do We Measure?

- facilitating private self-assessment and improvement
- evaluating project status (to facilitate management of the project or related projects)
- evaluating staff performance
- informing others (e.g. potential customers) about the characteristics (such as development status or behavior) of the product
- informing external authorities (e.g. regulators or litigators) about the characteristics of the product

-- Kaner and Bond

Why Do We Measure?

- To discover facts about the world
- To steer our actions
- To modify human behaviour

-- Tom DeMarco

DeMarco wonders if we, as an industry, are too focused on behaviour modification.

Why Do We Measure?

- To discover natural laws (third-order measurement)
- To refine and to tune (second-order measurement)
- To get the damned thing built (first-order measurement)

-- Jerry Weinberg

Weinberg suggests that, as an industry, we're obsessed with trying to make third- and second-order measurements, when first-order measurements are what we need.

Why Do We Measure?

Quality measurement depends upon
our skill at observation,
what we're comparing,
and the validity of the models
that we're using for assessment.

Why Do Buses Short-Turn?

- What supervisors seem to observe
  - schedule
  - location of the bus
- What they seem to compare
  - the scheduled position of the bus vs. the actual position of the bus
- The apparent model
  - Based on whether the buses are on schedule, vs.
    - Whether buses are full or empty
    - Whether passengers are being picked up and moved quickly or efficiently
    - Whether passengers are happy

Mine #2

Comparisons and assessments aren't necessarily numerical.

A Sign in Einstein's Office

Not everything counts that can be counted, and not everything that can be counted counts.

Lines of Code Metrics

- Might a given developer's style be highly explicit or terse?
- Might lines of code be calling into functions in the same program?
- Might lines of code be calling heavily into external libraries?
- Might a requirement be missing its implementation?
- Is more better? Is *less* better?

If you don't care about quality,
you can meet any other requirement.

--Jerry Weinberg

Mine #3

Numbers aren't as descriptive as words and stories.

Words can be vague or ambiguous, but numbers without clarifying words are just as bad or worse.

Mine #4

Many people in our field make sweeping, pseudo-statistical generalizations without proposing a measurement model.

The Pundits Speak

"Systems in general work poorly or not at all."

Peter Coffee, Keynote Speech, Agile 2006, quoting John Gall

- *Which* systems in general?
- What constitutes "working poorly"?
- What constitutes "working…not at all"?
- Don't all systems work for someone's purposes to some degree?
- If not, why don't the systems change?

A Data Point

"Complicated systems seldom exceed 5% efficiency."

Peter Coffee, Keynote Speech, Agile 2006, quoting John Gall

- *Which* systems?
- What constitutes a complicated system?
- What do you mean by "seldom"?
- How do you measure efficiency?

Mine #5

Most people in the testing and quality field haven't studied measurement theory or statistics, but some of us are asked to implement measurement programs anyway.

Mine #6

We often validate our assumptions with trivial surveys and analysis.

A Trivial Survey
Part 1: Missions

- How many are required to produce metrics in their job?
- How many will be required to produce or contribute to metrics in their job in the coming year?

A Trivial Survey
Part 2: Serious Study

- How many have studied statistics or measurement theory in high school, college, or university?
- How many have read a textbook on statistics?
- How many have read a textbook on economics?
- How many have read books about critical thinking?

A Trivial Survey
Part 3: Informal Study

- How many have taken a drive-by course in metrics?
- How many have read informal or self-teaching guides on metrics?
- How many have looked into metrics using online sources (e.g. Wikipedia)?
- How many have read “How to Lie with Statistics”?
- How many have read“Freakonomics”?

A Trivial Survey
Part 4: Basic Terms and Famous Stories

- How many know the difference between “dependent variables” and “independent variables”?
- What’s a “control” group vs. an “experimental” group?
- How many have heard of The Hawthorne Effect?

Mine #7

Good statistical work depends on isolating the dependent variables, and reducing or controlling the independent (free) variables.

Mine #8

In software development metrics, identifying the dependent variables is tough, and reducing or controlling all of the independent (free) variables is effectively impossible.

Exercise

Part 1: Identify the quantifiable variables involved in assessing the productivity of a tester. How might these be manipulated or gamed?
Part 2: Identify the *quality* attributes of a tester. How might *these* be gamed?

Assessing Testers – One Possible Approach

- Start with the quality attributes for a piece of software—capability, reliability, usability, scalability, security, performance, installability, compatibility, supportability, testability, maintainability, portability, localizability—and see where they take you.
- Some clearly apply to people; some do not
- Some sort-of apply—how might you remodel them to make them useful?
- Would we hang a *number* or a *description* on these attributes?

Mine #9

When something can't be proven, making predictions about it is somehow unsatisfying.

Predicting Software Defects

- "Testing can prove the presence of errors, but never their absence."
  - Edsger Dijkstra
- Suppose that we find the predicted number of defects in a risky product. *Are we done?*
- We can't know product quality for sure.
- Everything that we thought we knew about product quality on the last build can be undermined by the next.
- In the testing endgame, shouldn't we consider measuring the significance of change in terms of the whole project?

"It's just a one-line change."

Mine #10

The common method of calculating risk as "impact times probability" is just weird.

Impact times Probability

I \ P	1	2	3	4	5
1	1	2	3	4	5
2	2	4	6	8	10
3	3	6	9	12	15
4	4	8	12	16	20
5	5	10	15	20	25

Where are the big numbers clustered?

Why do we multiply? Why not add? Subtract? Divide? Or take an exponent?

Risk Times Impact

- The more serious problems include
  - High impact times low probability gives a low number
  - High probability times low impact gives a low number
  - The information as to which is which vanishes when we take the product of the two numbers
  - How does the impact *number* map to the impact *reality*?
  - Impact is a guess
  - Probability is a guess
  - Risk is therefore expressed in units of guesses[2]
    - Thanks to Cem Kaner for some of these insights

Mine #11

There are untold numbers of biases and errors involved with collecting and analyzing metrics.

Cause and Effect Errors

- Fundamental Attribution Error
  - "Things are this way"; based on incomplete observation and ignorance of context
- Confusing correlation and causation
  - Event A might cause Event B, but B might cause A
  - A might merely amplify B
  - A and B might be caused by C
- Single Cause Error
  - Things rarely are attributable to a single cause
- Confusing concurrence and correlation
  - Two things that happen at the same time might not be related

A Handful of Biases

- Evaluative bias of language
  - "Would you say our product is full-featured?"
  - "Would you say that their product is bloated?"
- Malicious compliance
  - might anyone be motivated to participate half-heartedly in a survey?
  - might anyone be motivated to distort the measurement actively?
- Collaborator bias
  - who is consenting to participate in the measurement?
  - who is getting left out, and why?

Mine #12

Software metrics seem especially subject to reification error.

Reification Error

- Reification error is the critical thinking error based on regarding, counting, or evaluating something abstract as a material or concrete thing

Test cases and requirements are ideas.
How do you count an idea?

When you divide a reification error by a reification error, the reification errors don't cancel.

Reification Error

- *Example:* $\frac{\textit{number of test cases}}{\textit{number of requirements}}$
- Are these two test cases of equivalent value?
  - "try 3 digits for the PIN, instead of the required 4"
  - "try withdrawing $10000000 to test input constraints"
- Are these two requirements of equivalent value?
  - "the ATM system must be able to handle transactions with all member banks in the Interac network"
  - "for transactions greater than $100, the ATM may dispense $50 and $20 bills in any combination, such that exactly the required amount is dispensed."

Counting Ideas

- Don't count test cases
  - test cases are ideas; counting test cases is classic reification
- Don't measure testers by bug reports
  - testers may be doing other things of great value besides writing bug reports

"If you evaluate testers by counting bug reports, I *guarantee* that your testers are misleading you."
Test cases are like briefcases; I'm a "famous testing expert", and I can't tell whether 1000 test cases are good or bad until I see them.
James Bach

Mine #13

Statistics is usually a sampling exercise.
It's easy to sample badly.

Samples

- Surveys are typically done on a *sample* of the population (otherwise, they're called censuses)
- The questions on the survey are *samples* of the set of questions that could be asked
- The answers to the questions are *samples* of the set of answers that people could provide
- The data is often collected in something called an interview, but it's usually just a list of questions with a set of constrained answers.
  - From "How to Lie With Statistics, by Darrell Huff (1954)

Mine #14

When we collect metrics on projects,
we tend to leave out certain projects.

Who's Left Out of the Sample?

- Most organizations have some projects that succeed, and some that fail.
- When a project fails
  - the people on it are dispersed to other projects
  - the metrics that have been collected for the failed project often vanish
  - a retrospective (a.k.a. post-mortem) doesn't happen for the failed project
  - in summary, it doesn't get counted and therefore it doesn't count
  - *yet this might be the most instructive project*
  - Quality Software Management, Vol. 2. First Order Measurement
    - Gerald M. Weinberg

Mine #15

Metrics often outlive their usefulness.

Dead Metrics

- In “Mad About Metrics”, Tom DeMarco tells a story about identifying the impact of interruptions at a large organization
- The metric successfully raised awareness of the problem, to the degree that people began to change their behaviour
- After a few months, the number of interruptions stabilized at a much lower level than before
- The metric stopped being useful, but years later, the company was still collecting (and presumably ignoring) the metric
- When a metric stops providing new information, think seriously about not collecting it

Mine #16

Many metrics are focused on tracking cost. Few metrics are oriented towards tracking benefits.

Cost vs. Value

- Weinberg asserts that the question he’s heard more than any other is “Why does software cost so much?”
  - to which he answers “Compared to what”?
  - DeMarco suggests that Weinberg’s answer is fatuous, since the questioner isn’t interested in the answer
- Yet DeMarco does suggest that we need to rejig metrics into a benefit focus, rather than a cost focus
- Many metrics compare the cost of two things, rather than the cost/benefit of two things
- This requires a “compared to what?” reframe

Mine #17

Metrics are collected for someone’s purpose. The metrics can and will be gamed to further that purpose.

An Instructive Story

- A colleague (who must remain nameless) tells the story of an organization that polled its internal customers for satisfaction
- The average score was 4.7 out of 10, which my colleague suggests was startlingly high—1s or 2s would have been more like it
- Why were these results being produced?
- At least one answer: bonuses across the board were calculated as a base value *multiplied by the average satisfaction score*.
- That is, the internal customers were given strong disincentives for speaking their minds.

Mine #18

People sometimes lie.
Or they sometimes leave out the truth.
Or they provide an answer that they think the questioner wants to hear.

Getting to the Truth

- When interviewers or reporters choose the next question, the choice is influenced by the answer to the last question.
- Police investigations don't use multiple choice tests.
- Successful lawyers plead cases by framing context.

Mine #19

When we measure people, they change their behaviour in ways that we might not expect or intend.

The Hawthorne Effect

- A set of studies on the influence of illumination in the workplace
  - done at Western Electric in Chicago, 1924-1933
- Three experimental groups, no controls
  - all showed increase in performance with more light
- Experimental and control groups
  - the control group got steady light, and the experimental group got gradually increasing light
  - performance increased for both groups
  - performance increases continued, even after the light level was *decreased* for the experimental group

The Hawthorne Effect

- More subtle experiments followed
  - If employees were told that bright is good, they tended to report that they liked the brighter light.
  - If employees were told that dim is good, they tended to report that they liked the *dimmer* light.
  - These results held *even after the groups were misled about the intensity of the light*

Mine #20

In order for statistics to work properly, we need a good population size and a good sample size.

Small Numbers and Large Numbers

- Large numbers tend to blur differences between elements, such that the differences are (heuristically) insignificant with respect to the measurement.
- We might be able to recognize and account for differences if numbers are sufficiently small
- What about the case of medium-sized numbers where differences may be *very* important?
- Might stories be more efficient informative in this case?

Even though it's anecdotal, it's still empirical.

Mine #21

Academic researchers, whom we might expect to do better science, can do some *really* shoddy science.

One Truly Scary Paper

GERT: An Empirical Reliability Estimation and Testing Feedback Tool
Martin Davidsson, Jiang Zheng, Nachiappan Nagappan, Laurie Williams, Mladen Vouk
Department of Computer Science
North Carolina State University, Raleigh
http://research.microsoft.com/users/nachin/papers/ISSRE_GERT.pdf

"GERT", the authors claim, "provides a means of calculating software reliability estimates and of quantifying the uncertainty in the estimate (a.k.a. the confidence interval)."

This is probably not the worst paper of its kind, but it's a paradigmatic example of bad metrics compounded. Let's have a look.

Metrics Collected by GERT

1. number of test cases / SLOC (R1);
2. number of test cases / number of requirements (R2);
3. test lines of code / SLOC (R3);
4. number of assertions / SLOC (R4);
5. number of test classes / number of source classes (R5);
6. number of conditionals / SLOC (R6);
7. SLOC / number of source classes (R7);
8. statement coverage (R8); and
9. branch coverage (R9).

The paper notes, "However, not all metrics have consistently demonstrated a correlation with software reliability." No kidding.

Lines of Code Metrics

Here are two lines of code from the same program. Are they equivalent?

```
obj.visibility=v;
```

```
for(i=0;!x&&d.layers&&i<d.layers.length;i++)
\ x=MM_findObj(n,d.layers[i].document);
```

In the first example, it appears as though one bit is being set.

In the second, multiple values are (conditionally) being initialized, compared, set, incremented, referenced, or dereferenced.

This is like counting tricycles and space shuttles as equivalent items.

The Study

- This study was based on projects done by 2nd and 3rd year computer science students.
- Each project was an open-source Eclipse plug-in, written in Java, that automated the collection of project metrics.
- Each project was developed by a group of four or five students during a six-week final class project.
- 22 projects were submitted; all were used in the analysis.

Table 2: Eclipse project size

Metric	Mean	Std Dev	Max	Min
SLOC	1996.9	835.9	3631	617
TLOC	688.7	464.4	2115	156

The Model

- "Plug-ins were tested using a set of 31 black-box test cases."
  - we're counting test cases here; there is no other description of them
- "Twenty six of these were acceptance tests and were given to the students during development."
  - what were the other five tests?
- "The actual reliability of the student programs was approximated by inputting the results of these black box test cases into the Nelson model."
- Using a randomly-chosen set of 18 programs, they built this scary equation.
  - Why only 18?

Reliability Estimate = 0.859 + 0.09459*R1 + 0.01333*R2 - 0.0404*R3 + 1.674*R4 + 0.01242*R5 - 1.222*R6 + 0.000867*R7

Expanding that out…

0.859 + (0.09459*(number of test cases / SLOC)) + (0.01333*(number of test cases / number of requirements)) - (0.0404*(test lines of code / SLOC)) + 1.674*(number of assertions / SLOC) + 0.01242*(number of test classes / number of source classes) - 1.222*(number of conditionals / SLOC) + 0.000867*(SLOC / number of source classes)

Note that they're using multiples on the resolution of 1 in 1,000,000 on a project with 18 samples.

Where are the constants coming from?

Branch coverage and statement coverage, collected by the tool, are left out. Not that it matters….

This is **mathturbation**.

Clearing the Minefield

Each of the following ideas
is a heuristic approach;
a suggestion, not an instruction.

Clearing the Minefield

You might find out some very important things just by trying them and failing.

Clearing the Minefield

- Don't produce, offer or accept a number without a comment
- "Never give a number to a bureaucrat"
  - Plum's Second Law
- Emphasize stories and narratives

Clearing the Minefield

- Remove *control metrics* that are linked to pay, bonuses, performance evaluation, etc.
  - control metrics are metrics that trigger some action, usually automatically
  - a metric that is used to control something will eventually be used to control you
- Foster *inquiry metrics*
  - inquiry metrics are metrics that prompt us to ask questions
- Relax measurement when the metric stops changing
  - if the results aren't satisfactory, try measuring something else for a while

Prefer Assessment to Measurement

- Don't feel that you have to render everything into a numeric value
- Observation can go directly to assessment without quantified measurement
- What other modes, beside numerical ones, can you use to assess progress?

DePree's Signs of Entropy (1)

- a tendency towards superficiality
- a dark tension among key people
- no longer having time for celebration and ritual
- a growing feeling that rewards and goals are the same thing
- when people stop telling tribal stories or cannot understand them
- a recurring effort by some to convince others that business, after all, is quite simple
- intolerance of complexity, ignorance of ambiguity

DePree's Signs of Entropy (2)

- differing understanding of words like "responsibility", "service", or "trust"
- when problem makers outnumber problem solvers
- confusion between heroes and celebrities
- leaders who seek to control, rather than to liberate
- concern for vision and risk superseded by daily pressures
- orientation towards the rules of business schools instead of value orientation
- when people speak of customers as impositions

DePree's Signs of Entropy (3)

- manuals
- a growing urge to quantify history and the future
- the urge to establish ratios
- leaders who rely on structures instead of people
- a loss of confidence in judgement, experience, and wisdom
- a loss of grace and style and civility
- a loss of respect for the English language

Max DePree, *Leadership is an Art*, Bantam Doubleday Dell, 1989

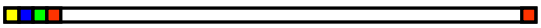
Quoted in Gerald M. Weinberg, *Quality Software Management Vol. 2, First-Order Measurement*, Dorset House Press, 1993

Other Modes of Assessment

- Try standup meetings or scrums
  - short meetings that identify needs for further meetings
- Try laddering exercises
  - ranking, rather than measuring
- Try temperature readings
  - appreciations
  - new information
  - puzzles
  - complaints
  - hopes and wishes
- Recognize the ways in which data can be converted to information, and vice versa

Clearing the Minefield

Try some exercises



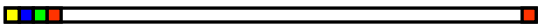
Exercise

Design two forms to collect information about satisfaction with a product or service.

Have one focus on numbers only. Have the other solicit written, narrative answers.

Which one provides a guide to improvement?

Meta-question: Do you evaluate the results of this exercise with a number, or with a discussion?

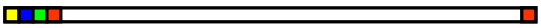


Exercises

Analyze a news report that quotes a lot of statistical data.

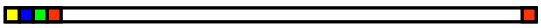
What information is missing from the report?

What information is plainly bogus?



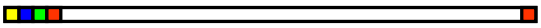
Exercises

Come up with a metric
for measuring tester performance,
such that the metric can't be gamed.



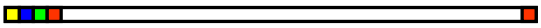
Exercises

Game the metric from the previous exercise.



Clearing the Minefield

Seek information, not just data



Clearing the Minefield

Always ask "Compared to what?"

Readings

- Quality Software Management, Vol. 2., “First Order Measurement”
  - Gerald M. Weinberg
- How to Lie with Statistics
  - Darrell Huff
- Visual Explanations
  - Edward Tufte
- Freakonomics
  - Stephen Leavitt

Readings

- Why Does Software Cost So Much?
  - Tom DeMarco
- Tools of Critical Thinking
  - David Levy
- “Software Engineering Metrics: What Do They Measure and How Do We Know?”
  - Cem Kaner and Walter P. Bond
  - http://www.kaner.com/pdfs/metrics2004.pdf