

The Myths Behind Software Metrics

Pacific Northwest Software Quality Conference
October 14, 2013

**Douglas Hoffman, BACS, MBA, MSEE,
ASQ-CSQE, ASQ-CMQ/OE, ASQ Fellow**
Software Quality Methods, LLC. (SQM)
www.SoftwareQualityMethods.com
doug.hoffman@acm.org

Doug Hoffman

Copyright © 2013 SQM, LLC.

1

Myths and Superstitions

- Myths are legendary stories that explain a practice
- Superstitions are beliefs or practices resulting from ignorance or trust
- Myths may be based on actual events that have been lost in history
- Superstitions usually remain unchallenged
- Myths and superstitions are often taken as fact and applied out of context

Doug Hoffman

Copyright © 2013 SQM, LLC.

2

Some Myths About Software Metrics



1. Measurement is easy
2. If we measure it we can control it
3. Measuring something doesn't change anything else
4. The arithmetic behind software metrics makes logical and mathematical sense
5. A metric is valid and useful if the measures are precise and accurate
6. Most metrics reflect attributes reasonably well
7. Commonly used software metrics are valid

Doug Hoffman

Copyright © 2013 SQM, LLC.

3

Measurement Terminology

- Entity (object or event we're trying to describe)
- Attribute (the property of an entity we're trying to measure)
- Measurement (assigning a value)
- Measure (value assigned)
- Reliability (repeatability of the measurement)
- Validity (freedom from bias)
- Measurement Error (systematic and random variation)
- Model (how the measures relate to the entity)

Doug Hoffman

Copyright © 2013 SQM, LLC.

Slide 4

Measurement vs. Metrics

- A *measurement* is an observation related to an entity, ascribing a value to it based on some model or theory, for the purpose of describing something about it.
- The *metric* is the application of a mathematical function that establishes or (or “maps”) the relationship between the measurement(s) and some attribute we want to describe.

$$\text{Transaction Rate} = \frac{\text{Transaction Count}}{\text{Time Interval}}$$

Doug Hoffman

Copyright © 2013 SQM, LLC.

5

1: Measurement Is Not Easy

- What are we measuring?
- What is our standard of measurement?
- How accurate is our measurement?
- How repeatable is the measurement?
- What is the natural variation in the attribute?



http://www.iso.org/iso/iso_cafe_measurement.htm

Doug Hoffman

Copyright © 2013 SQM, LLC.

6

Measurement Issues

- What entity are we trying to measure? (E.g., program size, bugginess, transaction rate)
- Do we understand what the entity is? (E.g., program size, complexity)
- Can it be measured directly? (E.g., program size, complexity)
- Is the measure repeatable (do we get the same value at different times)? (E.g., time, bug counts)
- What amount of variation should we expect? (E.g., time estimates, speed of a test)

Doug Hoffman

Copyright © 2013 SQM, LLC.

7

2: Measurement and Control

- Measuring something means we have some information
 - It might be good or poor information
 - It might or might not relate to what we want to know
- “You can't control what you can't measure” Tom DeMarco, *Controlling Software Projects* (1982)
- “I've become increasingly uncomfortable with it.” Tom DeMarco, *IEEE Software* (2009)
(http://www2.computer.org/cms/Computer.org/ComputingNow/homepage/2009/0709/rW_SO_Viewpoints.pdf)

Doug Hoffman

Copyright © 2013 SQM, LLC.

8

Observation Versus Control

- Depends on what we do with the metric
- An observation metric is one used to understand about something
 - Usually limited distribution
 - “Personal” metrics
- A control metric is one used to manage
 - Control may happen just because the measures are taken
 - Almost always has unintended side effects
 - Human behavior is hard to control

Doug Hoffman

Copyright © 2013 SQM, LLC.

9

3. Side Effects of Measures: Models

Without a mechanism that links changes in the attribute being measured to the numbers we get, we have a “measure” that can cause chaos:

- If we measure product quality by bug counts, do higher bug counts always mean less quality?
- If we measure the productivity of the test group by the number of bugs found, does fewer bugs always mean lower productivity? (What happens to in-depth analysis, strategic planning, bug follow-up, etc.?)
- If we measure failure rate to compute MTTF, does a longer time between failures always mean improvement in MTTF? (What about CPU speed, data entry rate, etc.?)

Doug Hoffman

Copyright © 2013 SQM, LLC.

Slide 10

Side Effects of Measures: Numbers

- Estimates usually back up already made conclusions
- Natural variation is usually ignored and measures treated as “correct”
- People will optimize what is tracked:
 - If you track numbers of bugs filed by the testers, the bug count numbers will go up.
 - If you track defect density, the number of LOC may go up without adding new functionality

Doug Hoffman

Copyright © 2013 SQM, LLC.

Slide 11

Side Effects of Measures: Human Behavior

- People will usually provide the numbers they think are expected:
 - It may be a conscious response to unreasonable expectations
 - It may be subconscious: for example, people will slack off when the numbers exceed expectations and will work harder to make the numbers when below expectations
 - Expectation is key: management doesn't have to have expectations, people only need to think they might

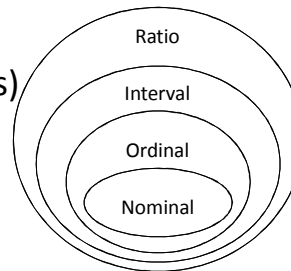
Doug Hoffman

Copyright © 2013 SQM, LLC.

Slide 12

4. Measurement Scales

- **Nominal** (categories)
 - Code objects or functions
- **Ordinal** (ranking; can be put in an order or hierarchy)
 - Bug severity (S1, S2, etc.)
- **Interval** (consistent differences)
 - Milestone dates
- **Ratio** (proportions)
 - Transaction rate (TPS)
 - Time intervals



Doug Hoffman

Copyright © 2013 SQM, LLC.

13

What Math Can You Do for Each Scale?

- **Nominal** (e.g., object names)
 - math using nominal values is unreasonable because they are only categories
- **Ordinal** (order by rank: 1st, 2nd, 3rd, etc.)
 - most math using ordinal values is unreasonable because they are only ordered categories
 - can identify the median (the middle element), divide into percentiles
 - no ratios, sums, or differences
 - e.g., for a four level bug severity ranking (Sev 1 through 4), the average Severity is not 2.5; the difference between Sev 1 and 2 is not the same as Sev 2 and 3

Doug Hoffman

Copyright © 2013 SQM, LLC.

14

What Math Can You Do for Each Scale?

- Interval (e.g., milestone dates)
 - doesn't require a true zero point: think calendars, thermometers
 - count units of distance between items
 - identify the mean, calculate the standard deviation, identify a rank order, etc.
 - but no zero-based stats like ratios
- Ratio (e.g., LOC)
 - all arithmetic operations apply
 - requires a true zero point
 - includes *fundamental* and *derived* measures

Doug Hoffman

Copyright © 2013 SQM, LLC.

15

What About Math and Metrics?

- Measures and computations may be helpful in context
 - they must be carefully chosen and fully explained
 - developers and testers may be able to improve if they are given unbiased numbers
 - understand the model and how well the metric represents the desired attribute
- Most computations are unhelpful and lead to dysfunctional behaviors
 - “42” is not the answer to any real question
 - people provide what they think you want
 - many metrics are generated to confirm beliefs

Doug Hoffman

Copyright © 2013 SQM, LLC.

16

5. Precision vs. Accuracy

- Precision is about the number of significant digits in a number (e.g., 4 seconds vs. 4.00 seconds)
- Accuracy is about having the correct value
- Estimates are approximations (guesses). The accuracy of an estimate is poor regardless of its precision
- Surrogate measures are neither precise or accurate measures of what we want to know
- Many metrics are computed with great precision from inaccurate or surrogate numbers

Doug Hoffman

Copyright © 2013 SQM, LLC.

17

Precision

- Precision is about the number of significant digits in a number. The number of decimal places of precision *implies* that the numbers are accurate and meaningful (e.g., defect density).
- Surrogate measures may provide precise and accurate measures of something sort of like what we really want to measure. They are neither precise nor accurate measures of what we want to know (e.g., program size)
- Many metrics are computed with great precision from inaccurate or surrogate numbers.

Doug Hoffman

Copyright © 2013 SQM, LLC.

18

Accuracy

- Accuracy is about having the correct value
- Estimates are approximations. The accuracy of an estimate is poor regardless of its precision. (e.g., likelihood of customers encountering a bug).
- Arithmetic using estimates compounds the inaccuracy
- Surrogate are necessarily inaccurate measures of what we want to know. They may not even have direct correlation with what we want to know.
- Many metrics are computed using inaccurate or surrogate measures. Computations using surrogate measures may have less meaning than “gut feel”

Doug Hoffman

Copyright © 2013 SQM, LLC.

19

6. Metrics Models

- A model describes a relationship between a real world attribute and measures/metrics
- Most models are approximations or use surrogate measures
- Without a rational model, metrics are meaningless
- With bogus arithmetic, metrics are false and misleading

Doug Hoffman

Copyright © 2013 SQM, LLC.

20

Some Models Are Better Than Others



To some degree



For some purposes

All models are wrong, but some are useful.

George E. P. Box

Doug Hoffman

Copyright © 2013 SQM, LLC.

21

Example Metrics Models

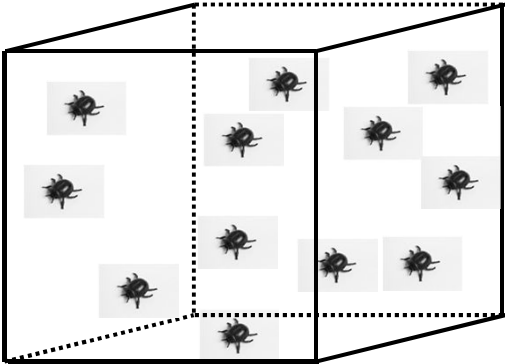
- Bug density: the number of bugs per unit size (e.g., defects/KLOC)
 - What does bug density mean? What does it measure?
 - How accurate is the bug count? How close to the actual number of bugs?
 - How is the size measured? Is this meaningful?
- Bug find rate: the number of bugs found per unit time
 - How accurate are time measures?
 - How are numbers of bugs counted?

Doug Hoffman

Copyright © 2013 SQM, LLC.

22

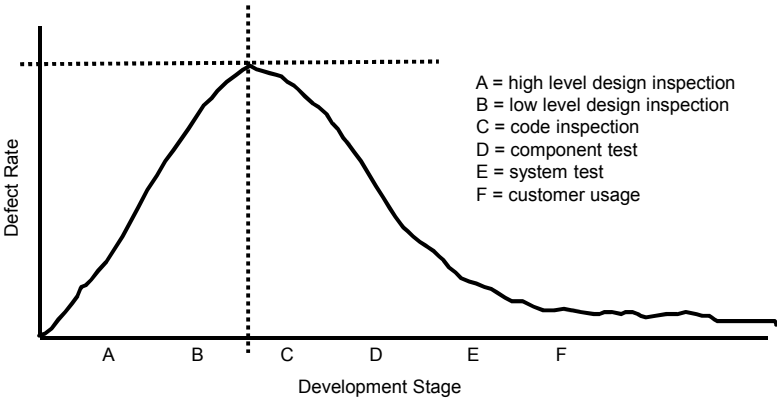
A Common Bug Model: Bug Density



$$\text{Bug Density} = \frac{\text{Bug Count}}{\text{Size of program}}$$

Doug Hoffman Copyright © 2013 SQM, LLC. Slide 23

A Common Bug Model: Bug Find Rate



Doug Hoffman Copyright © 2013 SQM, LLC. Slide 24

7. Common Measures and Metrics

- Bug counts, program size, and bug density
- Event timing
- Cyclomatic Complexity
- Number of test cases
- Number of tests passing or failing
- Percent completion of tests
- Predicted bugs

Doug Hoffman

Copyright © 2013 SQM, LLC.

25

Questions I Use to Test Metrics

- If the attribute improves, does the measure or metric always reflect the improvement?
- What's the underlying model to explain how the measure or metric maps to the attribute?
- Is the basic arithmetic valid?
- How can the measure or metric be distorted?
- What things can change the attribute without being reflected in the measure or metric?

Doug Hoffman

Copyright © 2013 SQM, LLC.

26

What Can We Do About Metrics?

- Discourage the use of bad measures and metrics
- Invent ones useful in your context
- Use them only for observation
- Keep them private to the people who generate them; encourage personal use
- Help people interpret the numbers
- Provide only aggregates to management

Doug Hoffman

Copyright © 2013 SQM, LLC.

27

What If You Have Metrics?

- Identify (create) metrics that are useful
- Use them only for observation
- Create balanced scorecards; anticipate side effects
- Use them to trigger questions, not actions
- Communicate the value (validity) of the metrics with managers
- Avoid direct application for project control

Doug Hoffman

Copyright © 2013 SQM, LLC.

28

Recapping



- Good measurement is hard
- We really can't control human behavior
- Measuring something changes other things
- Arithmetic for many metrics doesn't make mathematical sense
- Precision and accuracy are irrelevant for bad measures or metrics
- Few metrics have grounding in any kind of rational model
- The most commonly used software metrics don't tell us what we really want to know