Questioning Measurement

Michael Bolton, DevelopSense
Toronto, Ontario, Canada
mb@developsense.com
http://www.developsense.com
+1 (416) 656-5160

TASSQ
June 2011

Who I Am

Michael Bolton
DevelopSense, Toronto
Canada
mb@developsense.com
(416) 992-8378
http://www.developsense.com/past.html

Notes & Acknowledgements

- Acknowledgements to my colleagues and mentors:
  - James Bach
  - Jon Bach
  - Cem Kaner
  - Jerry Weinberg
- Thanks to our hosts at STAR East
- Get the complete version of this presentation
  - from my USB key
  - with a business card
  - from http://www.developsense.com/past.html

These notes are rough drafts. I won't talk about everything in them, and I will talk about stuff NOT in them.

I Don't Hate Numbers

- I love numbers *so much* that I can't stand to see them abused as they are by people in our profession.
- This talk is designed to help you spot critical thinking errors that might cause you to miss observations and mislead your client—or yourself.
- The intention is not to suggest that measurement is useless, but to expand our notions of what measurement might be.

> Imperfections in measurement are always a problem,
> but they're a devastating problem
> only when we don't recognize them.
>
> **---Daniel Gilbert, Stumbling on Happiness**

Exercise

Measure The Height of Your Chair

Why Do We Measure?

- self-assessment and improvement
- evaluating project status
- evaluating staff performance
- informing others about the characteristics of the product
- informing external authorities about the characteristics of the product

—Kaner and Bond

See http://www.kaner.com/pdfs/metrics2004.pdf
See Robert Austin, *Measuring and Managing Performance in Organizations*

But there's an over-riding reason…

Fear of irrationality

Quantifying something complex provides a kind of observational integrity, but with a side effect: *information loss.*
As Cem Kaner describes the social sciences, think of testing and measurement as providing *partial answers that might be useful.*

Ignore the risks of measurement, and distortion and dysfunction are likely. Manage to your metric, and dysfunction is *guaranteed.*

How Do We Measure?

- Third-order measurement
  - highly instrumented, used to discover natural laws
  - "What *will* happen? What *always* happens?"

- Second-order measurement
  - often instrumented, used to refine first-order observation
  - used to tune existing systems
  - "What's *really* going on here? What's happening right now?"

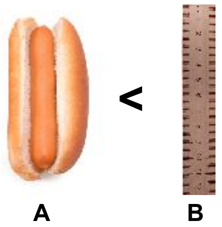
What Is Measurement?

"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."

- Source: "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
- Note that this is a description focused on *quantitative* measurement

Measurement vs. Metrics

- A *measurement* is two or more observations that are compared, based on some model or theory, for the purpose of making some distinction.
- The *metric* is the function that relates (or "maps") one observation to another.



Theory:

If (len)A < (len)B then A is not really a foot-long dog.

Exercise

How could this go wrong?

Questioning Measurement

What might we observe?
- What are we choosing NOT to observe?
- What are we not observing by accident?

- What comparisons can we make?
  - are those comparisons valid? relevant?
  - are those comparisons reliable?
- What are we trying to assess?
  - How will we respond to the information?
  - Will we use it to control? Or to learn?

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

Darrell Huff's Five Questions

- Who says so?
- How do they know?
- What's missing?
- Did somebody change the subject?
- Does it make sense?

From *How to Lie with Statistics*, originally published in 1954.

Measurement Scales

equality (🔑 identity; constructs; this is that)

nominal (🔑 labels; the number isn't quantity at all)

ordinal (🔑 order; number suggests a rank)

interval (🔑 differences, and equalities in them)

ratio (🔑 differences and proportions in

Mistake one kind of scale for another, and you can guarantee invalid, unsupportable conclusions.

derivative (🔑 combinations of the above)

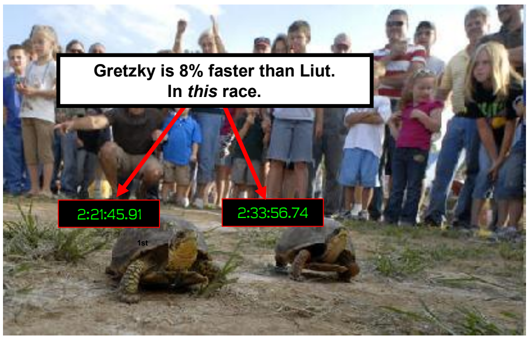
Nominal Scale

Ordinal Scale

Interval Scale

Gretzky is 2 minutes and 12 seconds faster than Liut.

Ratio Scale


Gretzky is 8% faster than Liut. In *this* race.

Kaner & Bond on Construct Validity

- What is the purpose of your measurement? The scope?
- What is the attribute you are trying to measure?
- What are the scale and variability of this attribute?
- What is the instrument you're using? What is its scale and variability?
- What function (metric) do you use to assign a value to the attribute?
- What's the natural scale of the metric?
- What is the relationship of the attribute to the metric's output?
- What are the natural, foreseeable side effects of using this measure?

The essence of good measurement is a model that incorporates answers to questions like these.

If you don't have solid answers, you aren't doing measurement; you are just playing with numbers.

How Else Do We Measure?

- First-order measurement
  - minimal fuss, direct observation, minimal instrumentation
  - used to inform a control action OR to prompt search for more refined information
  - "What's going on? What should we do? Where should we look?"

Weinberg suggests that, in software development, we're obsessed with trying to make third- and second-order measurements when first-order measurements might be all we need—and tend to be much cheaper.

What Is Measurement?

Measurement is the art and science of making reliable observations.
—Jerry Weinberg, Quality Software Management Vol. 2

- Since the time of Aristotle (at least), we've known about two kinds of measurement that inform decisions
  - "Two pounds of meat"

We waste time and effort when we try to obtain six-decimal-place answers to whole-number questions.

- http://www.developsense.com/articles/2009-05-IssuesAboutMetricsAboutPeople.pdf

A Synthesis

"Measurement is
the art and science of making reliable observations,
based on modeling and comparison
of objects, attributes, or events,
for the purposes of understanding distinctions,
making assessments, and informing evaluations."

- My intention here is to highlight
  - why we measure (assessment and evaluation)
  - how we measure (observation, and comparison based on models)
- Measurement might be qualitative or quantitative, but assessment and evaluation are always qualitative:

What do we want?

Why Prefer First-Order Measures?

- When you're driving, are you mostly concerned about…
  - your velocity, acceleration, vehicle mass, drag co-efficient, frictional force? (third-order)
  - your engine temperature, RPMs, and current rate of gas consumption? (second-order)
  - looking out the window to avoid hitting things?

I've observed *many* projects that have crashed because managers were overly focused on the dashboard instead of the traffic and obstacles around them, and the road ahead.

What kind of driver do you trust?

The Power of Limited Information

- Snap judgments and heuristics are central to our decision-making process
- Because they're fast and frugal, they may sometimes be more valuable than complex and rigourous analysis

People often make snap (first-order) observations or decisions
and then use (second-order) numbers
to *test* them.

Control vs. Inquiry Measurement

- A **control measurement** is a measurement that **drives decisions**.
  - *Any measurement you use to control a self-aware system will be used by that system to control YOU.*

- An **inquiry measurement** is any measurement that **helps you ask the right questions** at the right time.
  - Inquiry measurements are also vulnerable to gaming, but the stakes are far lower, so there's less incentive for manipulation.

This slide is taken from the work of my colleague, James Bach.
http://www.satisfice.com

Control vs. Inquiry

- Remove *control metrics* that are linked to pay, bonuses, performance evaluation, etc.
  - control metrics 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 prompt us to ask questions
- Relax measurement when the metric stops changing
  - if you're not obtaining new information, try measuring something else for a while

Counting Bugs? Or Identifying Issues?

- In software testing we talk about finding bugs, and reporting them.
- There's something worse than a bug: an *issue* —something that slows down or prevents your ability to find a problem.
- Mechanisms intended to provide management information often *ignore* issues.

- Bug reporting systems focus on bugs, and suppress reporting of issues
- Mechanisms intended to provide management control often *create* issues.
- For a description of issues and how we might report them, see http://www.developsense.com/blog/2011/01/youve-got-issues/.

What Are The Factors of a "Test Case"?

Power: will this test reveal a problem?
Validity: is problem revealed a genuine problem?
Value: is the information is useful to your product, project, or client?
Credibility: will clients believe it represents things people will actually do?
Representative: is it a good example of plausible similar tests?
Motivating: will the test make us want to fix the problem it reveals?
Performable: Can the test be performed as designed?
Maintainable: Can the test be revised easily when the product changes?
Reusable: It is easy and inexpensive to reuse the test for other products?
Pop: Will the test challenge our assumptions and reveal new information?
Coverage: Will the test exercise the product in a unique way?
Easy to evaluate: Is there a clear answer to the question the test poses?

Many of these ideas come from
Kaner & Bach's Black Box Software Testing Course

http://www.wtst.org/WTST7/BBSTwtst2008kanermeeting.pdf

What Are The Factors of a "Test Case"?

Supports debugging: Will it provide useful results for the programmer?
Repeatable: does the test reveal a problem consistently?
Mutability: can the test be adapted to other test ideas?
Complexity: are there interesting interactions between components?
Simplicity: does the test successfully isolate a problem of interest?
Accountability: can you explain, justify, and prove that you run the test?
Equipment cost: do you need specialized gear to run the test?
Development cost: what resources are required to design the test?
Setup cost: what time and resources are required to prepare for the test?
Execution time: how long does it take the test to run?
Reporting time: what effort is required to communicate the results?
Opportunity cost: what valuable work could you do instead of this test?

Many of these ideas come from
Kaner & Bach's Black Box Software Testing Course

http://www.wtst.org/WTST7/BBSTwtst2008kanermeeting.pdf

Don't Just Count Them!

- You've just seen at least 24 factors by which we might describe or evaluate a given test
- Bugs have similar numbers of factors, if only we pause to think about them
- Many factors aren't usefully quantifiable
  - yet they might be supremely important
  - people base decisions on politics and emotions
  - people have emotional reactions to software
- Models may leave out many dimensions
  - some of which might also be very important
- Testers are even more complex
  - tester effectiveness needs multi-dimensional measures

Counting *Ideas*

- Don’t count test cases
  - test cases are *ideas*; ideas aren’t things
  - counting ideas is reification error
- Don’t judge product health by bug counts
  - try bug stories instead
- Don’t measure testers by counting 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**

Lines of Code

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.

Precision vs. Accuracy

- What time is it, *exactly, right now?*
- My watch say 5:35:21, which is precise, but it’s *wrong*.
- Many people will offer precise numbers—but will they be *accurate*?

Heuristic: In software development processes, the more precise the number provided, the more likely it is to be based on an unsupportable model.

Testing is a Structured Process

In excellent testing, one structure tends to dominate all the others:

The Testing Story

Expert testers construct a compelling story of their testing. It is this story that gives testing a backbone.

To test is to compose, edit, narrate, and justify a story.

You must tell a story about the product…

…about how it failed, and how it *might* fail...

…in ways that matter to your various clients.

But also tell a story about testing…

…how you configured, operated and observed it…

…about what you haven’t tested, yet…

…or won’t test, at all…

…and about why what you did was good enough.

In our Rapid Testing Course, James Bach and I talk this way about exploratory testing—but the concept carries to any expert testing.

Hold It!

Aren’t stories *subjective* by nature?

Sure.

***All* observations—including quantitative observations—depend on observers, their choices about what to observe, and the quality of their observations.**

Break Down the Testing Story

- Who are the characters? (People? Bugs?)
- What matters to them?
- What happens over time?
- Why do they do what they do?
- When did the story happen?
- Where did they go?
- What did they do?

Why should we care?

Quantifying vs. Qualifying

- Comparisons and assessments aren't necessarily numerical (ask Goldilocks).
- 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.
- Could you tell convincing, motivating stories?
- Could you use ranking or laddering?
- Could you use reports and opinions from multiple people?

What if you asked for
"things you liked
and things you didn't like"?

Assessment Without Numbers

- Observation can go directly to assessment without quantified measurement
  - this is the first-order approach
- Ask what other modes, beside numerical ones, you could use for evaluation
  - start by asking *what problem you want to solve* or *what situation you'd like to assess*
- If you're worried that observations and assessments are subjective, ask *several* people who matter

Possibly Useful Measures

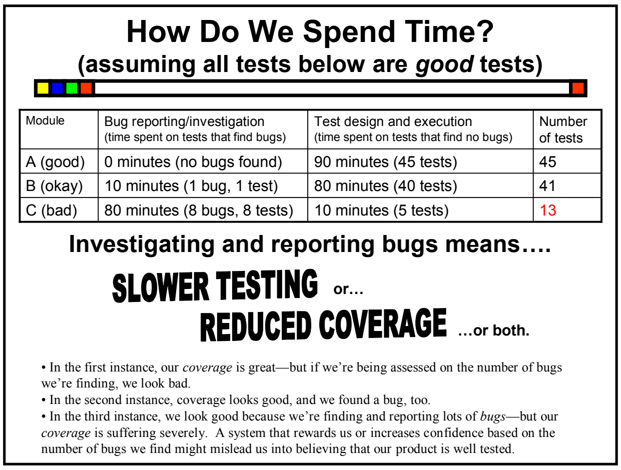
- The Binary Metric
  - "Any showstoppers?"
- The Issues List
  - list bugs and issues by importance to some stakeholder; optionally rank them first
  - note that testability issues may be most important; problems that prevent or slow down testing mean that other problems have more places to hide
- The Furrowed Brow Test (a.k.a. Squirm Test)
  - announce that we're planning to ship on schedule
  - observe posture, furrowed brows, and squirming

Possibly Useful Measures

- Try tracking time spent on
  - test design and execution
  - bug investigation and reporting
  - test setup
  - none of the above (non-testing work such as meetings, email, administrivia, etc.)
- Try tracking this to 5% - 10% granularity
  - finer granularity means that "time tracking" requires significant non-testing effort
  - do you include "time spent on time tracking" in your time tracking?
  - do you include "time to estimate" in their estimates?

Test Session Effectiveness

- A "perfectly effective" testing session is one entirely dedicated to test design, test execution, and learning
  - a "perfect" session is the exception, not the rule
- Test design and execution tend to contribute to test coverage
  - varied tests tend to provide more coverage than repeated tests
- Setup, bug investigation, and reporting take time away from test design and execution
- Suppose that testing a feature takes two minutes
  - this is a highly arbitrary and artificial assumption—that is, it's *wrong*, but we use it to model an issue and make a point
- Suppose also that it takes eight extra minutes to investigate and report a bug
  - another stupid, sweeping generalization in service of the point
- In a 90-minute session, we can run 45 feature tests—*as long as we don't find any bugs*



How Do We Spend Time?
(assuming all tests below are *good* tests)

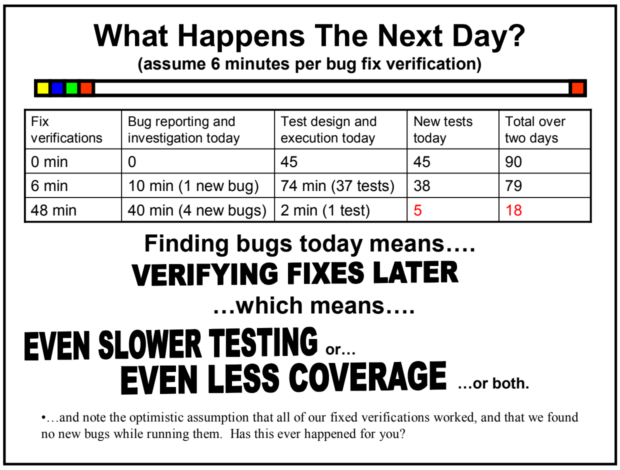
Module	Bug reporting/investigation (time spent on tests that find bugs)	Test design and execution (time spent on tests that find no bugs)	Number of tests
A (good)	0 minutes (no bugs found)	90 minutes (45 tests)	45
B (okay)	10 minutes (1 bug, 1 test)	80 minutes (40 tests)	41
C (bad)	80 minutes (8 bugs, 8 tests)	10 minutes (5 tests)	13

Investigating and reporting bugs means….

SLOWER TESTING or…

REDUCED COVERAGE …or both.

- In the first instance, our *coverage* is great—but if we're being assessed on the number of bugs we're finding, we look bad.
- In the second instance, coverage looks good, and we found a bug, too.
- In the third instance, we look good because we're finding and reporting lots of *bugs*—but our *coverage* is suffering severely. A system that rewards us or increases confidence based on the number of bugs we find might mislead us into believing that our product is well tested.



What Happens The Next Day?
(assume 6 minutes per bug fix verification)

Fix verifications	Bug reporting and investigation today	Test design and execution today	New tests today	Total over two days
0 min	0	45	45	90
6 min	10 min (1 new bug)	74 min (37 tests)	38	79
48 min	40 min (4 new bugs)	2 min (1 test)	5	18

Finding bugs today means….

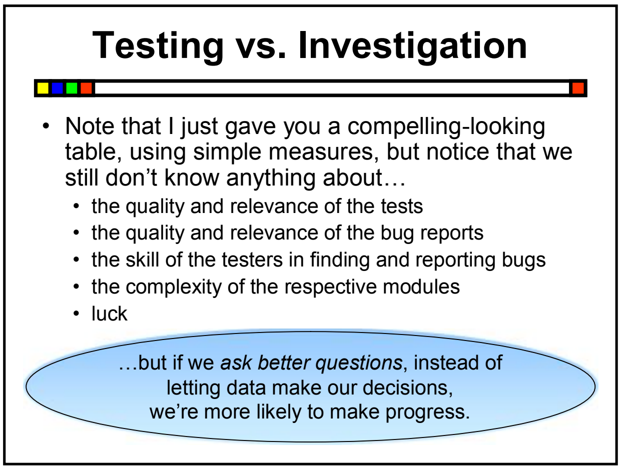
VERIFYING FIXES LATER

…which means….

EVEN SLOWER TESTING or…

EVEN LESS COVERAGE …or both.

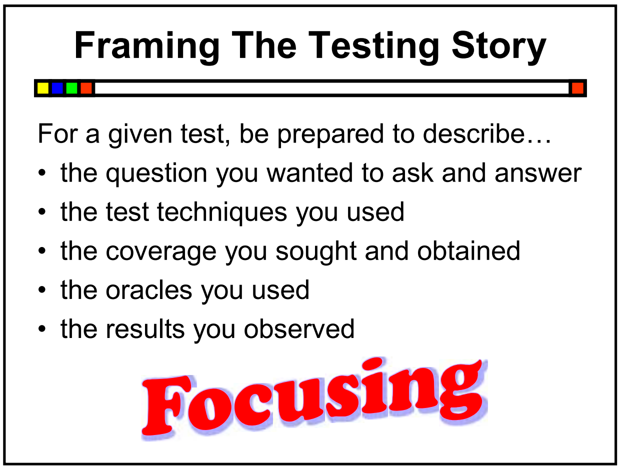
- …and note the optimistic assumption that all of our fixed verifications worked, and that we found no new bugs while running them. Has this ever happened for you?



Testing vs. Investigation

- Note that I just gave you a compelling-looking table, using simple measures, but notice that we still don't know anything about…
  - the quality and relevance of the tests
  - the quality and relevance of the bug reports
  - the skill of the testers in finding and reporting bugs
  - the complexity of the respective modules
  - luck

…but if we *ask better questions*, instead of letting data make our decisions, we're more likely to make progress.

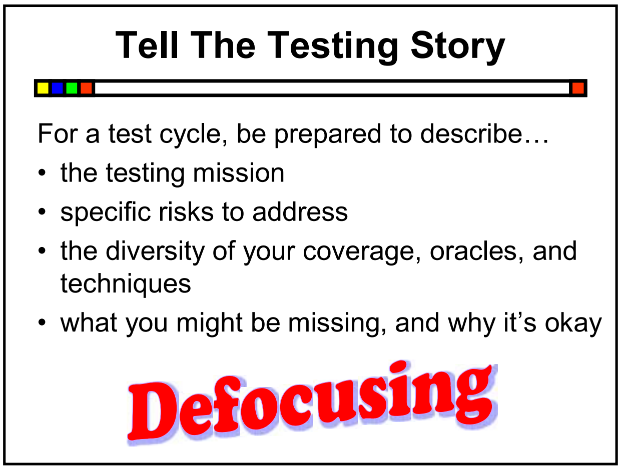


Framing The Testing Story

For a given test, be prepared to describe…

- the question you wanted to ask and answer
- the test techniques you used
- the coverage you sought and obtained
- the oracles you used
- the results you observed

Focusing

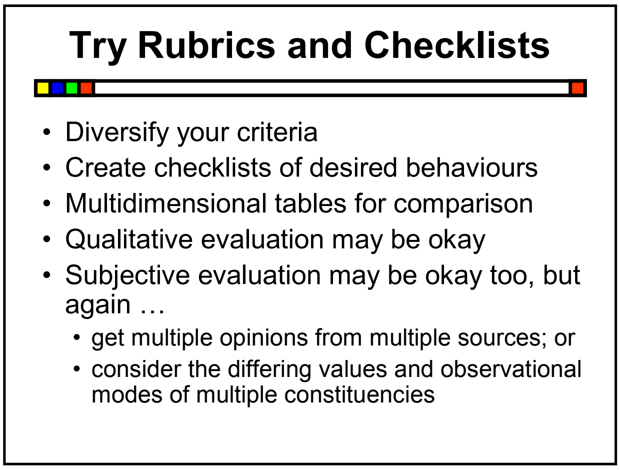


Tell The Testing Story

For a test cycle, be prepared to describe…

- the testing mission
- specific risks to address
- the diversity of your coverage, oracles, and techniques
- what you might be missing, and why it's okay

Defocusing



Try Rubrics and Checklists

- Diversify your criteria
- Create checklists of desired behaviours
- Multidimensional tables for comparison
- Qualitative evaluation may be okay
- Subjective evaluation may be okay too, but again …
  - get multiple opinions from multiple sources; or
  - consider the differing values and observational modes of multiple constituencies

Other Modes of Assessment

- Try temperature readings
  - appreciations
  - new information
  - puzzles
  - complaints
- Recognize the ways in which data can be converted to information, and vice versa
- When pushed to provide numbers, provide several alternative interpretations

Try Other Modes of Assessment

- Try private chats, standup meetings, or scrums
  - short meetings that identify needs for further meetings between smaller groups, to lessen wasted time
- Try laddering exercises
  - ranking, rather than measuring
  - if they're influenced by feelings, that may be OK
  - emotions are signals; look into them
- Try *talking to people*
  - try asking for descriptions instead of numbers
  - try retrospectives

Reporting

- "Never give a number to a bureaucrat"
  - Plum's Second Law
- Emphasize stories and narratives
- Don't produce, offer or accept a number without a story
  - lead with the story
  - show the multivariate nature of data
  - annotate charts or tables
  - note several possible interpretations
- Prefer direct observations and simple comparisons over derivative metrics

Estimation

- Change assumptions
  - Testing is a responsive activity
  - Think of prospecting: we don't know when we're going to find the gold
  - Most people, when estimating the test cycle, are really estimating the fix cycle
  - It could continue forever
  - It can be stopped at any time
- Ask "When do you want to ship?"
  - that's how much time you have, *and need*, to test

http://www.developsense.com/blog/2010/10/...
...project-estimation-and-black-swans-part-5-test-estimation

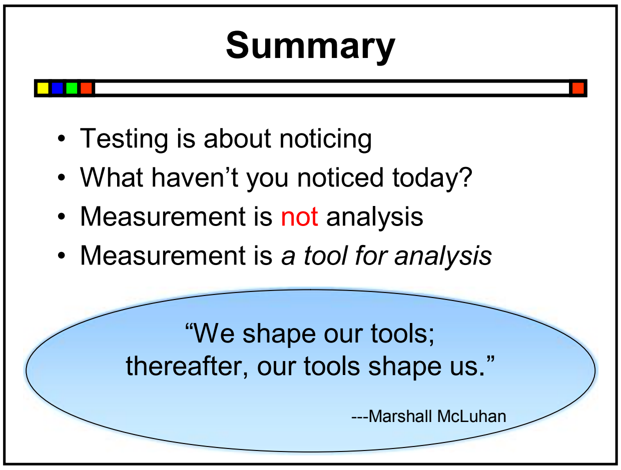
Let's Get Serious

- If our numbers are
  - multivariate
  - based on incomplete models
  - rough, first-order approximations

Let's not overwhelm ourselves wasting time and money seeking bogus precision.

Three Heuristics

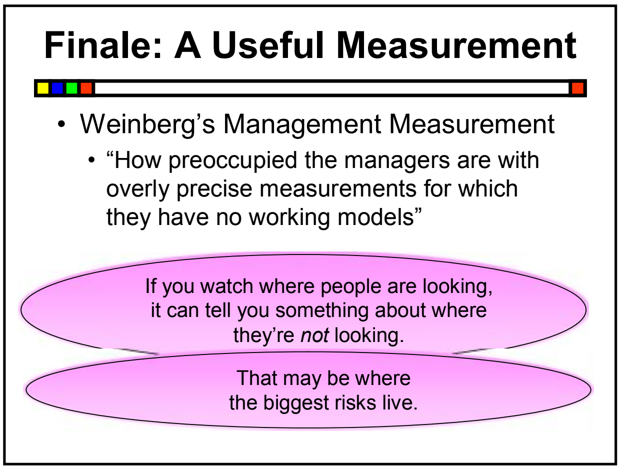
The numbers are not the story.
Seek information, not just data.
Always ask "Compared to what?"



Summary

- Testing is about noticing
- What haven't you noticed today?
- Measurement is not analysis
- Measurement is *a tool for analysis*

> "We shape our tools;
> thereafter, our tools shape us."
>
> ---Marshall McLuhan

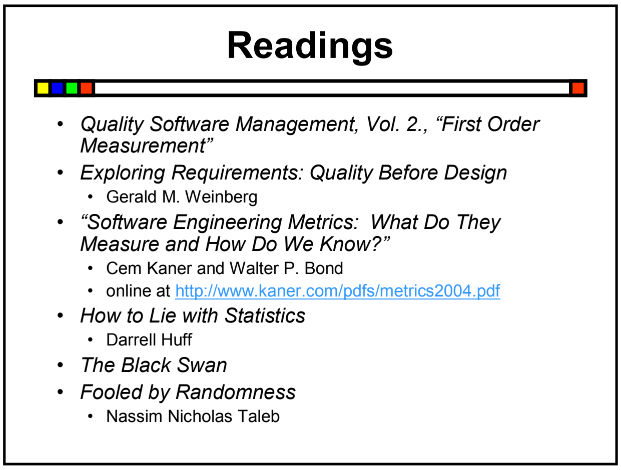


Finale: A Useful Measurement

- Weinberg's Management Measurement
  - "How preoccupied the managers are with overly precise measurements for which they have no working models"

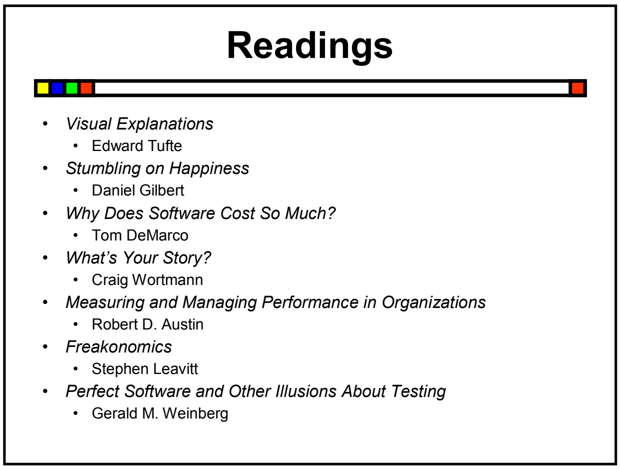
If you watch where people are looking,
it can tell you something about where
they're *not* looking.

That may be where
the biggest risks live.



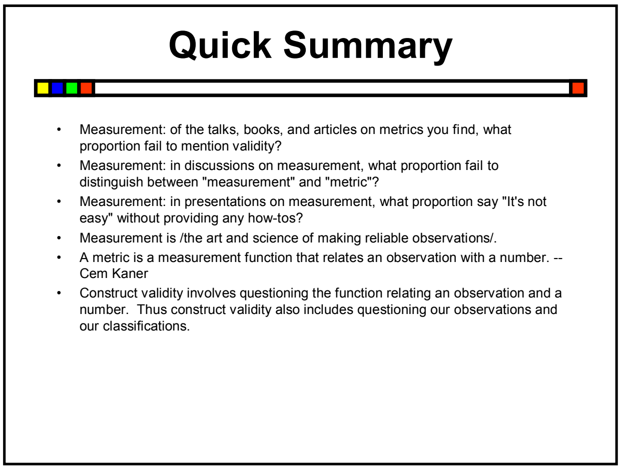
Readings

- *Quality Software Management, Vol. 2., "First Order Measurement"*
- *Exploring Requirements: Quality Before Design*
  - Gerald M. Weinberg
- *"Software Engineering Metrics: What Do They Measure and How Do We Know?"*
  - Cem Kaner and Walter P. Bond
  - online at http://www.kaner.com/pdfs/metrics2004.pdf
- *How to Lie with Statistics*
  - Darrell Huff
- *The Black Swan*
- *Fooled by Randomness*
  - Nassim Nicholas Taleb



Readings

- *Visual Explanations*
  - Edward Tufte
- *Stumbling on Happiness*
  - Daniel Gilbert
- *Why Does Software Cost So Much?*
  - Tom DeMarco
- *What's Your Story?*
  - Craig Wortmann
- *Measuring and Managing Performance in Organizations*
  - Robert D. Austin
- *Freakonomics*
  - Stephen Leavitt
- *Perfect Software and Other Illusions About Testing*
  - Gerald M. Weinberg



Quick Summary

- Measurement: of the talks, books, and articles on metrics you find, what proportion fail to mention validity?
- Measurement: in discussions on measurement, what proportion fail to distinguish between "measurement" and "metric"?
- Measurement: in presentations on measurement, what proportion say "It's not easy" without providing any how-tos?
- Measurement is /the art and science of making reliable observations/.
- A metric is a measurement function that relates an observation with a number. --Cem Kaner
- Construct validity involves questioning the function relating an observation and a number. Thus construct validity also includes questioning our observations and our classifications.