Burning Issues

Michael Bolton
DevelopSense
May 2010

Memo to those who want us all to speak a common language:
according to the Oxford English Dictionary, the word "standard"
has eight different definitions.

You want all testers to speak a common language?
How about Icelandic?
It's easy: it's pronounced "Eyjafjallajökull"

One kind of test automation is software development, often done by less capable programmers, or non-programmers.

Wasn't the problem that we already had lots of software that we weren't sure about?

Bad metrics are *not* "better than nothing". Friendly fire is not better than not shooting.

Test process improvement misses the point.
The point is productivity improvement or increased value, isn't it?

When a manager asks you to show him your test cases, ask him to show you his management cases.

When a manager asks you to show him your test scripts, ask him to show you his management scripts.

When a manager insists
that every test should have
an expected, predicted result
ask him if
every management action should have
an expected, predicted result.

When a manager insists that we
lower the cost of testing by bringing
in test automation, ask if we can
lower the cost of management by
bringing in management automation.

When a manager wants to evaluate
testers based on "defect escape
ratios", ask if we can evaluate
management by "bad management
decision escape ratios".

When a project manager asks "When
are you going to be done testing on
this project?", ask him "When are you
going to be done managing on this
project?"

I knew a guy who had a documented
process for everything. He starved
to death when he lost his process
document for making breakfast.

Plus he could never find his way
home without a step-by-step
procedure for it.

The Agilistas did not discover pairing,
or test-first programming.
They're like teenagers who've
just discovered sex.
It IS great, but calm down.

Process people:
please find something else to talk about
even if it's only for a few minutes.

At very least, talk about this:
Not all processes are linear.
Most processes that involve humans
are organic.

Jerry Weinberg: Decisions about quality are always political and emotional. We get all squeamish about that because we want to appear rational.

Humanity is okay. Can we please stop being embarrassed about it?

Testers should not be in the business of confirming, verifying, or validating. We're far more in the business of demolishing unwarranted assumptions and beliefs.

The code is not the product. The code is part of the product. The product is a problem solved for a customer.

Measurement is not necessarily quantitative.

Quantifying something removes information about it. What if the removed information is really important?

People are eager to please. They will change their behaviour to make the numbers look good.

Managers who do not observe their employees or the work being done should not call themselves managers. I suggest "pointy-haired."

Managers whose only role is to count things, judging the counts good or bad without asking how they can help should not call themselves managers. I suggest "clerks."

As a manager,
you don't have to be great at testing.
But you do have to be
good enough at it that
you can tell the difference between
good testing and bad testing.

It takes longer to perform a test and
investigate and report a bug than it
takes to perform a test.
Managers: have you noticed? What
are you going to do about that?

The test script is not the test.
The test is
what you think and *what you do*.

An important role of the tester is
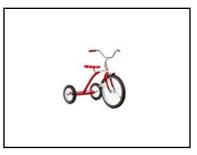to speak truth to power.

Many, even most of us agree that
current test certifications are bogus.
Our sense of ethics should require us
to speak out against bogus
certifications.

Certification is not about helping
people become qualified. It's about
disqualifying the majority.

If we can't
stand on the shoulders of giants,
maybe at least
we can stand on the toes of midgets.

Weekend Testers is about testers managing their personal development and building their skill. It's the most exciting thing to happen in testing, lately, maybe *ever*.

Metrics people: please stop misquoting Lord Kelvin. He said, "In the physical sciences." He was referring to physics, not bogus software metrics.

Metrics people: If you're going to go on about measurement, at least worry about measurement validity.

Example: how big is a vehicle?

Now: how big is a "test case"?

Passing test cases are *specific hopes with happy outcomes*. Failing test cases are *rumours of problems*.

Therefore expressed as a formula, passing vs. failing test case rates are
some number of specific hopes fulfilled
some number of rumours of problems
Is this really a valid metric?

Excellent testing isn't about a defined, predictable process. Excellent testing depends upon us putting ourselves in positions that expose us to the unpredicted.

Time, features, quality: you can pick only two *if and only if* you ignore things like motivation, innovation, organization, and skill.

Acceptance tests are misnamed. You don't know you're done when they pass; you know you're NOT done when they fail. They should be called "rejection tests".

The problem is not that testing is the bottleneck. The problem is that you don't know what's in the bottle. That's a problem that testing addresses.

There's a big difference between testing and checking.

A check has three linked parts:
1) An observation.
2) A decision rule.
3) The setting of a bit ("did the observation agree with the rule?")

A check can be applied non-sapiently, without human involvement, but…

Excellent checking is surrounded by sapient activities that require testing skill and programming skill.

Checking is very valuable when we don't fall asleep.

Even a well-checked program must still be tested if you want to know something new about it.

Is it scope creep, or have we simply discovered that we didn't understand the problem that well to begin with?

If you had understood everything perfectly to begin with, you wouldn't have needed to develop the product in the first place.

The test doesn't find the bug. The tester finds the bug, and the test has a role in finding the bug.
- Pradeep Soundararajan

Test automation doesn't find bugs. Testers (or programmers) find bugs, and the automation has a role in finding the bugs.

On the difference between preparation and planning: Your plan cannot predict when it will rain in Ireland. But if you're in Ireland, you had better be prepared for it.

I will say this about certification: The ISTQB is the finest form of personal reference that money can buy.

When you get certified by the ISTQB, you're paying money to make yourself indistinguishable from 130,000 other people.

Insanity: quoting Einstein over and over and expecting a different result.

Plus, it turns out that Einstein never said anything like that. So…

Insanity: misquoting Einstein over and over and expecting a different result.