might have

The Rapid Software Testing Guide to What You Meant to Say

Michael Bolton
http://www.developsense.com
@michaelbolton

An Apology

I MEANT for this talk to be funny.

Sorry.

Michael Bolton
DevelopSense

You said…

"We've just introduced Scrum!"

You might mean…

"We have shorter daily status meetings."

"Plus—hey, no chairs!"

…which means

Since nothing else has changed, basically, business as usual.

You said…

"Implement a mature software development process."

You probably meant…

"Implement **someone else's** software development process **model** (rather than your own)."

…which means

"Let someone else tell you how to do things."

…which means

"Remain infantile."

Michael Bolton
DevelopSense

You said…
"We have to do X."

You might have meant…
"**Someone says** we have to do X."

or…
"We **only know how** to do X."

or…
"We're **willing to tolerate** doing X."

…which means
"We **choose** to do X."

…which means
"**Maybe** we **don't** have to do X."

You said…
"This is objective."

You probably meant…
"**My *colleagues* and I agree** that this is objective."

…which means
"If **you disagree** with our assessment, you're either **not a colleague of ours** or you're **not being objective**."

…which means
"This is **subjective**."

You said…
"structured"

You probably meant…
"**formalized**"

…which means
You're probably only considering one kind of structure.

…which means
What you're doing may be more time-consuming and expensive than it's worth.

You said…
"high-level"

You probably meant…
"not really"

Examples:	**…which means**
"high-level process model"	"not really a process model"
"high-level description"	"not really a description"
"high-level evaluation"	"not really evaluation"
"high-level management"	"not really management"

(Credit to Tim Lister for this idea.)

You said…

"test"

You might have meant…

"fix"

Examples:

"This product is crap! They should have done more **fixing**."

"We have to start **fixing** earlier in the project!"

"How long will you need to **fix** the product?"

"Can't we just automate the **fixing**?"

You said…

"testing"

You might have meant…

"all of development"

Try it and see what happens!

"Why is **all of development** taking so long?"

"Why is **all of development** so expensive?"

"To prevent bugs from getting to the customers, we should work on improving **all of development**."

"Can't we just automate **all of development**."

You said…

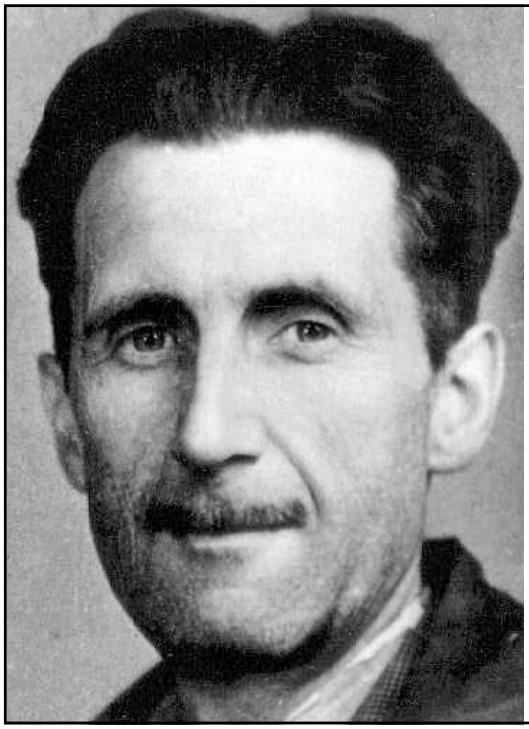
"We've just introduced DevOps"

You might have meant…

"Hey! We just realized that supporting the business and each other is our damned job."

I might have meant…

Hey, you kids, get out of my yard!



[The English language] becomes ugly and inaccurate because our thoughts are foolish, but the slovenliness of our language makes it easier for us to have foolish thoughts.

The point is that the process is reversible.

George Orwell
"Politics and the English Language"

Why don't people say what they mean?

Periodically, people equivocate to assert hegemony over the dialectic concerning the ontology.

Okay, sorry about that.

I meant to say…

Sometimes people use words that are confusing or weird to take control over conversations about ways we look at the world.

You said…	…which means
"You guys are so fussy about words!"	"I'm talking just fine, and I reject the idea that my speech is not working for me."

You might have meant…

- "You *know* what I mean; *you're* the one who is choosing to create a problem for *me*."
- "Something I thought was simple turns out to be complicated, and I'd prefer to make the complication go away."
- "You've identified something that is prompting me to think more deeply, and therefore I feel stupid (at least for now)."
- "I want to get the value out of something, but I want to avoid the effort or the controversy associated with getting it."
- "I'm okay with the risk of being misunderstood."

You said…	…which means
"Oh, that's just semantics."	"Oh, that's just *being specific about what we're talking about*."

You might have meant…

- "I don't understand the distinction you're trying to make."
- "I'm not aware of the risk of confusion here."
- "I *don't care* about the risk of confusion here."
- "I want to promote sloppy speech by reducing the concept of 'semantics' to 'nitpicking'."

So, what's the big deal?

Why We Think This Stuff Matters

- People use language as a scaffolding for their reasoning and their ontologies
- Misuse of language fosters a culture of ineffectiveness and dysfunction
  - sloppy testing work
  - disempowered, unskilled, marginalized testers
  - interference from people who think they understand testing
  - focus on the trappings, artifacts and tools instead of skills, exploration, experimentation, and learning

Moreover…

- Humans, for good and ill, “repair” gaps in communication
- LOTS of bugs result from misinterpretations of requirements, specifications, models,…
- LOTS of process problems result from overly narrow definitions of “testing”, “quality”, “bug”,…
- LOTS of process problems result from imprecise reference to “requirements”, “plans”, “strategies”,…

You said…	**…which means**
"Automate all the testing!"	"Automate all of the evaluation and learning and exploration and experimentation and modeling and studying of the specs and observation of the product and inference-drawing and questioning and risk assessment and prioritization and coverage analysis and pattern recognition and decision making and design of the test lab and preparation of the test lab
You might have meant… "Automate all the checking!"	

and sensemaking
and test code development
and tool selection
and recruiting of helpers
and making test notes
and preparing simulations
and bug advocacy
and triage
and relationship building
and product configuration

and application of oracles
and spontaneous playful interaction with the product
and discovery of new information
and preparation of reports for management
and recording of problems
and investigation of problems and working out puzzling situations

and building the test team
and analyzing competitors
and resolving conflicting information
and benchmarking

and oh… bring me a coffee.

Call this "Checking" not Testing

operating a product to check specific facts (mostly outputs)…

means

Observe → **Evaluate** → **Report**

Interact with the product in specific ways to collect specific observations.

Apply algorithmic decision rules to those observations.

Report any failed checks.

A Check Has Three Elements

1. An *observation* linked to…
2. A *decision rule* such that…
3. both observation and decision rule can be applied algorithmically.

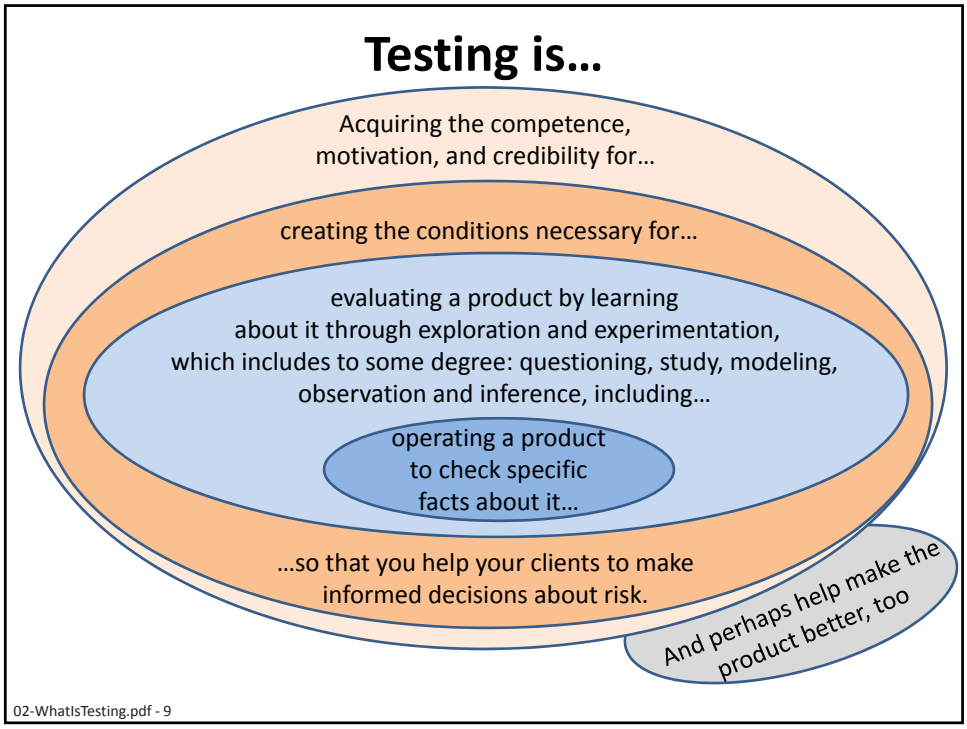
A ***check*** can be performed

by a machine
that *can't* think
(but that is quick and precise)

by a human who has been
instructed *not* to think
(and who is slow and variable)

02-WhatIsTesting.pdf - 7

Michael Bolton
DevelopSense



Why make this distinction?

- Because *checking* is mechanistic. It can be made completely **explicit** and automated.
- Because *testing* involves **tacit** skills that are developed through socialization, not via rote procedures.
- Because efficiency and effectiveness are very different conversations for explicit vs. tacit skills.
- Because for checking to be excellent, it must be embedded in excellent testing.
- Programmers have resisted marginalization for years! **(They no longer call compilers "autocoders" and programming languages are no longer called "autocodes".)**

Michael Bolton
DevelopSense

You said…

"manual testing"

You might have meant…

- interacting directly with the product, as a user might
- human checking
- testing that doesn't involve writing code
- testing without tools (but that doesn't happen)

We don't say "manual managing" or "manual science", because we don't expect tools to do those things.

Let's Talk About *Safety Language*

(aka "epistemic modalities")

- "Safety language" in software testing, means to qualify or otherwise draft statements of fact so as to avoid false confidence OR accidental deception.
- **Examples:**

I think…
It appears…
So far…
It seems…
I infer…
apparently…
I assumed…
I observed…

~~The feature worked~~ ⇨ I have not yet seen any failures in the feature…

Safety language is for listening, too.

- Try applying safety language moves to what you **hear**, as well as to what you **say**.
- People are often naïve, careless, or incompetent at expressing themselves
  - sometimes they are manipulative, too
- Just as speaking clearly is difficult, listening and interpreting is too
  - we *repair* what we hear, but not always very well
- Speaking and listening carefully may effortful
  - but it can become easier with practice and some helpful heuristics

Safety Language Vocabulary

A vs. THE

When trying to explain something,
prefer "a" to "the".

- Example: “A problem…” instead of “THE problem…”
- Using “A” instead of “THE” helps us to avoid several kinds of critical thinking errors
  - single path of causation
  - confusing correlation and causation
  - single level of explanation

Safety Language Vocabulary
Unless…

At the end of a statement,
try adding "unless..."

- When someone asks a question based on a false or incomplete premise, try adding “unless…” to the premise
- When someone offers a Grand Truth about testing, append “unless…” or “except in the case of…”

Safety Language Vocabulary
Also…

Whatever is happening,
something else may ALSO be happening.

- The product gives the correct result! Yay!
- …It also may be silently deleting system files.
- There may be *more where that come from*.

Safety Language Vocabulary
"So far" & "Not yet"

Whatever is true now
may not be true for long.

- The product works… so far.
- We haven't seen it fail… yet.
- No customer has complained… yet.
- Beware of "always" and "never"
  - There can never be a valid test for "always". Or "never".

Safety Language Vocabulary
"Or…"

Whatever your theory about "what is", other theories could describe "what is" as well, or better.

- Practice asking "What else could this be?"
- Consider opposite alternatives.
- "Mary had a little lamb."
- The Rule of Three (Jerry Weinberg): If you haven't thought of at least three plausible and non-trivial interpretations of what you've taken in, you probably haven't thought enough.
- See also *How Doctors Think*, Dr. Jerome Groopman

You said…

“It works!”

It would be more accurate to say…

SOME ASPECT of SOME FEATURE

APPEARS

to meet SOME REQUIREMENT to SOME DEGREE

based on SOME THEORY

based on SOME OBSERVATION

that SOME AGENT (I) made

under SOME CONDITIONS

ONCE or MAYBE MORE

on MY MACHINE.

You said…

“It works!”

It would be more accurate to say…

It CAN work.

I’ve SEEN IT work.

(Jeremy Clarkson CAN be a nice guy.)

Michael Bolton
DevelopSense

You said…

"I break the software."

You might have meant…

"I **find places where** the software **is broken**."

"I break **people's illusions about** the software."

It might be safer to say…

"I **explore and experiment** to discover where the software might present problems that threaten value to people who matter."

Or, more simply "I **test** the software."

It might be safer to say…

"I test the software."

Why it's important

When you say "I break the software", you set yourself up for a potential public relations problem.

Others may "repair" "I break the software."

- "The software was fine **until the testers broke it.**
- "We could ship our wonderful product on time **if only the testers would stop breaking it**."
- "Normal customers wouldn't have problems with our wonderful product; it's just that **the testers break it**."
- "There are no systemic management or development problems that have been leading to problems in the product. No way. **The testers broke it.**"

Michael Bolton
DevelopSense

You said…
"Cannot reproduce."

You probably meant…
I do not yet know how to reproduce it.

I have not yet been able to reproduce it.

Why it Matters
Some may interpret "cannot reproduce" as "**never really happened before**".

This gets repaired into "**will never really happen again**."

You said…
"Customer expectations"

You probably meant…
Customer desires

Why it Matters
If the customer likes it, **the expectation is not an issue.**

If the customer expected not to like it and *still didn't like it*, **the point is that they still don't like it.**

The expectation is a middleman.

Michael Bolton
DevelopSense

You said…
"Acceptance tests"

You probably meant…
Checks, not tests.

Rejection, not acceptance.

Rejection checks, not acceptance tests

Why it Matters
We don't want to be fooled into believing that something that "passes" **acceptance checks** is accept**able**.

You said…
"Definition of done"

You probably meant…
"Definition of **not done yet**"

Why it Matters
We can anticipate in advance **some** things that would cause us to say "we can't ship yet!"

We **cannot** anticipate all of them at the beginning of the project (sprint, cycle, whatever you call it).

BUT we can agree to be humble, and recognize that we'll learn things along the way—some of which might cause us to conclude we're **not done yet**.

Some final hints
Listen for…

- Empty labels (beware: "Agile")
- Clichés, stale images, canned phrases
  - like "canned phrase"
- Mysterious omissions
- Evaluative bias
  - "full-featured" vs. "bloated"
- Hidden controversy
- Imprecision
- Equivocation

And finally…

- Don't forget to say thank you!

Thank you!