Where Do You Want To Go Today? No More Exploratory Testing

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

Where Do You Want To Go Today? No More Exploratory Testing

Wha..?

The Motive for This Talk

We're not changing the definition of testing.

The meaning of "testing" was stolen decades ago.
It's time to steal it back.

"We're making a product!"
"We need you to start testing it right now!"

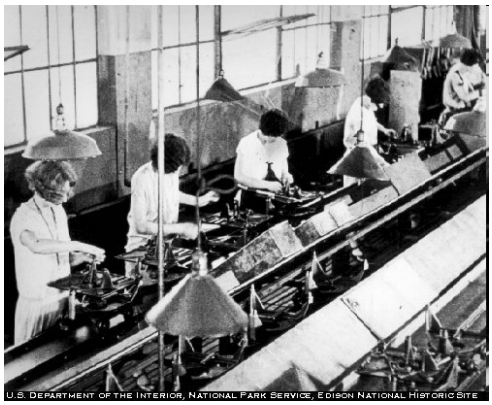
What do you do?

Testing in two easy steps!

1. Prepare test cases.
2. Execute test cases.

Maybe it's more like this…

1. Read the specification.
2. Identify specific items to be checked.
3. Prepare test cases.
4. Execute test cases.



Or maybe it's more like this…

1. Read the spec.

1.1. OMG there is no spec!

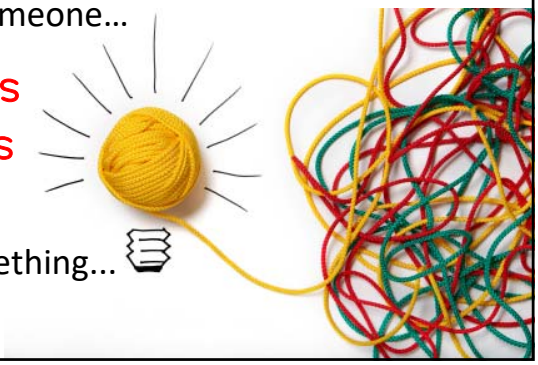
1.2 Oh wait, there is a spec! I'll just read it.

1.2.1 OMG the spec is old and confusing and maybe WRONG…

1.3 Maybe I should ask someone…

1.3.1. OMG Nobody seems to know how this thing is supposed to work!

1.3.2. Wait… is there something… *anything* I can test?



Yes! You CAN test…

- …the product
- …a mockup of the product
- …some document describing the product
- …a diagram that models the product
- …a product *like* this product
- …somebody's ideas about the product

Testing is the process of evaluating a product by learning about it through exploration and experimentation.

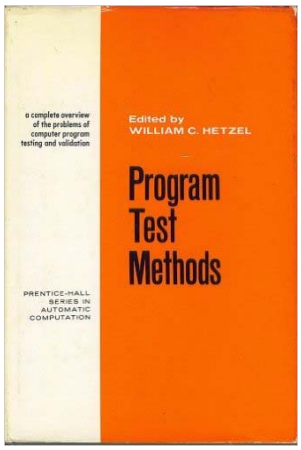
This Talk is About the Arc of Ideas

- How easy it is for design flaws to hang around in the program
- How long it takes to change the world, despite electronic media (media accelerate bad ideas as quickly as good)
- "Bad idea" is relative to some person, at some time
- Figuring out what we mean is intertwingled with figuring out what we're doing.

Disclaimer: This talk is to a great degree a personal memoir, and should not be considered authoritative in any historical sense.

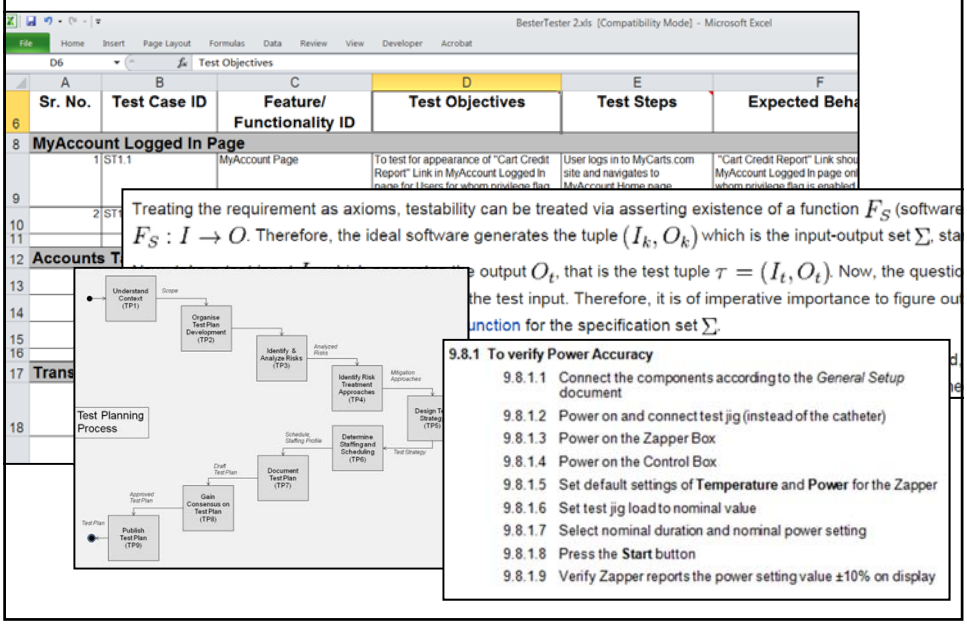
In The Beginning, There Was Testing.

Program Test Methods (1972)

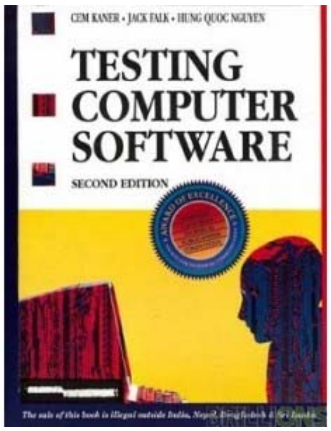


Testing became confused with the *artifacts* of testing.

Lo, here be testing!



Pre ET 1.0: Awakening (1983)



Cem Kaner coined the term “exploratory testing” (ET) in 1983 in preparation for Testing Computer Software, published in 1988… and no one really noticed.

ET 1.0: Rebellion

- It was not clear how many people were DOING very informal testing.
- It *was* clear that talk of testing focused on “the test case”, as it still does.
- The word “exploratory” appeared in *TCS*, but few seemed to notice it.
- Those who did treated ET as a distinct activity, or a technique.
- This was a mistake*, serving to marginalize ET!

* in hindsight

When Did You Get On The Bus?

I Got On the Bus Without Realizing It

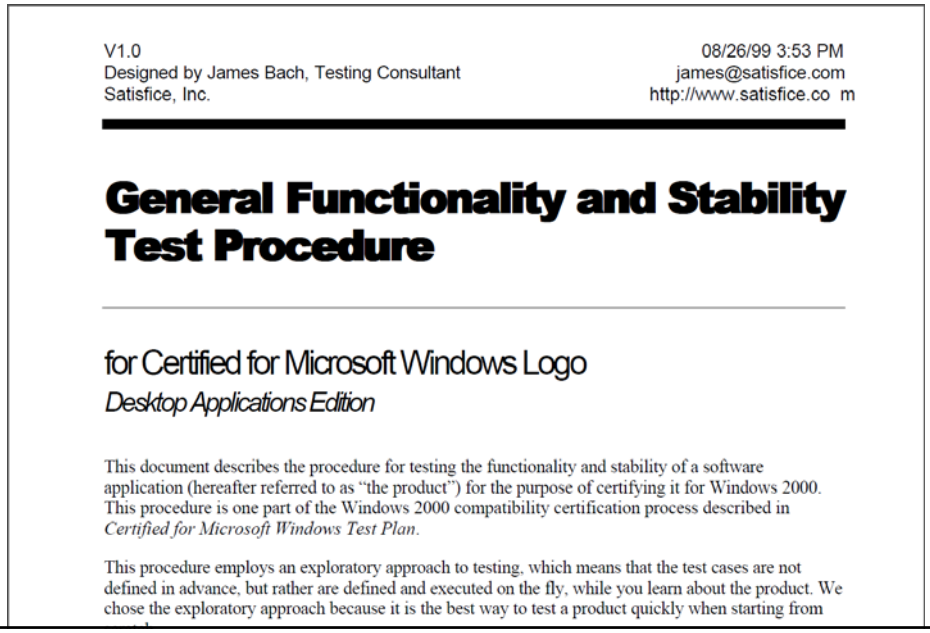
- At Quarterdeck, we practiced ET but just thought of it as "testing".
- We didn't use formal scripts, and when we did, they yielded pretty terrible results.
- Our test manager claimed that a dedicated tester was 10 times more efficient than a non-dedicated tester
  - but those were heavily scripted untrained, unsupervised testers from tech support.
- We might have done better to ask support techs to experiment, and then debrief them about problems they found.
  - We didn't notice that *we* did that inside the test group.

What IS Exploratory Testing?

- Simultaneous test design, test execution, and learning.

—James Bach, 1995

ET 1.5: Explication (1999)



V1.0
Designed by James Bach, Testing Consultant
Satisfice, Inc.

08/26/99 3:53 PM
james@satisfice.com
http://www.satisfice.co m

General Functionality and Stability Test Procedure

for Certified for Microsoft Windows Logo
Desktop Applications Edition

This document describes the procedure for testing the functionality and stability of a software application (hereafter referred to as "the product") for the purpose of certifying it for Windows 2000. This procedure is one part of the Windows 2000 compatibility certification process described in *Certified for Microsoft Windows Test Plan*.

This procedure employs an exploratory approach to testing, which means that the test cases are not defined in advance, but rather are defined and executed on the fly, while you learn about the product. We chose the exploratory approach because it is the best way to test a product quickly when starting from

ET 1.5: Explication (1999)

- Things really started to get rolling in 1999
- Appearance of the General Functionality and Stability Test Procedure (formalized ET)
- Some interesting writing from Kaner, Bach, Marick, Agruss & Johnson
- Failed experiments with undoing confusion between "ad hoc" and "exploratory" testing
  - really a distinction between skilled and unskilled

The "Ad Hoc" Business

- "Ad hoc" doesn't mean
  - sloppy
  - slapdash
  - unstructured
- "Ad hoc" does mean
  - literally "to this"
  - implicitly "to this *purpose*"
- Since many people seem reluctant to use words precisely, there's a marketing problem.

SBTM (2000): ET Sessions as Things

- For some managers, an obstacle to accepting ET is that although engineering is complex, cognitive, exploratory work, they prefer to think in terms of discrete units of production.
- Workaround: put the complex cognitive work inside a discrete unit of production (the session), as Jon and James Bach did with "High Accountability Exploratory Testing" (later, "Session-Based Test Management")
- Backfire: ET was further entrenched as a discrete activity; a *thing*.

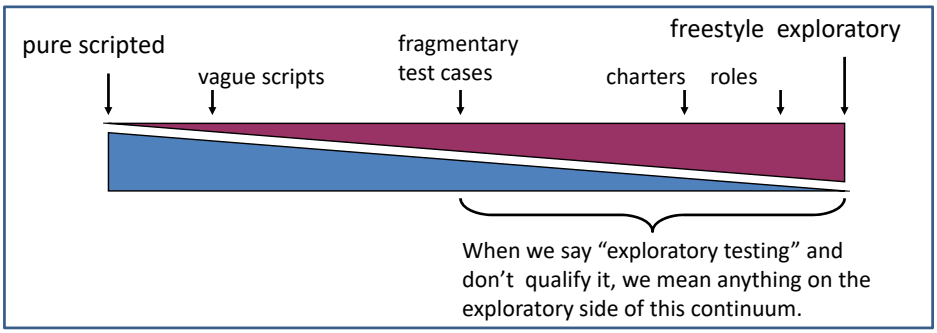
Surprise! Applying Session-Based Testing to Medical Software, Medical Device and Diagnostic Industry, May 1, 2003
http://www.mddionline.com/article/applying-session-based-testing-medical-software

HAET (2000): Testing in Sessions

```
CHARTER
-----------------------------------------------
Explore a decision created with QuickBuild -- the wizard that guides the user
through the options, criteria, and weights needed to calculate the best
decision.

#AREAS
OS | Win98
Build | 1.2
DecideRight | QuickBuild
DecideRight | Report Generator
Strategy | Exploration & Analysis

START
-----------------------------------------------
4/17/01 1:30pm

TESTER
-----------------------------------------------
Jonathan Bach

TASK BREAKDOWN
-----------------------------------------------

#DURATION
short

#TEST DESIGN AND EXECUTION
70

#BUG INVESTIGATION AND REPORTING
```

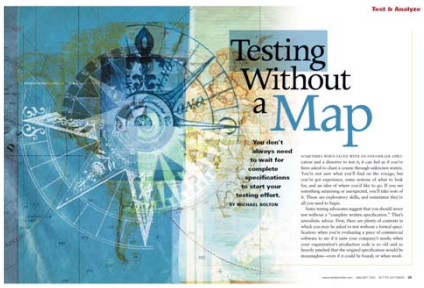
The ET Continuum (2003)

- Changed ET from “technique” to “approach” that applies to any technique
- Like a style of cooking, a style of dancing, an adjective, a mindset
- Turns out there was a huge bug in the ET Continuum that we didn’t notice *for years*.



James Bach: The Scripted/Exploratory Continuum from 2003

Testing Without a Map (2005)

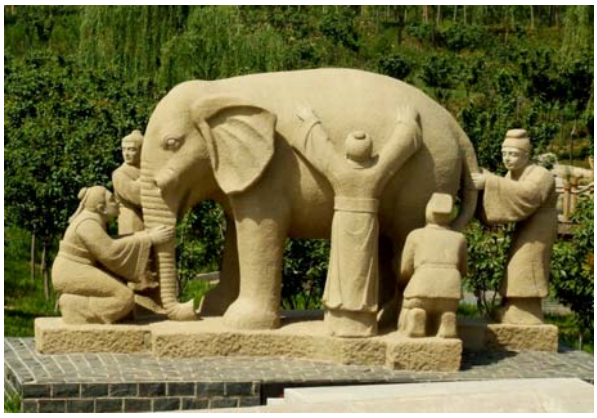


- With Brian Marick's very helpful editing and James Bach's HICCUPPS heuristics, I was able to show that ET *is* a structured process
- Proponents were starting to get attention
- Meanwhile, others were talking about "exploratory testing" and "error guessing" in the same breath, as techniques.
- Detractors usually compared "*scripted*" testing to *unskilled* testing. They left out the skilled exploration that went into preparing scripts!

Exploratory Testing Research Summit (2006)
Participants











James Bach, Jonathan Bach, Scott Barber, Michael Bolton, Elisabeth Hendrickson, Cem Kaner, Mike Kelly, Jonathan Kohl, James Lyndsay, Rob Sabourin, and an impressively large elephant in the room.

The Exploratory Testing Research Summit

- We all accepted the definition "simultaneous test design, test execution, and learning (with an emphasis on learning)".
- Everyone agreed on what the definition *was*, but not on what it *meant*. (We had reached *shallow agreement.*)
- We could have taken this a sign that ET pervaded all of testing, but we didn't quite take it that way.
- The adjective "exploratory" (contrasted with "scripted") added the dimension of agency.
- In other words: *self-directedness*.

What is Scripting?

- A script, in the general sense, is something that constrains our actions in some way.
- A *formal* script is one that must be followed in a specific way to enact a specific procedure.
- An *explicit* script is a formal script that has been expressed, written, or told.
- *The essence of scripted testing is that the tester is not in control, but rather is being controlled by some other agent or process.*
- This one simple, vital idea took us years to apprehend!

See "On Scripting", http://www.developsense.com/blog/2015/07/on-scripting/

What is Scripting?

- A script, in the general sense, is something that constrains our actions in some way.

See "On Scripting", http://www.developsense.com/blog/2015/07/on-scripting/

ET is *Self-Directed Testing*

- We now recognize that by "exploratory testing", we had been trying to refer to *rich, competent testing that is self-directed*.
- In other words, in all respects other than agency, skilled exploratory testing is not distinguishable from skilled scripted testing.
- *Only agency matters*, not documentation, nor deliberation, nor elapsed time, nor tools, nor conscious intent.
- We're still dealing with the implications of that realization, too.

ET 2.0: Freedom and Responsibility (2007)

- In the year following ExTRS, Cem Kaner produced this definition, to which many agreed:
- Exploratory software testing is
  - a style of software testing that
  - emphasizes the personal freedom and responsibility of
  - the individual tester to
  - continually optimize the value of her work by
  - treating test-related learning, test design, and execution as
  - mutually supportive activities that
  - run in parallel
  - throughout the project.

See Kaner, "Exploratory Testing After 23 Years", www.kaner.com/pdfs/ETat23.pdf

Automated and Sapient Testing (2007)

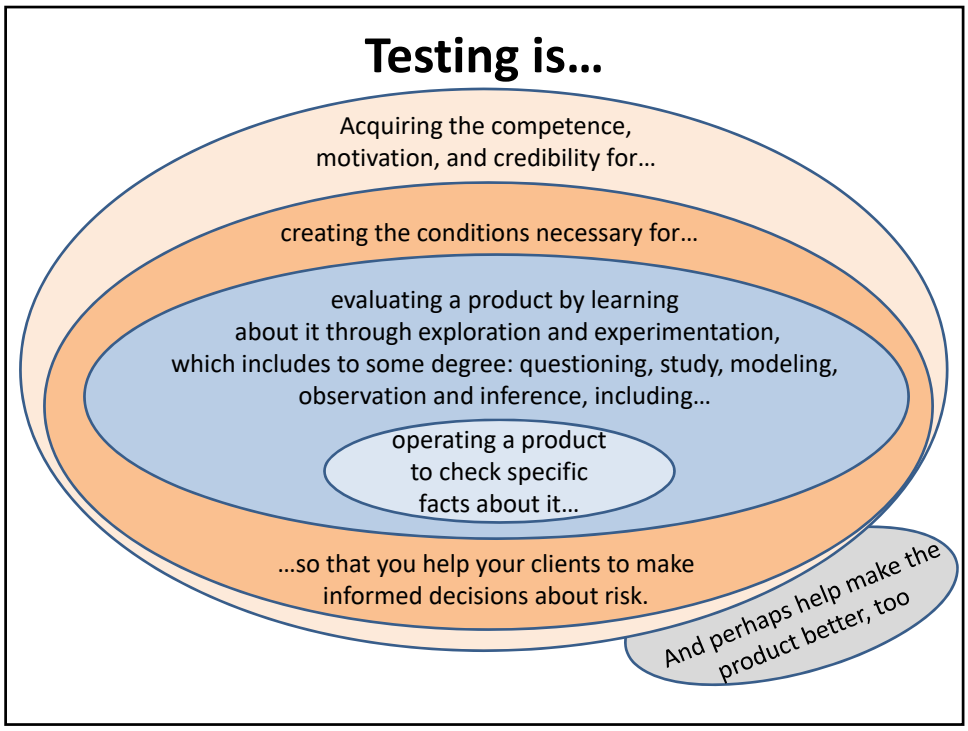
- "***A sapient process is any process that relies on skilled humans.***"
- Sapient and non-sapient didn't work because "non-sapience" *sounds like* "stupid" to some
- So it sounded like we were condemning checking by calling it non-sapient
  - we didn't have a word for checking
  - people who designed and coded checks got upset
  - machines and programs didn't take offense
    - they're non-sapient

Testing vs. Checking (2009)

- The "vs." part got a bad reaction (even, initially, from colleagues), and still does from some
  - yet no one complains about "trees vs. leaves", "biting vs. eating", "toasting vs. cooking"
  - and no one complains about "compiling vs. programming", either
- We still have to deal with the problem that people see "checking" as an insult, rather than as a *tactic* of testing.

Testing and Checking Refined (2013)

- ***Testing*** is the process of evaluating a product by learning about it through exploration and experimentation, which includes to some degree: questioning, study, modeling, observation, inference, etc.
- ***Checking*** is the process of making evaluations by applying algorithmic decision rules to specific observations of a product.



Do you notice how fine distinctions of language and thought can take *years* to work out?

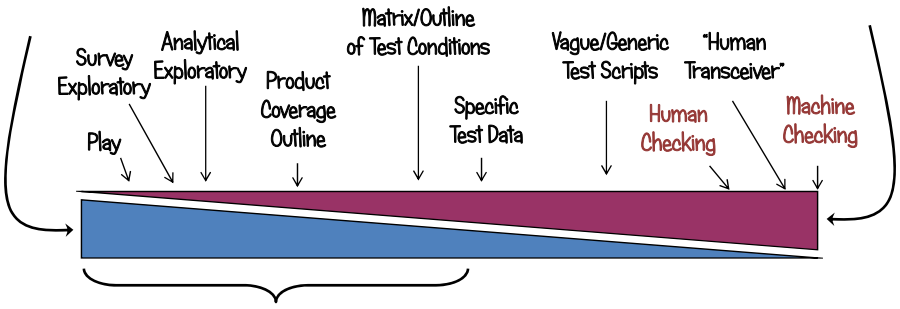
On the Road to ET 3.0

- McLuhan's ideas about media and tools
- Karl Weick, *Sensemaking in Organizations*
- Venkatesh Rao, *Tempo*
- James C. Scott, *Seeing Like a State*
- Harry Collins, *Tacit and Explicit Knowledge* and *The Shape of Actions*
- The concept of "responsible tester" (defined as a tester who takes full, personal, responsibility for the quality of his work).
- The recovery of "testing" within the Rapid Software Testing namespace

The *Formality* Continuum, 2014

INFORMAL
Not done in any specific way, nor to verify specific facts.

FORMAL
Done in a specific way, or to verify specific facts.



When we say "exploratory testing" and don't qualify it, we mean anything on the exploratory side of this continuum.

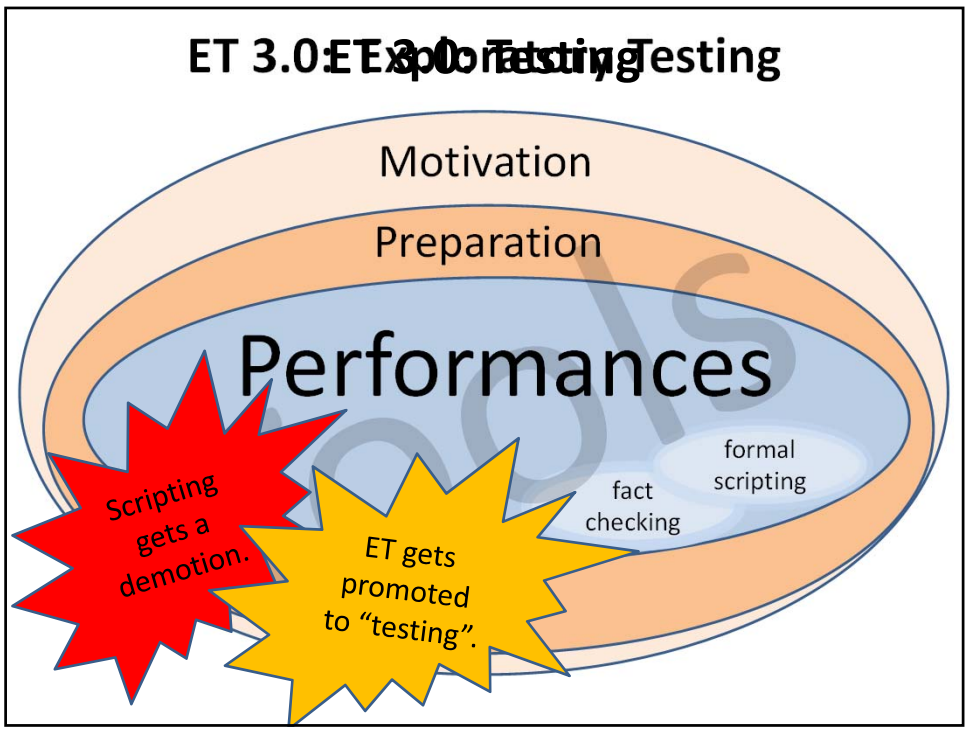
Bug fix: formality may intensify over time, thus left-to-right makes more sense.

One Big Problem in Testing

Formality Bloat

- Much of the time, your testing doesn't need to be very formal*
- Even when your testing *does* need to be formal, you'll need to do substantial amounts of informal testing in order figure out how to do *excellent* formal testing.
  - Who says? The FDA. See http://www.satisfice.com/blog/archives/602
- Even in a highly regulated environment, you do *formal* testing primarily for the auditors. You do informal testing to make sure you don't lose money, blow things up, or kill people.

* *Formal testing means "testing that must be done to verify a specific fact, or that must be done in a specific way."*

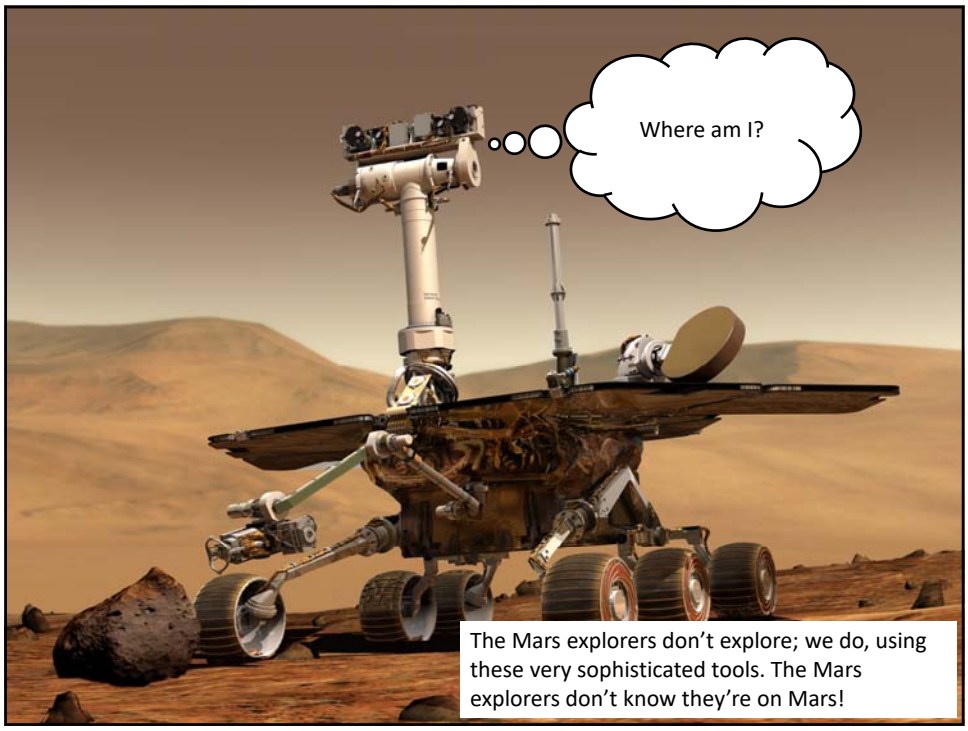


ET 3.0 (2015)

- With ET 3.0, we demote scripting to a technique—a *tactic* of testing, not its centre—and we promote exploratory testing to, simply, *testing*.
- Scripting, not exploration, is the guest in the house of testing.
- We are deprecating "exploratory testing" within the Rapid Software Testing namespace.
- We can still talk about "exploratory testing", but we also look forward to people recognizing that testing is *fundamentally* an exploratory activity: we're continuously investigating, discovering, and learning—mapping out territory anew, or in more detail.

Why not say "exploratory testing"?

Why not say "vegetarian cauliflower"?



The Mars explorers don't explore; we do, using these very sophisticated tools. The Mars explorers don't know they're on Mars!

Objections

- "Isn't it all just testing?"

  Yes, it is. And we would prefer to know what we're talking about when we analyze "testing".

- "Why bother with all this philosophy-talk? Can't we just get back to work?"

  Go ahead. **But what kind of work do you want to do?**

"A verbal cloak of ignorance is a garment that often hinders progress."
—Topley & Wilson, 1929

We're not there yet!

- In many shops, testing is still ruled by test cases
- "Automation" is seen to be
- Testers are often very poor at telling all three key parts of the testing story
- Testers (in the main) have not yet developed sophisticated use of tools, nor are tools embraced for exploratory purposes.
  - see Kaner and Hoffman, and early stuff by Marick

Two *Fundamental* Testing Questions

Is there a problem here?

Are we okay with this?

Managers only ask for test cases, pass/fail ratios, and other silly things when testers are not answering these questions, and when (therefore) managers don't trust the testers.

What do managers and developers really want from testers?

An answer to this question:

Are there problems that threaten the on-time successful completion of the project?

Our job is to investigate the product through exploration and experimentation, seeking problems and risk.

Me Talk Better, Me Think Better

Replace	with…
Verify that…	Challenge the belief that…
Validate	Investigate
Confirm that…	Find problems with…
Show that it works	Discover where it *doesn't* work
Pass vs. fail…	Is there a problem here?
Test case	Test conditions and test ideas
Counting test cases	Describing coverage
Automated testing	Programmed checking
Test automation	Using tools in powerful ways
Use cases	Use cases AND *mis*use cases AND *ab*use cases AND *obt*use cases…
KPIs and KLOCs	***Learning from every bug***

Conclusions (for now!)

- There is no manual testing.
- There is no automated testing.
- TESTERS USE TOOLS.
- Some testers specialize in building and applying tools. Call them “toolsmiths” or “technical testers”.
- Talk of “manual testing” and “automated testing” confuses everyone, creates a bogus class structure, and damages the craft.

DON’T DO THAT.

Conclusions (for now!)

- Are you doing testing? *Then you are already doing exploratory testing*.
- Are you doing scripted testing? If you’re doing it responsibly, you are doing *exploratory testing with scripting* (and perhaps with automated checking).
- If you’re *truly only* doing “scripted testing,” then you are just doing *unmotivated checking,* and we would say that you are not really testing. You are trying to behave like a machine, not like a responsible tester.

Conclusions (for now!)

Are you doing testing?

Then you are already doing exploratory testing!

What happens when your family becomes obsessed with the musical "Hamilton"?

Some people who helped me…

- Jerry Weinberg, Cem Kaner, Brian Marick, James Bach, Jonathan Bach, Scott Barber, Elisabeth Hendrickson, Jonathan Kohl, James Lyndsay, Rob Sabourin, David Gilbert, Rikard Edgren, Henrik Emilsson, Selena Delesie, Petter Mattsson, Doug Hoffman, Erik Peterson, Anne-Marie Charrett, Aaron Hodder, Oliver Erlewein, Eric Jacobson,…
- My sincere apologies to those I've forgotten