Test Framing

Michael Bolton
http://www.developsense.com
KWSQA
September 2010

Acknowledgements

- Ideas about test framing have been developed in collaboration with James Bach
- Some material in this presentation is from Rapid Software Testing, a course by James Bach and Michael Bolton.
  - http://www.satisfice.com/rst.pdf
  - http://www.satisfice.com/rst-appendices.pdf

Important Questions

Why run that test?

- Variations:
  - Why are you planning to run that test?
  - Why are you running that test *right now*?
  - Why did you run that test?

Important Questions

Why NOT run THAT test?

- Variations:
  - Why aren't you planning to run that test?
  - Why aren't you running that test *right now*?
  - Why didn't you run that test?

Important Questions

Why didn't you find that bug?

- Variations:
  - Why didn't you find that bug earlier?
  - Why did you apparently ignore that requirement?

Important Questions

Why do you think that's a bug?

- Variations:
  - Why do you say that this isn't working properly?
  - What requirement is being left unfulfilled here?
  - Why do you think that's a requirement?
  - For whom might this be a problem?
  - Do you think a user would ever do that?

Even more generally…

Why are you doing this?

- Variations:
  - Why are you not doing that?
  - How does this test relate to a requirement?
  - How does this test relate to a risk?
  - How does this test relate to your mission?

What test framing?

Test framing is
the set of logical connections
that structure and inform a test.

Framing ~= Traceability

- Framing is, in essence, traceability…
- …but typically we hear people talk of traceability in an impoverished way: between *tests* and requirements *documents*
- *Can you demonstrate traceability between tests and implicit requirements?*

Much More Traceability

1. Product traces to specifications.
2. Specifications trace to standards.
3. Test sessions trace to product versions.
4. Test sessions trace to specifications.
5. Test sessions trace to logs which trace to product, playbook and specifications.
6. Test sessions trace to charters and charters to playbook.
7. Playbook traces to standards.
8. Playbook traces to specifications.
9. Playbook traces to risks which trace to specifications…
10. Tests trace to risk…
11. Tests trace to implicit requirements…
12. Tests trace to other tests…

How Do We Know What "Is"?

We see the signs!

"If I see X, then probably Y, because *probably* A, B, C, D, etc."

- THIS CAN FAIL:
  - Getting into a car
    - Ooops, not my car.
  - Drunk driving
    - People with diabetes can seem drunk
  - Irrational decisions
    - rational from a different set of values
  - I can never find the sugar
    - I have a preset model for sugar—and it's always the other one

Vocabulary

- structure
  - that which forms the unchanging parts and relationships of a system; "that which remains"
- logic
  - A means of convincing or proving e.g. "the logic of the situation", the facts which dictate what action is rationally to be taken
- narration
  - telling a story that fits in time
- framing
  - placing the test, via logic and narrative, in logical relationship with the structures that inform it

Vocabulary

- galumphing
  - exploiting variability by doing something in a deliberately over-elaborate way
  - adding lots of unnecessary but inert actions that are inexpensive and shouldn't (in theory) affect the test outcome
  - …but sometimes they do affect it!

To test is to compose, edit, narrate, and justify two parallel stories.

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…

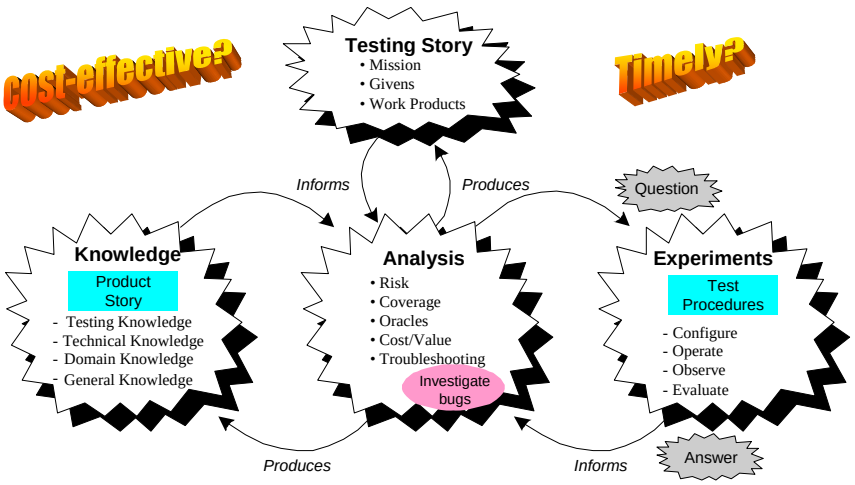
…about what you haven't tested, yet…

…or won't test, at all…

…and about why what you did was good enough.

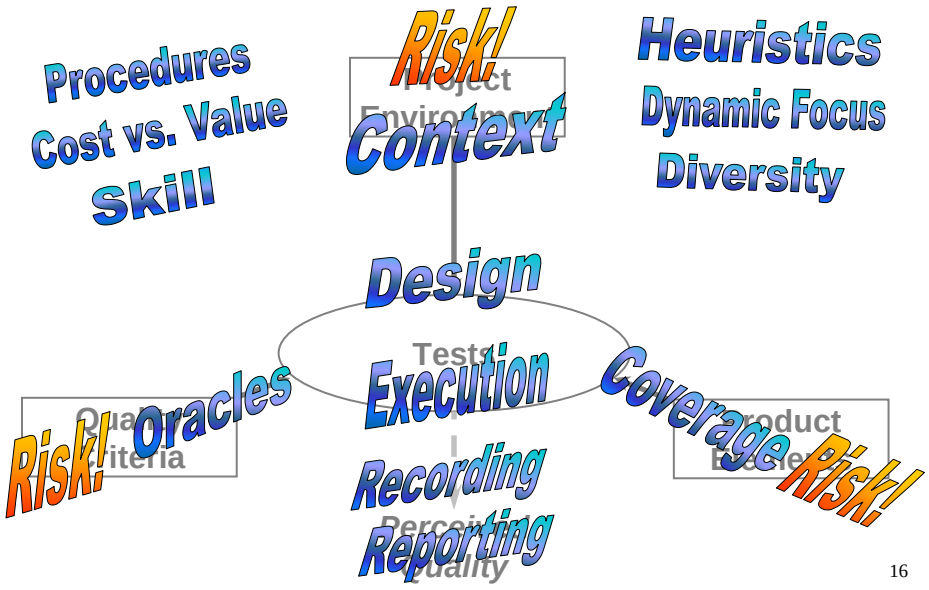
14

One View of Testing Structure



15

A Heuristic Test Strategy Model



16

What if you have
an unframed test?

Try framing it!

But if you can't do it perfectly, or right away, that might be okay. Why?

How can you justify
an unframed test?

All of the hidden frames!