







- 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?

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

- 1. 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.
- 2. But also tell a story about how you tested it...
- ...how you configured, operated and observed it ...
- ...about what you haven't tested yet...
- ...or won't test at all ...
- 3. And also tell a story that explains how good your testing was...
- ...why your testing has been good enough...
- ...why what you haven't done (so far) doesn't matter...
- ...what the risks and costs of testing are...
- ...how testable (or not) the product is...
- ...what you need and what you recommend.



Vocabulary

system

- · a set of things in meaningful relationship
- structure
 - that which forms the unchanging parts and relationships of a system; a consistent pattern for the system; "that which remains"
- narration
 - telling a story that fits in time
- framing
 - via logic and narrative, placing the test in logical relationship with the structures that inform it

Vocabulary

logic

- a formal means of convincing or proving facts via valid arguments
 ...using a set of propositions linked by connectives or operators
- proposition
- a simple statement of fact or inference
- connectives
- formal logic: if, and, or, and not, else, if and only if, therefore
 less formally: because, unless, otherwise...
- Framing provides traceability, but testers often limit traceability as being between *tests* and requirements *documents*—explicit requirements.
- 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–oops, not my car.
- · Bad driving—why?
- · Bad work—why?
- Inexpicable behaviour-why?
- · I can never find the sugar-why



























Galumphing A Defocusing Heuristic to Exploit Variability

- · doing things 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
 - · bring up a dialog and dismiss it
 - modify an option and rescind it
 - · perform an action and reverse it
 - · re-selecting default options

- · inserting an expression where a single value would do
- · over-filling an input field, then fixing it

Exploiting Variation To Find More Bugs

Micro-behaviors

- Unreliable and distractible humans make each test a little bit new each time through.
 Randomness
 - Can protect you from unconscious bias (but be careful; humans almost always act non-randomly)

Data Substitution

- The same actions may have dramatically different results when tried on a different database, or with different input.
- Timing/Concurrency Variations
- The same actions may have different results depending on the time frame in which they occur and other concurrent events.
 Platform Substitution
- · Supposedly equivalent platforms may not be

Exploiting Variation To Find More Bugs

Scenario Variation

- The same functions may operate differently when employed in a different flow or context.
- State Pollution

- Hidden variables of all kinds frequently exert influence in a complex system. By varying the order, magnitude, and types of actions, we may accelerate state pollution, and discover otherwise rare bugs.
- Sensitivities and Expectations
 - Different testers may be sensitive to different factors, or make different observations. The same tester may see different things at different times or when intentionally shifting focus to different things.