Test Process in 20 Minutes

Michael Bolton
DevelopSense
http://www.developsense.com

TESTREK Toronto Managers' Workshop
October, 2009

1

What Is Testing?

- Questioning a product in order to evaluate it
  - James Bach
- Gathering information with the intention of informing a decision
  - Jerry Weinberg
- An empirical, technical investigation of a product, done on behalf of stakeholders, with the intention of revealing quality-related information of the kind that they seek.
  - Cem Kaner

But in the wild, to our clients, much of testing is MOSTLY about…

FINDING PROBLEMS

How do we find problems?

We use *oracles.*

Hold it…
What is an oracle?

4

No, not the database

ORACLE

Not the database

An oracle is…

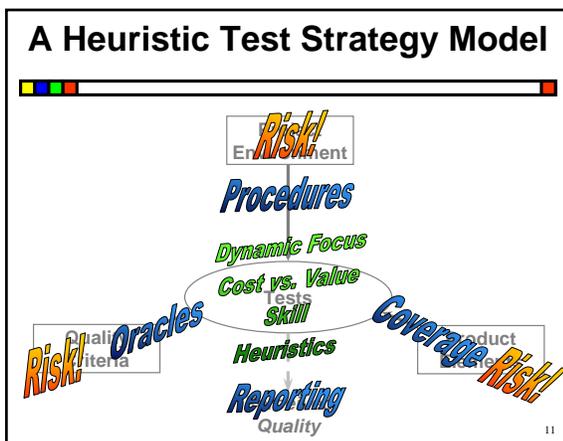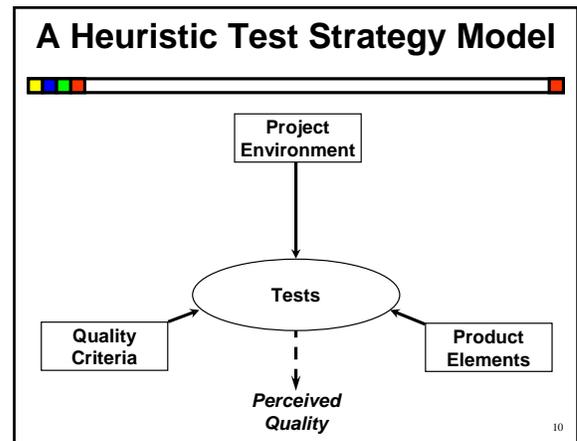a principle or mechanism
by which
we recognize a problem

1

But wait…

How can we be
SURE
that we're seeing a problem?

WE CAN'T.

Certainty isn't available.

But we DO have *heuristics*

Heuristics are fallible, "fast and frugal" methods of solving problems, making decisions, or accomplishing tasks.

*"The engineering method is the use of *heuristics* to cause the best change in a poorly understood situation within the available resources."*

Billy Vaughan Koen
Discussion of the Method

A Heuristic Test Strategy Model

Project Environment

Tests

Quality Criteria

Product Elements

Perceived Quality

10

A Heuristic Test Strategy Model

Risk!
Environment
Procedures
Dynamic Focus
Cost vs. Value
Tests
Skill
Quality Criteria
Oracles
Heuristics
Coverage
Product Elements
Risk!
Reporting
Quality

11

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…
   [12]…and about why what you did was good

The Structure of Testing
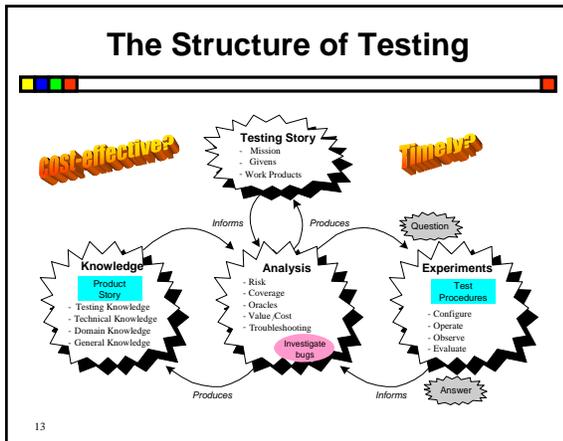
What IS Coverage?

Coverage is "how much of the product we have tested."

It's the extent to which we have traveled over *some map* of the product.

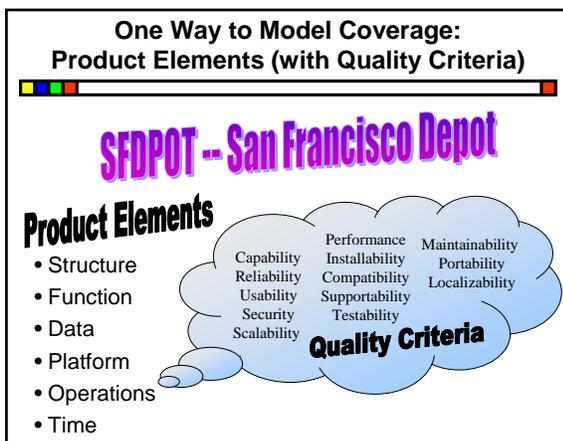...but what does it mean to "map" a product? Talking about coverage means talking about **models**

Models

- **A model is a heuristic idea, activity, or object…**
  such as an *idea in your mind*, a *diagram*, a *list of words*, a *spreadsheet*, a *person*, a *toy*, an *equation*, a *demonstration*, or a *program*

- **…that represents (literally, *re*-presents) another idea, activity, or object…**
  such as something complex that you need to work with or study

- **…whereby understanding something about the model may help you to understand or manipulate the thing that it represents.**

  - A *map* is a model that helps to navigate across a terrain.
  - *2+2=4* is a model for adding two apples to a basket that already has two apples.
  - *Atmospheric models* help predict where hurricanes will go.
  - A *fashion model* helps understand how clothing would look on actual humans.
  - *Your beliefs about what you test are a model of what you test.*

There are as many kinds of test coverage as there are ways to model the system.

Structure **Business Risk** **Time**
Functions **Platform** **Technical Risk**
Data **Operations**

And each could be...

Intentional or... **Accidental**

One Way to Model Coverage:
Product Elements (with Quality Criteria)

SFDPOT -- San Francisco Depot

Product Elements
- Structure
- Function
- Data
- Platform
- Operations
- Time

Capability
Reliability
Usability
Security
Scalability
Performance
Installability
Compatibility
Supportability
Testability
Maintainability
Portability
Localizability

Quality Criteria

Extent of Coverage

With respect to each model...

- Smoke and sanity
  - Can this thing even be tested at all?
- Common and critical
  - Can this thing do the things it *must* do?
  - Does it handle happy paths and regular input?
  - *Can* it work?
- Complex, extreme and exceptional
  - Will this thing handle challenging tests, complex data flows, and malformed input, etc.?
  - *Will* it work?

Modern Project Cycle

Project Start | Alpha | Beta | Beta | Beta | "Code Freeze"

Ongoing Requirements Definition

Release to Manufacturing

Incremental Design

Design...

Incremental Coding

Testing and Retesting

How does Rapid Testing compare with other kinds of testing?

When testing is turned into an elaborate set of rote tasks, it becomes ponderous without really being thorough.
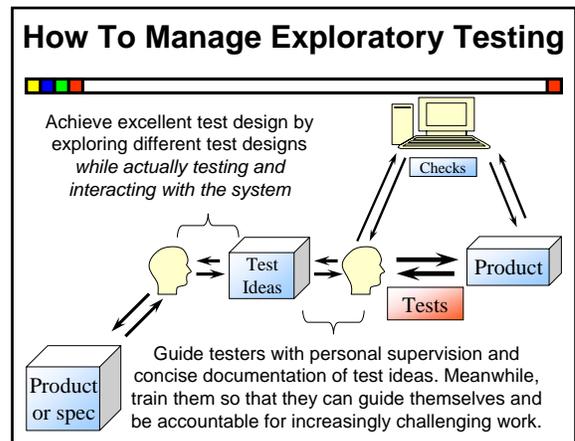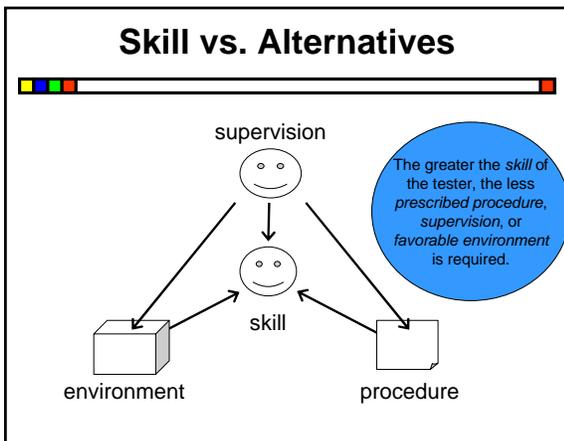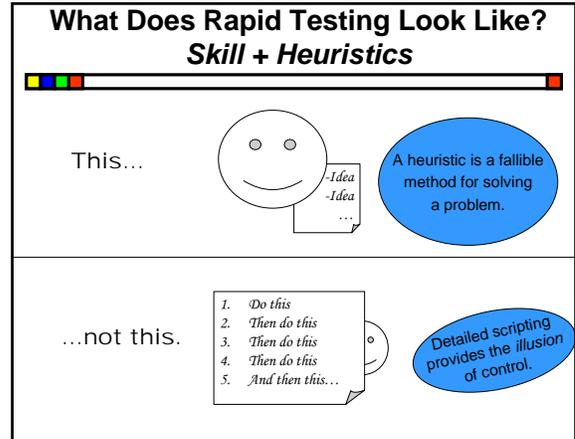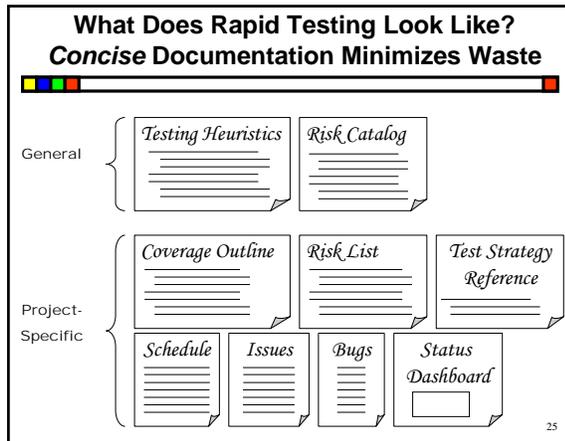
Management likes to talk about exhaustive testing, but they don't want to fund it and they don't know how to do it.

More Work & Time (Cost)

Ponderous
Slow, expensive, and easier

Exhaustive
Slow, **very** expensive, and difficult

Slapdash
Much faster, cheaper, and easier

Rapid
Faster, less expensive, still challenging

Rapid testing may not be exhaustive, but it is thorough enough and quick enough. It's less work than ponderous testing. It might be less work than slapdash testing.

It fulfills the mission of testing.

You can always test quickly... But it might be poor testing.

Better Thinking & Better Testing (Value)

The Themes of Rapid Testing

- Put the **tester's mind** at the center of testing.
- Learn to **deal with complexity** and ambiguity.
- Learn to **tell a compelling testing story.**
- Develop **testing skills** through practice, not just talk.
- **Use heuristics** to guide and structure your process.
- **Be a service** to the project community, not an obstacle.
- **Consider cost vs. value** in all your testing activity.
- **Diversify** your team and your tactics.
- Dynamically **manage the focus** of your work.
- Your **context should drive your choices**, both of which evolve over time.

What Does Rapid Testing Look Like?
Skill + Heuristics

This...

-Idea
-Idea
...

A heuristic is a fallible method for solving a problem.

...not this.

1. Do this
2. Then do this
3. Then do this
4. Then do this
5. And then this...

Detailed scripting provides the *illusion* of control.

Skill vs. Alternatives

supervision

skill

environment

procedure

The greater the *skill* of the tester, the less *prescribed procedure, supervision,* or *favorable environment* is required.

How To Manage Exploratory Testing

Achieve excellent test design by exploring different test designs *while actually testing and interacting with the system*

Checks

Test Ideas

Product

Tests

Product or spec

Guide testers with personal supervision and concise documentation of test ideas. Meanwhile, train them so that they can guide themselves and be accountable for increasingly challenging work.

What Does Rapid Testing Look Like?
Concise Documentation Minimizes Waste

General

Testing Heuristics   Risk Catalog

Project-Specific

Coverage Outline   Risk List   Test Strategy Reference

Schedule   Issues   Bugs   Status Dashboard

25

A Quick Note on "Maturity"

- In process-speak the notion of maturity is *backwards*
- A mature entity, in biology, is one that
  - can survive and thrive without parental support
  - has achieved a stage where it can reproduce (and mutate, and diversify)
- A genuinely mature process incorporates
  - freedom
  - responsibility
  - diversity
  - adaptability
  - self-sufficiency
  - repeatability, but only as appropriate to context

Acknowledgements

- Much of the material in this presentation is from the class "Rapid Software Testing", by James Bach and Michael Bolton