Testing and Noticing

Michael Bolton
DevelopSense
michael@developsense.com

James Bach
Satisfice
james@satisfice.com

STAR East
May 2009

Acknowledgements

• Mark Federman → Terrence Gordon → Marshall McLuhan for the original title (“What Haven’t You Noticed Lately”)
• Cem Kaner
• Jerry Weinberg
• Adam White
• Pradeep Soundararajan
• …and the authors of the works cited in the references.
Updates

• This presentation is ALWAYS under construction
• Updated slides at http://www.developsense.com/past.html

A Gospel of Liberation
The Gospel of No Gospels

• No appeals to
  • best practices
  • heavyweight processes
  • management support
• Excellent testing starts with us
  • this depends upon the personal freedom, responsibility, and skill of the tester
  • we don’t need anyone’s permission to sharpen our skills
  • let’s invest ourselves in our own education
The Question We All Dread

Why didn't you find that bug?!

One Possible Answer

Uh... we didn't notice it.
Yet have you ever noticed why?

• “We didn’t notice it.”

• “We did notice it, but we didn’t know that it was a problem.”

• “We did find it, and we knew that it was a problem, but we didn’t think it was important enough to worry about.”

• “We were busy looking for other bugs.”

• “We were busy reporting other bugs that we found.”

Let’s look at…

NOT noticing.
Notice Anything About The Flowers?

What About When We Change the Question?

• Notice anything when you look *between* the flowers?
• What happens when we treat the flowers as *ground* and treat the background as *figure*?
What is Noticing?

- Recognizing events, objects, or properties
- Starts with direct sensory intake, then moves to meaning, significance, and response
- An internal reaction
- Can be managed by dynamically managing observation and focus
- Often triggered by emotional reactions
- Can be trained and improved

What Might We Notice?

- Problems or other kinds of information about products
- Things about ourselves
- Things about others
- Things about our environment
Feedback Loop: Perception & Conception

What we believe

What we've been trained to do

What we can see

I wouldn't have seen it if I hadn't believed it.

Conceptual Priming Ideas

- The skill of factoring
- Your technical knowledge and beliefs
  - beliefs about what is (im)possible
  - beliefs about what is (un)likely
- Guideword heuristics
- Patterns of familiar bugs
Snap Judgments, Rapid Cognition

- Snap judgments are central to our decision-making process
- Snap judgments are vulnerable to corruption by forces that we’re not aware of and that we don’t understand
- Sometimes we can improve the quality of our snap judgments by *removing* information, rather than adding it
- Instead of changing the decision-maker, change the context in which the decision is made.

Have you ever noticed…?

*Diversity supports better noticing.*
The Law of Requisite Variety

• A controlling system needs to have more states than the system it controls
• “Complicate yourself if you want to understand something complicated.”
  - G.F. Smith
• “A tester is someone who knows things can be different.”
  - Jerry Weinberg

Practice Factoring!

• “List all the dimensions of (some common object) that may be relevant to testing it.”
• “dimensions” means attributes of the object that may vary from one to another or within the same object over time.
• “relevant to testing” means that there is probably some value to some client, with respect to some testing mission, of manipulating or observing a particular dimension.
• Try it with others; compare notes; notice categories of dimensions (and categories of categories)
There Are Factors To Observation, Too!

- the thing being observed (the *system*)
- the environment (all the things around us)
- our knowledge and models (conceptual)
- our senses (perceptual)
- our experience (experiential)
- our feelings (affective)
- our mission and our client

...and there are factors to each one of these factors!

---

Consistency ("this agrees with that")

*an important theme in oracles*

- History
- Image
- Comparable Products
- Claims
- User Expectations
- Purpose
- Product
- Standards

Consistency heuristics rely on the quality of your models of the product and its context.
History

If a product is inconsistent with previous versions of itself, we suspect that there might be a problem.

Okay, so how the #@ do I print now?

Image

If a product is inconsistent with an image that the company wants to project, we suspect a problem.
Comparable Products

WordPad

When a product seems inconsistent with a comparable product, we suspect that there might be a problem.

Word

Claims

New! Supports Mac OS/X!

When a product is inconsistent with claims that important people make about it, we suspect a problem.
User Expectations

When a product is inconsistent with expectations that a reasonable user might have, we suspect a problem.

Purpose

When a product is inconsistent with its designers’ explicit or implicit purposes, we suspect a problem.
When a product is inconsistent internally—as when it contradicts itself—we suspect a problem.

When a product is inconsistent with laws or widely accepted standards, we suspect a problem.
But How Do I Keep Track? HICCUPPS!

- History
- Image
- Comparable Products
- Claims
- User Expectations
- Purpose
- Product
- Statutes

...plus for “Familiar Problems”, add that inconsistent F!

Thirty-Six Things You COULD Notice

"cidtestdsfdpotcrusspicstmplfdsfscura"

Customers  Information  Developer relations
Team  Equipment & tools  Schedule  Test Items  Deliverables

Product Elements
- Structures
- Functions
- Data
- Platforms
- Operations
- Time

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

Test Techniques
- Function testing
- Domain testing
- Stress testing
- Flow testing
- Scenario testing
- Claims testing
- User testing
- Risk testing
- Automatic testing
Sharpening Perception

- Priming our senses
- Noticing our emotional triggers
- Training the intuition
- Watching others test or use a product
- Practice!

Have you ever noticed…?

It's sometimes hard to know how to notice.
Listening for Biases (Just a Sampler)

- **Evaluative Bias of Language**
  - “*Our* product is full-featured. *Theirs* is bloated.”

- **Representativeness Bias**
  - “It’s just a one-line change. Why bother testing?”

- **Automation Bias (machines over people)**
  - “The green bar tells us we’re done.”

- **Reification Error (counting the uncountable)**
  - “We’ve got 10,487 test cases for our 492 requirements.”

Assimilation Bias: That’s (not) Testing

- Testing is “questioning a product in order to evaluate it” or “gathering information with the purpose of informing decisions”

- By these definitions, what is testing and what is not?
Assimilation Bias (a.k.a. Lumping)

How can we estimate "the testing phase" when we're mostly waiting for other people?

Assimilation Bias (a.k.a. Lumping)

How can we estimate "the testing phase" when it doesn't really exist anyway?
Have you ever noticed…?

Noticing can be triggered by emotions and feelings.

How Do People React to Software?

- Impatience
- Frustration
- Amusement
- Surprise
- Confusion
- Annoyance
Feelings Provide Clues

- An emotional reaction is a trigger to learning
- Without emotion, we don’t reason well
  - See Damasio, *The Feeling of What Happens*
- When you find yourself mildly concerned about something, someone else could be very concerned about it
- Observe emotions to help overcome your biases and to evaluate significance

An emotion is a signal; consider looking into it

Emotional Triggers

**What might feelings be telling us?**

- Impatience ⇒ an intolerable delay?
- Frustration ⇒ a poorly-conceived workflow?
- Amusement ⇒ a threat to someone’s image?
- Surprise ⇒ inconsistency with expectations?
- Confusion ⇒ unclear interface? poor testability?
- Annoyance ⇒ a missing feature?
- Boredom ⇒ an uninteresting test?
- Tiredness ⇒ time for a break?
Affective Priming
Preparing Your Emotional Mindset

• building confidence
• developing tolerance for mistakes
• tolerance for confusion
• stress inoculation
• embracing and celebrating the new
• avoiding learned helplessness
• recognizing and dealing with environments that might be unsupportive or hostile

Have you ever noticed…?

Noticing can be influenced by experience.
Heuristics in Medicine: A Case Study

- There are at least three ways to direct people to treatment

Clinical Intuition
Complex Statistics
Fast and frugal rules of thumb (heuristics)

What To Do?

- A man is rushed to hospital with severe chest pain
- Decision:
  - coronary care unit?
  - regular nursing bed (with a heart monitor)?
- Problem:
  - based on long term risk factors (family history, male, advanced age, smoking, diabetes, high cholesterol, hypertension) doctors sent about 90% of patients to coronary care.
  - care unit became crowded, quality of care decreased, cost went up

A story from Gut Feelings, by Gerd Gigerenzer.
Research

• Doctors sent most patients to the CCU
• Sent patients who should have been there just as often as those who shouldn’t have
• The decision was no better than chance.

Heart Disease Predictive Instrument

This table, with the aid of a long formula and a pocket calculator, helped doctors to make better CCU assignments.
Method

- The doctors were told to
  - find the right probabilities for each patient
  - type these into a calculator with a long formula
  - press ENTER
  - read off the result
  - compare it to a threshold number
  - route the patient to the CCU or a regular bed

Accuracy went up.

But...

- Even though accuracy was up…
- Even though overcrowding eased…

The doctors hated it.
They didn't understand it.
Testing the Conclusions

- The researchers tested the efficacy of the method and the calculations.
- They took the tables and calculators away.

Accuracy remained high.

- After the doctors had been exposed to the chart, their intuitions improved permanently.

Conclusions on the HDPI Approach

- When systems with heavy calculations and many probabilities conflict with intuition, people tend to resist the complex solution.
- When there is high uncertainty, simple diagnostic methods tend to be more accurate.
- Practice with the complex solution appeared to train doctors’ intuitions subconsciously.
- This led to the recognition and development of a heuristic…
The Fast and Frugal Tree

EKG ST Segment Changes?
- Yes
- No

Chest pain the chief complaint?
- Yes
- No

Any one of NTG, MI, ST $\leftrightarrow$, ST $\uparrow$, T?
- Yes
- No

Coronary Care Unit

Regular Nursing Bed

Coronary Care Unit

Regular Nursing Bed

Heuristics Win!

Patient CCU Assignments

Ideal (all hits, no false positives)

Individual doctors (mostly right, lots of false positives)

Fast and frugal tree (top result for safety AND reduced false positives)

HDPI (complex calculation, all over the place)
Practice!

- Testing as interaction with the product
- Playing with the software
- Being a real user
- Experiencing patterns of (eventually) familiar problems
  - on the job (scary, career-threatening)
  - experiential training (scary, less risky)
- Learning seemingly unrelated to testing
  - new skills, hobbies, games, ...

Environmental Priming

Preparing By Situating Yourself

- Minimizing disruptive distractions
- Maximizing productive distractions
  - taking a break? taking a shower?
- Creating a safe environment for noticing
  - notice the environment, and notice when it's hostile to observation
- Introduce equipment and tools to assist observation
Have you ever noticed…

Some products are hard to test?

Object Priming

*Preparing The Thing To Be Observed*

- log files and debug output
- scriptable interfaces
- built-in error detection and correction
  - so we don’t have to notice
- better error and status messages
- consistent user interfaces and workflows

Testability ~ Usability
One More Big Thing

• Have you noticed how much time you spend on investigating and reporting bugs?

A well-tested program gives us more time to notice!

How Do We Spend Time?
(assuming all tests below are good tests)

<table>
<thead>
<tr>
<th>Module</th>
<th>Bug reporting/investigation (time spent on tests that find bugs)</th>
<th>Test design and execution (time spent on tests that find no bugs)</th>
<th>Number of tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>A (good)</td>
<td>0 minutes (no bugs found)</td>
<td>90 minutes (45 tests)</td>
<td>45</td>
</tr>
<tr>
<td>B (okay)</td>
<td>10 minutes (1 bug, 1 test)</td>
<td>80 minutes (40 tests)</td>
<td>41</td>
</tr>
<tr>
<td>C (bad)</td>
<td>80 minutes (8 bugs, 8 tests)</td>
<td>10 minutes (5 tests)</td>
<td>13</td>
</tr>
</tbody>
</table>

Investigating and reporting bugs means….

SLOWER TESTING or...

REDUCED COVERAGE …or both.

• In the first instance, our coverage is great—but if we’re being assessed on the number of bugs we’re finding, we look bad.
• In the second instance, coverage looks good, and we found a bug, too.
• In the third instance, we look good because we’re finding and reporting lots of bugs—but our coverage is suffering severely. A system that rewards us or increases confidence based on the number of bugs we find might mislead us into believing that our product is well tested.
What Happens The Next Day?
(assume 6 minutes per bug fix verification)

<table>
<thead>
<tr>
<th>Fix verifications</th>
<th>Bug reporting and investigation today</th>
<th>Test design and execution today</th>
<th>New tests today</th>
<th>Total over two days</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 min</td>
<td>0</td>
<td>45</td>
<td>45</td>
<td>90</td>
</tr>
<tr>
<td>6 min</td>
<td>10 min (1 new bug)</td>
<td>74 min (37 tests)</td>
<td>38</td>
<td>79</td>
</tr>
<tr>
<td>48 min</td>
<td>40 min (4 new bugs)</td>
<td>2 min (1 test)</td>
<td>5</td>
<td>18</td>
</tr>
</tbody>
</table>

Finding bugs today means….
VERIFYING FIXES LATER
...which means….

EVEN SLOWER TESTING or...
EVEN LESS COVERAGE ...or both.

...and note the optimistic assumption that all of our fixed verifications worked, and that we found no new bugs while running them. Has this ever happened for you?

Alternating Strategies

- Variation vs. Repetition
- Pausing vs. Rushing
- Reversing
  - Seeking noise in the signal
  - Seeking signal in the noise
- Simplifying and Complicating
- Focusing and Defocusing
- Visualizing vs. Verbalizing
Have you ever noticed…?

We notice different things at different times.

Repetition

- What does it mean to repeat a test?
- *Exact* repetition means
  - looking at exactly the same things
  - performing exactly the same actions
  - in exactly the same order
  - using exactly the same data
  - making exactly the same observations
- Humans aren’t so good at exact repetition
  - but they can be good at noticing new things
- Machines are excellent at exact repetition
  - but they’re lousy at noticing
When Things Are Changing Rapidly…

…exact repetition might be important to detect the changes.

Use machines to aid exact repetition!

When Things Aren’t Changing Rapidly…

…variation might be important so that we notice things that we hadn’t noticed before.

Use human interaction to foster variation!
Managing Attention

• To test for *anticipated* problems…
• To test a *simple* product, or *part* of a complex product very thoroughly…
• To *pinpoint* an observed problem…
• To *confirm* that a fix has been made…
• To maximize test integrity…
• To stay in grooves…

Focusing Heuristics

1. Start the test from a *known* (clean) state.
2. Prefer *simple, deterministic* actions.
3. Vary One Factor At a Time (OFAT)
4. Trace test steps to a *specified model*.
5. Follow *established and consistent* lab procedures.
6. Make *specific* predictions, observations and records.
7. Make it *easy to reproduce* (automated input may help).
Managing Attention

• To find *unexpected* problems…
• To find *elusive problems* in sustained field use…
• To *test* whether a fix has broken something else…
• To discover new dimensions of the product or the testing mission…
• To get out of ruts…

**De-Focus!**

Blink Tests

• A blink test is a test that leverages human pattern recognition
• Example: ProSum
• Example: Change Blindness Test
Defocusing Heuristics

1. Start from a variety of different states (not necessarily clean).
2. Prefer complex, challenging actions.
3. Vary Many Factors At a Time (MFAT).
4. Generate tests from a variety of models, or without reference to a conscious model.
5. *Question* your lab procedures and tools.
6. Try to see everything with open expectations.
7. Make the test hard to pass, instead of easy to reproduce (automatic logging and screen recording may help).

Enhanced Brainstorming:
Try McLuhan’s Laws of Media

• To McLuhan, a medium was *any* object or thought or tool or idea that has an effect
• Every medium has four effects we can probe
• The “probes” are questions that we can ask about any and every medium:
  • What human capabilities or media does the medium *extend, enhance, accelerate, intensify*?
  • When stretched beyond its limits, how will the medium *reverse* into the opposite of those effects?
  • What ideas about older media does this medium *retrieve*?
  • What media are *obsolesced* by this medium?
### Laws of Media Tetrad Example: The BlackBerry™

**Extends…**
- Presence, availability
- Communication
- Organization
- Meetings and community
- Memoranda
- Navigation

**Retrieves…**
- Telephone
- Appointment book
- Personal letters
- Notepad
- Teletype
- Pony Express
- Tamagotchi
- Tour Guide

**Reverses into…**
- Absence, unavailability
- Distraction, interruption
- Trivial communication
- Inability to delegate
- Dependency
- Repetitive Stress Injury
- Loss of navigation skill

**Obsolesces…**
- Cell phone
- Desktop computer
- Stationery
- Privacy
- Memory
- Face-to-face conversation
- Maps

---

### Metacognition

- observe what you’re observing
- to some degree, just being aware of the pitfalls helps you to defend against problems
- Beware the Meaning Problem
  - you think a signal means one thing, but it means another
  - “That alarm only goes off when there’s a fire drill. This must be just another fire drill”
- Beware the Significance problem
  - “If it were a really serious fire, there’d be an announcement. This must not be a serious fire.”
- Beware the Fill-in Problem
  - our brains automatically compensate for missing information
- pair up!
Change the Observation

• choose something specific to observe
• choose another sense (hearing? touch? smell?)
• Ask
  • “What other things are going on?”
  • “Are there more things like this one?”
  • “What do I believe is the cause of this effect?”
  • “Are there different causes (other than the one that I have inferred) of the same effect?”
  • “Are there different effects from the same cause?”
  • “What other meanings or significance we could take from the thing we've just observed?”

It’s Okay Not To Notice Everything

• We can’t notice everything, even if we wanted to
• That’s why we have development teams
• That’s why we have test teams
• That’s why we have review and testing
• That’s why we test using many different approaches
Have you ever noticed…?

Sometimes we notice things later. When all else fails, take a break, do something else, come back to the problem.

References

- *Blind Spots*
  - Marilyn Von Hecke
- *Introduction to General Systems Thinking*
- *General Principles of System Design*
- *Perfect Software and Other Illusions About Testing*
  - Gerald M. Weinberg
- *Blink*
- *The Tipping Point*
  - Malcolm Gladwell
References

- *Lessons Learned in Software Testing*
  - Kaner, Bach, Pettichord
- *Gut Feelings*
  - Gigerenzer
- *Sensemaking in Organizations*
  - Weick
- *McLuhhan for Managers*
  - Federman

References

- *The Feeling of What Happens*
  - Damasio
- *Tools of Critical Thinking*
  - Levy