

# The Craft of Bug Investigation

**Jon Bach**

Manager for Corporate Intellect

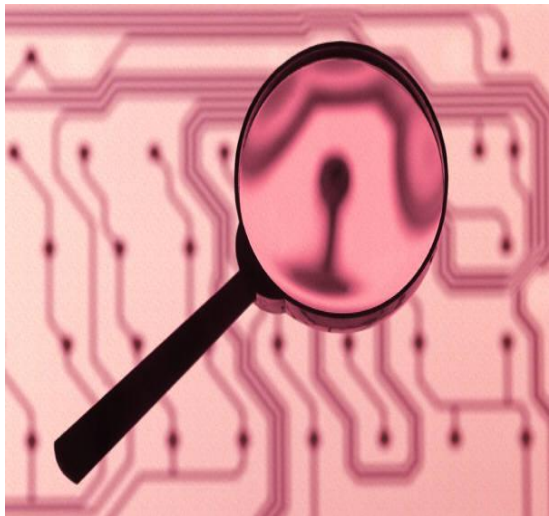
[jonb@quardev.com](mailto:jonb@quardev.com)

**STAR East 2010**



# Preamble

---



**These next 3.5 hours are designed to lead you through a series of exercises that help you narrow down a bug once you've seen something strange.**

**We'll discuss techniques for bug isolation, reproducing intermittent bugs, and skills to help you know when you've gone as far as is reasonable before handing it off to the programmer.**

# A morning of investigation

---

**Art Show**

**Psychic Reader**

**Gettysbug Address**



**Mysterious Spheres**

**Emergence, see?**

**Out-of-Bounds**

**This App Can Break**

*Your mind is a black box...*



- 1) No one can read your mind.
- 2) It takes skill to explain your testing so that you can be accountable for it.
- 3) That requires a lot of practice.

# How did you *\*find\** that?

## Some Investigation Skills and Tactics

***“MR.Q COMC GOARABC R&R?”***

Modeling

Resourcing

Questioning

Chartering

Observing

Manipulating

Collaboration

Generating/Elaborating

Overproduction/Abandonment

Abandonment/Recovery

Refocusing

Alternating

Branching/Backtracking

Conjecturing

Recording

Reporting

***Investigation is a mindset using this skillset.***

# Abductive Inference

---

*Abductive inference means finding the best explanation for a set of data.*

1. Collect data.
2. Find several explanations that account for the data.
3. Find more data that is either *consistent* or *inconsistent* with explanations.
4. Choose the best explanation that accounts for the important data, or keep searching.

# Exercise: Psychic Reader

---



# Lessons

---

**You can be psychic, too**

**You may be clue-less, for now**

**Try “defocusing”**

**“I was just following orders”**



# Exercise: Mysterious Spheres

---



# Lessons

---

**It's not your fault you're a failure**

**Assumptions need nourishment**

**Oracles are people, too**

**“Elsewhere”, is a useful thought**

# Exercise: This App Can Break

---



# Lessons

---

**“It’s not you, it’s me.”** (perception)

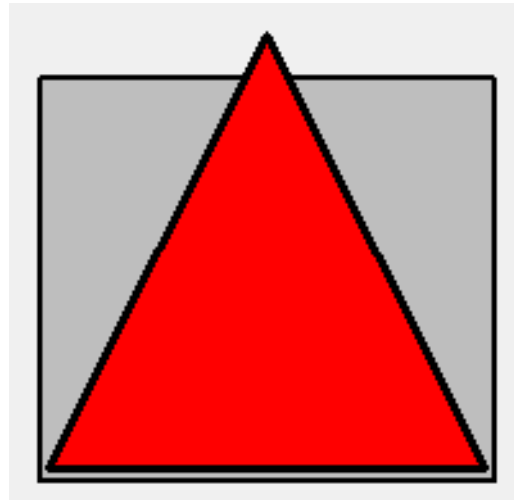
**Hexual healing?**

**Get thee to wikipedia**

**“It’s not me, it’s you.”** (environment)

# Exercise: Out-of-Bounds

---



# Lessons

---

**Be a “model” tester**

**Stagehands work behind curtains**

**Outside data is not inside data**

**Computers are our slaves**

# Exercise: Art Show

---



**What is my Operating Rule?**

**What do you notice?**

**What (crazy) test haven't you tried?**

# Lessons

---

**“You’re so one dimensional”**

**Jump to \*conjectures\***

**Yes, I \*do\* look FAT in this. (M vs. O)**

**Breaking “rules”!**



# Exercise: Gettysbug Address

```
P:\Talks and papers\Exercises\ip_address.exe
Enter a valid IP address: 123.123.123.123
That is a VALID ip address

Enter a valid IP address: 999.999.999.999
That is an INVALID ip address

Enter a valid IP address: _
```

Does it work?

What is the hidden feature?

What story does the data tell?

# Lessons

---

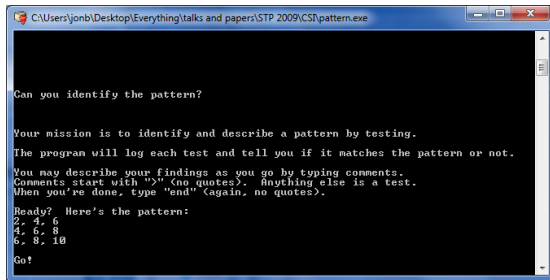
**The Rumble Strip heuristic**

**Missing a mission?**

**Be “thoughtless” (random is good)**

**Do \*anything\*, \*differently\***

# Exercise: Emergence, see?



This app asks you for the next item in a sequence of numbers.

What is the next number?

Operating rule?

# Key Ideas

---

**Data emerges during testing**

**A test is an answer for “I don’t know”**

**Beware of “premature celebration”**

**Cheaters always win** (ask Eric Cartman)

# Intermittent Bugs: Observation

---

Bad observation  
Irrelevant observation  
Bad memory  
Misattribution  
Misrepresentation  
Unreliable oracle  
Unreliable communication

[Link](#)

# Intermittent Bugs: System

---

Purposeful change, and then back to original

Accidental change

Platform change

Flaky hardware

Trespassing system

Executable corruption

Component competition

# Intermittent Bugs: Machine State

---

Frozen conditional  
Improper Initialization  
Resource denial  
Progressive data corruption  
Progressive destabilization  
Overflow  
Occasional functions  
Different mode or option setting

# Intermittent Bugs: Input

---

Accidental  
Secret boundaries or conditions  
Different profile  
Ghost input (alternative source)  
Simultaneous action as others  
Compromised input  
Timing issues  
Crazy Combinations



# Intermittent Bugs: You?

---

You may not be aware of...

- ...variables of influence
- ...sources of distortion in your observations
- ...available tools that might help
- ...boundaries and their characteristics
- ...the system's missing / extra functions
- ...complex / competing algorithms

# Dynamic Quality Paradigm

It's more important to work on *Item B*.



# How much investigation?

---

- 1) Sufficient benefits
- 2) No critical problems
- 3) The benefits outweigh the problems
- 4) In the present situation, and all things considered, improvement would be more harmful than helpful

*The answer must be “Yes” to all four criteria*

# **Your Context...**

---

**Stories?**

# More info

---

- ET Dynamics:  
<http://www.satisfice.com/articles/et-dynamics.pdf>
- Test Heuristics and Planning  
<http://www.satisfice.com>
- Context-Driven Software Testing  
<http://groups.yahoo.com/group/software-testing>
- Center for Software Testing Education and Research  
<http://www.testingeducation.org/BBST>
- Books related to Exploratory Testing skills and tactics  
<http://www.testingreflections.com/node/view/3190>
- Scenario testing examples  
<http://www.testing.com/test-patterns/patterns/pattern-scenario-testing-kaner.html>