

Becoming a Software Testing Expert

James Bach, Satisfice, Inc.

james@satisfice.com

www.satisfice.com

Are you a Testing Expert?

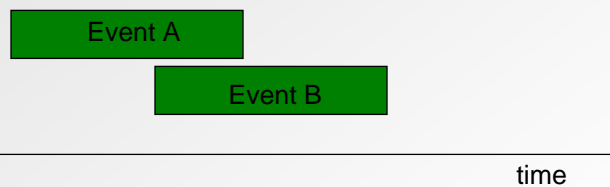
- Analyze these claims:
 1. “You should write a test plan”
 2. “It’s important that testing be repeatable”
 3. “Each test case should have an expected result”
 4. “Test automation saves money and time”
 5. “All testing is based on a model of what being tested”
 6. “Good enough quality is not good enough”
 7. “An undocumented test cannot be improved”
 8. “Exploratory testing is a useful practice”
 9. “It’s better to use the term *defect* than *bug*”
 10. “Ambiguity should be removed from requirements.”

Take the Expert Challenge

1. You are using a calculator.
2. You press the keys “2+2=”

What is the expected result?

Take the Expert Challenge



- You want to test the interaction between two potentially overlapping events.
- What are the test cases?

Take the Expert Challenge

What is *boundary testing* and how is it done?

Please explain your testing methodology,
coherently, in five minutes or less.

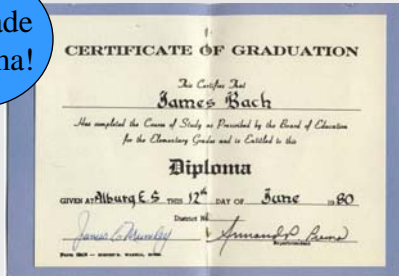
Experts vs. Non-Experts

I believe... for any given claim or problem...

- **Non-Experts are More Likely to Say...**
 - Yes, that's what the books say.
 - This is right.
 - That is wrong.
 - I don't know. {awkward silence}
- **Experts are More Likely to Say...**
 - Tell me more about the context.
 - I can think of how that might be true and I can think of how it might be false. Let's think it through...
 - Let me reframe that...
 - Here are some possible answers...
 - Here's one way I've solved this...
 - I don't know. Here's how I will find out...

These are my credentials

8th grade diploma!



Name of Student..... Bach, James

STUDENT ACADEMIC PERMANENT RECORD

Year	Grade	FM	CR	R
1980-81	Gr. 9			
	English	78	1	
	Social Studies	84	1	
	Physical Science	94	1	
	Math 10			
	Math 11			
	French 11	70	1	
	Physical Education	85	1/2	
1981-82	Grade 10			
	English	63		
	Social Studies	83	1	
	Calculus II	83	1	
	Physics	49		
	Math 10	91	1	88
	Math 11	81	1	72
	Physical Ed.	87	1	

- I also have:
 - Level 1 paraglider pilot certification
 - PADI open water diver certification
 - Driver's license (state of Virginia)
 - Student private pilot license (*expired*)
 - Motorcycle license (*expired*)

Resigned high school in '82.

These are my credentials

**PADCOM, INC. vs. NETMOTION WIRELESS, INC.
NO. 03-983 (SLR)
March 16th, 2006**

Page 371...

- 12 MR. ERNEST: Your Honor, Padcom would ask to
 13 have Mr. Bach recognized as an expert in computer
 14 software and software testing, including networking, as
 15 well as computer programming.
 16 THE COURT: All right. Any objection to that?
 17 MR. SHELTON: No objection, your Honor.
 18 THE COURT: All right.

By "expert tester" I mean any of the following...

- Someone who's *very good* at testing.
- Someone who's *considered to be* an expert.

**You may already be an expert tester.
What I want to do is help you become even better,
according to whatever standard that matters to you.**

Perfect testing is...

Testing is the *infinite process*
of comparing the *invisible*
to the *ambiguous*
so as to avoid the *unthinkable*
happening to the *anonymous*.

In other words, perfect testing is a **challenge**.

A more tractable definition...

Testing is *questioning* a product
in order to *evaluate* it.

Through the *cognition* of the tester, good testing emerges from the infinite space of perfect testing.

What's Special About Testing

- There are few people around to teach you how to test.
- Most of what is taught as “testing” is unreliable or misleading folklore.
- Testing is a complex problem-solving activity.
- Learning testing on your own doesn't cost you much, you don't need anyone's permission, and it generally poses no threat to life or property.

However...

- It's hard to know if you are doing it well.
- Good testing varies quite a lot with the context.

What's Special About Testing

Testing is not part of Computer Science.

(But knowing Computer Science helps)

I belong to the Context-Driven Testing Community.
We follow certain principles.

1. The value of any practice depends on its context.
2. There are good practices in context, but there are no best practices.
3. People, working together, are the most important part of any project's context.
4. Projects unfold over time in ways that are often not predictable.
5. The product is a solution. If the problem isn't solved, the product doesn't work.
6. Good software testing is a challenging intellectual process.
7. Only through judgment and skill, exercised cooperatively throughout the entire project, are we able to do the right things at the right times to effectively test our products.

source: Lessons Learned in Software Testing, by Kaner, Bach, Pettichord

Some Cool Things That Experts Have and Do

■ Experts have...

- Situational awareness
- Confidence in confusion
- Colleague network
- Trained reflexes
- Awareness of limitations
- Diverse experiences
- Relevant knowledge
- Mental models for problem-solving
- Reputation

■ Experts do...

- Avoid traps and dead ends
- Systematic inquiry
- Confront authority and convention
- Self-training and retraining
- Self-criticism
- Pattern matching on experience
- Coherent explanations
- Justify methodology
- Write, speak, teach

What an Expert Sounds Like *Steve McQueen, Towering Inferno*

[talking to fire commissioner] What do we got here, Kappy?

Fire started, 1st floor, storage room. It's bad. Smoke's so thick, we can't tell how far it's spread.

Exhaust system?

Should've reversed automatically. It must be a motor burnout.

Sprinklers?

They're not working on .

Why not?

I don't know.

[talking to architect] Jim? Give us a quick refresher on your standpipe system.

Floors have 1/2 and 3/4-inch outlets.

GPM?

Fifteen hundred from ground to 1st floor from 1st floor to 2nd floor and 2nd floor from there to the roof.

Are these elevators programmed for emergencies?

Yes.

What floor are your plans on?

My office.

That's two floors below the fire. It'll be our Forward Command. Men, take up the equipment. I wanna see all floor plans, through .

Expert Performance is Situational.

Expertise is situated...

- ...socially. If your clients and peers don't like your work, you won't be able to function as an expert in their eyes.
- ...psychologically. If you're tired, angry, or indifferent, you won't perform well.
- ...technically. If you don't know anything about databases, for instance, you will be limited in how well you test them.

A Personal Vision of Testing Expertise

■ "I can..."

- test *anything*...
- under *any* conditions...
- in *any* time frame...

HOW IS IT POSSIBLE?!

AMAZING CLAIM!

A Personal Vision of Testing Expertise

■ “Relative to...”

- my standing in the local community...
- how hard I try...
- my technical insight...

■ “...I can...”

- test *anything*...
- under *any* conditions...
- in *any* time frame...



A Personal Vision of Testing Expertise

...that's do-able.

■ “Relative to...”

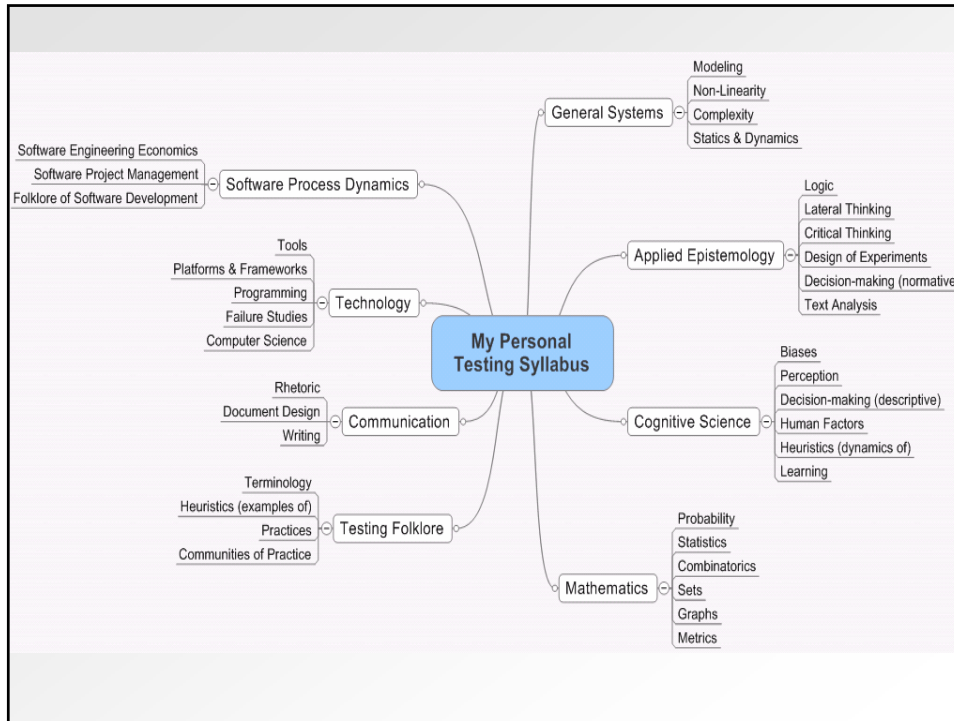
- my acceptance within the local process culture...
- how much I want to succeed...
- my technical insight...

■ “...I can...”

- test *anything*...
- under *any* conditions...
- in *any* time frame...

■ “...such that,”

- I deliver useful results in a usable form.
- I perform at least as well as any other expert would.
- I choose methods that fit the situation.
- I can explain and defend my work on demand.
- I collaborate effectively with the project team.
- I make appropriate use of available tools and resources.
- I advise clients about the risks and limitations of my work.
- I advise clients about how they can help me do better work.
- I faithfully and ethically serve my clients.



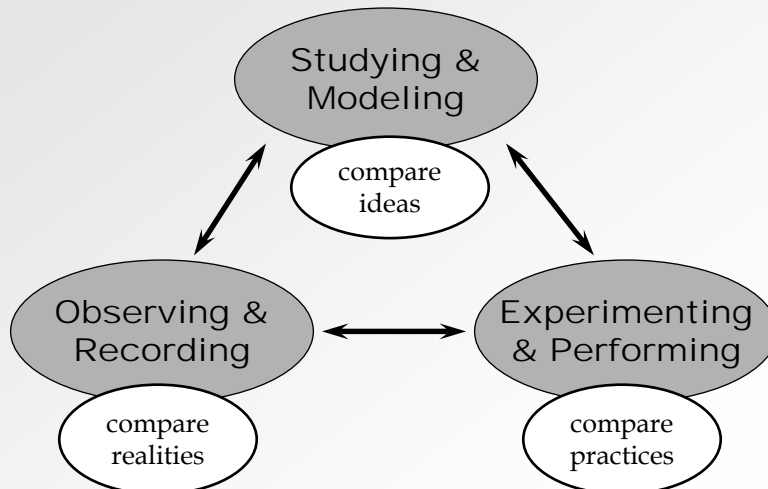
Notice? No "Testing" Books!

- Gödel, Escher, Bach: An Eternal Golden Braid , Douglas Hofstadter.
- The Sciences of the Artificial, 3rd Ed., 1996, Herbert A. Simon
- Introduction to General Systems Thinking, 1975, Gerald M. Weinberg.
- Secrets of Consulting, 1986, Gerald M. Weinberg
- More Secrets of Consulting, 2002, Gerald M. Weinberg
- Quality Software Management, Vol. 1: Systems Thinking, 1991, Gerald M. Weinberg
- General Principles of Systems Design, 1988, Gerald M. Weinberg, Daniela Weinberg
- Tools of Critical Thinking, 1997, David A. Levy
- Exploring Requirements: Quality Before Design, 1989, Don Gause, Gerald M. Weinberg
- The Social Life of Information, 2000, John Seely Brown, Paul Duguid
- How to Solve It, 1945, George Polya
- How to Read and Do Proofs, 1990, Daniel Solow
- Judgment and Decision Making, 2000, Terry Connolly, et al
- Cognition in the Wild, 1996, Edwin Hutchins
- Thinking and Deciding, 1994, Jonathan Baron
- Lateral Thinking: Creativity Step by Step, 1990, Ed De Bono
- Abductive Inference: Computation, Philosophy, Technology, 1996, John R. Josephson, Susan G. Josephson
- Time Pressure and Stress in Human Judgment and Decision Making, 1993, Ola Svenson, A. John Maule
- Conjectures and Refutations: The Growth of Scientific Knowledge, 1992, Karl Popper
- Proofs and Refutations, 1976, Imre Lakatos
- The Pleasure of Finding Things Out, 1999, Richard Feynman
- Rethinking Systems Analysis and Design, 1988, Gerald M. Weinberg
- Quality Software Management, Vol. 3: Congruent Action, 1994, Gerald M. Weinberg
- Becoming a Technical Leader: An Organic Problem-Solving Approach, 1986, Gerald M. Weinberg

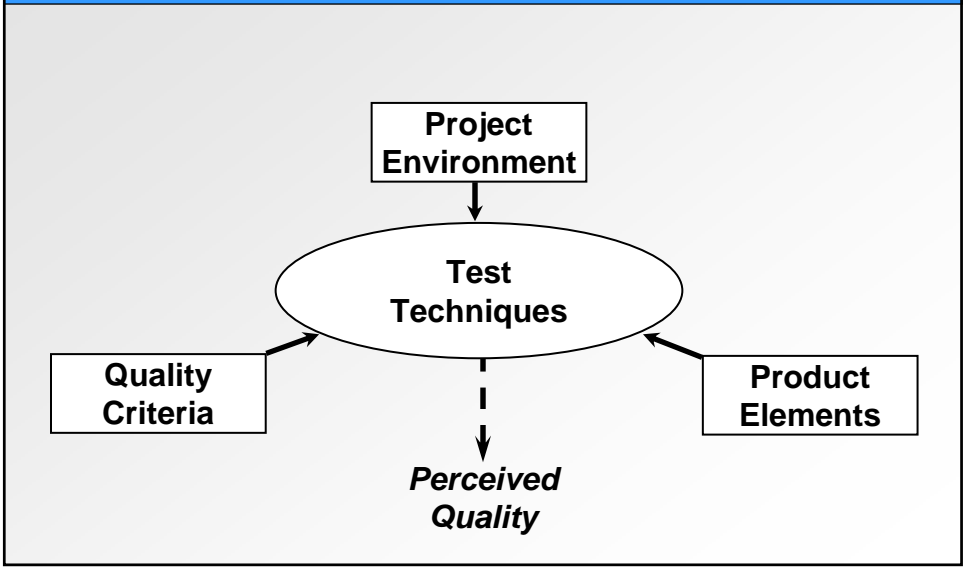
What Level of Learning?

- **Level 0:** “I overcame *obliviousness*.”
 - I now realize there is something here to learn.
- **Level 1:** “I overcame *intimidation*.”
 - I feel I can learn this subject or skill. I know enough about it so that I am not intimidated by people who know more than me.
- **Level 2:** “I overcame *incoherence*.”
 - I no longer feel that I’m pretending or hand-waving. I feel reasonably competent to discuss or practice. What I say sounds like what I think I know.
- **Level 3:** “I overcame *competence*.”
 - Now I feel productively self-critical, rather than complacently good enough. I want to take risks, invent, teach, and push myself. I want to be with other enthusiastic students.

Developing your expertise



All Product Testing is Something Like This



Thirty-Six Testing Heuristics

"cidtestdsfdpotcrusspicstmpldfsfscura"

- | | | | |
|---------------------|------------|-----------------|-------------------|
| Customers | Structures | Capability | Function testing |
| Information | Functions | Reliability | Domain testing |
| Developer relations | Data | Usability | Stress testing |
| Team | Platforms | Security | Flow testing |
| Equipment & tools | Operations | Scalability | Scenario testing |
| Schedule | Time | Performance | Claims testing |
| Test Items | | Installability | User testing |
| Deliverables | | Compatibility | Risk testing |
| | | Supportability | Automatic testing |
| | | Testability | |
| | | Maintainability | |
| | | Portability | |
| | | Localizability | |

What About Certification?

- A lot of people feel that college degrees comprise some kind of certification. I haven't felt a need for them.
- I have not seen a "Certified Tester" program that I respect. The ones I've seen are an embarrassment to the craft and an insult to skilled testers everywhere.
- *Certification by association* has worked well for me.
- *Certification by body of work* also works well.
- **When you make a name for yourself, your name trumps any "certification" you might also have.**

My Advice to New Experts

- Practice, practice, practice!
- Don't confuse experience with expertise.
- Don't trust folklore— but learn it anyway.
- Take nothing on faith. Own your methodology.
- Drive your own education— no one else will.
- Reputation = Money. **Build and protect your reputation.**
- Relentlessly gather resources, materials, and tools.
- Establish your standards and ethics.
- Avoid certifications that trivialize the testing craft.
- Associate with demanding colleagues.
- Write, speak, and *always tell the truth as you see it.*