Debunking Myths in Software Testing

Considerations for a Career in Critical Thinking

University of Waterloo

21 November 2007

By Paul Carvalho

http://www.STAQS.com/

What do *you* want?

A successful career requires *intrinsic motivation*

Corollary: Anything easy becomes difficult if done reluctantly

- Far East Fortune Cookie Co. Ltd.

Career vs. Job

Career:

- a long-term, intentional focus on a field or type of work
- personal choice, personal goals (may change)

Job:

- a temporary assignment with a particular employer
- may or may not apply to your career

➢ *Explore different job opportunities before deciding upon a particular career*

Agenda

- What motivates me in Software Testing
- What is Testing
- Suggestions to help prepare for an IT career

What motivates *me*?

Top 10 things I like about Software Testing:

1. work with new Technology
2. work with People
3. somewhere between Programming and Tech Support
4. both Science and Art
5. job security
6. lots of career/specialisation opportunities
7. learning and growth opportunities
8. challenging and changing
9. diversity of approaches
10. make a difference

1. Technology

- Work with computers:
  - Often high-tech, the latest and greatest
  - New tech, devices and systems
  - Opportunity to gain in-depth knowledge
- Software:
  - New applications, new tools, new solutions
  - Reverse Engineering and Systems Thinking skills give you detailed insights into how software works
  - Internet is your friend

➢ ***NB:*** *The Medium is NOT the message!*
(i.e. you don't have to be a programmer to work with computers)

2. People

- Other employees
- Customers

- Every problem has 2 sides:
  - *Technical*
  - *Emotional*

"No matter what the problem is, it's always a people problem."

- Jerry Weinberg

2. People

- Develop ‘soft’ skills on the job:
  - Listening, giving/receiving feedback
  - Sales, negotiation, persuasion
  - Understand yourself & others (e.g. MBTI, team dynamics, group problem-solving, etc.)

  ➢ *Applied psychology*

- Gerald (Jerry) Weinberg: author and teacher of the *psychology* and *anthropology* of computer software development.

3. Programming ⇔ Tech Support

- *Been there:* Programmer:
  - Creating new algorithms and code was fun
  - Felt out of touch with customer in team env't

- *Done that:* Tech Support:

  - Found it exhausting after 6 months (I vs E preference)
  - Felt less able to influence product development

- QA/Test:
  - Close to customer needs, able to influence Dev't
  - More variety, "Super user", little bit of both worlds

4. Science & Art

- The *Science*:
  - Good Testing practices come from understanding the **Scientific Method**
  - Nature of observation, bias, doubt
  - Research skills, problem-solving, analysis
  - Specific Test Techniques
  - **Math:** Statistics, Combinatorics
  - Modelling and Systems Thinking

> *Test Techniques* <

- Test Techniques are *ways* of looking at a system.
  - *(Ideally, they help you find bugs that are important to stakeholders.)*
- Some techniques include:

popular

- Functional analysis
- Cross-function analysis
- Specification-based Path analysis
- Code-based Path analysis
- Statement coverage
- Branch coverage
- Decision tables
- Transaction flows
- Equivalence partitioning
- Boundary value analysis
- Risk-based
- Prior defect history
- Pareto analysis of defect patterns
- Operational profiling
- Statistical sampling
- Combinatorial testing
- Failure modes and effects analysis (FMEA)
- Error guessing
- Database integrity testing
- Exception handling
- Stress & limit tests
- Test factor analysis / Pairwise comparison
- Orthogonal arrays
- State-Transition diagrams
- Cause-Effect graphing
- User scenario tests
- Flow tests
- Hazard or Threat analysis
- Upstream/Downstream test
- Interface test
- Input constraint attacks
- Stored data constraint attack
- Computation constraint attack
- Output constraint attacks
- Smoke test
- Monkey testing
- Random testing
- Performance
- Usability
- Confidentiality
- Security
- Installability
- Compatibility
- Reliability/Stability
- Claims tests
- Maintainability
- Testability
- Supportability
- User documentation
- Localization (L10n)
- Internationalization (I18n)
- . . .

How many do you use?

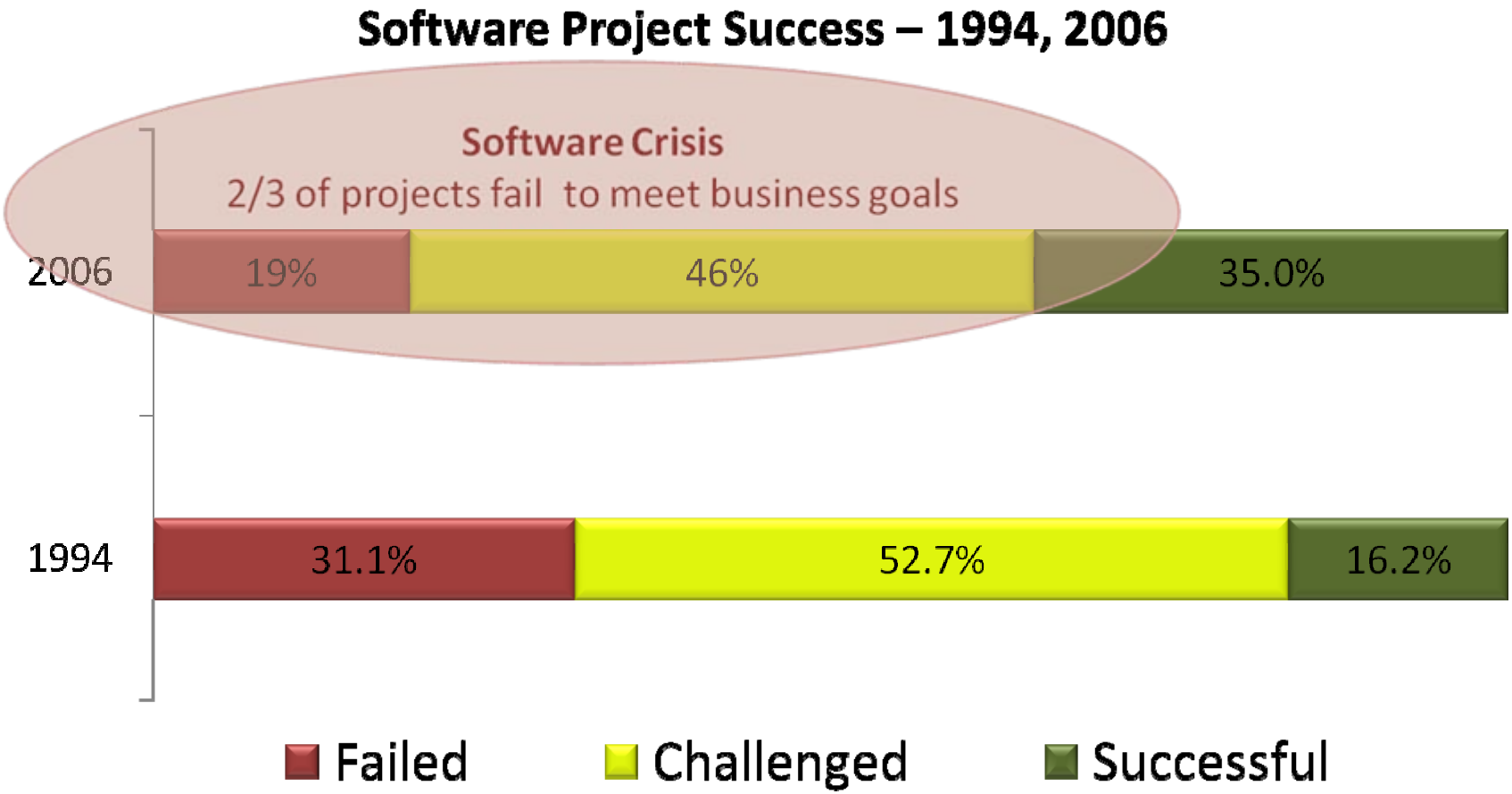
Regularly?

4. Science & Art

- The *Art*:
  - How you express yourself, style
  - Give meaning to the information generated & presented
  - Sources of inspiration
  - Testing can be a Symphony!

"First with the head then with the heart"
- Bryce Courtenay "The Power of One"

5. Job Security



"*The CHAOS Chronicles*" 1994, 2006 The Standish Group

(slide courtesy Declan Whelan © 2007)

5. Job Security

- Software Testing not really taught in school
  - *maybe* one course/school if you look hard enough
- Lack of education and improper or lazy development practices = poor Quality work
- In many places, bad testing prevails

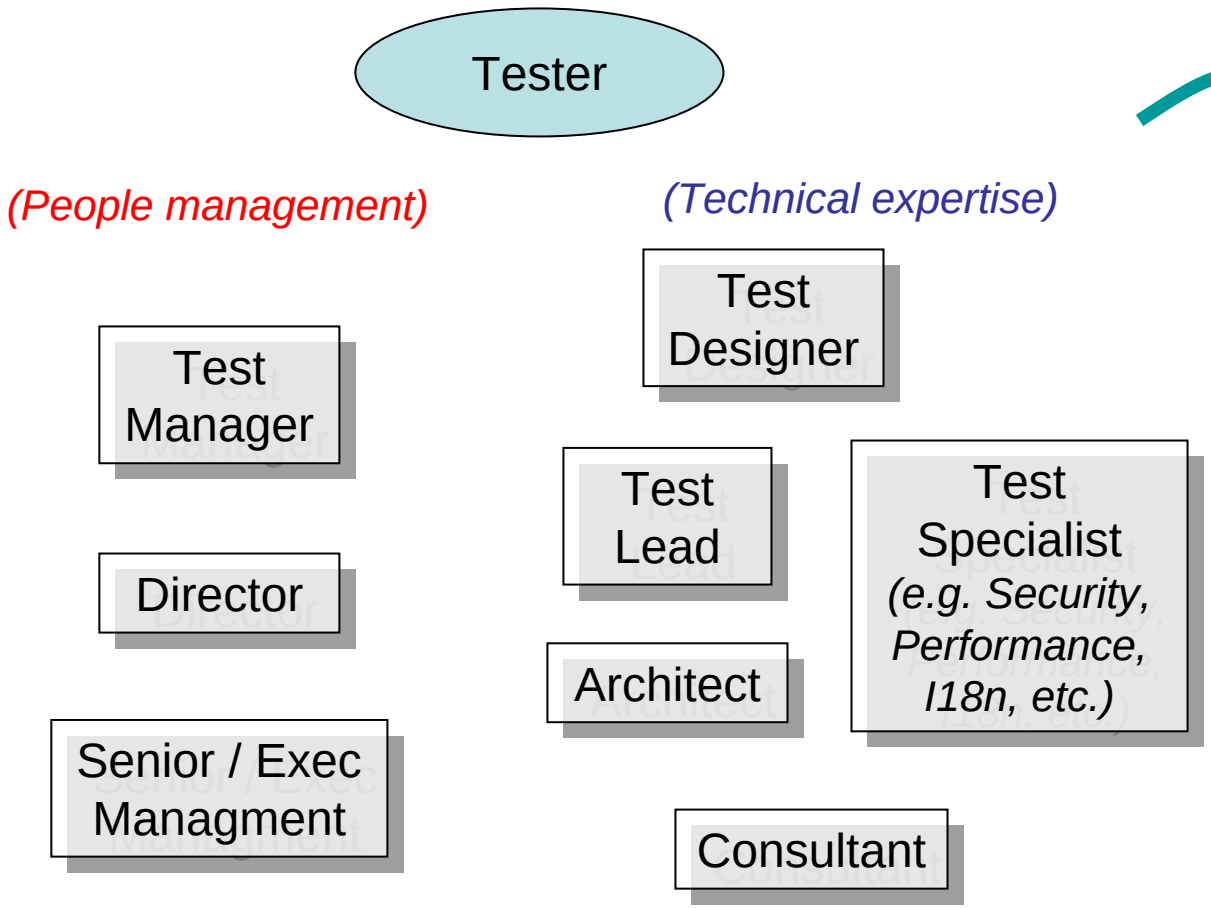
➢ If you can think and *want* to do good work, there is no shortage of good Testing opportunities in the foreseeable future

6. Lots of Career Opportunities

- Limited by your imagination
- *Some examples:*
  - *I love languages → Internationalisation (I18n) Testing specialist*
  - *Generalist? → Black-Box (System) Testing specialist*
  - *Psych background? → Usability Testing & Profiling*
  - *Programming? → TDD and Automation*
  - *Business? → Use cases, Business Analyst*
  - *Like People? → Test Manager, Director, etc.*
  - *also: Security Testing, Performance Testing, etc.*

6. Lots of Career Opportunities

- *Possible career path considerations:*



or, move on..

Within IT:

- Programmer
- System Analyst
- Project Manager
- Tech writer
- Support specialist
- etc.

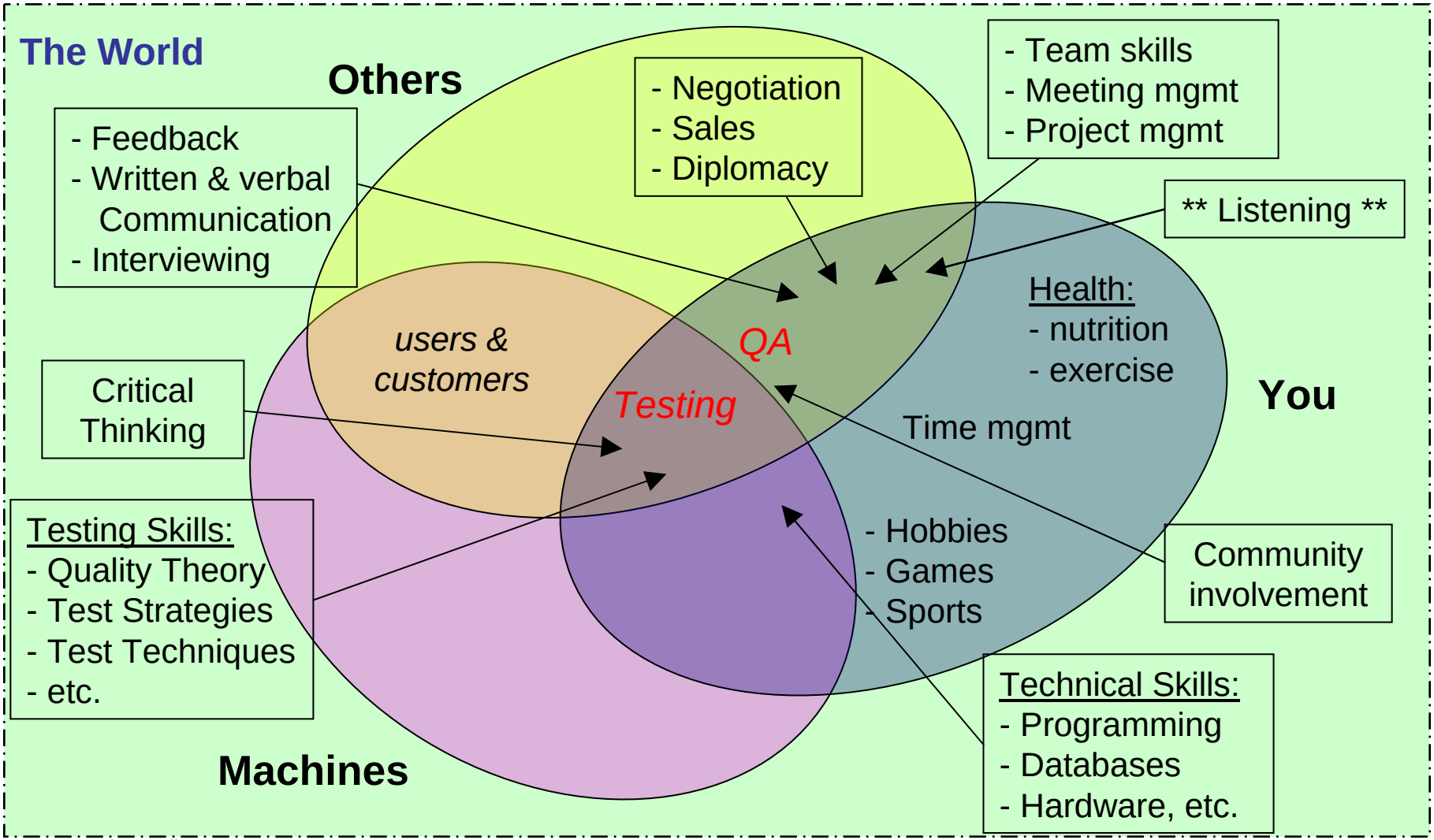
Outside of IT:

- anything you want (e.g. Secret Agent, Private Investigator...)

7. Learning and Growth

- I love to learn. Lifelong learner.
- *Receive knowledge:*
  - Learn something new every year
  - Take courses, attend workshops, conferences
  - Learn from the industry and situations you're in at work
- *Share knowledge:*
  - Give presentations
  - Write articles, blog, participate in online communities
  - Teach classes and courses
  - Participate in Workshops – International community
  - Contribute to growing Body of Knowledge (e.g. TestingEducation.org)

7. Learning and Growth:
Good Testing Skills Include



8. Challenging & Changing

- More than meets the eye
- Complete testing of any system is ***impossible***
  - How do you decide which tests to do and which to skip?
  - How do you know when your test effort is *good enough*?
- No Silver Bullets
- The *context* may change everything
  - Very rarely are two projects the same

> *Testing is done in Context* <

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.

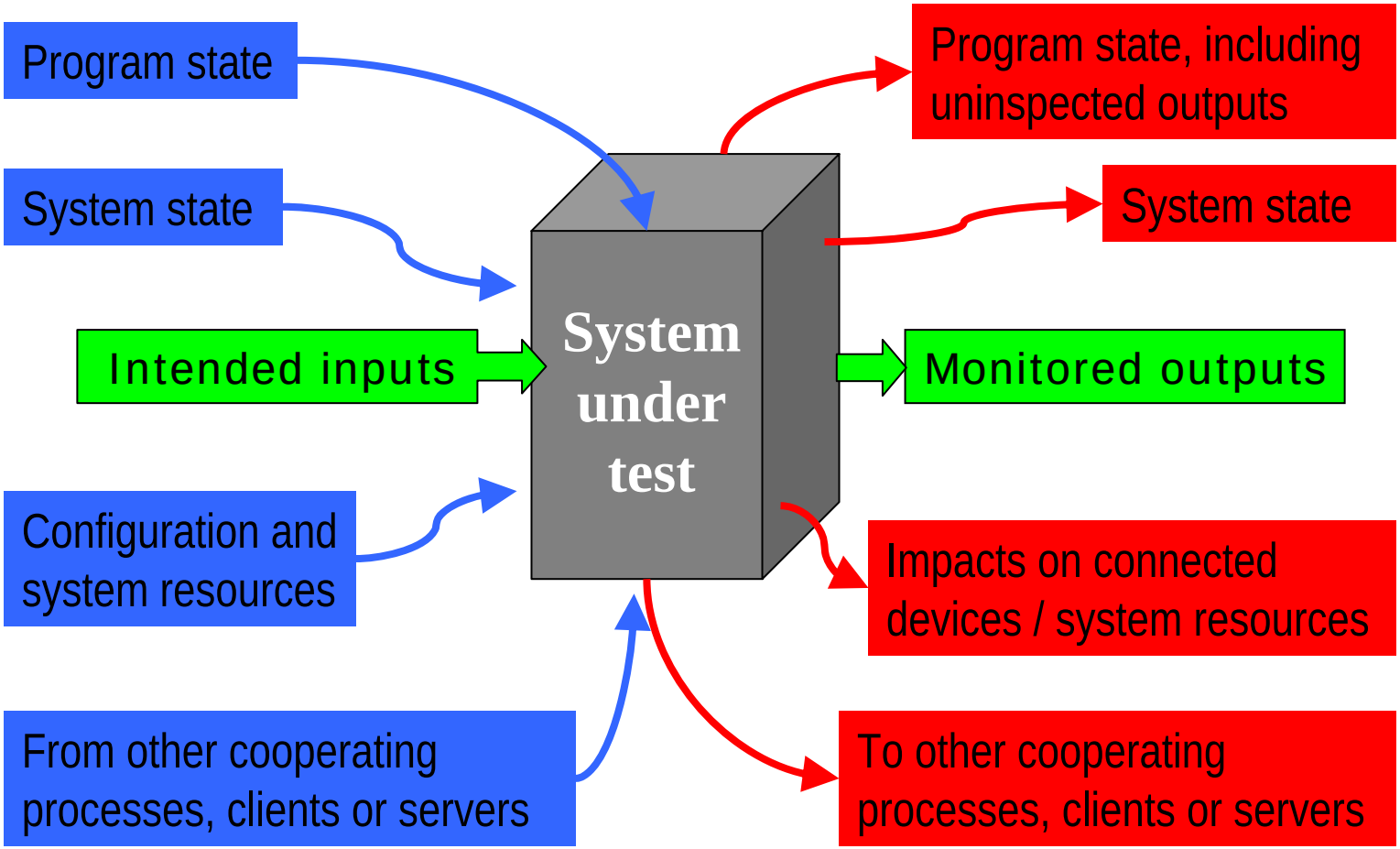
8. Challenging & Changing

- Are you sure about what you think you *know*?
  (about the requirements, system, ...)
    - Epistemology might provide insights
- Are you sure about what you think you *saw*?
    - **Inattentional Blindness** affects everyone
    - See short video (online) or colour-changing card trick
- Oracles
    - How can you tell if you have a problem?

> *Oracles* <

- Useful but fallible heuristics (principle or mechanism) that help you recognise a problem
- Consistent with:
  - **History:** Present function behaviour is consistent with past behaviour.
  - **Image:** Function behaviour is consistent with an image that the organization wants to project.
  - **Comparable Products:** Function behaviour is consistent with that of similar functions in comparable products.
  - **Claims:** Function behaviour is consistent with what people say it's supposed to be.
  - **User's Expectations:** Function behaviour is consistent with what we think users want.
  - **Product:** Function behaviour is consistent with behaviour of comparable functions or functional patterns within the product.
  - **Purpose:** Function behaviour is consistent with apparent purpose.

Oracles and Blindness:
A program can fail in <u>many</u> ways



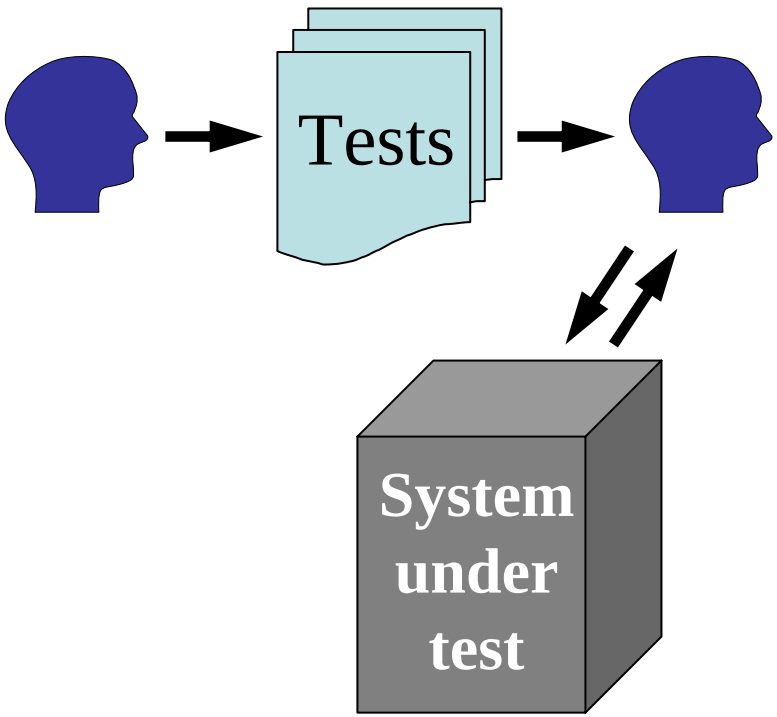
9. Diversity of Approaches

- Many different ways of Testing
- Bret Pettichord: 5 ‘Schools’ of Software Testing:
  - Analytical School
  - Standard School
  - Quality School
  - Context-Driven School
  - Agile School
- The Great Debate: *Scripted* vs *Unscripted* testing

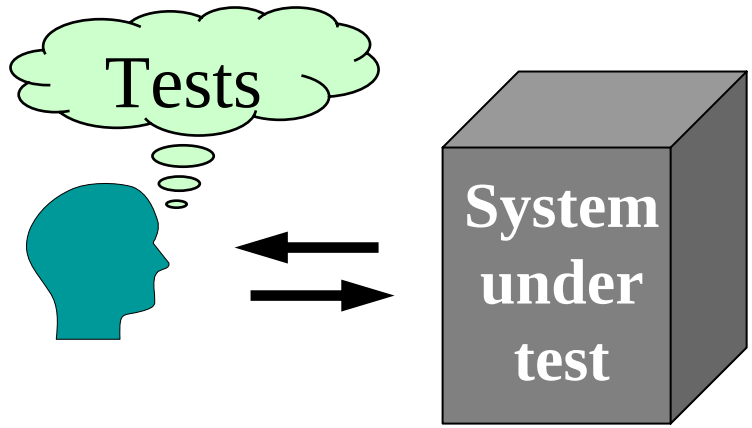
Comparing Testing Approaches

vs.

In *scripted* testing, tests are first designed and recorded. Then they may be executed at some time later or by a different tester.



In ***exploratory*** testing, tests are designed and executed at the same time, and they are often not recorded.



(Critical Thinking applied *at the time* of test execution)

(Based on slide from James Bach)

10. Make a Difference

- To your company
- To your customers (and by extension, the world)

1. Safety

a) of Lives. Bugs can kill.

- In November 2000, 21 patients died after being overdosed by a Cobalt-60 radiotherapy machine at the National Cancer Institute in Panama.

b) of Information, Finances. (Security)

- Identity theft
- Safeguard your savings

10. Make a Difference

2. Microeconomics

a) Competition.
   - You want better Quality than your competitors

b) Maximise Profit.
   - Cost to fix bugs increases exponentially over time (Waterfall model)

3. Macroeconomics

- US Commerce Dept estimated $100 billion spent from 1995-2001 testing and repairing computers affected by the Y2K problem
- In 2002, the National Institute of Standards and Technology (NIST) reported that software bugs cost the U.S. economy an estimated $59.5 billion annually, or about 0.6 percent of the gross domestic product

10. Make a Difference: *Recent Bugs in the News*

- BlackBerry Server software upgrade problem affected 8 million users (April 2007)

- Air Canada's computer reservation system bug delayed flights and affected thousands of passengers across Canada (November 2007)

What is Testing?

- Software *Development* is the creative process of transforming an idea into a working product

- Software *Testing* is the feedback mechanism to check that expectations are being met
  - Employ tools and techniques. *Do you know the right techniques?*
  - Make observations. *Do you know what you saw? Are you sure? Do you know what you didn't see?*
  - Compare against expectations. *Do you know what they are?*
  - ➢ Use Critical Thinking and judgement to make inferences
  - Report findings to the stakeholders responsible for the product

What is Testing?

- A tester generates information (the ‘i’ in ‘IT industry’)
- A *skilled* tester generates good information on important things *fast* (i.e. in time for someone to act on the info)
    - Testing is a service
- Cem Kaner: “Testing is a technical investigation done to expose quality-related information about the product under test”
- Jerry Weinberg: “Quality is value to some person”
- Mike Emeigh: “Testing doesn’t build quality - it reveals it”

What makes Testing worth spending time on -- as a job and maybe as a career?

"We are professional investigators. Rather than building things, we find ways to answer difficult questions about the quality of the products or services we test.

Our job--if we choose to do it well--requires us to constantly learn new things, about the product, its market, its implementation, its risks, its usability, etc.

To learn these, we are constantly developing new skills and new cognitive structures in a diversity of fields. It also requires us to communicate well to a diverse group of people.

We ALSO get to build things (test tools), but very often, we build to our own designs, which can be more satisfying than building an application that does something we'll never personally do (or want to do).

Learning to do good software testing requires learning to do critical thinking well, and to back it up with empirical research."

- Cem Kaner, 'Software-Testing' discussion forum, November 2007.

Preparing for the Real World

- Whether *in school* or *in the work force*, use Gardner's theory of Multiple Intelligences / Talents to guide your learning:

Typical focus

1. Linguistic
2. Logical-Mathematical
3. Spatial or Visual
4. Musical
5. Bodily-Kinesthetic
6. Interpersonal
7. Intrapersonal
8. Naturalistic
9. Philosophical - Ethical

➢ Expand your focus, expand your mind

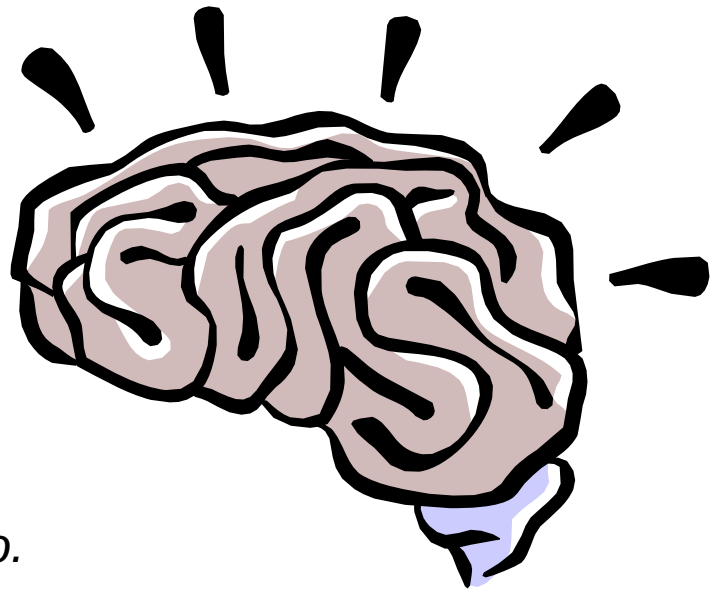
Build your own
Undergrad Software Testing Program

- Recommend the following courses as a baseline to prepare for a Testing career (*check out UW BKI program*):
  - **Philosophy** – Ethics, Logic, Critical Thinking, Creative Thinking
  - **English** – Technical communication, presentations, Rhetoric
  - **Psychology** – Intro, Cognitive Processes
  - **Science** – courses to learn and practice the Scientific Method
  - **Math** – Statistics, Algebra, Combinatorics
  - **CS** – Programming principles, Pattern recognition
  - **Economics** – Micro, Macro
  - **Engineering** – Intro, Testing courses, Software Engineering
- Include a Minor or Specialisation in an area of Interest

➢ People need ***diversity*** of skills to be adaptable today

For Your Future?

- Choose to use your brain
  - Good testing requires critical thinking & creativity
  - Explore, Think, Learn, Grow, Adapt, Be Successful
- *Turn your brain on and leave it on!*



Ubi dubium, ibi occasio.

Want to know more?

- Books:
  - "Better Software" magazine
  - Weinberg, G. "Quality Software Management, Volumes 1-4."
  - Kaner, C., et al. "Testing Computer Software, 2nd ed."
  - Kaner, C., et al. "Lessons Learned in Software Testing"
- Web Sites:
  - www.TestingEducation.org
  - www.StickyMinds.com
  - www.Satisfice.com, www.Exampler.com, www.Pettichord.com
- Local Community:
  - KW Software Quality Association (kwsqa.org)
  - Toronto Association of Systems & Software Quality (tassq.org)