Re-**Brand**ing Software Testing
A Beginning

Kitchener-Waterloo Software Quality Association

26 March 2008

By Paul Carvalho

http://www.STAQS.com/

Pop Quiz

- What’s the first thing ***you*** think of when you hear the term ‘Software Testing’?

- What’s the first thing *other* (IT) people say when *they* hear ‘Software Testing’?

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 (right product built right)
  - 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, decisions
  - 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)
    - e.g. like an ER doctor

- 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.

Top 10 realities and perks of Software Testing

1. work with new Technology
2. work with People
3. it’s a Service
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. New & Cool Technology

- Work with computers:
  - Often high-tech, the latest and greatest
  - New tech, gadgets, 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 (ideas, solutions, tools & more)

➢ ***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

"Yes, I think I have good people skills. What kind of idiot question is that?"

- 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 Behavioural psychology*

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

3. It’s a Service

- Treat your team and function like a Business
- Each project has a Budget, Schedule, and Goal
  - Budget: “You have no money for extra equipment, tools, training or people.” *Does that stop you?*
  - Schedule: “The Release date is fixed.” *Plans change.*
  - Goals: Some people might not know what these are. *Ask. Infer. Pick something good and work with it.*
- A Test Plan is a Statement of Work, a contract
- Your product is *Information* *(not paperwork)*
- What **value** are you providing?

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
  - There are Methods and Processes

> *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
- Statistical sampling
- Operational profiling
- Statistical process control (SPC)
- Failure modes and effects analysis (FMEA)

- Error guessing
- Database integrity testing
- Exception handling
- Stress & Load tests
- Test factor analysis
- Orthogonal arrays
- State-Transition diagrams
- Cause-Effect graphing
- User scenario tests
- Flow tests
- Hazard or Threat analysis
- Upstream/Downstream test
- API / Interface test
- Input constraint attacks
- Stored data constraint attack
- Computation constraint attack
- Output constraint attacks
- Formula analysis

- 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)
- Shoe test
- ...

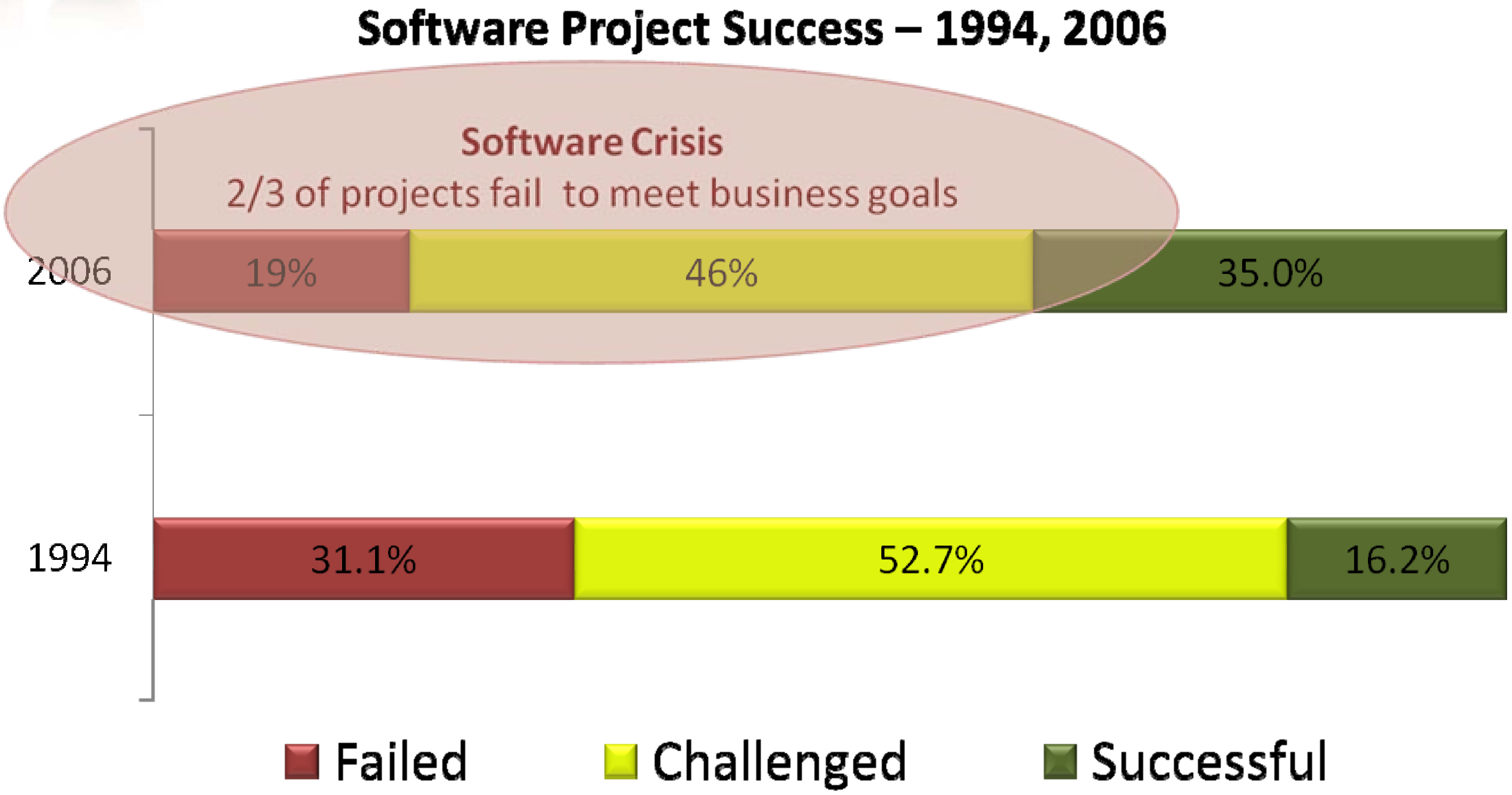
4. Science & Art

- The *Art*:
  - Improvisation
  - How you express yourself, style
  - Give meaning to the information generated & presented
  - 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 ☹

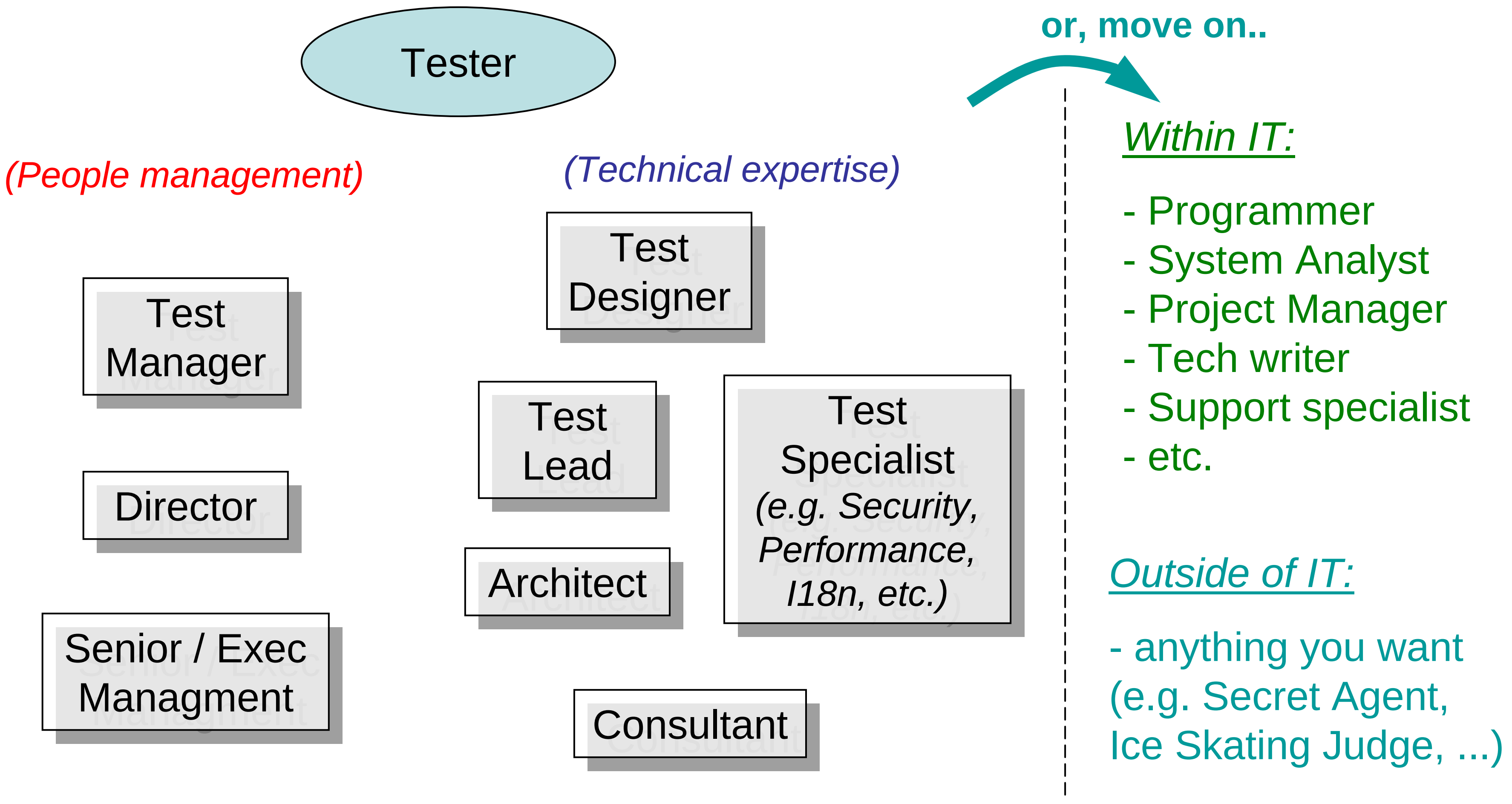
➢ If you can think and *want* to do good work, there is no shortage of good Testing opportunities in the foreseeable future (i.e. Demand > Supply)

6. Career Opportunities

- Only limited by your imagination
- *Some examples:*
  - *Knack for 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? → Social Engineering, Test Manager, etc.*
  - *also: Security Testing, Performance Testing, etc.*

6. Career Opportunities

- *Possible career path considerations:*



Tester

(People management)

Test Manager

Director

Senior / Exec Managment

(Technical expertise)

Test Designer

Test Lead

Architect

Test Specialist *(e.g. Security, Performance, I18n, etc.)*

Consultant

or, move on..

Within IT:

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

Outside of IT:

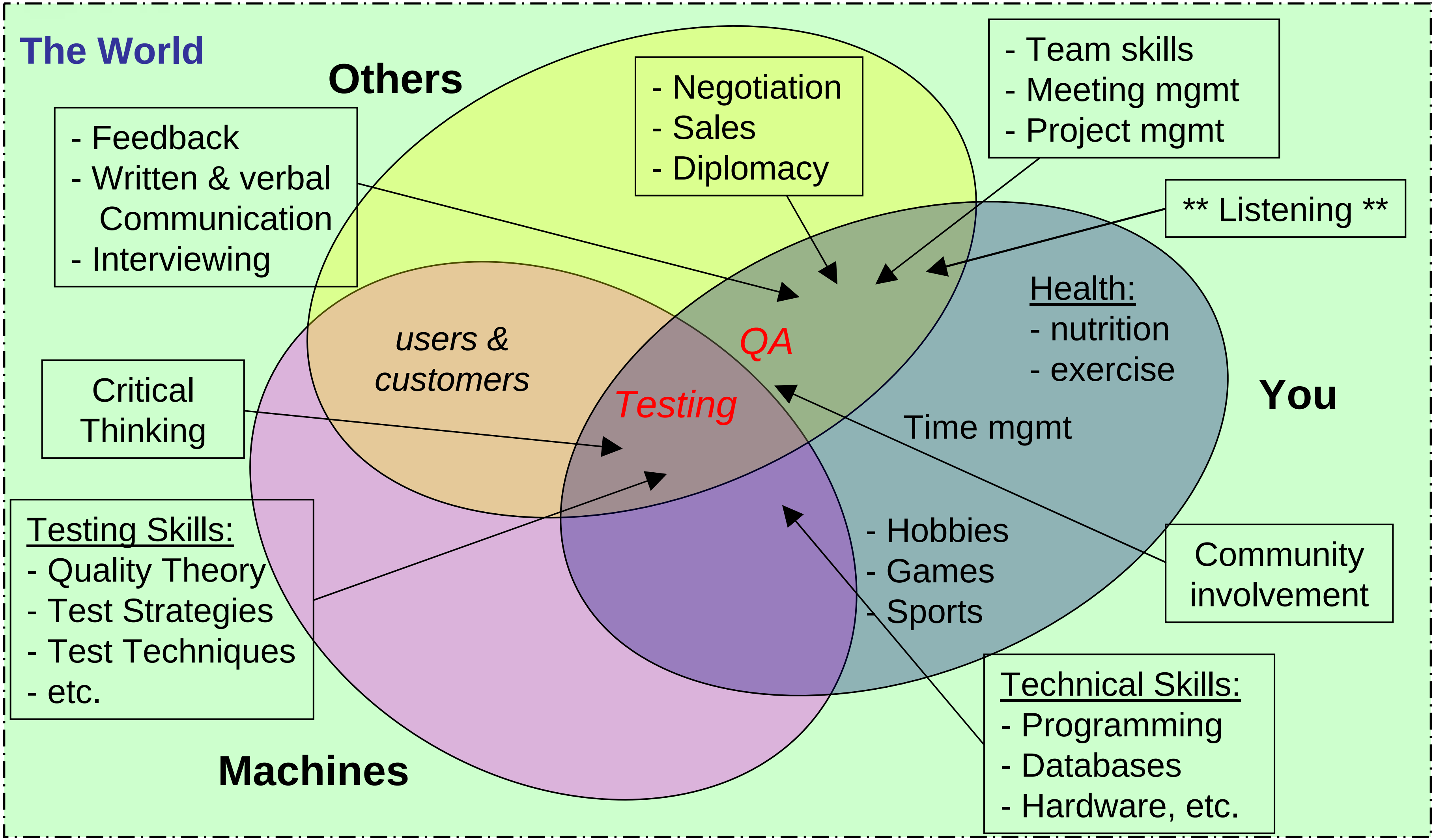
- anything you want (e.g. Secret Agent, Ice Skating Judge, ...)

7. Learning and Growth

You have opportunities to:

- *Receive knowledge:*
  - Learn something new every year, read books, articles, blogs
  - Take courses, attend workshops, conferences
  - Learn from the industry and situations you're in at work
- *Share knowledge:*
  - Give presentations, train others
  - 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...



The World

Others

You

Machines

users & customers

QA

Testing

- Feedback
- Written & verbal Communication
- Interviewing

- Negotiation
- Sales
- Diplomacy

- Team skills
- Meeting mgmt
- Project mgmt

** Listening **

Critical Thinking

Health:
- nutrition
- exercise

Time mgmt

Testing Skills:
- Quality Theory
- Test Strategies
- Test Techniques
- etc.

- Hobbies
- Games
- Sports

Community involvement

Technical Skills:
- Programming
- Databases
- Hardware, etc.

7. Learning and Growth*:* *Multidisciplinary Education Req'd*

- Need Holistic view of world. No more silos. Diversify!
- For example, study all these for a start:
  - **Philosophy** – Ethics, Logic, Critical Thinking, Creative Thinking
  - **English** – Technical communication, presentations, Rhetoric
  - **Psychology** – Intro, Cognitive Processes, Behavioural
  - **Science** – courses to learn and practice the Scientific Method
  - **Math** – Statistics, Algebra, Combinatorics
  - **CS** – Programming principles, Pattern recognition
  - **Economics** – Micro, Macro, Entrepreneur
  - **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

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.

http://www.context-driven-testing.com/

8. Challenging & Changing

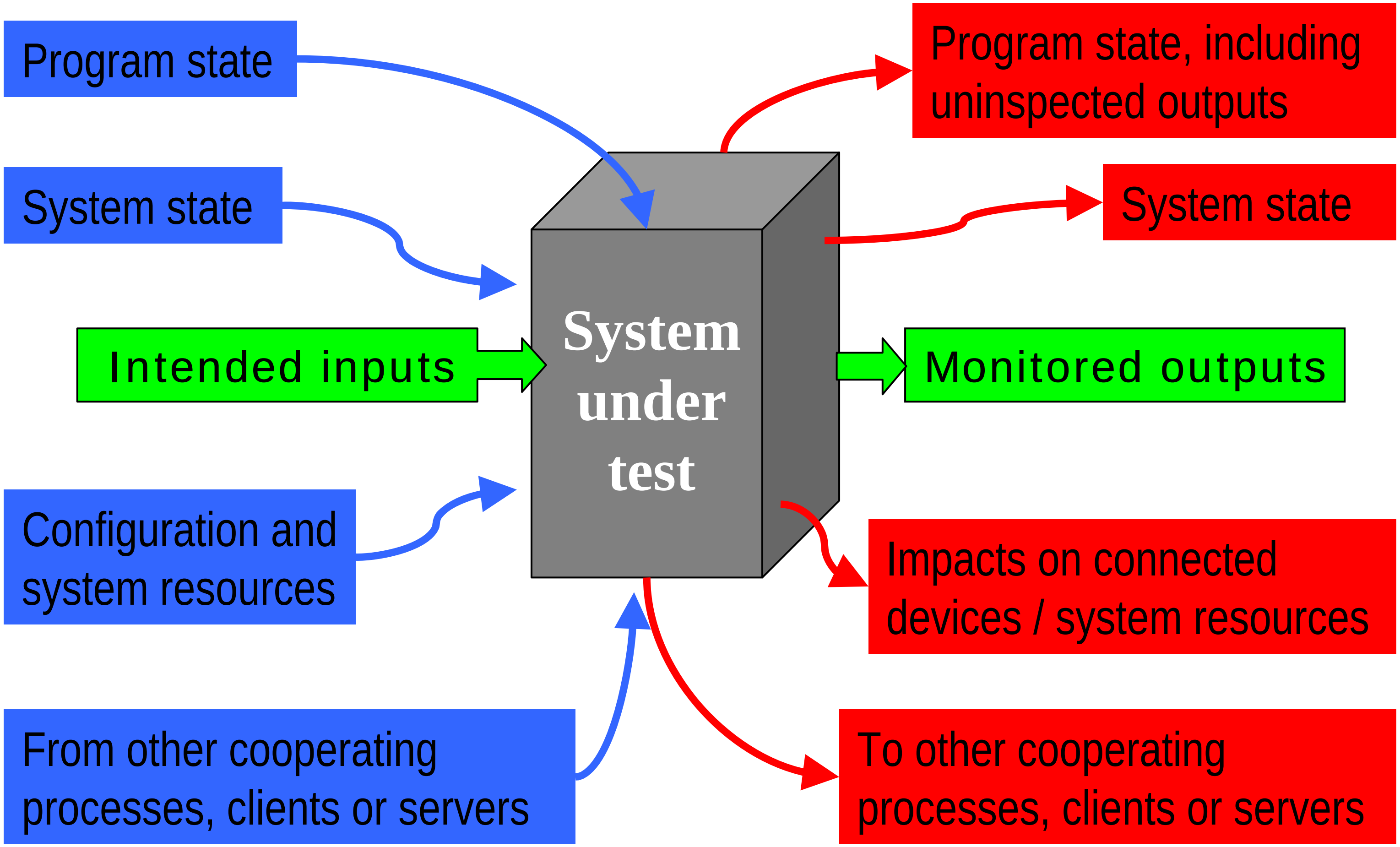
- Are you sure about what you think you *know*?
  (about the requirements, system, ...)
  - Cognitive Psychology – insights into problem solving
  - Epistemology – insights into nature of knowledge
- Are you sure about what you think you *saw*?
  - **Inattentional Blindness** affects everyone
  - See short video: DoTheTest (online) or Quirkology 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.

(From James Bach, james@satisfice.com)

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



Program state

System state

Intended inputs

Configuration and system resources

From other cooperating processes, clients or servers

System under test

Program state, including uninspected outputs

System state

Monitored outputs

Impacts on connected devices / system resources

To other cooperating processes, clients or servers

(© Cem Kaner, kaner@kaner.com)

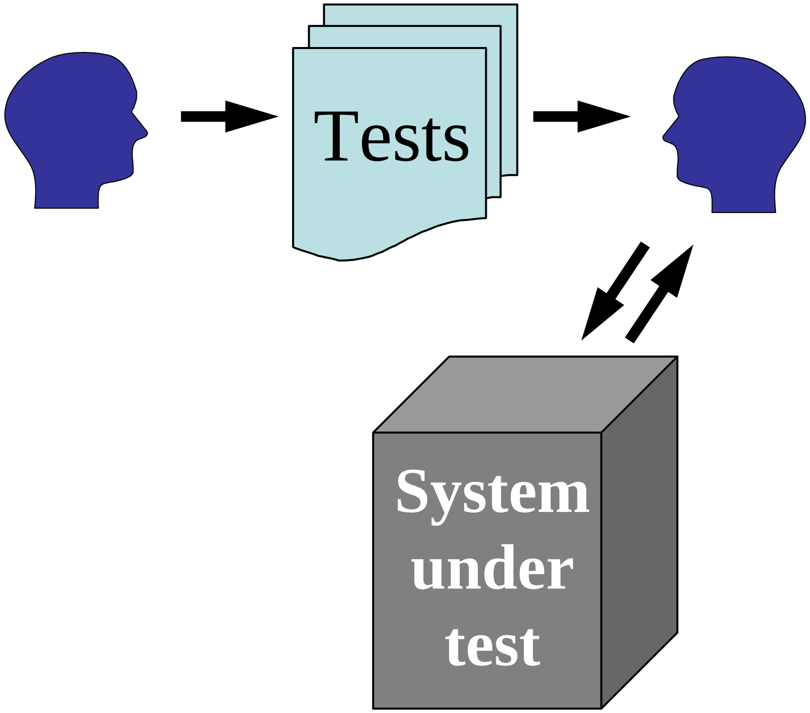
9. Diversity of Approaches

- Many different ways of approaching Testing
- Bret Pettichord: 5 ‘*Schools*’ or Views of Software Testing today:
  - Analytical
  - Standards
  - Quality
  - Context-Driven
  - Agile
- The Great Debate: *Scripted* vs *Unscripted* testing

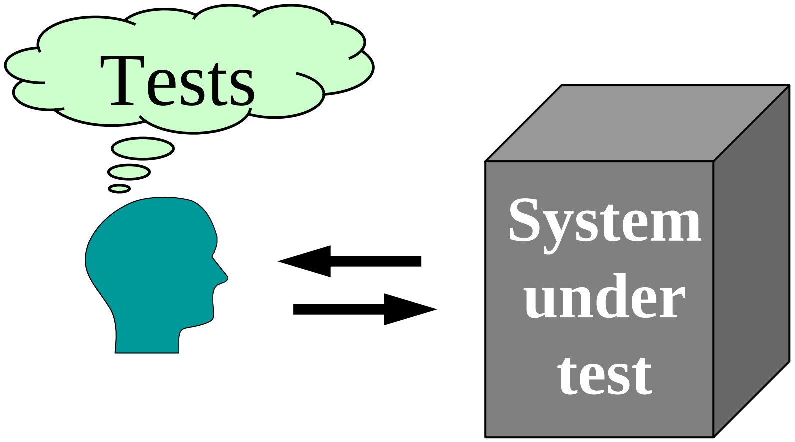
Comparing Testing Approaches

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

4. Convenience

- BlackBerry email outage (April 2007, Feb 2008)
- Air Canada's computer reservation system bug delayed flights and affected thousands of passengers across Canada (November 2007)

So, What is a Brand?

- Not just Marketing. It's not a logo.
- Can you articulate it?
- Do you deliver on your promises?
- What do your customers say?

What it is:

The sum of all the *experiences* someone will go through when interacting with Testers, Software Testing or the QA Function.

What is the Testing Brand?

- What *experience* do people have with...
    - Testers? Software Testing? Poor Quality systems?
- What do they see? hear? get?
    - Managers, Project Leads, Developers, ...
    - Customers
    - General Public, Media (News, print, blogs, etc.)
    - Educational Institutions (bad/incomplete/no info)
- What do others learn about it from ***you***?
    - What you tell others? How you act / interact?
    - What value you *deliver* (i.e. the bottom line)

Building Differentiation (Convert)

- It’s all about better knowing your target audience’s needs and how you can cater to them!
- Do you know the specific expectations of each of your target audiences?
- How does your Value Proposition deliver toward their specific expectations?
- What’s your Brand Experience for these people?

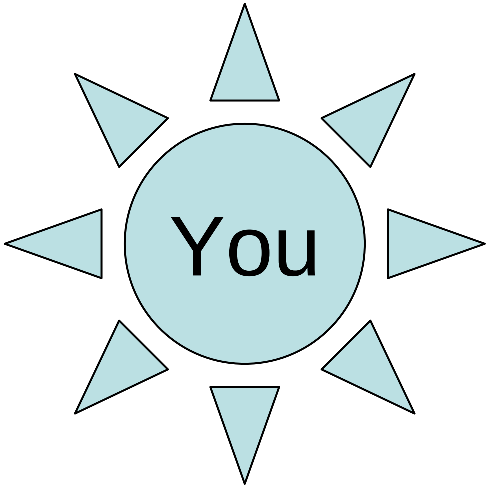
Target Audiences

Educational Institutions

Senior/Executive Management

Other Managers & Leads

Developers



You

Project Managers

Other Testers/QA !!!

Customers

General Public & Media

(e.g. TV, News, Movies, Friends, Family, Social Networks, etc.)

Other Stakeholders

Print Media:
Books, Blogs, Journals, articles, etc.

Re-Branding:

Deliver a Positive Brand Experience

- Be Professional
- Don’t stop learning and growing
- Speak up! Spread the word
- Educate and Inspire others
- Lead by example (actions speak louder than words)

Give everyone a great impression.

Leave them with a Quality experience!