Two Futures of Software Testing

Michael Bolton

DevelopSense

expo:QA

Madrid, Spain

November 2006

Who I Am

Michael Bolton

(not the singer, not the guy in Office Space)

DevelopSense, Toronto, Canada

mb@developsense.com

+1 (416) 992-8378

http://www.developsense.com

Acknowledgements

- James Bach
  - some of the material comes from the Rapid Software Testing Course, of which James is the senior author and I am co-author
- Cem Kaner
- Bret Pettichord
- Jerry Weinberg
- Richard Feynman

Building a Skilled Community is Important!

Thank you to inQA:labs!

Ground Rules

- There are many strong opinions expressed in this presentation
- You are welcome and encouraged to challenge and debate them at any time
  - you can share them, too, of course
- It’s up to you to decide whether to apply these ideas in your context
  - don’t let anyone (other than, perhaps, your boss) tell you what to do

What is Testing?

- Testing is questioning the product in order to evaluate it (James Bach)
  - this is the operational definition that we use when we’re talking amongst skilled testers
- Testing is a process of technical investigation, intended to reveal quality-related information about a product (Cem Kaner)
  - this is the definition that we use when we’re trying to underscore the idea that *testers need skills*

The Four Schools of Testing

- Bret Pettichord, The Four Schools of Software Testing. Presentation at the Workshop on Teaching Software Testing, Melbourne, FL, 2003. This was an important attempt to understand testing cultures and their principles.
- http://www.testingeducation.org/conference/wtst_pettichord_FSofST2.pdf
- Bret is also a very cogent thinker and writer on the subject of test automation. http://www.pettichord.com

The Four Schools of Testing

- The Analytical or Mathematical School
  - formulas and flowcharts and state diagrams will save us.
- The Factory or Process School
  - lots of planning, scripting, and other paperwork will save us.
- The Quality Control or Quality Police School
  - testers are the gatekeepers
  - telling other people how to do their jobs (even when we don’t do ours that way) will save us.
- The Context-Driven School,
  - our skills, our ability to think critically, and our ability to choose practices appropriate to serving the testing mission, will *help* us (not save us)
  - the other schools may have valuable things to teach us
  - nothing will *guarantee* perfect testing

The Dark Future:
The Plan is Everything

- Testing shall be a rigourously planned and controlled process
  - all planning and tests prepared in advance
  - test only according to documented requirements
  - all testing is verification testing
- All projects shall take two years (minimum)
- All requests for change politely refused

If the context doesn't fit the plan,
change the context to fit the plan.

The Dark Future: Change is Controlled

- Nothing is more important than following our process strictly
  - our clients will understand, of course
  - if they have change requests, they should have known from the beginning
- By insisting that requirements don’t change, we can eradicate risk

The Dark Future:
Remove the Human Element

- By eliminating the human element, we can eliminate variability, uncertainty
- Testing doesn’t require skilled labour
- All testers have the same (limited) skills, but skill doesn’t matter for clerical work
- All tests must be scripted

The Dark Future: Automation is King

- Humans are too fallible to detect defects
- We must slow down development to let automation catch up
- Ad hoc testing is banned
- Exploration and investigation are luxuries that we cannot afford

The Dark Future: Testers Own Quality

- Testers are the quality gatekeepers
- Testers will be isolated from developers
- Management can count on its testers to assure quality
- We measure test coverage by counting test cases
- We judge testers by counting bugs

The Dark Future: Pathologies

- Places knowledge and learning up front, at the beginning of the project
  - when we know the *least* about it!
- Thinking and learning through the project are ignored
- Treats testing as unskilled work
- Machines are trusted; human cognition is devalued
- Testers implicitly run the project *when it’s convenient*
  - even though they’re powerless, they get blamed for lapses
- If there are problems, it’s because the testers should have better requirements or better bosses
  - you naughty testers!

The worst thing about the dark future is that it’s so much like today.

The Bright Future: Skill Is Central

- Tester skill is at the centre of testing
- Important skills include
  - critical thinking – recognizing bias and thinking errors
  - general systems thinking – coping with complexity
  - context-driven thinking – coping with changing situations
  - scientific thinking – designing and performing experiments
  - cognitive skills – learning and using lots of observational modes
  - programming
    - but not necessarily for all testers

What IS Quality?

> *Quality is value to some person.*
>
> *---Jerry Weinberg*

- Value is what someone will do (pay) to have their requirements met
- Quality is subjective
- Decisions about quality are political decisions
  - who has the power and authority to make them?

The Bright Future: Testing is a Service

- The primary role of testing is usually to provide quality-related information to management
- Testing provides services for the rest of the project community
- Testers do excellent work
  - with insufficient or barely sufficient information
  - under extreme time pressure
  - with the tools that are available (or that they develop quickly)
  - in a way that withstands scrutiny
  - *even when everyone else is breaking the rules*

The Bright Future: Information In Context

- Testers are expert communicators
- Testers identify and frame context
- Test managers encourage management to reject deceptive quantitative measures
  - example: counting test cases
  - example: counting bugs
- Testers seek alternative explanations and perspectives
- Testers are skeptical
  - but not cynical

The Bright Future: Test Activities

- Testers *actively question the product* in order to evaluate it
- Testers *focus on risk*
- Testers *continuously develop new questions*
- Testing is *investigative*, as well as confirmatory
  - in the agile model, developers handle the bulk of the confirmatory testing effort at the unit level
- Testers develop skill in exploratory testing
  - parallel test design, test execution, and learning

The Bright Future: Change Happens

- Testers expect and embrace change
- Confirmatory testing serves as a change detector
  - We would ***love*** it if the developers embraced unit tests and TDD
  - …but that’s their business (and management’s), NOT ours
- Investigative testing changes rapidly depending on what is being investigated, so test artifacts are
  - lightweight, where appropriate
  - efficient (leaving us more time to run tests)
  - produced in service of a genuine purpose
  - not clerical or bureaucratic
  - not overinvested
  - not wasteful

The Bright Future: Machines Do Mechanical Work

- Automation *assists* the testing effort
  - machines do high-speed, simple-oracle tasks
  - *people* do high-skill, high-cognition tasks
- Testers do not work from scripts
  - testers do excellent work with limited guidance
  - where recording is important, keep records
- Testers work from more than requirements documents
  - testers apply inference, conference, and reference heuristics
  - testers are general-systems thinkers

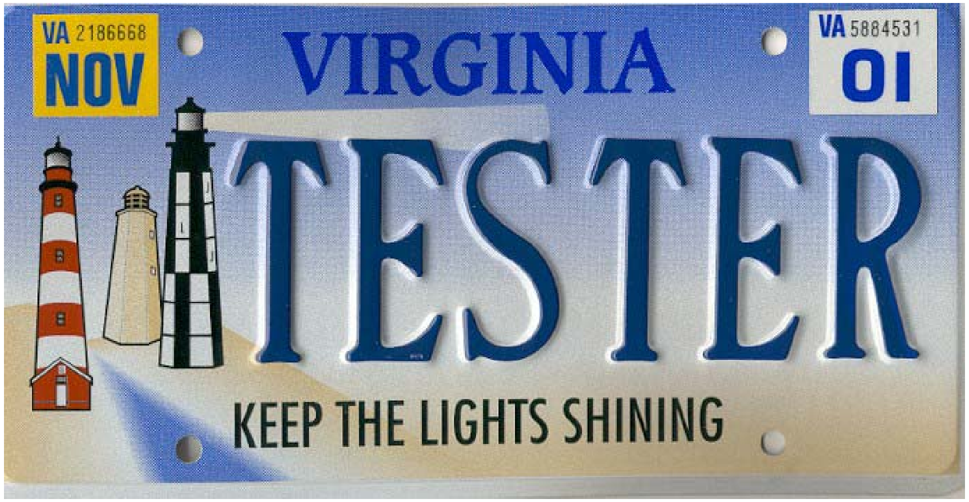
The Bright Future: Testers Collaborate

- Many (most) testers are co-located with developers
- Testers provide extremely rapid feedback
- Testers and developers collaborate on testability
  - controllability (scriptable interfaces, easy reconfiguration)
  - visibility (on-screen status, log files)
- Many testers learn to program
  - many, but not all
  - some testers learn regular expressions and spreadsheet skills
  - test scripting is flexible and lightweight

The Bright Future: Testers Focus on the Mission

- If the mission *requires* lots of documentation and data, we supply it
  - but we regularly check to make sure that it's adding value
  - I guarantee that someone will leave this presentation claiming that I advocate no documentation, ever; I *don't* advocate that
- If the mission *requires* lots of automated testing, we develop it
  - but we don't stop manual exploratory tests
  - we don't make it rigid
- If the mission *requires* us to suspend our skills, we do
  - on the understanding that someone else is responsible for the quality of our work
- Testing is NOT *mythodology*

Testers Light The Way



This is our role.

We see things for what they are.

We make informed decisions about quality possible, because we think critically about software.

We find out how the software really *works.*

Learning More: Finding Bugs

- Lessons Learned in Software Testing
  - by Cem Kaner, James Bach, and Bret Pettichord
- Testing Computer Software
  - Cem Kaner, Jack Falk, and Hung Quoc Nguyen
- How to Break Software
  - Whittaker
- How to Break Software Security
  - Whittaker and Thompson
- Testing Applications on the Web
  - Hung Quoc Nguyuen
- Hacking Web Applications Exposed
  - Joel Scambray and Mike Shema

Learning More: Testing Philosophy

- The Pleasure of Finding Things Out
  - Richard Feynman. Read his Appendix to the Challenger Report.
- Surely You're Joking, Dr. Feynman! Adventures of a Curious Character
  - Richard Feynman
- What Do You Care What Other People Think?
  - Richard Feynman
- Quality Software Management Vols. 1 – 4
  - Jerry Weinberg
- *Anything* by Jerry Weinberg

Learning More: Other Wonderful Stuff

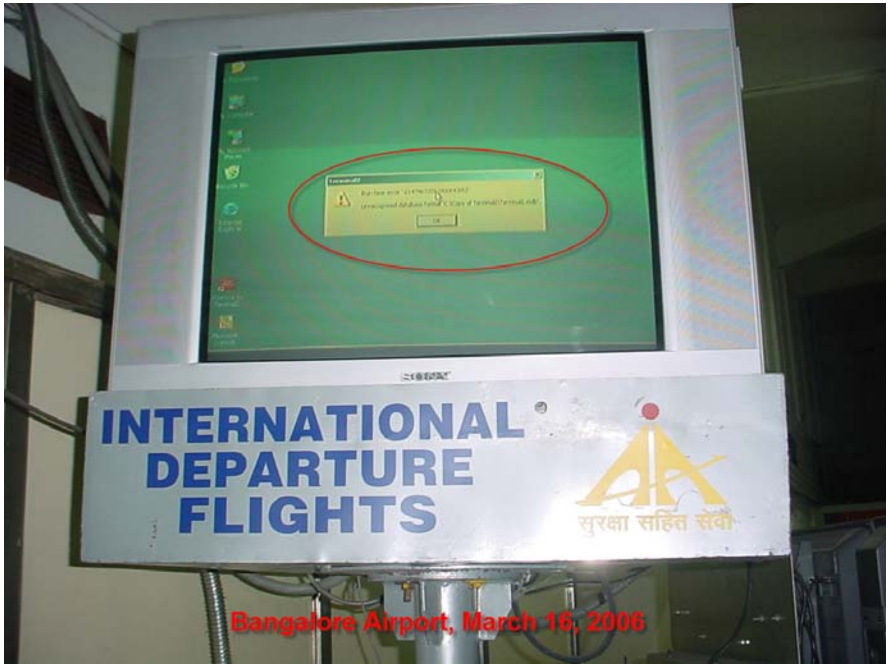
- Please Understand Me
  - David Kiersey
  - The Myers-Briggs Type Inventory, which provides insight into your own preferences and why *other people* seem to think so strangely
- The Visual Display of Quantitative Information
  - Edward Tufte
  - How to present information in persuasive, compelling, and beautiful ways
- A Pattern Language
  - Christopher Alexander et. al
  - A book about architecture
  - even more interesting as a book about thinking and creating similar but unique things—like computer programs and tests for them
- Better Software
  - a most unfortunate name of a most wonderful magazine
- The Amplifying Your Effectiveness Conference
  - held every November in Phoenix, AZ
  - See http://www.ayeconference.com for details

Learning More: On the Net

- StickyMinds http://www.StickyMinds.com
- Risks Digest http://catless.ncl.ac.uk/risks
- Cem Kaner http://www.kaner.com
- James Bach http://www.satisfice.com
- Michael Bolton http://www.developsense.com
- The Florida Institute of Technology
  - http://www.testingeducation.org
  - http://www.testingeducation.org/BBST/index.html

Our Work Is Never Done...

Mumbai Airport, March 15, 2006



Bangalore Airport, March 16, 2006

Who I Am

Michael Bolton

(not the singer, not the guy in Office Space)

DevelopSense, Toronto, Canada

mb@developsense.com

+1 (416) 992-8378

http://www.developsense.com