



## Two Futures of Software Testing

**Michael Bolton**  
**DevelopSense**

*PNSQC*

October, 2007

## Who I Am



Michael Bolton

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

DevelopSense, Toronto, Canada

[mb@developsense.com](mailto: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

## 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.

## What is Testing?

- Testing is a process of
- empirical
- technical
- investigation
- done on behalf of stakeholders
- with the intention of revealing
- quality-related information
- of the kind that they seek

--- Cem Kaner

This is the definition that we use when we're trying to emphasize that *testers need skills*.

*These definitions are not intended to disagree.*

## 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.testineducation.org/conference/wtst\\_pettichord\\_FSofST2.pdf](http://www.testineducation.org/conference/wtst_pettichord_FSofST2.pdf)
- Bret is also a very cogent thinker and writer on the subject of test automation; maintainer of WATIR
- <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 rigorously planned and controlled process
  - all planning and tests prepared in advance
  - we test only according to documented requirements
  - all testing is verification testing
- All projects shall take two years (minimum) to give us time to plan
- All requests for change politely refused

### **The Dark Future: Change is Controlled**



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

## The Dark Future: Measurement



- We measure
  - requirements scope by counting requirements
  - developer output by counting lines of code
  - complexity by counting code branches
  - test coverage by counting test cases
  - product quality by counting bugs
  - the value of testers by counting bug reports

## The Dark Future: Putting The Testers In Charge



- Testers are the quality gatekeepers
- Testers refuse to test until they have been supplied with complete, unambiguous, up-to-date requirements documents
- Testers “sign off” on project readiness
- Testers can block releases

### **The Dark Future: Promoting Orthodoxy**



- Testers must be certified
- Testing doesn't require skilled labour
- All testers have the same skills
- Testers must be isolated from developers
- All tests must be scripted
- Ad hoc testing is banned; variation suppressed
- Exploration and investigation are luxuries that we cannot afford

### **The Dark Future: Automation is King**



- Humans are too fallible to detect defects
- By eliminating the human element, we can eliminate variability and uncertainty
- Sure, automation takes time and effort to prepare, therefore...
- ...we must slow down development to let testing (particularly automation) catch up

## 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
- Measurement is riddled with basic critical thinking errors
  - primarily reification error and rotten construct validity

## The Dark Future: Pathologies

- Testers implicitly run the project *when it's convenient* for management to let them
- Even though testers are essentially powerless, they *do* get blamed for lapses
  - even though bugs have been created by others
  - even though bugs are hidden
- When testers fail, it's because
  - they should have better requirements,
  - they should have told their bosses and developers what to do

**You naughty testers!**



**The worst thing about  
the dark future is...**

***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
  - writing, recording, and reporting
  - diversification of skills and tactics
  - rapid learning
  - programming
    - but not necessarily for all testers

## What IS Quality?

*Quality is value to some person.*


*---Jerry Weinberg*

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


## The Bright Future: Testing is a Service

- Testing provides services for the rest of the project community
- The primary role of testing is to provide quality-related information to management
- 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 stands up to scrutiny
  - *even when everyone else is breaking the rules*
- *Management* makes management decisions


## The Bright Future: Information In Context

- 
- Testers are expert communicators
  - Testers identify and frame context
    - to make sure that testing matches the mission
  - Testers seek alternative test approaches, explanations, and perspectives
    - to reduce the possibility that we'll miss important problems
  - Testers are skeptical (but not cynical) empiricists


## 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
    - as in the agile model, developers handle the bulk of the confirmatory testing effort at the unit level...
    - where automation is inexpensive and feedback is immediate
  - 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
    - ...because it would allow us to obtain greater coverage in less time
    - ...but ultimately that's their business (and management's), NOT ours
  - Investigative testing changes rapidly depending on what is being investigated and discovered, 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 concise 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 (even 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
  - most testers learn regular expressions and spreadsheet skills
  - test ideas are captured using flexible and lightweight tools

## The Bright Future: Narratives vs. Numbers

- Testers compose, edit, and narrate cogent stories
  - about the product
    - how it can work
    - how it might fail
  - and about their testing
    - what they did, and why
    - what they didn't do, and why
    - and why the testing was good enough
- Test managers encourage management to reject deceptive quantitative measures
  - example: counting test cases
  - example: counting bugs
- Testers don't supply numbers without a story

## 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, and this is the **documented** proof
- If the mission *requires* lots of automated testing, we develop it
  - but we don't stop brain-engaged exploratory tests
- 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

## 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 let management make those decisions.*

## Testers Light The Way



*We also think critically about our own work.*

*We question our context and our choices,  
both of which evolve over time.*


*We question testing folklore.*

**Testing is NOT methodology.**


## 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 About 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>