



Running with the Test Revolutionaries

An experience report by
Paul Szymkowiak
paul@szymkowiak.com

Bio: Paul Szymkowiak

◆ Career Profile

- 18 years in software development. Work in New Zealand, Asia Pacific, Canada, US , UK.
- Independent Test consultant '94 to '97
- Rational Software '97 to '04
 - Field consultant '97 to '00
 - Rational Unified Process (RUP) developer '00 to '04

◆ Focus Areas

- **Coaching Iterative Development teams**
 - Skill transfer in Project Management, Testing & Test Automation, Requirements
- **RUP adoption and customization**
 - Organization Assessment, Tailoring, Implementation Support, RUP Tools.

Agenda

◆ Setting the scene

- Testing: some basic issues
- Established Notions
- Changes in Software Development
- Signs of a revolution

◆ Changes in Software Testing

- An overview of test community work
- A quick look at some new thinking

◆ Discussion

Basic Testing Issues

◆ What is Testing?

- "Running new or revised programs to determine if they process all data properly" Computer Desktop Encyclopedia
- "To subject to a procedure that ascertains effectiveness, value, proper function, or other quality" American Heritage Dictionary
- "Questioning a product for the purpose of evaluating it" Bach
- "A technical investigation of a product, an *empirical search* for *quality-related information* of *value to a project's stakeholders*." Kaner

◆ Why is Testing difficult?

- How do you quantify the different ways in which a given program can behave?
 - AND given the increasing complexity of Hardware and Software?
 - ... AND given the increasing sophistication/ demands of product users?

Basic Testing Issues

◆ Why is Testing difficult? (continued)

- When can you start?
 - "Testing" Interim Artifacts through inspection and review?
 - Run tests against the first executable software build?
- When are you finished?
 - Out of Time? No more bugs? Met coverage goal? No more test ideas?
- How do you isolate, reproduce and effectively communicate problems?

◆ A couple of general truths

- You can't test everything
- You can never know that you've found all the bugs
(You can prove the presence of bugs, but not the absence of them)

Established Notions

- ◆ "Testing suffers everyone else's poor work. If only everything else were done well, good testing would be possible!" (A common conclusion testers reach)
 - The Project Manager should *use a different process*
 - The Project Manager should *provide more time* in the plan
 - The Analysts should *document the requirements* better
 - The Developers should *unit test & debug their code* better
 - The DBA should *provide and maintain a test database*
 - The Release team should *create and document builds* better
- ◆ In your project, these points may be valid observations...

Established Notions (Stop the World, I want to get off!)

- ◆ ...and wanting the world to be better isn't a bad thing. However, the problem with testers thinking this way is:
 - *Testers aren't empowered* to make or affect these changes.
 - *Testers alienate themselves* from the rest of the team:
 - Testers are seen as spectators/ commentators – not active participants and stakeholders in the software projects success.
 - **OR** Testers are seen as "Quality Police" – some testers actively seek this role, wanting to "own" all aspects of product quality. You can't own the responsibility for quality in someone else's work.
 - *Some Testers refuse to work* until deficiencies in other areas are addressed.
- ◆ Improvements in these areas might help, *but do they solve the fundamental problem of testing?*

Established Notions: Reality Check

- ◆ Even when the planets aligned, the traditional testing process guidance (e.g. IEEE829) didn't serve testers well:
 - If you *started documentation* of test artifacts *early*:
 - Your tests often needed significant rework before test execution.
 - You spent significant time in test documentation maintenance.
 - You used up much of your budget in documentation, not execution.
 - If you *started documentation* of test artifacts *late*:
 - You had insufficient time to document all your tests
 - You had insufficient time to execute the tests and report bugs.
 - Any feedback on critical problems came too late to be economically addressed by the development team.

Changes in Development

"Hey, no fair! Somebody went and changed the rules!"

- Shifts in software market dynamics:
 - Birth of consumer software market
 - Cyclic reduction in hardware and software costs
 - Increasing user numbers, sophistication and expectations
 - Software becomes increasingly critical to business survival and success
- Software industry makes moves to improve efficiency
 - Iterative Development: Spiral Model (Boehm), RAD, RUP & Agile Development.
 - Advances in Software Development technology: support tools, languages etc.
- Software Testers reacted by:
 - Most kept complaining about the same old concerns (and some new ones)
 - Some saw that existing processes, skills and techniques weren't 100% applicable
 - A few saw potential opportunity to make improvements to testing efficiency

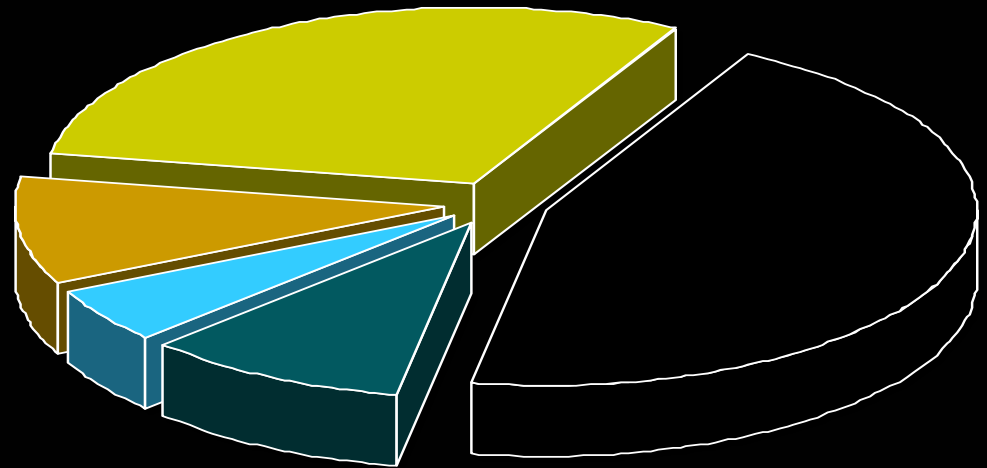
Changes in Development: New test challenges

- ◆ Iterative Development addresses a number of problems, but introduces some new test issues:
 - Project artifacts are available earlier but intentionally incomplete: they continue to evolve (no artificial, premature "completion" and "sign-off").
 - Focus shifts from interim artifacts to the executable code as a real objective measure of progress and quality.
 - Software builds occur earlier and often throughout the lifecycle.
 - Developer tool improvements reduce the time and increase the number of builds.
 - Early project focus is on proving key architectural elements, late focus is on achieving completeness.
 - Increasing market pressures on commercial and business software leads to increasingly compressed project timetables

Signs of a Revolution: What Solutions are there?

Different groups saw different solutions:

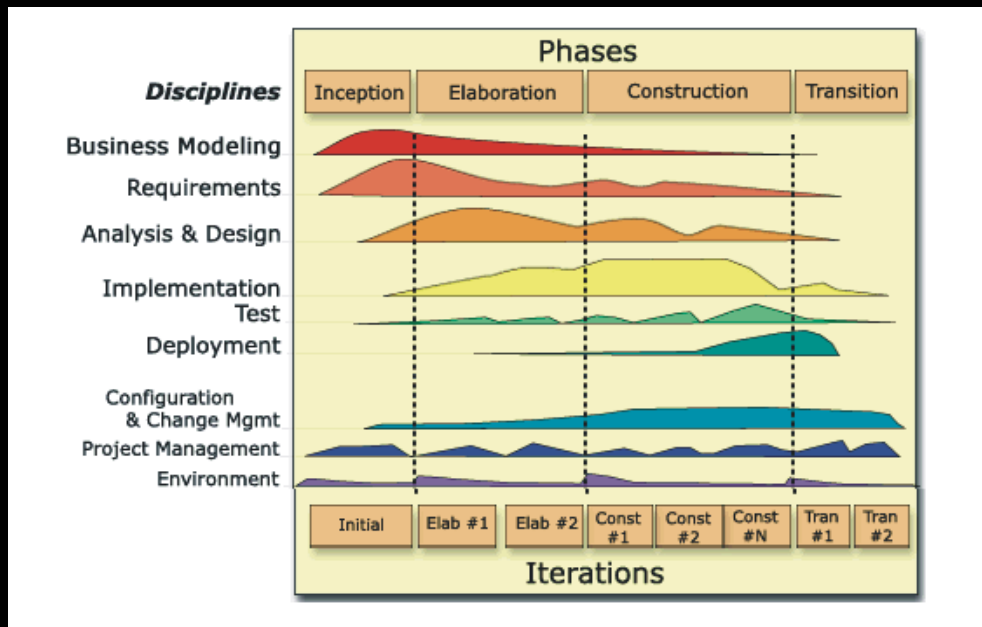
- "How do testers work effectively in iterative development?"
- "Can testers develop a context-driven approach?"
- "Can developers and customers own the testing?"
- "Can we shoe-horn our existing test process into iteration time boxes?"
- "What was the question?"



Test Revolution: Iterative test approach?

How do testers work effectively in iterative lifecycles?

- What test strategies, techniques and tools do they use?
- Perhaps ROP (Rational Objectory Process) addresses this?
 - (... no, not really ...) Existing content based largely on SQA Process
 - '97-'99 Collection of iterative test experience within Rational Software



(... if I were King for a day ...)

Test Revolution: Context-driven Testing?

- ◆ How do you know what contexts your skills can be applied in?
- ◆ Is there a core set of skills that testers could apply in various contexts?
- ◆ Can testers be "Context Driven" and determine when to apply context-specific skills?
- ◆ Can testers develop testing strategies that are flexible in the face of project process inconstancy?



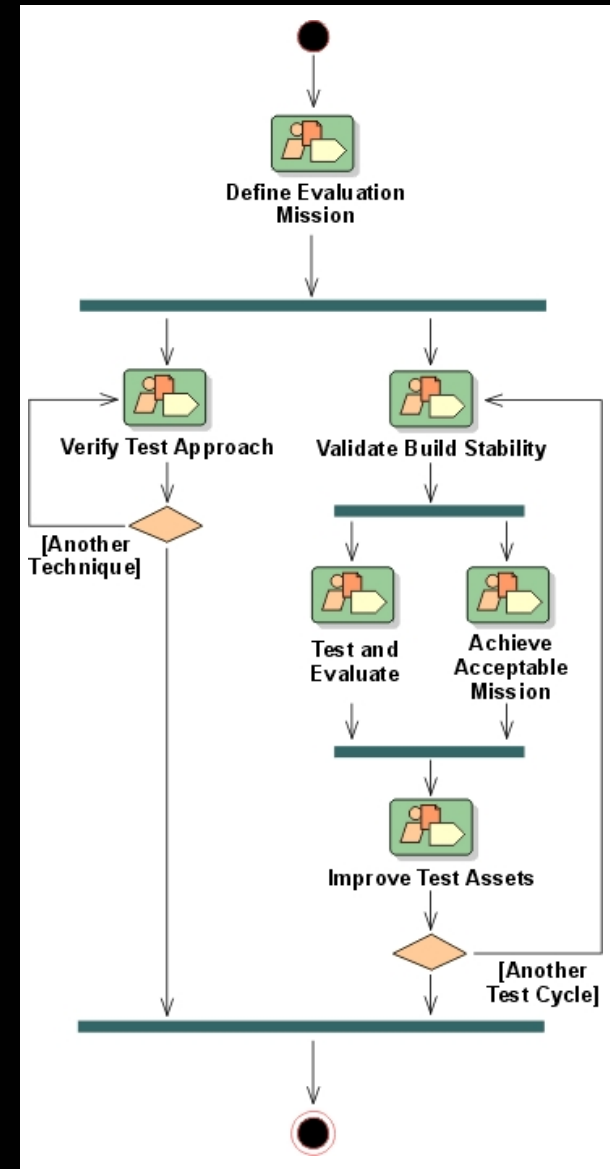
Agenda

- ◆ Setting the scene
 - Testing: some basic issues
 - Established Notions
 - Changes in Software Development
 - Signs of a revolution
- ◆ **Changes in Software Testing**
 - An overview of test community work
 - A quick look at some new(?) thinking
- ◆ Discussion

Changes in Test: Enhancements to RUP

(sometimes you get what you wish for ...)

- '99-'00 Critical Mass achieved for changes to RUP (Rational Unified Process)
- In '00, I joined the RUP team in Vancouver, Canada. My mission was to improve testing elements and structure within RUP to better support iterative development:
 - Consolidated the internal opinion and experience within Rational Software.
 - Canvassed the broader test community for applicable experience and practices.
- Major RUP development effort in 2001
- Test improvements released in RUP 2002



Changes in Test: Context-Driven Testing

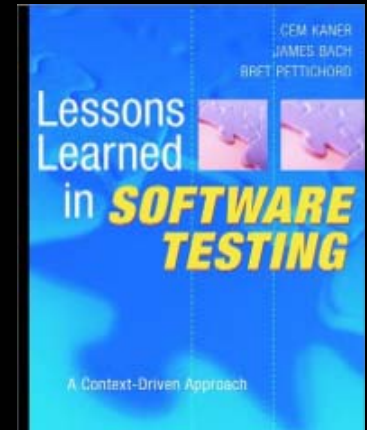
◆ Key philosophies:

- Testing provides an information service to the software development team.

◆ See <http://www.context-driven-testing.com/>

■ Seven Basic Principles of the Context-Driven School:

- The value of any practice depends on its context
 - There are good practices in context, but there are *no best practices*
 - People, working together, are the most important part of any project's context
 - Projects unfold over time in ways that are often not predictable
 - The product is a solution. If the problem isn't solved, the product doesn't work
 - Good software testing is a challenging intellectual process
 - 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.
- A look at some key community personalities ...



Changes in Test: Cem Kaner J.D, Ph.D.

- ◆ Respected academic with practical focus and experience.

- ◆ Key positions/ Ideas:

- Better tester education; accountability for bad software
- Testing is a skilled, Intellectual undertaking; Testing isn't QA.
- IEEE doc standards are damaging; Bug Advocacy.

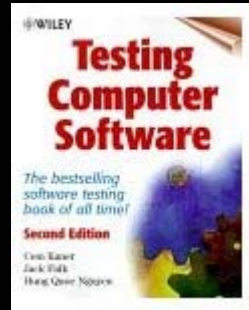


- ◆ Current Position:

- Professor of Software Engineering at FIT, Director of FIT CSTER
- Teaches courses on testing, software measurement, empirical research methods, and computer law and ethics

- ◆ Publication Highlights:

- Testing Computer Software 2nd Ed.(Kaner et. al. 1993)
- Bad Software (Kaner, Pels 1998),
- Lessons Learned in Software Testing (Kaner, Bach, Pettichord 2002)



- ◆ Website: <http://www.kaner.com>

Changes in Test: James Bach

◆ Key positions/ Ideas:

- Good Enough Quality/ Testing; Test Missions
- Socratic & Cognitive Thinking
- Exploratory Testing; Rapid Testing

◆ Current Position:

- Founder of Satisfice, Inc., a test training and consulting company

◆ Publication Highlights:

- Lessons Learned in Software Testing (Kaner, Bach, Pettichord 2002)
- Better Software, STOE, IEEE Computer ('95 - '99)

◆ Website: <http://www.satisfice.com>

◆ "You are the headlights of the project"



Changes in Test: Brian Marick

- ◆ Focus on practical experience rather than theory
- ◆ Key positions/ Ideas:
 - Programmer Testing; XP-Agile practices; test patterns
 - Context-driven test strategies; test-idea catalogs
 - Coauthor of the Manifesto for Agile Software Development
- ◆ Current Position:
 - Founder of Testing Foundations, a test consulting company
- ◆ Publication Highlights:
 - The Craft of Software Testing (Prentice-Hall,1995)
 - Better Software, STQE (Editor/ Author), Contributor to RUP content
- ◆ Website: <http://www.testing.com>
- ◆ "Project documents are interesting fictions: Useful, but never sufficient"



Changes in Test: Some other interesting folks

◆ Bret Pettichord

- Test Automation expert, Agile Development & Testing, Testing hot list
- <http://www.pettichord.com>

◆ James Whitaker

- Books: How to Break Software (2002), How to Break Software Security (2003)
- Holodeck – Dynamic runtime fault injection and testing tool
- Associate Professor at FIT <http://www.cs.fit.edu/~jw/>

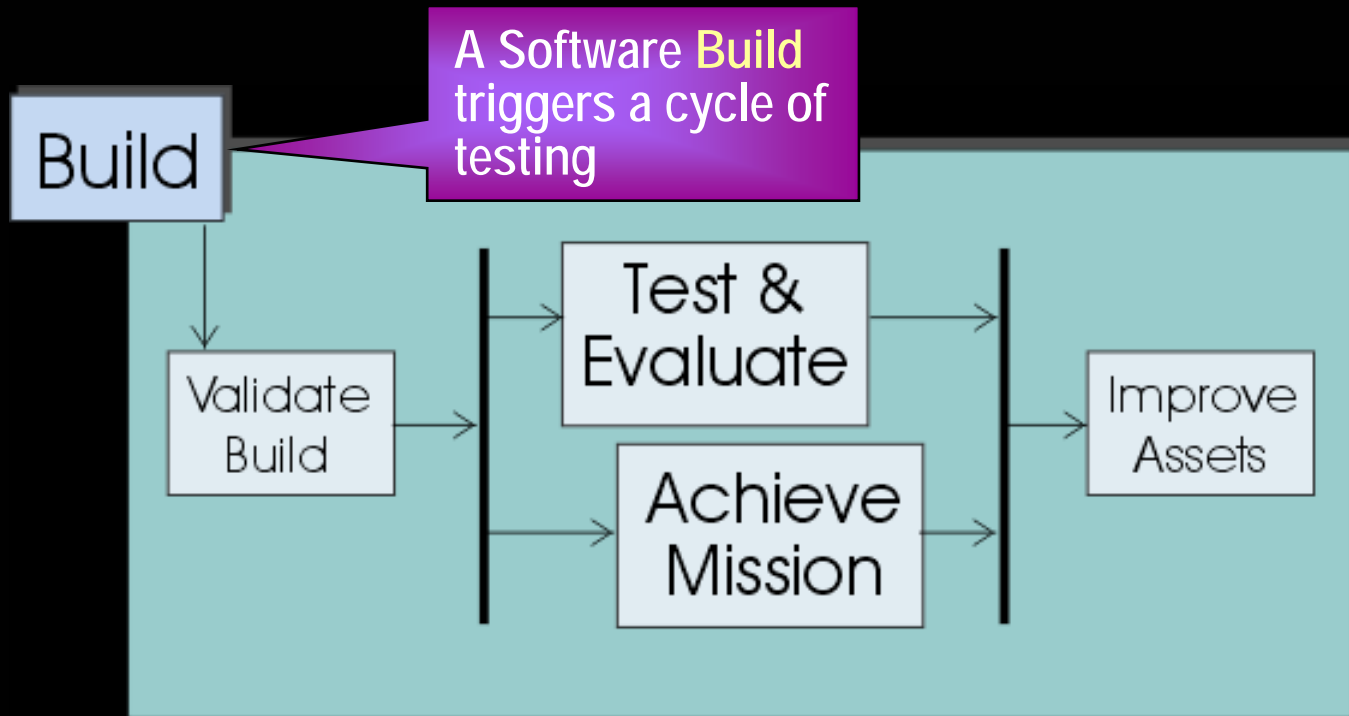
◆ Hans Buwalda

- Soap-Opera Testing, Automation with Action Words
- <http://www.logigear.com>

◆ Elisabeth Henderickson

- Bug Hunter, Creative Software Testing
- <http://www.qualitytree.com>

New Thinking: Test in each Iteration



New Thinking: Good Enough Quality/ Testing

- ◆ "Good Enough testing is the process of developing a sufficient assessment of quality, at a reasonable cost, to enable wise and timely decisions to be made concerning the product. [...] I'm asserting that, at a high level, it's useful to think about the value of testing as a dynamic with four parts:
 - Assessment of product quality (How accurate and complete is it?).
 - Cost of testing (How reasonable is it? Is it within project constraints? Is there a good return on investment, such as the extent of information gained per test?).
 - Decisions (How well does the assessment serve the project and the business?).
 - Timing of all the above (Is it soon enough to be useful?)."

Bach, A Framework for Good Enough Testing, IEEE Computer Oct 1998

New Thinking: What is the Mission of Testing?

- ◆ Given that you can't test everything, how do you focus the test activity? What goal(s) will drive the trade-off decisions for the test effort?
 - Verify a specification (Requirements, User-Manual Claims)?
 - Find Bugs (as many as possible, important ones fast)?
 - Advise about perceived Quality?
 - Assess fitness for use?
 - Certify to a standard?
 - Satisfy stakeholders?
 - Fulfill process mandates?
- ◆ Different Test Teams have different Missions.

New Thinking: Exploratory Testing

- ◆ "Any testing to the extent that the tester actively controls the design of the tests as those tests are performed and uses information gained while testing to design new and better tests." James Bach
- ◆ Some common characteristics of practitioners views about exploratory testing:
 - Interactive
 - Concurrence of cognition and execution
 - Creativity
 - Drive towards fast results
 - De-emphasize archived testing materials

New Thinking: Contrasting Test Communities

◆ The Analytical School

- Predominant focus on finding "The Best" Techniques and Tools.
- "Testing is the math problem I know how to solve within a specified micro-world. My work could lead to a profound breakthrough... further research is needed."

◆ The Quality Control School

- Predominant focus on Quality Assurance.
- "Testing? I am a quality engineer. Have you heard the good news about prevention? Zero defects is my motto! Who's up for a fishbone diagram? ... Hey, come back!"

◆ The Factory School

- Predominant focus on adherence to Routine Procedures.
- "It is important to minimize process variation by restricting the human factor. Therefore, testing is better when it is expressed and controlled through artifacts according to documented, detailed and unambiguous procedures."

New Thinking: Contrasting Test Communities

◆ The Context-Driven School

- Predominant focus on the Human Factor.
- "Good testing is so challenging to do, and varies so much with the context of the project, that consistently good testing requires that we exploit the human factor, not restrict it. Grow the skills of testers, and give them the authority and resources to solve the problem in the most appropriate ways."

◆ The "Oblivious" School

- Awareness of testing as a task assignment, not a skilled practice.
- "What? We just Test it."

Adapted from "Skilled Testers and Their Enemies", James Bach, Satisfice, Inc., 2004

Discussion

- ◆ Hopefully you've seen some thought provoking ideas (or perhaps you've had your own practices vindicated)
 - Thoughts / Comments?
 - Questions?

Thank You!

Paul Szymkowiak

eMail me: paul@szymkowiak.com

References

- ◆ The Impossibility of Complete Testing, Cem Kaner, Law of Software Quality Column, Software QA magazine Copyright © Cem Kaner <http://www.kaner.com/imposs.htm>
- ◆ Testing: The RUP Philosophy, Paul Szymkowiak & Philippe Kruchten, Feb 2003, The Rational Edge http://www-128.ibm.com/developerworks/rational/library/content/RationalEdge/feb03/RUPphilosophy_TheRationalEdge_Feb2003.pdf
- ◆ The Context-Driven Testing principles <http://www.context-driven-testing.com/>
- ◆ A framework for Good Enough Testing, James Bach, IEEE Computer Oct 1998
- ◆ What is Exploratory Testing, James Bach, Jan 23, 2001, <http://www.stickymind.com>
- ◆ Exploratory Testing and the Planning Myth, James Bach, Mar 6, 2001, <http://www.stickyminds.com>
- ◆ The Nature of Exploratory Testing, James Bach & Cem Kaner, May 7 2004
- ◆ Skilled Testers and Their Enemies, James Bach, Satisfice, Inc., 2004, <http://www.satisfice.com/presentations/skilledtesters.pdf>