



Testing and Tester Evolution -



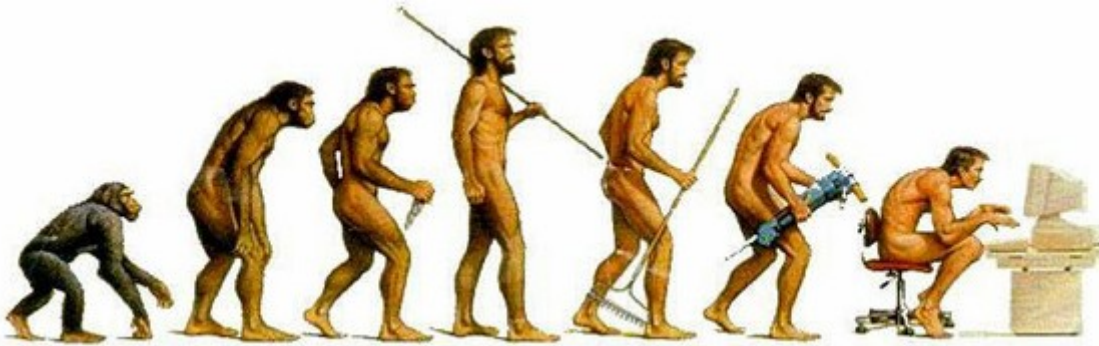
Are we Regressing, Stagnating or Advancing?

Originally Created for:

***TISQA 2006: Tomorrow's Testing -
Agility, Security and Performance Conference
October 5, Chapel Hill, NC***

Scott Barber
Chief Technologist
PerfTestPlus, Inc.





Satire, from the blog of

Christopher Allen

Founder, Alacrity Ventures

Software Testing has been evolving since the dawn of computing as a science, an art and a profession.



Moses receiving the Ten Commandments,

as depicted by Raphael

Software Testing is not bestowed upon us by a mystical deity of testing from atop a mountain.



Software Testing Evolves



More importantly, Software Testing evolves through human choices that are frequently influenced by:

Business Drivers, Profit and Schedules

Fear, Uncertainty and Doubt

Hearsay and Rumor

Incomplete Data

Vendors and Tool Availability

Unrelated or Incomplete Research or Training

Reluctant Testers



And occasionally:

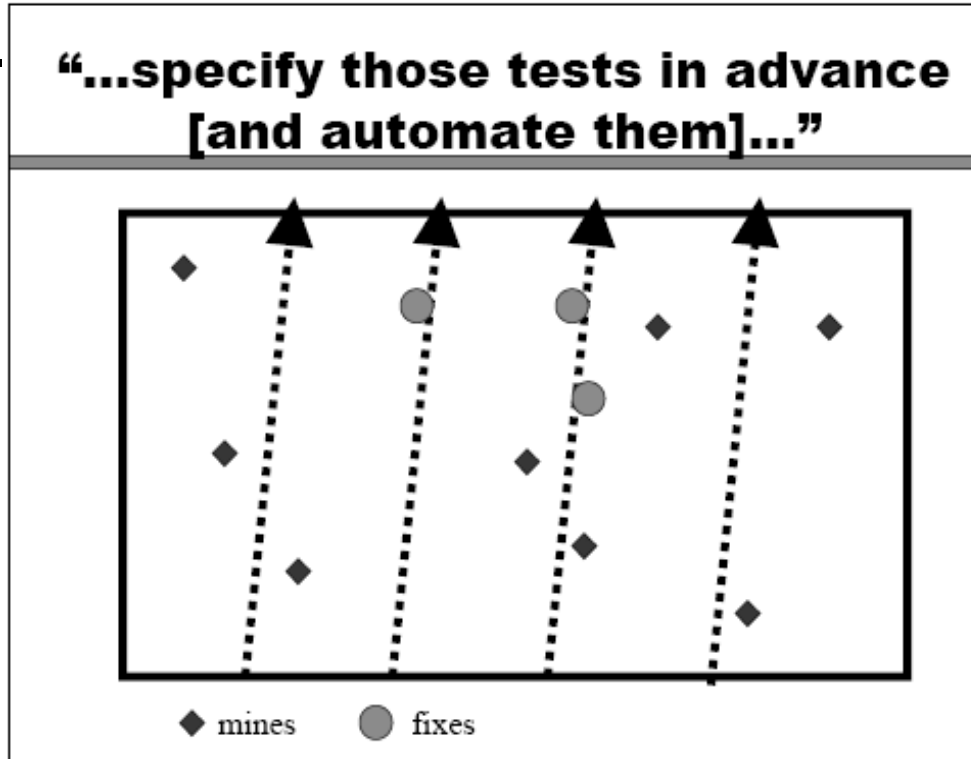
An Altruistic Desire for Better Software Quality



Functional and Integration Testing became a virtual “given” in software development around Y2K.
(Coincidence?)



For example, Automated Regression Testing as a method of determining “goodness” of subsequent versions.



My favorite James Bach slide from:

“Explaining Software Testing To Anyone”

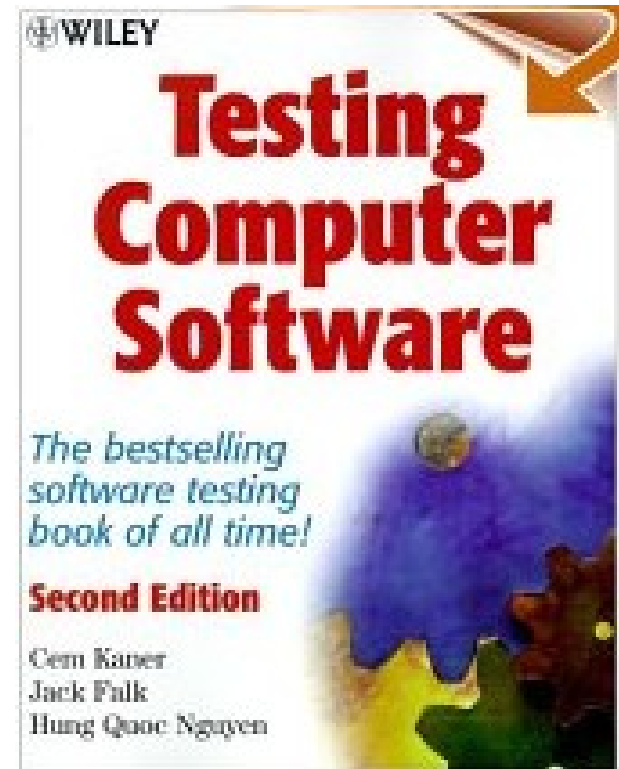
26 years ago, Cem Kaner started writing the first edition of Testing Computer Software, intending to strip away many of the excuses used to justify bad testing; Such as *You can't do good testing:*

without a specification

without reviewing the code

if the programmers keep adding functionality while you test

if you get along too well with the programmers



- Functional & Integration Testing Gain Acceptance
- Testing Tool Vendors Dominate... Then Disintegrate
- Waterfall Falls Out of Favor
- Certification Wars
- Testers Get Techy
- Extreme & Agile Become Cool
- Performance & Security Make Waves
- Test First & Test Driven Infect Development
- OpenSource & Communities Make a Splash
- Top Testers Attend Peer Workshops & Form AST



In about 2001, making test an integral part of the complete development team seemed to reach critical mass in the market.

Many of the integrated teams followed a Waterfall and/or “validate before you ship” model.

Automated regression testing was commonly thought of as the universal answer to the challenge of complete testing.



© 2004 International Institute for Software Testing.



Testing Tool Vendors



1999 marked the start of the rise to power and profitability of the test tool vendors.

Vendors sold their wares as answers as opposed to enablers to “C” level managers who were often impressed with the sales demo.

Prices and maintenance contracts soared, while more and more tools became “shelf-ware”.

Test teams were mandated to make use of the tools, often over their objections about the quality of testing they produced.



MERCURY



Rational
the software development company



Teams realize that, while good in theory, the Waterfall method doesn't handle the inevitability of change and leads to expensive late project changes and delays.

In response, iterative development models like RUP and models with feedback loops like the V-model and STEP become popular...

Unfortunately, many “process improvement” initiatives fail, Waterfall practices persist and these new processes don't deliver as promised.



Vendors, consultants and organizations “come to the rescue” with a myriad of multiple choice tests to “certify” testers.

Once again, a nice theory, the certifications are widely viewed as superficial money making efforts by the certification provider that prove little more than a working vocabulary and/or tool feature familiarity, but demonstrate little correlation to competence.

Good testers get “left behind” because of the time and expense required to keep up with certifications that don't improve their skills.

Even the certifications fail to agree on basic definitions.



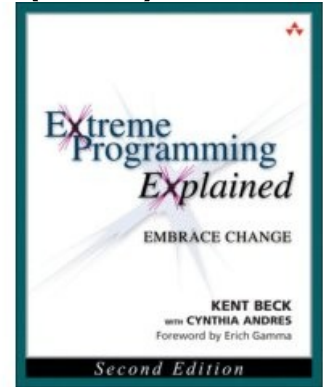
As technologies advance and automated regression testing fails, testers are forced to get more and more technical to find critical defects fast.

Instead of providing opportunities for technical training for testers, many employers opt for Software Development Engineers in Test (SDETs)

SDETs prove technically competent, but frequently under-skilled and under-interested in testing as a craft.



Kent Beck publishes Extreme Programming (XP) Explained, which becomes *the* reference for many development teams.



A group of expert developers and testers, coined the Agile Manifesto.

A group of prominent testers, started the Context-Driven School of software testing based on the principles of the Agile Manifesto.

A group of prominent developers began the Agile Movement and the Agile Alliance.



Unfortunately, XP and Agile downplay the roles of testers who are separate from developers.



Victoria's Secret simulcasts a pay-per-view fashion show on the web, leading to a high profile performance disaster.

9/11, identity theft, phishing scandals and hacked networks make security testing front page news.

Some vendors bring tools to the market to assist non-security experts in testing software for common security flaws.

Still, many development shops continue to accept the risk of minimal performance and security testing with the intent of saving \$.



Testing performance and security is inherently agile.



Test First and Test Driven Development have become common tenets of XP and Agile development.



Both movements minimize the role of non-developer testers outside of acceptance and usability testing.

Most Test First and Test Driven organizations don't see the value in pairing technical testers with developers to enhance the overall completeness and quality of testing.

Many Testers still demonstrate a reluctance to test code directly.



The rise and fall of tool vendors and the increasing capabilities and variety of OpenSource testing tools have led to greater acceptance of their use.

On-line communities are booming across the internet. Both general information and specific tools have their own communities.



On-line communities are dominated by newbies who drive off more senior altruists.

OpenSource tools still bring maintenance and training challenges.



Top Testers have organized and attended nearly 100 peer workshops since Y2K.

The Association for Software Testing was formed in 2005 by some of the top testers in the industry to promote advancement in software testing and a partnership between practitioners, academics and students.





Most software development and testing organizations are reporting that it's "business as usual" with isolated pockets of advancement.

Software testing consultants are reporting noteworthy advancements in a handful of organizations.

Software testing thought leaders are reporting that testing is poised for a significant lurch forward.

However, advancements are contingent on individual testers embracing change and enhancing their technical and testing skills.



The Good:

New publications and conferences are increasing awareness of the challenges and realities of performance and security testing.

Peer conferences and communities are enhancing knowledge sharing across the industry.

Tool vendors are (finally) building tools that enable performance and security testers to do their job well.

More and more stakeholders are asking for performance and security testing and accepting that they are rare and technical skills.



The Bad:

Even with increased awareness, performance and security testing lags about 5 years behind functional and integration testing in frequency and completeness.

Knowledge sharing is limited to a small number of testers and managers.

The new tools are more powerful, but also more challenging to use and are still, well, new.

Many stakeholders are still unwilling to take the time or spend the money to test performance and security the way they know they should.



The Ugly:

Increased awareness does not equate to increased knowledge about **how** effective performance and security testing should be conducted.

Knowledge sharing is great, but good training is hard to find.

New tools, without good training make the lives of the already elite testers easier, but further complicate the lives of everyone else.

Until stakeholders are willing to accept, support and pay for new paradigms in performance and security testing, little is going to change.



My Take (Part 1):

Early adopters are reaping the benefits of advancements today.

The next major tool releases will make most performance and security testers quite happy.

High quality training, books and reference material are a must to assist today's journeymen achieve mastery.

We must earn the trust of stakeholders to allow us to do our job the way it needs to be done.



My Take (Part 2):

Performance and security testers must step out of our comfort zone to learn and apply lessons from:

Human Psychology

Probability and Statistics

Operations Research

Graphical Presentation of Information

Retrospective Data Analysis

Human Factors and Usability



Summary



Testing and Testers are absolutely evolving.

Evolution is not always good.

Performance and security testing is gaining ground on functional and integration testing, but still has a long way to go.

New tools are enabling us to do our job better, but only if we really know what we are doing to start with.

The existence of advancements does not imply that they are commonly in use.

I am enthusiastic about the ongoing evolution of testing







Want More Information?



<http://www.PerfTestPlus.com> (My site)

<http://www.PerformanceTester.com> (Various resources)

<http://www.TestingReflections.com> (QA blog collection)

<http://www.QAForums.com> (Huge QA forum)

<http://www.LoadTester.com> (Good articles and links)





Scott Barber
Chief Technologist
PerfTestPlus, Inc

E-mail:
[*sbarber@perftestplus.com*](mailto:sbarber@perftestplus.com)

Web Site:
[*www.PerfTestPlus.com*](http://www.PerfTestPlus.com)

