

1. Software Testing Hotlist

Resources for Professional Software Testers

Bret Pettichord, editor
Updated 8 December 1999

Contents

- [Test Automation](#)
 - [Using Silk](#)
 - [Testing Management](#)
 - [Reporting Defects](#)
 - [Software Testability](#)
 - [Using Expect](#)
 - [Notorious Bugs](#)
 - [Testing and Quality Assurance](#)
 - [A Career in Software Testing](#)
 - [Software Process](#)
 - [Forums](#)
 - [Interesting Sites](#)
-

Test Automation

Techniques and ideas for automating software testing.

[Seven Steps to Test Automation Success](#), by Bret Pettichord

Important steps for planning and running a test automation project.
Cites pitfalls of GUI test automation.

[Software Test Automation and the Product Life Cycle](#), by Dave Kelly

Advice on test automation focussing on when to start automating.

[Lessons in Test Automation](#), by Elfrieda Dustin

Test managers often can find themselves surprised at the difficulty of test automation. This article collects lessons learned on various projects after they started using automated tools. Corrective actions are suggested.

[When Should a Test be Automated?](#), by Brian Marick

Considers relevant factors in deciding which tests to automate.

[Integrated, Effective Test Design and Automation](#), by Ed Kit

Describes "third generation" table-driven test automation. Includes spreadsheet templates.

[How to Automate Testing of Graphical User Interfaces](#), by Tilo Linz and Matthias Daigl

Describes important considerations in the development of test procedure libraries.

[Totally Data-Driven Automated Testing](#), by Keith Zambelich

Evaluates two useful architectures for test automation.

[When to Automate Testing: A Cost-Benefit Analysis](#), by James Hancock

Common sense about what tests should be automated.

[Styles for Making Test Automation Work](#), by Mike Powers

Coding standards and techniques for test automation.

[Improving the Maintainability of Automated Test Suites](#), by Cem Kaner

Descriptions of the pitfalls and challenges of automated testing.

[Classic Testing Mistakes](#), by Brian Marick

A litany of mistakes made in testing, including a section on test automation mistakes.

[Test Automation Snake Oil](#), by James Bach

An argument against some of the naive claims made for test automation by someone who has been burned many times. A [newer version](#) of article is available in PDF.

[Test Automation System Checklist](#), by James Bach

Advise on features to include in a test automation system.

[Success with Test Automation](#), by Bret Pettichord

Description of the development of test automation that used a table-driven approach.

[Automated Software Testing - A Perspective](#), by Kerry Zallar

Other listings of articles on test automation.

[Papers on Automated Testing](#), from Automated Testing Specialists

[Papers for Automated Testing](#), from the Chinese Association for Software Quality

Using Silk

Silk is a tool for automating GUI test execution, developed and published by [Segue Software](#). Earlier versions of the tool have been called called QA Partner, SilkTest, or QualityWorks. It uses a proprietary language called 4Test.

[Classes, Objects, Dynamic Instantiation and Constructors](#) by Jeff Hemsley, 1999

If you've used Silk's 4test language much, you've noticed that it has some object-oriented capabilities. But the only objects allowed are window objects, and even these may only be defined at compile time using window declarations. Or so you might think. Actually general purpose objects can be created during runtime using some clever, previously undocumented techniques. These were discovered and developed by Jeff Hemsley of Autodesk. He describes them in this paper published at Segue's website. (Access requires Segue customer account.)

[Using Object Oriented Design for Testing Custom Tree Controls](#), Cynthia Sadler, 1999

(Access requires Segue customer account. Also available as a [zipped word doc](#) by anyone.)

Describes how rich support was provided for a custom control.

[How to Create an Expert System from the Recovery System](#), by Gerry Lyon, 1999
(Access requires Segue customer account. Also available as a [zipped word doc](#) by anyone.)

Describes an expert system for software verification.

[Applying the Object-Oriented Paradigm to the QualityWorks Recovery System](#), Andrew Townley, 1999? (zipped word doc)

Explains an architecture that allows a recovery system to be quickly configured to support different applications or different test logging policies.

[Easy to use GUI generates QA Partner scripts](#), Tony Venditti, 1998

Describes his [Script Wizard](#), a tool which facilitates the maintenance and developed of Silk testscripts. The Script Wizard is available for download.

[Automated Testing with Segue's QualityWorks](#), Elisabeth Hendrickson, 1998

Suggests a high-level outline for using Silk on a project.

Other sites featuring information about Silk.

[Tony Venditti's Silk Automation Page](#)

Includes source code and the archives of QAPUSER and SILKTEST

[Tony Venditti's Silk/QA Partner Automation Home Page](#)

Features his Script Wizard, a tool for assisting with testscript development.

[David Reed's QA Partner Page](#)

4Test code and tips.

[Charlie Hedstrom's TestMap Extensions](#)

Describes a commercial add-on to Silk. Also includes tips and 4test code for download.

[Austin Segue User Group FTP site](#)

Includes presentations.

Forums for Silk Users

[BetaSoft SilkTest Forum](#)

[BetaSoft QA Partner Forum](#)

[Silktest mailing list](#)

Testing Management

[The Testing Team's Motto](#), by Brian Marick

A testing team's mission is to reduce the uncertainty regarding the quality of the software.

[Software Testing Best Practices](#), by Ram Chillarege

This concise report describes 28 best practices for software testing. They are divided into three groups. Those in the first group should be implemented before starting with those in a later groups. Ram Chillarege is the head of IBM's Center for Software Engineering. His group is researching many of the practices in the latter groups. This roadmap as to what needs to be done first is quite reasonable.

[Negotiating Testing Resources: A Collaborative Approach](#), by Cem Kaner

[Risk-Based Testing](#), Keith McMahon

[System Test Pattern Language](#), by David DeLano and Linda Rising

Reporting Defects

[Software Defect Isolation](#), by Prathibha Tammana and Danny Faught
Discusses techniques for isolating, reproducing and reporting defects.

Software Testability

[Improving Software Testability](#), by Joy Shafer
Identifies factors affecting the ability to test software

Using Expect

Expect is a Unix tool for testing command line interfaces.

Notorious Bugs

[Are there still major computer system failures caused by software bugs?](#)
A long list of software failures from the Software QA/Test Resource Center.
[10 Great Bugs of History](#), by Cnet.
Part of a [special series](#) on bugs.

Ariane 5, June 1996. The maiden launch of the Ariane 5 rocket blew up 40 seconds from liftoff. The rocket and its four satellites were uninsured and worth \$500 million. The proximate cause of the crash was an overflow error due to an attempt to convert a 64 bit floating point value into a 16 bit integer. This error occurred in code that was non-functional after liftoff, when the error occurred. The [Inquiry Board report](#) provides details regarding the software failure and the design policies that lead to it. A [note](#) in the Risks Digest indicates that a

complete system test would have found the problem but was vetoed for budgetary reasons.

Therac-25, June 1985 - January 1987. The Therac-25 medical linear accelerator was responsible for six accidents involving massive overdoses of radiation, three of which lead to deaths. The proximate causes of the accidents were eventually attributed to two separate and hard-to-reproduce faults caused by race conditions in the data entry system. IEEE Computer published a [summary report](#) by Nancy Leveson and Clark Turner. Nancy Leveson also published a [detailed account](#) as a chapter in her book *Safeware: System Safety and Computers*.

Boris Beizer's List. Boris Beizer has said, "Remember that the most notorious bugs in the history of software development were all unit bugs -- bugs that would have been found by proper unit testing." (Letter to swtest-discuss, 10 June 1997). Here is his list:

1. The Voyager bug (sent the probe into the sun).
2. The AT&T bug that took out 1/3 of US telephones.
3. The DCS bug that took out the other 1/3 a few months later.
4. The Intel Pentium chip bug (it was software, not hardware).
5. The Ariane V bug.

In a later letter, he agreed that the Therac 25 bug was also notorious, but not really a unit bug.

John Shore [collected](#) a number of reports, often apocryphal, of famous bugs.

Testing and Quality Assurance

Notes on the difference between software testing and software quality assurance.

[What is the difference between SQA and Software Test](#), from the Revision Labs FAQ
[What is Software Quality Assurance and What is Software Testing?](#), from the Software QA/Test Resource Center

A Career in Software Testing

[Software Tester Salaries](#), by the Software Testing Institute

Software Process

Ideas on how to organize software development.

[The Challenge of "Good Enough" Software](#), by James Bach

This article describes an alternate discipline for delivering quality products in a timely fashion.

[The Immaturity of the CMM](#), by James Bach

This article criticizes the Capability Maturity Model as a useful methodology for organizing software development.

Forums

Discussing testing and bugs.

[Testers Network](#), by Data Dimensions

This monthly newsletter frequently contains concise summaries of useful test techniques. Written by lead testers at the consulting company, the articles never amount to puff pieces. Some consulting companies are afraid that if they tell you too much about what they do, you'll realize you can do it yourself. Not here. Lots of good ideas can be found here.

The newsgroup [comp.software.testing](#) has lively debates on topics such as the proper relationship between testers and developers. This is a place where testers and testing managers go to for advice when they get squeezed particularly hard. Many experienced and helpful software testers frequent this newsgroup. The group's list of [frequently asked questions](#) contains much useful information.

The moderated newsgroup [comp.risks](#) contains concise and technically precise descriptions of software failures, especially when loss of life is involved. This is an excellent place to learn about the kinds of systemic problems that plague software.

[BugNet](#)

This newsletter tracks bugs in commercial PC software. Can you find problems like these *before* your product ships?

[Software Testing and Quality Engineering](#), formerly Software QA Magazine.
A magazine for the professional tester. The articles are written by experienced professionals. I get all kinds of ideas from this.

[SQA Suite Users Mailing List](#)

Interesting Sites

Sites with essays, articles or lists of general interest.

[Bret Pettichord](#)

Features papers on test automation and other testing topics.

[Center for Software Engineering Research at the Florida Institute of Technology.](#)

Features papers by James Whittaker, the center's director.

[Danny Faught](#)

Includes articles on testing, Unix, and Perl.

[Satisfice, James Bach's consulting company](#)

Includes many of Bach's [articles](#) on software testing and process.

[Software Research, Inc.](#), a test tool vendor.

Includes [Test Techniques Newsletter](#) and [Software Quality HotList](#).

[CenterLine Software](#), a test tool vendor.

[Rational](#), a test tool vendor that gobbled up Pure, Atria and MS test.

Includes dozens of [technical papers](#), including [Quality on Time](#), by Aki Fujimura, an interesting article on scheduling development and testing.

[Software Testing Laboratories, Inc.](#), consulting and contracting company.

Includes a collections of [articles](#). They publish a newsletter, [The Testers Network](#).

Also available are [Quality Matters](#), a newsletter of the Washington Software and Digital Alliance.

[Marick's corner](#)

Includes papers and lists of consultants, tools and trainers.

[RBSC Corporation](#), consulting, training and contracting.

Includes lots of information about object-oriented testing: [articles](#) by Bob Binder, a [bibliography](#), and a [list of resources](#).

[Reliable Software Technologies](#), consulting, training, and a test tool vendor.

Includes a few hundred [papers](#) documenting original research and a [reliability hotlist](#).

[Cem Kaner](#), Testing Consultant and Attorney at Law

Includes papers on [software testing](#) and [the law of software quality](#).

[Software Testing Resources](#), by Kerry Zallar

This web site also includes links to other QA resources.

[Johanna Rothman](#)

Includes papers on software testing and risk management

[Rice Consulting Services](#)

Includes a couple [articles](#).

[Systeme Evolutif](#)

Includes testing [resources](#).

Software Quality Engineering

Online catalog of software quality books, David Gelperin

Storm

Includes [Who's Who in Software Testing Research](#), [Directory of Software Testers and Consultants](#) and an index of [monographs and opinions](#).

4QA, also known as SVP.

Includes [sample test code](#) for QA Partner.

Software Testing Institute

Includes a [salary survey](#)

Technical Thoughts, by Howard Fear

Short essays on software engineering and testing.

Quality Tree Consulting

[Articles](#) and short essays ("[ruminations](#)") on software quality and testing by Elisabeth Hendrickson.

The Center for Software Testing Research at Florida Tech

Includes course notes by [James Whitaker](#) on software testing, and in particular stochastic (random) techniques.

Harry Robinson

Danil Suits ("Mr. Cluey")

Information about QA Partner, including an [archive](#) of the QAPUSER mailing list.

The Carl Group

Maintains a page illustrating [problems](#) with SilkTest.

Chinese Association for Software Quality (CASQ)

A comprehensive web site. Includes excellent linked bibliographies. [Automated Software Testing](#).

Quality World

Thousands of links

Automated Testing Specialists

Includes [index](#) to articles on automated testing

Aptest, providing test automation services

Includes [Software Testing Resources](#)

BetaSoft, providing test automation services

Includes forums and downloads for popular test tools. Also contains a job board.

Tantara Management Services, software process improvement and quality assurance consulting.

Includes a [hotlist](#) of resources.

Other Commercial Sites

Note: The Software Testing Hotlist no longer lists commercial websites unless they contain articles or other information of interest to the industry.

- [Marick's list of test tools](#)
- [Marick's list of consultants](#)
- [Yahoo's directory of software testing companies](#)

- [Software QA/Test Resource Center](#)
 - [Software Methods and Tools listing of test and test management tools](#)
-

Book Review

Testing Computer Software, by Kaner, Falk & Nguyen

If you read only one book on software testing, read this book. Many books will tell you how to test when you have enough time and cooperation. This book tells you what to do when the schedule is tight, the specification is missing, and the developers are tired of your negative attitude. It has sound advice and is a pleasure to read. I keep coming back to it. Feel like you have an impossible job? Read this book.

About the Editor

Bret Pettichord is a test automation engineer at [Tivoli Systems](#) in Austin, Texas. He develops test automation strategies and tools for Tivoli's product teams. He specializes in GUI test automation, scripting languages, and test methodology. Before joining Tivoli, Bret developed automated tests for several commercial software vendors, including Unison, BMC, Segue and Interleaf.

Bret is a frequent speaker on software testing and test automation. He is on the advisory board of [Software Testing and Quality Engineering](#) magazine and is [certified in software quality engineering](#) by the [American Society of Quality](#). He is a participant in the [Los Altos Workshop for Software Testing](#) and a member of the [IEEE Computer Society](#). He has a bachelor's degree in philosophy and mathematics from [New College](#) of the University of South Florida. Both of his parents were IBM programmers. His mom wrote better code than his dad.

Please send updates and comments to the editor at

b.pettichord@computer.org

Software Testing Hotlist, Copyright © 1995 - 1999 Bret Pettichord