

---

**Stephen R. Savitzky**, Computer Curmudgeon  
<https://Stephen.Savitzky.net/> <steve@savitzky.net>  
<https://www.linkedin.com/in/ssavitzky> <https://github.com/ssavitzky/>  
Highly experienced programmer looking for full- or part-time remote work.

---

## Summary

Software toolmaker and craftsman, with decades of experience that includes Java microservices at Amazon, website and web infrastructure development with Java and Perl, embedded systems programming on Debian Linux, build management with git, GNU make and Bourne Shell, server monitoring with Nagios, class libraries in Java and C++, and small projects in Javascript, C, LISP, PHP, and Python. Masters degree in Computer Science from Stanford.

Experienced communicator, presenter, and mentor; introduced many leading-edge software concepts and methods to my employers: meta-assemblers and real-time kernels at Zilog; object-oriented programming and cross-platform development at Advansoft Research; application frameworks, open-source, Linux, CVS, git, XML, and web services at Ricoh Innovations. An IT person from India gave me the title of "guru". Helped two Amazon teams convert from Perforce to git.

## Skills

Expert: Java, git, Perl, Bourne Shell, GNU make; Linux; web development: HTML, JSON, HTTP, ReST. Less recently, C, C++, assembler; XML.

Competent: PHP, Javascript, Python, Smalltalk, Lisp, Pascal, Fortran; Unix;  $\LaTeX$ , SIP; mentoring, tutoring; technology transfer; technical communication; audio recording and editing; dev/ops.

## Experience:

**tutor** Apr 2017-present: Freeland, WA

- teaching a home-schooled middle-school student math, science, and programming (with Racket)

**Software Development Engineer** Mar 2013-Apr 2017: Amazon, Seattle, WA

- Implemented a complete web service, and parts of others, in Java.
- Interface and class design, unit and integration testing.
- Cobertura, FindBugs, Mockito, EasyMock, Checkstyle, Eclipse, S3

**ReSTful Engineer** Oct 2012-Mar 2013: Disney ID group, Seattle, WA (via KForce)

- Implemented ReST APIs, unit tests, and functional tests using Java, Spring, Mockito, Eclipse, and Maven in an agile environment.
- One of the local experts on Git.

**Sr. Software Architect** 2011-2012: California Innovations, Cupertino, CA

- Added automatic SIP audio conferencing to interactive whiteboard system for vastly improved user experience.
- Implemented client and server monitoring with Nagios, including simplified web-based installation procedure for MacOS-based clients in remote locations. Provides convenient overview of whiteboard usage and server health.

**Principal Software Engineer** 2010-2011: E-Writer Systems, San Jose, CA

One of six researchers who went to help found a new subsidiary, Ricoh E-Writer Systems, turning what had been a research project into a product, the Ricoh eQuill electronic clipboard.

- Designed and implemented prototype web service with REST API on Apache and Perl, used for experiments and demos while the production version was being built.
- Rewrote and improved the PDF conversion implementation in PHP with Zend framework.
- Developed a reliable, secure document transfer protocol based on ssh and git, and supervised the consultant implementing it.
- Modified the dropbear ssh daemon to handle a large number (tens of thousands) of keys by replacing flat file with memcachedb.
- Architected and wrote the factory test suite for eQuill tablet, based on bash shell and test programs written in bash and python, to meet requirements from the manufacturer in China.

**Chief Software Scientist** 1992-2010: California Research Center, Menlo Park, CA:

- Object-oriented programming, Linux, and open-source software “guru”.
- Designed, and led 3-person group developing, a C++ framework for remote service applications. Organized and help teach a 10-week technology-transfer course. The transparencies used were derived from the C++ header files using a “documentation generator” that was later extended to include Perl.
- Designed and implemented (in Java and Perl) an XML-based server-side scripting language (PIA), which became the basis for an “experimental” forms system that was in daily use for a decade. Added a multi-language “loose parser” for code formatting.
- Co-invented the GeoFi geographic tagging format (US patent 8,089,405: *Applications for geographically coded access points*); implemented versions in Perl and Javascript.
- Co-inventor on 27 issued patents.
- Introduced object-oriented programming, application frameworks, Linux, WWW, and open-source to Ricoh.

**Prior Experience**

**Scientist:** Advansoft Research Corp., Santa Clara, CA

**Principal Engineer:** Zilog, Inc., Campbell, CA

**Staff Engineer:** Zilog, Inc., Cupertino, CA

**Senior Engineering Programmer:** American Microsystems, Inc., Santa Clara, CA

## Publications

- Open Source software: PIA <http://github.com/ssavitzky/pia-server>  
A web server and templating engine for HTML and XML, with functional-programming semantics and XML syntax. It makes a single pass through the input file, making it suitable for low-memory applications
- Book: *Real-Time Microprocessor Systems*, Van Nostrand.
- “Notes, Advice, and Warnings for Kids on the Web”, on my prize-winning web site “Interesting Places for Kids”.
- 27 US patents, listed at [Stephen.Savitzky.net/patent-list.html](http://Stephen.Savitzky.net/patent-list.html)
- White Paper: *Web Applications and the PIA*, RiSource.org. Presented as a conference paper at XTech 2000.
- “Automatic Discovery and Execution of Personal Applications from Shared IO Devices”, Bradley Rhodes, Sergey Chemishkian, Edward L. Schwartz, Stephen Savitzky and Haixia Yu, in 2010 7th IEEE Consumer Communications and Networking Conference (CCNC), Jan 9-12 2010.
- Conference papers on “ZRTS (Zilog Real-Time Software)” (Wescon and Midcon), and “Cross Software Development under UNIX” (Electro).
- CD: *Coffee, Computers, and Song!*, HyperSpace Express.

## Education and Memberships

**Stanford University:** MS in Computer Science

**Carleton College:** BA in Mathematics

**Memberships:** ACM, Sigma Xi, Phi Beta Kappa, AAAS, EFF, SVLUG.

## Addendum: Projects

### ISIL (Team member, Amazon)

A small but complete internal web service, written in Java. Design, implementation, unit and integration testing, deployment, operations, maintenance.

### IWB (Team member, CIC)

Added automatic SIP-based audio conferencing to an interactive whiteboard project, greatly improving the user experience when sharing a whiteboard between two or more locations. Deployed and configured Nagios for monitoring the status and performance of both the 14 installed whiteboards and the associated web services.

### eQuill (Team member, EWS)

An e-ink based tablet for business use, running Debian GNU/Linux internally and backed by a cloud service written in PHP. Built the prototype web service in Perl, designed the second version of the document transfer protocol (using git over SSH), and wrote the factory diagnostics in a combination of Bash and Python. Ported the routines that converted between PDF and the internal (image-based) document format from Python to PHP, and modified the `dropbear` ssh daemon to accept a large number (tens of thousands) of keys by replacing the flat `authorized_keys` file with a `memcachedb` “no-SQL” database.

### GeoFi (Implementor, CRC)

Co-invented (US patent 8,089,405: *Applications for geographically coded access points*) a way of encoding a geographic location, accurate to a couple of feet, in 9 alphanumeric (base-60) characters so that it can be incorporated into a WiFi SSID string for use as an indoor GPS replacement. Implemented the algorithm both in Perl as a module and CGI, and in JavaScript as a Google Maps mashup.

### PIA (Team lead, CRC)

Designed PIA, an experimental server-side scripting language with XML syntax and functional programming semantics. Initially implemented in Perl, re-implemented in Java, and eventually ported to C++. Convinced Ricoh to become the first Japanese company to release an open-source software project. Supervised the development of a C implementation (cPIA, available on SourceForge). Note that this project was actually started *before* XML, using an SGML syntax that was actually considerably simpler and more user-friendly. It was roughly contemporary with PHP.

PIA was used to implement a form-processing system that included such standard business forms as vacation requests, timecards, expense reports, and so on. It was written in 1998 and remained in use with only minimal maintenance for a decade, until a series of Java upgrades introduced incompatibilities that would have taken researchers away from other projects to fix. Administrative personnel were able to easily create new forms. It was eventually replaced by forms in the Metadot intranet portal, which most users agreed were less convenient.

Another unique (for its time) feature of the PIA is that it uses HTML elements as data structures, e.g. by treating DL elements and tables as associative arrays, and entities as variables. Unlike most such systems, it stores filled-out forms as digitally-signed documents rather than as database entries.

### REST (Team lead, CRC)

Led a three-person team in developing a C++ programming framework for building an application that remotely monitors copiers and FAX machines. Introduced object-oriented programming and frameworks to Ricoh. Technology transfer included organizing and co-teaching

a the equivalent of graduate-level 10-week class. Cross-platform on Windows NT and Linux, and included GUI front ends for web and X, generated from the same source code. Wrote a “documentation generator” (in Perl, predating JavaDoc by several years) to create the transparencies used in the course from the C++ header files. Later extended the documentation generator to include Perl, Python, and PHP.

**WABE** (Sole developer, Advansoft)

A C++ class library that includes arrays, hash tables, run-time class descriptors, X-based graphics, Motif-based GUI wrapper objects, and a self-describing object-oriented file format. Used internally on three projects: a data analysis and display program, an IC-tester utility, and a test-case generator for a compiler.

**RD3, RD7, RWi** (Consultant, HyperSpace Engineering)

A series of three MS-DOS operating system shells (similar to X-tree), each capable of reading (and in one case writing) disks written by another operating system as well as MS-DOS (Perkin Elmer 3500, Idris, and iRMX86 respectively). These were done as a spare-time consulting project; I received royalties on each copy sold. The implementation was in C and included an object-oriented, text-based user-interface toolkit; I also implemented a subset of the *courses* library and emulators for the three different filesystems. The menus, manual, and online help were all generated from the C source code.

**Assemblers, linker** (Project leader, Zilog)

Sole developer of the linker and cross-assemblers for two of Zilog’s three microprocessor families, and project leader for the third. The assemblers were written in C, portable between MS-DOS and Unix, and entirely table-driven; the third assembler was completed by a junior programmer in a matter of weeks. The linker was an early implementation of the IEEE standard for microprocessor object files, and was unique in allowing a mix of word lengths (up to 256 bits) and address ranges in a single project.

**ZRTS** (Project leader, Zilog)

Designed ZRTS (Zilog Real-Time Software), a real-time operating system kernel for the Z8000 microprocessor. Implemented the task-switching and message-passing microkernel, logical I/O system, and terminal driver; other device drivers were done by team members. ZRTS was moderately successful both inside and outside of Zilog.

**AMI MDC** (Sole developer, AMI)

Wrote most of the software for AMI’s 6800-based Microprocessor Development Center including the ROM monitor, bootstrap loader, debugger, and text editor, all in 6800 assembler. Added macros to the assembler. The text editor was written using object-oriented macros, and included variable length strings with garbage collection.

**CD:** *Coffee, Computers, and Song!* (singer/songwriter, engineer, producer, publisher)

Did everything for this CD but the manufacturing and artwork, using open source tools: Makefiles, perl and shell scripts, LaTeX, and Audacity. The web page, track list, disk table of contents, and songbook are all made from the same set of source files.

**Interesting Places** (Author/webmaster)

“Interesting Places for Kids on the Web” was an early personal website that was listed as one of the top 100 websites in 1995, and widely referenced both online and in print. Part of the Ricoh website, it drew traffic to the corporate site in the early years of the commercial Web. Included “Notes, Advice and Warnings for Kids on the Web”.