

Résumé of Flash (K. J.) Sheridan

500 East 77th Street #3314
New York NY 10162-0034

flash@pobox.com
<http://pobox.com/~flash>
<http://www.linkedin.com/in/flashsheridan>

Summary Twenty years experience in software quality assurance, including three in static analysis; two years experience in software development. Skilled at finding, reporting, and tracking software bugs, via whitebox testing, test harness development, user-level testing, load testing, and static analysis. Published authority on compiler testing and static analysis.

Skills Software quality assurance, automated testing, static analysis, embedded systems, GCC, LLVM/Clang, Xcode, C/C++, Lisp, Perl, Python, Swift, Coverity, Klocwork, Git, Kafka, sh/tcsh/bash, HTML, LaTeX, TCP/IP, Linux, Macintosh, Windows.

Employment

- 2017 Bloomberg L.P./Open Systems Technologies, Inc.† Software Engineer in Test.
Implemented and evangelized software quality best practices in financial software: continuous integration, automated test validation, and static analysis.
- 2012–2015 Apple. Compiler Quality Engineer.
Testing of Swift and the LLVM/Clang compiler/static analyzer via a Python test harness. Filed bugs in an internal database against compilers and the Xcode development environment; evaluated and reduced bugs filed in the public LLVM/Clang Bugzilla database. Test lead for the Apple Command Line Tools. Added validation tests to the open source Swift compiler unit tests. Wrote GUI automation tests for the Xcode IDE.
- 2011–2012 Bell Labs/Alcatel-Lucent.† Software security static analyst.
Static analysis using Coverity and Klocwork for the detection of security vulnerabilities in mobile telephony infrastructure.
- 2009–2010 Qualcomm/Code Integrity Solutions.† Senior consultant.
Static analysis using Klocwork, PC-lint, and Coverity for the detection of bugs and security vulnerabilities in mobile telephony software. Evaluated and compared competing static analysis tools.
- 1998–2009 Palm/PalmSource.† Test programmer.
Senior Quality Lead for the PalmSource ARM-native compiler; no bugs in the compiler were reported after its release, and an account of my testing methodology was published by a leading academic computer science journals; see “Publications.” Corrected Coverity integration with existing build system to improve automated detection of software defects; diagnosed analysis failures; adjusted analysis options to increase relevance and quantity of defect reports. Implemented reporting system to track responsibility for detected defects, which resulted in hundreds of bugs being fixed.
- 1995–1997 Apple.† Test programmer.
Low-level Quality Lead, Newton MessagePad 2100. API Quality Lead, eMate 300. Low-level Quality Lead for an unreleased POP/SMTP client. Maintained and extended test harness for automated API and user-level testing. Developed test program for Newton Backup Utility data integrity. As semi-official Usenet representative, handled third-party bug reports, user data collection, rumor management, and spin control.
- 1986–1989 The University of London. Research Assistant, Inference under Uncertainty.
Writings on Artificial Intelligence, listed below.

† Contractor or contractor to employee

iPhone Software (Objective-C)

EncycloClip. A faceless app to expedite search in the Encyclopedia Britannica web site.

Palm OS Software (CodeWarrior C)

Small Talk 1.02 & Japanese Language Module (LandWare). Added support for Japanese OS extension; supervised and implemented translation of phrase database into Japanese.

Newton Software (NewtonScript)

KwikMenu™ (LandWare). Provided immediate, universal menu access to a variety of common tasks. Reviewed in *NewtNews* February 1996, *Gecko* April 1996, and *Pen Computing* May 1996. Gecko rating: 4½ out of 5.

Register (Kagi). Allowed a user to pay for software with a credit card via electronic mail; provided an API used by approximately 150 third-party packages.

URLCop (freeware). Provided an API to dispatch a URL to the appropriate application. Supported by Newscape, Newt's Paper, pURL, and Shuffler.

Publications

“Deploying Static Analysis,” cover story, *Dr Dobb's Journal*, July 2012.

“Static Analysis Deployment Pitfalls,” short paper, IEEE Symposium on Software Reliability Engineering, 2010.

“Static Analysis in a Fallen World,” talk at the Stanford University Computer Science Department, 2010.

“Practical Testing of a C99 Compiler Using Output Comparison,” *Software: Practice and Experience*, 2007 [37] pp. 1475–1488.

“Access Linux Platform for Linux Geeks: an Introduction,” PalmSource Developer Newsletter, 2006 [25].

“urlCop: Proposed Standard for Inter-Application URL Handling,” presentation, Apple Newton Developer Conference, 1996.

“A Survey of Techniques for Inference under Uncertainty,” *Artificial Intelligence Review*, 1991 [5] pp. 89–119.

Three articles in *Non-Standard Logics for Automated Reasoning*, Academic Press, 1988, edited by Phillippe Smets, E.H. Mamdani, and Didier Dubois.

“A Variant of Church's Set Theory with a Universal Set in which the Singleton Function is a Set,” *Logique et Analyse*, 2016 [59].

“Fixing Frege's Set Theory,” talk at the Stanford University Mathematics Department, 2014.

Education

Balliol College, University of Oxford. Thesis for a doctorate in mathematical logic completed and published by *Logique et Analyse*. The results are summarized in sections 4.1 and 4.3 of *Set Theories with a Universal Set*, by T.E. Forster (Oxford Logic Guides 20, Clarendon Press, Oxford, 1992).

Yale University. B.A., Mathematics and Philosophy. Anthony D. Stanley Award for Excellence in Pure and Applied Mathematics.

Phillips Exeter Academy. Graduated *cum laude*.

Languages

Latin, Greek, French (fair), Russian (fair), German (fair).

User and Developer Groups

I was webmaster and co-founder of the Stanford Newton/iPhone User Group from 1994 to 2014, president and founder of the Stanford PalmPilot User Group from 1997 to 2009, chief judge for the PalmHack IV-VII programming contests, and founder of the LinkedIn Static Code Analysis Group.