

Jason LeCount
jason@lecount.org, (510) 470-0321
linkedin: <http://www.linkedin.com/in/jasonlecount>

SUMMARY

I have 28 years experience with build and test infrastructure and automated testing methodologies. I strongly believe in Agile software development and am interested in supporting TDD by making it easier to test code, through framework/APIs, infrastructure, mentoring, etc. Working with a codebase written in a functional language would be a huge plus.

SKILL PROFILE

Operating Systems: OS X, Linux, Android, Windows
Technologies: Java J2EE, ruby, scala, python, selenium-webdriver, docker, ScalaTest, junit, TestNG, rspec, Jenkins, ant, maven, bash, git. javascript/node.js

EXPERIENCE

Sungevity **June 2014 – present**
Test Architect

Led team of 5 people focused on automated testing, and test infrastructure at Sungevity. Mentored and did training sessions for entire QA team (18 people) on Webdriver, automated testing, REST API testing, docker, git, Python and Scala.

Wrote Webdriver-based UI automation framework in python. Features included Salesforce integration, page objects, transparent support for single-page javascript apps, cross-browser, and mobile support. Test results persistence is automated and uses TestRail as a system of record.

Wrote requests-based API testing framework in python.

Wrote scala library for functional test utilities of the Sungevity back-end. Features included functional test support with the same semantics as the Play controller unit/integration tests, Siren response parsing, Testrail test results integration, JSON request construction.

Wrote Hipchat integration for automated post-release validation test result notification into Hipchat rooms

Wrote a python utility for automatic generation of test users used by Dev and QA for internal testing.

Implemented docker-based Jenkins stack in which master and slaves are docker containers. All changes to Jenkins configuration (master and jobs) are encrypted (in cases of sensitive data) and stored in GitHub repo.

Implemented docker-based tooling to provide a consistent **local** (OSX / Windows) test

environment regardless of host OS. Implemented with a mixture of bash and python. Maintained Jenkins server and jobs creation and updates.

As part of a hack day, wrote a Javascript app, which became an internal tool that simplified test triage by providing a dashboard view of failing tests.

Wrote python tree-based test framework for test cases structured around a model of tree traversal. Nodes represented application states and edges represented actions to transition between states. Test setup consisted of declaring rules around how the tree may be traversed, then the framework walked all valid paths through the application-under-test given such constraints.

Organized and ran a 6 week Haskell lunchtime study group.

Off hours: completed successfully the following Coursera classes in Scala: ***Functional Program Design in Scala, Parallel Programming***, and ***Principles of Reactive Programming***. Currently learning Clojure.

Okta**June 2013 – June 2014**

Senior Software Engineer in Test
Internal Build and Tools Engineer

Wrote a node.js / Firebase-backed server to provide real-time view data of Okta's Jenkins CI pipeline

Wrote node.js scripts to collect historical metrics on Jenkins build jobs.

Wrote a Jenkins plugin in Java to do setup and teardown on Jenkins slaves

A part of a team of three, enhanced and maintained a large (120 slave, 2000 builds/day) Jenkins EC2-based build/test infrastructure.

Ported a large (4000 test) test suite from JUnit to TestNG.

Converted a large set of Jenkins jobs with inline job definitions to a bash-based job framework.

Wrote Ruby scripts to help automate triage of failing tests.

Maintenance of a Python Gearman-based Jenkins job queue.

Owned and extended a Ruby-based tool to set up Engineer's laptops upon joining Okta and automate periodic laptop updates of various tools (e.g. new git versions, new internal dependencies such as memcached, etc...)

Netpulse**October 2011 – June 2013**

QA Architect

Responsible for improving testability of all Netpulse application components, as well as growing an Agile and test-driven Engineering culture. My primary goal is to ensure that all application components are easily testable, and that staff (Dev, QA) understand how to write unit and functional tests and that impediments to doing so are removed.

Android Team:

Wrote and maintained all build / CI / release automation with ant and Jenkins

Integrated Guice (IoC framework) and Robolectric with our application.

Mentored Dev team in test-driven development including topics such as testable design, dependency injection, unit vs. integration testing, etc.

Trained team in transitioning to git from svn.

Wrote POJO and Robolectric tests.

Wrote functional test framework with monkeyrunner and added longevity tests and graphed 'top offenders' -- an aggregate list of exceptions caught from the prior monkeyrunner run.

Qt (legacy client) Team:

Added functional UI automation tests via FrogLogic's Squish tool.

Legacy Server Team:

Wrote a Ruby based tool for HTTP API testing

Mined production logs to generate a regression suite of 2500 distinct inputs and correct XML responses for existing legacy server functionality for the purpose of testing backwards compatibility of new server.

New Server Team:

Added functional tests for REST API via Ruby framework.

Began work in Scala to automate build with SBT and add tests with specs2. Scala server port was cancelled due to management change so, sadly, this project never took off.

Wrote selenium-webdriver-based UI automation of web portal.

Performed load testing with Tsung. Graphed and published run results as well as historical trends.

Cisco Systems

February 2007 – July 2011

Senior Automation QA Lead

Was responsible for leading team of 3 automation engineers in an Agile (Scrum) team of developers, product managers, QA engineers, and OPS engineers. Was responsible for automation tool choice, implementation and design of test frameworks, working with junior engineers to build out suites, and general oversight of junior team members.

Wrote Selenium-based UI test framework to automate UI regression of Cisco Eos social network. Test framework has the following features:

- Distributed and parallel test execution across multiple selenium servers.

- Fast, multi-browser execution using CSS selectors.

- Test result bridge to HP QualityCenter, which allows real-time integration with black-box test inventory and results.

- Allows black-box team to execute automated suites on their own machines, then push results back to QualityCenter. This has saved tremendous manual test time.

- Groovy and TestNG based test API allows for flexible and powerful tests.

- TestNG results stored in mysql database to enable analysis of historical test run data.

- With one other automation team member, wrote 1500 testcases for the Eos social network.

Wrote an ETL (extract, transform and load) regression test framework and suite for testing historical data.

Features include:

- Execution of real user activity on QA, Stage and Production on hourly basis.

- All cookies and contextual information from test user activity is written to an audit database for subsequent validation.

- Validation suite compares stored test metadata with ETL historical information via SQL test suite. Data comparison framework allows very simple data-driven testcases to be written. Current suite validates 3000 data points.

Led initiative to transition CMSG group off of Subversion and onto git. Gave presentations, trained engineers and was an overall advocate for its adoption.

Fiveacross.com

October 2006 – February 2007

Senior QA and Build Engineer

Developed automated deployment scripts for QA, Stage and Production using Capistrano. Developed rspec tests for a Ruby on Rails social network that hosted sites for NASCAR, the NHL and Televisa. Tested and deployed infrastructure updates for all environments.

Northstar Systems International February 2003 – October 2006

Senior Tools and Build Engineer

Implemented scripts to do client deployment (appserver configuration, deployment, database creation and population.)

Implemented automated build / deploy / test infrastructure for internal Dev / QA use. Infrastructure is responsible for cross-platform (Linux and Win XP) continuous and daily builds, unit test execution and website update, javadoc updates, emailing build failures and unit test results and automated creation of TeamTrack defects from unit test failures.

Trained Indian QA team in New Delhi on the Northstar API and unit-testing best practices.

Co-wrote and maintained the NorthStar ant build. Build integrates with eclipse and determines build dependencies via Ecilpse project dependencies.

Code-generated junit testcases using Velocity to test CRUD methods on data objects. Implemented a variety of other testcases using junit / Cactus to test business logic.

Migrated current junit test framework to spring-based framework, allowing testing of application services without requiring an appserver.

Implemented Development and Customer Support bug tracking workflows using Serena TeamTrack.. Set up end-to-end security through firewall via an Apache proxy that

performs an additional layer of authentication for external users.

Performed various release control duties (branch management, label creation, integrations / reverse integrations) using Perforce.

Configured and managed ASP monitoring scripts with nagios.

Technologies: Java/J2EE, junit/Cactus, QuckTest Pro, ant 1.6, Perforce, bash, linux

E2Open, Inc.

June 2002 - November 2002

Senior Software QA Engineer

Responsible for developing automation infrastructure for QA group. Designed and implemented an XML content generation program in Java for generating data conformant to a specific DTD. Wrote an extensible data mapping validation tool in Java for use by other QA engineers. Implemented a junit-based testing framework for QA whitebox testing. Performed content validation on an XML transformation engine. Wrote a distributed messaging component using log4j to provide communication between a server application and a junit testcase. Wrote a variety of bash/Python utilities to automate system level testing.

Technologies: Java, XML, jUnit, Python, log4j, bash

Kenamea, Inc.

July 2001 - February 2002

Senior Software QA Engineer

Wrote automated testcases in Java using Macaca (a Junit-like testing framework) to unit test a JMS messaging implementation. Wrote system-level tests to realistically test multiple users sending to/from JMS queues and topics.

Wrote automated testcases for a transaction processing system for Kenamea's message switch product. Testing verified that transactions were durable, isolated, could be committed and rolled-back in a variety of situations. Designed a multi-machine environment to perform system crash testing to verify that uncommitted transactions could be rolled back with no lost messages / corrupted state.

Designed and coded modules to support testcase / test results entry into an Oracle database with a web-based front-end. Scope of work included designing database schema, coding stored procedures, coding Java classes/JDBC access layer and JSPs to display the testcase / results data.

Automated existing performance testing harness to run on its own and do diagnostics of common types of failures (hung server, etc.) and to add testcase results to the database at the end of the test run. Finished system automatically installed server-side and client-side components when new builds were available, ran performance tests, and inserted results into database (see above.) Thus, up-to-date performance results were always available company-wide via web-based interface.

Technologies: Java/J2EE, Oracle 8, JSP, Macaca

Foundation Software Group July 1996 - July 2001

President

Client: Instinet January 1999 - August 2000

Designed and coded a load-testing application in Java to create and manage test clients for an EJB-based Fixed Income global trading application. Load testing application was distributed world-wide (New York, London, Frankfurt and Paris) and used to simulate up to 2000 simultaneous users. Analyzed results from test output to identify performance bottlenecks.

Designed and coded a Java test suite for the purpose of regression testing the Fixed Income trading system back-end.

Designed and implemented a GUI regression test suite in Silk. Test suite automated the testing of all major GUI features and was used to automate Smoke testing of new builds of the Fixed Income system.

Technologies: Java 2 (including EJB 1.0 and JMS), Silk/4Test, Oracle 8

Client: Reuters October 1998 - January 1999

Designed and implemented a regression test suite for a Java-based bond trading application. Wrote extensions to the application in Java to enhance testability of application. Trained personnel in automated testing methodologies and 4Test. Wrote extensive documentation (~100 pages) on the process of automated testing, as well as best practices for implementation using Silk.

Technologies: Silk/4Test, Java

Client: Merrill Lynch

October 1997 - July 1998

Designed and implemented a regression test suite for the purpose of testing a Solaris-based Foreign Exchange trading system. Features include testing of ticket entry, correct blotter behavior (activity, credit exposure, positions, etc.), start of day, and end of day. Coded in C++ tests for the validation of real-time messages between applications across a network. Wrote a Tcl/Tk script for the Systems group to launch different versions of the FX trading application from a GUI interface.

Technologies: Solaris, Windows NT, Sybase, 4Test, C++, Sybase, Tcl/Tk, Perl

Client: Goldman Sachs

November 1996 - September 1997

Designed, and implemented a generic object library in 4Test for testing applications written with a proprietary API. Testing objects are configured via ini files, and are instantiated dynamically for greater flexibility. Object library is extremely flexible, and provides a novice QA Partner user with an interface which encapsulates application logic within objects.

Designed, coded and implemented a test suite with QA Partner / QA Organizer to validate a client/server equities order entry system. Order system is distributed among four client machines (both Solaris and Windows clients), including Sales, Trader, Floor, and SIAC, with a Sybase database server back-end. Test suite simulates order entry, execution, and messaging between all front-end machines, and performs GUI validation as well as database validation.

Coded a C++ real-time price provider for a SIAC simulator, for the purpose of accurately simulating automated order handling. Price classes were implemented with Rogue Wave Tools.h++.

Technologies: Solaris, Windows NT, Sybase, 4Test, C++, Perl, Korn Shell

Client: Wall Street Systems November 1996 - December 1996

Managed project to validate GUI functionality of a Futures trading system comprised of over 600 screens. Provided an easily configurable test suite which can be easily extended, and has support for languages other than English.

Technologies: Windows 3.1, 4Test

Client: Dow Jones / Telerate July 1996 - October 1996

Designed and coded utilities to test market feed distribution software. Utilities were written in C++, Perl, and Korn shell. Rewrote Dow Jones' existing tests to increase efficiency. Testing utilities included software to translate data formats, and compare log data from test runs.

Technologies: HP-UX, C++, Perl, Korn Shell.

Coopers & Lybrand August 1993 - July 1996

Senior Software Consultant

Client: AIG Trading Corporation, September 1995 - July 1996

Designed and coded a C++ test suite to unit-test API functions for a Foreign Exchange trading application. API functions interfaced with an Oracle database to manipulate trade and currency position data. Aspects of validation included functional, stress and efficiency.

Designed, coded, and maintained a test suite using QA Partner to perform unit, integration, and regression testing of an entire Foreign Exchange trading system. All phases of foreign exchange trading were modeled, from front-office trading to back-office confirmation and instruction. Aspects of validation included functional, stress, efficiency, and integration.

Trained personnel in the 4Test language and use of its IDE, QA Partner.

Technologies: Solaris 2.4, Windows NT, C++, Oracle, QA Partner, Perl, Korn Shell

Client: Smith Barney July 1994 - July 1995

Designed and implemented a test environment for a 1,600 turret trading floor as part of a

team: wrote test plans, advised client staff on hardware and software needs, wrote test specification documents, designed, developed and analyzed test suites written in XRunner (TSL), C, Perl, Korn and C shell. Gave seminars on developing automation tests and the use of automation test suites to improve software quality. Automation tests written included NIS+ and NFS stress tests, IBM AS/400 stress tests, application functionality, application interoperability, Market Data, and Foreign Exchange trade simulation. The automation test environment created was able to realistically simulate and control all aspects of trading on a 1600-turret trading floor from one workstation.

In C, added a Motif front-end to a TCP/IP based workstation control application. Application was used to control all aspects of automation testing in extremely large networks.

Technologies: Solaris 2.3, C, XRunner, TCP/IP, Motif, Perl, Korn Shell, C shell.

Client: Chemical Bank

July 1993 - June 1994

Designed and coded a test suite to perform automated testing on a 400 turret trading floor. Wrote tests to verify network functionality and bandwidth, application functionality and resiliency to system stress, and system integration.

Maintained a Perl application which was responsible for remote test execution on 150 machines. As a result of enhancement to the application's communication with its clients, overall performance increased 200%.

Technologies: Solaris 2.3, C, XRunner, TCP/IP, Perl, Korn Shell, C shell.

Pyramid Technology July 1990 - July 1993

Senior Software Quality Engineer

Designed and coded test software for two Ethernet device drivers, disk and tape device drivers, and compilers, including C, C++, COBOL, FORTRAN 77, and Pascal. Test development included writing software test plans, writing technical specifications, developing test suites, executing test suites, and maintaining test software.

Developed an Ethernet test suite to cover the user, streams, and protocol layer (TCP and UDP) levels. Executed test suite against a single-interface and a dual-interface Ethernet device driver.

Wrote and maintained a test automation harness in C++ to control execution and error

reporting of tests distributed throughout a network.

Designed and implemented a utility to automatically back up and restore critical data to and from a network.

Represented Pyramid Technology as liaison to Olivetti Corp., ensuring that their UNIX port was compliant with various conformance test standards. Trained personnel there in use of the X/Open Conformance Test Suite. Responsible for running and analyzing conformance test suites, including POSIX, X/Open, System V Verification Suite (SVVS), and gABI (Generic Application Binary Interface) tests. Approved all operating system releases based on results of conformance testing.

Technologies: UNIX SVR4 (DC/OSx) and BSD 4.3 (OSx), C++, C, TCP/IP, Korn Shell, C shell.

Stanford Linear Accelerator Center October 1988 - August 1989

Programmer / Analyst

Designed and implemented an electronics schematic wirelist translation program. Modified and maintained a voicemail/telephone paging system to route phone communication building-wide.

EDUCATION

California State University Chico June 1989 - June 1990

Coursework included Pascal, Ada, C, and data structure theory.

De Anza Community College June 1991 - January 1993

Coursework included C++, SVR4 Kernel programming, and TCP/IP network programming.