Christopher C. McAfee

400 Sierra St., Moss Beach, CA 94038
Resume formats: html, pdf
mcafee@mocha.com

Specialty Improving the Developer Experience by delivering tools backed by solid, unit-tested frameworks.

Experience **Apple, Inc.** 1/17 - Current

Build & Tools Automation Engineer, GPU Software

Maintained and added missing functionality to a legacy build system implemented in Perl/CGI.

- Designed and implemented a constraints-based pruner service that evaluates a list of rules and computes an aggregate action for candidate artifacts. Code was written in python with a reusable framework architecture. 100+ unit tests covered most of this code, and a dry-run prune mechanism allowed for deletes to happen easily in a testing context.
- Added stable asr behavior to the legacy build client. Feature was written in python and bridged to perl via a command-line interface. Code was unit tested and lives in a reusable module.
- Wrote a client command broadcast utility in python/paramiko to maintain our 300+ builder hosts. Prior to this tool, the team was mostly doing maintenance by hand without ssh keys.
- Currently designing a modern replacement for the legacy build system.

TangoMe, Inc. 5/14 - 1/17

Manager and Technical Lead, Tools and Automation Group

Responsible for all builds at Tango, as well as the automated end-to-end testing experience.

- Designed and implemented an extendable build pipeline that could execute any build in a consistent manner. This Pipeline was implemented as a python framework, and was deployed to engineering production for a majority of product builds. This framework accelerated our ability to add builds for new products and deliver new functionality for all builds by sharing code effectively. Framework design stressed class encapsulation and test-driven development, with 2500+ unit tests. I hired a team of 4 core python developers to help design and implement this framework.
- Recruited two key former co-workers to add VMware experience our IT department. Together we built a scalable, licensed VMware cluster of hosts deployed on Synology/NetApp/Blade hardware with 10G backplane connectivity. This cluster provided a stable build pool of 165 Linux and macOS builders with no noticeable slowdown compared with bare metal versions.
- Chartered a task force to design and implement a parallel execution path for our end-to-end test framework (see previous Tango role, below)
    - Test results were changed to aggregate results to a database server, this provided flexibility that enabled a better design for the parallel execution effort.
    - We compressed 26 hours of test time down to 2.5 hours by parallelizing across 12 hosts. This reduction in test execution time dramatically improved feedback for

    developers, and these test results now comprise an important engineering requirement for integrating new features to the main-line codebase.
    - Designed a live, priority-based view of the end-to-end test failure signal for active branches. Many wall displays in engineering now cycle through these views, large failure events are very visible and get traction quickly.
    - Introduced architectural boundaries in the end-to-end test framework to foster stronger test development. This has helped reduce cut-and-paste among tests, removed test-specific code from the harness execution, and improved test execution efficiency.
- Maintained the iOS Development Portal configuration for all products at Tango. Created device provision policies and tracked devices with group owners. Managed certificates for devices, APN push notifications, and distribution builds. Created new certificates and profiles for new products.
- Created requirements and policy documents to help new processes stabilize faster.
- Worked with IT to move critical engineering services (jenkins, jira, artifactory, svn, github enterprise) to more-stable VMware VMs. Audited backup plans, tuned filesystems/config for better I/O performance.
- Retooled the build pipeline to work with a github repository. Worked with IT to deploy a corporate version of Github Enterprise.

TangoMe, Inc. 5/13 - 5/14
Senior Test Automation Engineer

Responsible for the design and implementation of a comprehensive test harness to drive end-to-end tests that covered all areas of the Tango mobile experience.

- Designed a test harness architecture to run any test, written in any language, with a full report that captured as much state from the test result as possible.
  Harness was implemented as a python framework and is running the end-to-end test suite today.
- Test results were collected and presented to the user in html form. All server logs and client state was captured, and any core files detected were saved with stack traces written to a file as part of the test report. All results and test artifacts were published to a server so that any test result could be referenced in an email or bug report. Once this system was deployed, it was possible to debug a very complicated test failure at the log level from an email report on a mobile device, from any location. This accelerated test failure traction and fix rates for both client and server teams, and dramatically increased confidence in the features we were developing.
- This harness ran each test in a separate process to guarantee test integrity and provides a clean timeout behavior for tests that failed to complete. This execution model enabled parallel execution which happened in the second phase of development (see above role).
- For tests that did not complete in time, partial results were provided as much as possible. This has allowed the team to debug hangs in a productive manner.
- Added profile support to optimize both harness-side and test-side code paths to minimize overall execution time.
- The python code for this implementation was organized into shared frameworks that were made available to other groups in the company.
  More than 2500+ unit tests were written to cover functionality at all levels.
  Build operating systems were macOS, Ubuntu, Centos; clients were iOS, Android, Java

(server).

NookMedia / Barnes & Noble, Inc. 1/12 - 5/13
Senior Release Engineer

Responsible for all Nook client builds.

- Created and maintained a cross-platform build framework in python. Unit tests were added for each class in the framework.
- Clients included the Nook brand of devices (Android), Android for Google Market, iOS, Window8 Metro, Windows8 Phone.
- Deployed a Jenkins server for all client build status, introduced a consistent naming scheme for jobs, and created an archive process independent of the Jenkins server.
- Built out a small osx build farm when there was very little to start with.
- Helped plan and deliver a maven binary distribution process for sharing core library components among different builds. This dramatically simplified the builds for many of the client builds.
- Used maven to download and install Nook-specific Android SDK at build-time, instead of relying on engineers to install the same version by-hand on a daily basis, which was the previous behavior. This saved the client engineers countless hours of toolset management and aligned all client development to the same toolset as it changed.
- Decoupled the Nook client build from the OS build. Prior to this, any client change needed a complete build of the operating system and took a very long time. This required some creative negotiating with the OS platform build team.
- Rescued confusing bash-script driven builds for the Android and iOS clients. These scripts were very error-prone and were not committed to SCM anywhere.
- Drove the initial Windows8 Metro build system bring-up. Initial effort was x86-based, I implemented much of the ARM port. I was able to port the build system to Windows8 in about 4 weeks./li>
- Managed branch and build policies for all client groups.
- Administered the svn and git repositories for the client engineering team.

TangoMe, Inc. 6/11 - 12/11
Release Engineer

Responsible for design and implementation of a cross-platform build and test automation frameworks in a very interrupt-driven, fast-paced startup environment.

- Developed and deployed a Build framework implemented using python and scons. Unit tests were written for each class, along with system-level tests. Client builds targeted Android, iOS, wp7, Windows 7 desktop. Server build was java/ivy-based.
- Ported the build system to windows/wp7 in about 6 weeks, unit test coverage paid a lot of dividends for this project.
- Maintained a small build farm of Centos5, osx hosts.
- Cleaned up and re-deployed a floundering jenkins server. Renamed jobs and products with a consistent naming scheme, and externalized the build archive mechanism used to publish builds to the engineering team.
- Cleaned up the release tracking process for QA.
- Managed the branch strategy for the entire engineering team, communicated and updated the branch policy as needed.

- Helped triage build and test failures on a daily basis.
- Provided build assistance late at night, on weekends, and supported a remote development team in Beijing.

doubleTwist Corporation 6/10 - 6/11
Release and QA Automation Engineer

Responsible for maintaining and improving the existing build process of three client applications on osx/iOS, Windows, Android/ARM.

- Planned and executed a significant repository layout re-structuring to enable better branching practices and encourage cross-platform development.
- Introduced Jenkins as part of a CI build server to cover builds on Android, osx/iOS, and Windows.
- Wrote a Graph server to store and present performance data. Server was implemented in python using CGI and ChartDirector.
- Deployed two linux rack servers to host IT services: DNS, confluence wiki, Jenkins, Graph server, svn repository. Added a regular backup process for both servers.
- Identified the need for a VPN solution, purchased and configured a Cisco 5505 VPN router for the company.
- Served as primary network engineer during our switch from a residential dialup connection to T1 solution. Untangled some legacy network mis-configurations.
- Helped plan and execute a move of the company (25 people) to a new location during my 3rd week of employment. Set up new server room and network with attention to detail and safety of equipment. Nine months later I was again the point person for moving the bulk of our infrastructure during our second move.

Sentilla Corporation 7/09 - 6/10
Release Engineer

Responsible for creating and executing a repeatable, trackable release process for the engineering team. Product was built on three tiered platforms - Java EE & Linux on X86 server/appliance, Embedded Linux on COTS ARM hardware, and Java VM on custom ARM hardware.

- Designed and implemented a new install procedure, reducing the install time for a given release from 5 days to about 2.5 hours.
- Created a mechanism for tracking and managing changes in the base Linux OS, this had not been done before I arrived.
- Re-wrote an existing installer in Python to build and install on top of a known, managed Linux OS target. Added improved error handling and remote execution status to help debug failed installs.
- Created a tool to discover and describe the state of the target operating system configuration. This proved very useful for tracking binary releases over time, and for detecting discrepancies between developer and production and installations.
- Proposed, designed, and executed a major reworking of the build-time and run-time file system layout during the 3.0 release. Work was done on a branch over 2 months and landed without major problems in a few hours. This greatly simplified both the build and the customer experience on the target OS.

Apple, Inc. 1/07 - 5/08
Release and QA Engineer, Vector and Numerics Group

Responsible for testing and integration of the Accelerate vectorization framework and the math library (libm) for Leopard, Tiger updates, and Leopard++ for ppc, i386, x86_64, AppleTV, and iOS/arm architectures.

- Reduced the build time of a critical main project from 3 hours to 15 minutes by isolating the long part of the build into a separate project.
- Designed and wrote a nightly build system in Python. This provided nightly roots, as well as current build status for trunk and branch builds for all of the projects in the group.
- Designed and wrote a performance tracking mechanism in Python. This allowed us to track code performance for various projects in a regular way. This was a simple design, flat files were parsed into a format that ChartDirector could understand, and a CGI/JS/Python interface allowed the user to query specific parts of project performance for a given architecture.
- Extended the performance tracking mechanism to track correctness tests for our projects. Line charts tracked correctness errors vs. time (build).
- Regularly upgraded OS builds across all architectures and ran tests to monitor correctness and performance, this helped us find regressions from code checkins, OS changes, and compiler changes.
- Automated the release and approval process with interactive Python scripts, this saved a considerable amount of time.

Pixar Animation Studios / Disney 1/06 - 1/07
Automation Engineer, Studio Tools group

Designed and implemented a test harness written in Python. Test descriptions and schedules were specified in xml format, and tests were executed in parallel using a fork/exec timeout mechanism. Overall test reports were generated as html pages with pointers to performance graphs and test baseline data. The new test harness improved the engineers' ability to create, manage, and track test results in our continuous build environment.

- Developed and deployed a 2.0 version of this test harness in a way that supported the 1.0 version. Assisted engineers during this transition, and helped phase-out the 1.0 usages over time.
- Refactored an existing make system to move test dependency actions from test-time to build-time, and added make targets for commonly-used test harness commands.
- Upgraded an existing and heavily-customized Bugzilla/mysql 2.19 installation to 2.20, and again to 2.22. Added numerous schema+UI customizations in a way that enabled the whole code base to continue to accept merges from the Bugzilla main-line. Both upgrades were performed with little or no engineering downtime.

Apple Computer, Inc. 7/04 - 12/05
Release Engineer, Java Group

Primary release contact for the Java group. Responsibilities included improving automated build infrastructure, streamlining tools/processes for the engineers, and integrating changes with the main OS build group.

- Provided release support during the entire Tiger/10.4 development cycle. This included daily installs of new operating systems, compiler changes, and changes to other subsystems that the Java product depended on.
- Inherited and managed a complicated set of cvs branches. Introduced a more formal branching methodology that included naming conventions and base tags.
- Improved a 1995-era build system by carefully converting sh scripts to perl, added usability improvements like switching to HTML mail status, added JavaScript to some CGI tools, checked the entire system into cvs, and added comments to a mostly uncommented set of scripts.
- Adopted a gnu-plot graph tool from mozilla.org to use for performance tracking, and improved it to compare groups of test results across two or more machines.
- Led other engineers to standardize on more-modern scripting languages such as perl to improve productivity and increase the maintainability of the tools we were creating.
- Communicated build integration status and build failure status with the engineers in my group on a daily basis.

Scalix Corporation 2/04 - 6/04
Tools/Quality Assurance Engineer

Primary QA engineer for a 1.0 release of a web-based mail server administration application.

- Discovered a few bad security/architecture problems early enough in the development cycle so that things got fixed in time.
- Provided critical bug support and UI improvements for this product. Wrote test cases, learned the product, and tested it in various local, remote, and secure (SSL/kerberos) configurations.
- Took the initiative to set up some harder test scenarios: different versions of Linux (Redhat, SuSE), Tomcat 5.0, SSL.
- Started documenting a complicated and changing installation process in html, previously the group had no publishing of any kind.
- Tried to foster a fun environment for a short-staffed and over-worked group.

Netscape Communications / Time Warner, Inc. 12/02 - 12/03
Developer, Engineering Feedback Tools Group

Maintained existing crash-catching mechanism (mozilla.org's Talkback app).

- Introduced continuous feedback for Talkback server statuses.
- Rewrote an existing nmake build system to use GNU make, consolidated Makefile rules and streamlined the build.

Netscape Communications / AOL Time Warner, Inc. 4/01 - 12/02
Performance Engineer, Embedded Client Group

Responsible for monitoring and improving the performance of Communicator 6.0.

- Instrumented our continuous build system with performance tests, added graphing cgi support to help identify regressions.
- Helped identify and chase down performance regressions when they occurred. Championed new performance regression rules to a group of 350 open-source

engineers.

- Helped organize a schedule and policy for "sheriffs", people elected to help marshal mozilla.org cvs checkins from 350+ engineers into a single source code base.
- Designed and Implemented a dynamic build mechanism for pulling and building any module in the mozilla codebase in a minimal fashion.

Netscape Communications / America Online, Inc. 7/98 - 4/01
Developer, Cross-platform Browser Group

Helped implement the 6.0 browser front-end using XML and JS.

- Helped sanitize Netscape code for initial open source release to mozilla.org.
- Active in mozilla.org newsgroups. Encouraged private Netscape discussions to go outside the firewall.

Netscape Communications Corp. 10/96 - 7/98
Unix Browser Engineering Lead, Client Division

Team lead for the Unix front-end, Communicator

- Led the team through difficult crunch periods in a WinNT-centric development environment.
- Often represented the entire Unix client team in team-lead meetings.
- Front-line distributor of many, many bugs and features for the Unix client.
- Maintained a constant presence in comp.infosystems.www.browsers.x
- Shipped Communicator 4.x on 20+ versions of Unix.

Netscape Communications Corp. 12/95 - 10/96
Lead Developer, Server Division

Ported a Win32/MFC backend to Solaris and implemented a Motif/C++ front-end for the LiveWire SiteManager application.

- Hired and managed a contractor to assist with the workload.
- Shipped SiteManager 1.0 on 6 flavors of Unix.

Bay Networks, Inc. 9/94 - 12/95
Developer, Network Management Group

Co-developed an RMON Threshold Manager application in Motif/C++ for Solaris, HP-UX, AIX.

- Shared the work with one other person.
- Ran the first http server inside Bay Networks firewall, educated people about the emerging www experience.

Integrated Systems, Inc. 8/92 - 9/94
Developer, Xmath application

Maintained a MATLAB-like application in Motif/C++ in a team of 4.

Skills

Languages / Tools	Python, bash, scons, make, perl, JavaScript, HTML, C++, emacs.
Operating Systems	macOS/iOS, Android, Linux (Ubuntu, Centos), familiar with most other unixes. Some Windows.
Version control	git, svn, p4, cvs
iOS Developer Portal	Certificate, AppID, profile configuration and maintenance.
Engineering services	jira, confluence, artifactory, VMware, reviewboard, github, bugzilla

Education

Stanford University	MS, Computer Science	1992
Stanford University	MS, Mechanical Engineering	1991
Cornell University	BS, Mechanical Engineering	1989

Interests

Bass guitar (Rock, alternative, punk, blues), sailing, running, biking, hiking, gardening, and playing with my two boys, ages 7 and 5.
Most recent international travel was to to Niagara Falls, Canada (2016).