John Wetherill's Software Portfolio

* In developer/evangelist role at ActiveState I prepared and delivered many PaaS-focused tech talks targeted at developers and devops at major industry tradeshows including CloudExpo, JavaOne, QCon, OSCon, CloudBeat, Velocity, Cisco Toolapaloosa. Delivered several full-day in-depth developer-oriented PaaS talks at ACM, eBay, NASA, Carnegie Mellon University (Silicon Valley), and at a community college in Orange County. Visited countless high-tech companies in the Bay Area, usually with CxOs senior engineers/architects, to deep-dive on PaaS and Cloud issues. Wrote a handful of technical blogs.

* In developer role at ActiveState I enhanced the core Stackato codebase by adding support for Java7 and Tomee+, and implementing Stackato's Java debugging capabilities. Additionally fixed a number of bugs in the core product, and provided code-level support to many key F-500 customers.

* Managed and drove the implementation of the "Stackato-CPBM" connector enabling the Stackato PaaS to be orchestrated directly by Citrix's Cloud Portal Business Manager. Integration points included authentication and SSO, resource provisioning, service lifecycle, and billing/reporting. Worked with the India-based Citrix engineering team to spec out the design of the integration, and managed and mentored a Vancouver-based intern (whom we since hired at my recommendation) in building up the connector implementation. Created a screencast and wrote a blog describing the results.

* Led Agile team of 5 senior User Interface Engineers specializing in JavaScript, CSS, HTML, PHP, and Ruby. Built a server-side JavaScript framework as the foundation of a major upcoming product release, while maintaining an existing MVC web portal application built using an in-house PHP-based MVC framework.

* Built a server-side JavaScript based REST protocol analyzer and testing tool that was instrumental in testing and debugging the REST interactions between the front- and back-end.

* Conceived, architected, developed and delivered entire software stack for major AMI vendor, including messaging-based backend based on standard Enterprise Integration Patterns, nosql persistence (mongodb), Java front-end (based on SpringMVC and Play!). Managed and drove the development of the powerful Spring/MVC user interface.

* Single-handedly managed and implemented all components of entire SLDC, including deployment, continuous integration, bug tracking, source code management, test frameworks, TTD coding practices, distributed logging, database configuration, system health monitoring, backups, and event notifications.

* As Agile core Java development team member for AMI vendor, implemented auditing and rate plan management code.

* Conceived, designed, and architected "Duke's Photo Booth," an elaborate and highly regarded JavaFX-based green screen application used at JavaOne 2009 to create custom photographs of attendees with special guests and chosen backgrounds. Technologies included JavaFX, Java, MySQL, NetBeans, JavaCard, and Sun Ray. This application generated hundreds of leads and was and deemed as one of the main attractions of the event.

* To drive developer adoption of JMonkeyEngine, a powerful Java 3D rendering engine, conceived, designed, architected, implemented, and ran the "Virtual Flying Dukes" 3D coding contest where JavaOne attendees built and submitted a 3D avatar using the jMonkeyEngine Java 3D rendering engine.
The attendee solved a basic projectile motion physics problem to catch a virtual t-shirt, and was judged in ingenuity and entertainment value. Created lots of buzz. Based on Java, jMonkeyEngine, and NetBeans.

* Built event tracking application to track events in which Sun participates. Event managers, tech leads, requirements, owners, and status can all be managed online, and multiple metrics can be collected. Built with Java Server Faces, MySQL, and NetBeans.

* Re-implemented the above management application in Ruby On Rails, achieving a 90% code reduction with significant application enhancements. The app became the standard tool for booking events within Sun.

* Designed and implemented an RFID-based attendee session tracking system which, using RFID-enabled JavaCards, tracked and controlled attendance at multiple world-wide Sun events. Additionally built a sophisticated AJAX console used to manage and monitor the reader devices, servers, and database tables. The session tracking system, first deployed in 2002, is still in use and has evolved over the years, now extended to support mobile devices.

* Conceived, developed and delivered the Spinning Duke game used for several years at JavaOne, where attendees spin a virtual 3D wheel to win various prizes. Thousands of attendees used the application which is based on Java and jMonkeyEngine, and built using NetBeans.

* Designed, implemented and produced a JavaFX-based prototype Video Surveillance Security system which showed multiple camera feeds and tracked security guards carrying JavaFX-mobile cellphones.

* Built an extensive social-networking website for a local church using Ruby on Rails (see http://skylandchurch.com).

* Researched and managed the implementation of a complete RFID asset tracking system that tracks thousands of hardware assets used by the events team for various conferences and tradeshows throughout the world.

* Mentored and managed three college interns, overseeing and guiding the building of a program to integrate the JavaMedia APIs, and modifications to an existing internal application.

* Created and delivered several in-depth technical sessions at multiple JavaOne conferences.

* Architected and implemented a client-side J2ME / JXME (aka JXTA for J2ME) "treasure hunt" application for JavaOne allowing a large network of Sharp Zaurus Linux-based PDAs to join a JXTA peer group and share jar files. This included designing and coding a developer contest which challenged JavaOne attendees to write a Java application that reads the time on an analog clock at report it back to a servlet.

* Designed and implemented a client-side MIDP application as part of an architecture for trusted transactions using MIDP/J2ME interfacing with a J2EE backend. This included parsing XML streams from MIDP, managing session keys, and persisting preferences in a MIDP RMS database.

* Architected and developed an accessibility chat demo for JavaOne 2000, based on Jini, JavaSpaces, KVM (on Palm VII and RIM Blackberry devices), Servlets, Swing, and the Accessibility APIs. Enhanced this considerably for JavaOne 2001 to support wireless MIDP clients (Motorola i85s phone, Palm devices).

* Designed and implemented a Jini-enabled house in the Santa Cruz mountains, including building and configuring network infrastructure, and architecting and developing Java applications and Jini services to control and monitor multiple sensors and actuators in the house including appliances, a large salt-water aquarium, and many more. Through its sensors, actuators, and cameras, this house could be observed and controlled.

* Designed and build a powerful Geographical Information System vaguely resembling Google Earth (ten years earlier) for NEXTSTEP using Objective C and InterfaceBuilder.

* Designed and implemented an IBM/370 CMS Software Virtual Machine that completely emulated the entire IBM/370 instruction set.

* Created an "IPL (Initial Program Load) Tape" for IBM Mainframe system failure. Upon machine crash, the system immediately boots from the tape which loads a tiny app in an unused memory frame, and executes the app which proceeds to dump all memory to the remainder of tape.

* To showcase JMonkeyEngine, built a 3D presentation tool with jawdropping visuals and navigation effects.