Eric Dobbs

eric@dobbse.net
Boulder, Colorado

Creating resilience in complex systems and software teams.

- Personal and professional integrity
- Demonstrated ability to identify and deliver systemic improvements
- Exceptional written, verbal, and visual communication
- Enthusiasm for learning and teaching
- Extremely versatile
- Over two decades in web development

Manager of Software Development – DevOps   Itron, Inc. (was Comverge)
December 2013 to March 2018

Impact   Applied diplomacy and role-modeling to streamline communications and greatly increase collaboration between QA, Development, IT, and Solutions Delivery.

Increased productivity for 20 developers and 4 technicians. Drastically reduced build times and deploy times while also increasing reliability.

Successfully lead adoption of Docker throughout the organization.

Leadership   Fostered deep respect and trust from my team. The director of development received repeated requests that they remain on my team. Several senior devs expressed specific interest in joining my team.

Nurtured emotional safety and courage. Established weekly retros and bi-weekly one-on-ones where we focused on social and emotional aspects of software development. Modeled speaking hard truths and admissions of failure. Gave specific and concrete applause frequently.

Impacted the entire development team when my fellow managers also adopted retros and one-on-ones.

Mentorship   Successfully encouraged professional development in all levels of technical experience: high-school intern, junior dev fresh out of a code school, intermediate dev transitioning from QA to engineering, and two senior developers.

One exceeded their goal of submitting a proposal when their talk was not only accepted, but also well received at Rocky Mountain Ruby 2015.

DevOps   Reduced build times from 8 hours to 30 minutes by a combination of expanding the jenkins cluster from 20 to 70 VMs, and identifying and fixing the worst-offenders in the test suite.

Improved our deploys with significant increase in consistency and reducing time-to-deploy from one month to a few hours. Instituted standardized deploy process with team-wide adoption of runbooks, and made semi-automated deploys into a consistent part of the software

delivery life-cycle.

Social and technical engineering of Docker adoption. Enabled early adopters to experiment and learn containers in development. Deployed first containers into our Continuous Integration cluster. Created first containerized tests and deploys. Designed and evangelized effective patterns of use. Planned, negotiated, persuaded, and executed deployment in stages from dev, to test, QA, UAT, and production.

Senior Software Engineer   Mobilecause
February 2012 to November 2013

Impact	Fostered a culture of continuous improvement through honest and courageous communication, by speaking hard truths, and applauding when others did the same.
Emergent Design	Refactored all outbound messaging, the core purpose of the software, into a delivery service without disrupting any existing message flows. Thanks to this layer of abstraction, only two weeks later we were able to easily improve throughput just in time for a customer's big campaign messaging over 330,000 numbers.
	Introduced a command dispatcher and handlers for message processing.
	Created a foundation for modularized html, css, and javascript which enabled migration of the existing UI from a home-grown CSS-soup to Bootstrap.
	Refactored one particularly messy controller with just a handful of method extractions. A few weeks later those changes enabled a colleague to clarify the routing and models and complete the cleanup without me.
Scale	Profiled outbound messaging performance and throughput. Identified garbage collection and IO as primary bottlenecks.
	Threaded message delivery for an immediate twelvefold increase in SMS message throughput. Introduced retries and fallbacks to ensure message logging and persistance and expand the ability to identify and diagnose failures.
	Enabled message workers to run under JRuby as a strategic move to open other performance opportunities via JVM based tools and languages.
Front-end Engineering	Streamlined unit testing of javascript front end code using Jasmine.
	Created dynamic and thoroughly customizable fundraising thermometer and pledge wall for use in high-profile gala events. Produced the SVG graphic for dynamic display and employed Ember.js for updates.
Methodology	Full-stack agile methodologies including continuous integration, full-time pair-programming, retrospectives, etc.

Software Engineer   Pivotal Labs

November 2010 to September 2011

Impact	Initiated an informal usability analysis for Groupon's editorial tools via direct observation of editors and deal creators.
	Consequently created dramatic improvements in deal creation and editorial workflows. Streamlined communications between sales and editorial teams by integrating with Salesforce.
Learning	Mastered Ruby within a month. Immediately effective maintaining and updating existing Rails application for Groupon Merchant Center team.
Scale	Refactored Groupon's merchant analytics into an independent service-oriented architecture. Improved largest deal analytics reports from a query time of 7 minutes down to 15 seconds.
Methodology	Full-time pair-programming, behavior-driven design (BDD), continuous integration, collective ownership, small releases, planning game, retrospectives, on-site customer.

Consultant   **bivio Software, Inc.**
December 2003 to May 2010

Impact	Changed the company culture: Testing became a core value. Extended internal unit testing framework to simplify testing core business logic. Number of test files grew from 85 to 585 or 20% to 60% coverage.
	Drove successful adoption of XHTML and CSS for web interfaces, and wikis and blogs for content management, both internally and for bivio's customers.
Learning	In order to build a python interface to a C++ scientific visualization toolkit, learned on demand the toolkit's API, C++, Qt, PyQt, OpenGL, PyOpenGL, NumPy, python distuitls, SIP, and GNU autotools.
	Kept pace with bOP, an extremely complex and abstract web framework under continual evolution: The framework grew from 390 classes to 900+ through 720 releases with a median days-between-release of 1.
Teaching	Trained programmers with wildly diverse levels of experience from high-school student interns with no previous programming experience to veteran C++ and Java developers with decades of experience.
	Trained customer support staff in investment club accounting software, managing customer expectations, and in the use of CRM systems. Most of the training was by teleconference or online via WebEx or GoToMeeting.
	Lead a three-day bOP training course for a client's entire development team whose programming skills ranged from systems administrators to senior embedded systems programmers. Also created the overall training plan and most of the supporting documentation. Received strong positive feedback from all participants.
Client Impact	Saved face for the company when the president forgot a meeting with the biggest client. Took his place offering valuable contributions around complex design and security issues. Client was seriously impressed

with my ability to step in unprepared and to turn a potential train wreck into a productive meeting.

Complexity   Maintained and extended applications in diverse and complex domains including a state-wide insurance accounting system, a non-profit grant application system, a formal balloting system for industry standards, an alumni association membership system, and an investment club tax accounting system.

Aikido Instructor   **University of Colorado Recreation Center**
August 2005 to Present

Impact   Three students gave unsolicited feedback that my teaching changed their lives.

          Increased typical class enrollment by 50%. Inspired many students to join Boulder Aikikai.

Third Degree Black Belt
Boulder Aikikai
May 2013

Bachelor's of Environmental Design, emphasizing computer methods in design
College of Architecture and Planning, University of Colorado at Boulder
May 1994