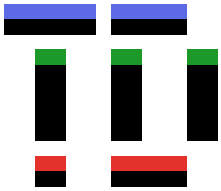

Tim Doerzbacher

Software Engineer

412•758•0179
tim@tim-doerzbacher.com
tim-doerzbacher.com
github.com/timmd909/

Technologies

- Angular
- Ansible
- Apache
- Bootstrap
- CodeIgniter
- DataTables
- Durandal
- ejabberd
- FreeCAD
- Git
- Grep
- Grunt
- Jenkins
- Jira
- jQuery
- Knockout
- MariaDB
- MySQL
- Nagios
- Nightwatch.js
- NodeJS
- PHPUnit
- PostgreSQL
- Python
- QUnit
- Rasbian
- React
- RequireJS
- Selenium
- Subversion
- Symfony2
- Thrift
- Ubuntu
- WebDriver
- WordPress

Skills

- Agile Development
- Hobbyist Roboticist
- Database Design
- Graphic Design
- LAMP Environments
- Linux Administration
- /Regular Expressions/
- Website Design

Languages

- Bash
- C/C++
- Erlang
- HTML/CSS
- JavaScript
- LESS
- PHP
- Python
- SASS/SCSS
- SQL
- Yaml

Operating Systems

- GNU/Linux
- Mac OS X
- Windows

Work History

Now

Opus One Interactive

Senior Software Engineer

- Created continuous integration environment to ensure build quality
  - Built off of Gitlab's CI/CD functionality using multiple Gitlab runners for testing and deployment
  - Automated PHPUnit backend tests and Karma front end tests
  - Nightwatch.js was used to create headless Chrome integration tests
- Leading the graduation migration from legacy PHP code to PSR compatible code using Composer libraries
  - Moving to PSR compatible code was done for better code analysis, coverage, and code completion for quicker development.
- Fully automated release process
  - Symfony based console commands are used to initiate new releases via Jira, Gitlab, and Slack integrations.
  - After initiating a release, all the developer has to do is approve 2 merge requests and wait for the new release to be baked and ready for deployment.
  - Updates are then pushed live with a single click per customer allowing greater flexibility strafing releases.
- Lead effort to reskin and improve the UI/UX and underlying software architecture.
  - Used Knockout, Bootstrap, and RequireJS on the front end to effectively create an ad hoc Durandal
  - Switching to Bootstrap alleviated styling inconsistency issues. Bootstrap's excellent documentation also frequently answers the question of "How should X look?" the majority of the time, making development and reskinning much easier for developers.
  - Twig templating was used on the backend to simplify and speed development times.
  - Previous architecture was using bare metal PHP templates with a mix of jQuery and Prototype to handle more complex UI requirements
- Design and implementation of new DataTables based list views
  - Complete rewrite of list based pages in the software to use DataTables and server side JSON
  - New view based data structures and rendering pipeline architecture was developed to generate the JSON in a flexible and consistent manner.
  - This flexibility was expanded to also support PDF and CSV output formats for report generation.
- Overhaul of PDF report and business document generation
  - Used Twig to generate HTML which is then fed through TCPDF to generate business documents. Using this approach, we'll be able to more easily migrate to another PDF generator when it becomes necessary.
  - Simplified and generic report pipeline was created to simplify the layout and generation of all row based reports.
- Database clean up with Doctrine ORM
  - Used Doctrine migrations to progressively sync Doctrine's expectations with the underlying schema.
  - Removed or refactored code that relied on invalid IDs to implement tri-state logic.

2016

2016

Intermedix

Front End Engineer / Release Engineer

- Redesigned and implemented new release engineering process.
  - Fully automated publishing of release assets.
  - Automated release emails by using ChangeLog.
  - Organized and developed a branching strategy for release and development.
  - Redesigned the Grunt project files into a modular structure and more easy to maintain structure.
  - Integrated Stash with Jenkins to enforce testing before merging.
  - Created automatic release process that is triggered by new commits or merges to the master branch.
  - Built and maintain Jenkins build slave.
  - Added additional linting rules and code to enforce consistency in the code.
- Oversaw development of new major release of our product.
  - The original architecture was of a large, monolithic package which was prone to regressions.
  - Chopped the package in many smaller, more focused packages to allow downstream implementers more flexibility with upgrading and feature sets.
- Designed and implemented visual regression tests to improve continuous integration.
  - The testing framework is shared, making any updates or the testing framework to be affected immediately in all existing and new modules
  - All demos in the documentation are automatically checked for visual regressions.
- Evaluating and integrating additional third party libraries in the shared toolkit.
  - Automated patching and tweaking (when necessary) so that all libraries can be used seamlessly with RequireJS.

2014

2015 – 2007

PiCon Web Design & Marketing

Subcontractor

- Developed a web-based commercial employment application designed for the trucking industry, yet generalized enough to be used by any industry.
- Rewrote .NET and Microsoft based websites for use on LAMP servers.

2014 – 2013

Walking Thumbs

Full Stack Developer

- Integrating PHP and ejabberd based nodes and their respective database nodes into a custom XMPP-based messaging system.
  - Apache Thrift was used to enable direct communication between the PHP and ejabberd nodes.
  - ejabbard calls originally queried the database through Thrift calls to PHP. Rewrote to have the ejabberd nodes directly query the PostgreSQL cluster when possible.
- Created syncing algorithm for finding other users on the system.
  - Hashing was used for matches instead of unencrypted contact information.
  - This was done to protect privacy concerns while still being able to match other users.
  - Normalized email and phone numbers to prevent false positives and maximize matches. This was extremely important since all contacts were stored only as hashes.
- Improved and expanded automated testing suite; increased the code coverage and reduced the execution time.
- Handled all server related tasks to support other developers working on the Android and iOS clients.
  - Set up Nagios on an AWS instance.
  - Installed, configured, and upgrading of servers.
- Built automated server build scripts for use by the rest of the team, decreasing the server-side ramp up time for new app developers.
- Handled overseeing implementing company websites.

2013 – 2008

Kb Port

Server & Web Developer

- Handled UI design and implemention on several major software revisions.
  - Utilized HTML5/CSS3 for maximum compatibility in the future.
  - Previous versions were hacked to work on Internet Explorer with a proprietary Active X control.
- Developed RAID-based archival system for compiling video sessions from network connected video recording products.
  - System has up to 14TB of storage allowing hundreds of days of recording video to be quickly searched, indexed, and shared.
  - Automated backups to the archival unit eases consolation of student videos and prevents the video records from filling up.
- Developed customized customer management and inventory management software.
  - Features included allowing automated upgrades to systems out in the field to facilitate up-selling new products and features.
  - The system interfaced with copy protection code on the systems to prevent unauthorized cloning.
- Designed a new video package format to allow exchange of data between the company's various products.
  - Designed to be easily extendable at future times to allow any type of metadata or otherwise to be added without breaking backwards compatibility.
  - Utilized OSS tools for maximum inter-compatibility on different platforms.

References

References available upon request.