Tim-Doerzbacher.com

I've developed a skill set varied enough to take any web-based idea from initial concept, to design, and all the way through development to the finished implementation. I'm most comfortable with open source software.

Contact Info

1 • 412 • 758 • 0179
136 North South Drive
Pittsburgh, Pennsylvania 15237
tim@tim-doerzbacher.com

What I Can Do

Here's an annotated list of software, languages, and so forth that I've used. The battery icons indicate my level of comfort with a particular item.

Technologies

- 🟦 Agile Development
- 🟦 Building robots for fun
- 🟩 Database Design
- 🟫 Graphic Design
- 🟦 LAMP Environments
- 🟩 Linux Administration
- 🟦 /Regular Expressions/
- 🟩 Website Design
- 🟦 DurandalJS
- 🟦 Eclipse
- 🟫 ejabberd
- 🟫 Facebook APIs
- 🟩 FreeCAD
- 🟦 Git
- 🟦 Grep
- 🟦 Grunt
- 🟩 Jenkins
- 🟦 jQuery
- 🟦 Knockout
- 🟫 MediaWiki
- 🟦 MySQL/MariaDB
- 🟫 Nagios
- 🟩 NightwatchJS
- 🟩 NodeJS
- 🟫 Phabricator
- 🟩 PHPUnit
- 🟫 PostgreSQL
- 🟥 Python
- 🟦 QUnit
- 🟦 Rasbian
- 🟫 React
- 🟦 RequireJS
- 🟩 Selenium
- 🟦 Subversion
- 🟩 Symfony2
- 🟫 Thrift
- 🟦 Ubuntu
- 🟫 WordPress

Languages

- 🟦 Agile Development
- 🟦 Building robots for fun
- 🟩 Database Design
- 🟫 Graphic Design
- 🟦 LAMP Environments
- 🟩 Linux Administration
- 🟦 /Regular Expressions/
- 🟩 Website Design
- 🟦 DurandalJS
- 🟦 Eclipse
- 🟫 ejabberd
- 🟫 Facebook APIs
- 🟩 FreeCAD
- 🟦 Git

Skills & Techniques

- 🟦 Agile Development
- 🟦 Building robots for fun
- 🟩 Database Design
- 🟫 Graphic Design
- 🟦 LAMP Environments
- 🟩 Linux Administration
- 🟦 /Regular Expressions/
- 🟩 Website Design

Operating Systems

- 🟦 Agile Development
- 🟦 Building robots for fun
- 🟩 Database Design

Where I've Worked

2017 — Present

Opus One Interactive
Senior Engineer

- Added continuous integration tests to our build process
  - Integration tests are run using custom GitLab runners
  - Configured runners to run Selenium headless using Xvfb
  - NightwatchJS is used to drive the Selenium 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
  - Used Symfony based console commands to automate the creation of release branches, and their associated merge requests via Gitlab's API
  - Upon successfully passing all tests and every thing is merged, a release tag is automatically cut and pushed.
  - Added Slack integration to alert developers to the progress of the release process.
- Created and implemented roadmap to move the UI off of legacy CSS towards using Bootstrap 4 for styling
  - I did any necessary adjustments and modifications to Boostrap 4 to make it appear similar to the legacy styling.
  - Switching to Bootstrap alleviated styling inconsistency issues caused primarily from conflicting CSS rules in the old styling.
- Added Bootstrap support to gradually move away from legacy libraries
  - Previously jQuery and Prototype were used to provide interactive functionality.
- Database clean up
  - Previously, no foreign key checks were used causing database inconsistencies
  - Refactored code that relied on invalid IDs to implement tri-state logic
  - Added foreign keys and tests to ensure database sanity

2014 — 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.

2013 — 2014

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.

2008 — 2013

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.

Education

2005 — 2009

Art Institute of Pittsburgh

Graduated with Bachelor degree in Graphic Design

References

References available upon request.