Kris Hicks

San Francisco
krishicks@gmail.com (mailto:krishicks@gmail.com)
krishicks.com (http://krishicks.com)

tl;dr

10 years of experience as a software engineer. I have experience and a continuing interest in solving the sort of problems Go is suitable for, continuous integration/delivery, build and release, and product/project management. I've done test-driven development and various flavors of agile since 2009, including full-time pairing.

Timeline

2007

Started teaching myself software engineering with Ruby.

2009

Joined ThoughtWorks as a consultant and gained VCS, TDD, CI, and pairing experience, in addition learning how to write well-designed software.

2011

Joined Pivotal Labs where I did more consulting in the same spirit as at ThoughtWorks, and also gained PostgreSQL experience.

2013

Joined DaisyBill, a startup, to take a break from consulting.

2014

Joined Pivotal to work on Cloud Foundry, where I gained experience in Go, BOSH, Cloud Foundry, deployment ("build and release"), distributed systems, product and project management.

Recent work

- `yaml-patch` (https://github.com/krishicks/yaml-patch) : a library for creating RFC6902 ("JSON Patch") patches, but in YAML

- `pg2mysql` (https://github.com/pivotal-cf/pg2mysql) : a library for migrating data from PostgreSQL to MySQL

- `pcf-pipelines` (https://github.com/pivotal-cf/pcf-pipelines) : a collection of Concourse pipelines for installing and upgrading Pivotal Cloud Foundry and its addons

What makes me tick

- Delivering business value first and foremost
- Solving problems with a pragmatic approach to testing and implementation
- Deep-dives into code to root out a bug or improve understanding of the code
- Building new systems, refactoring old ones
- Teaching people about Git and Git workflows, Vim
- Improving the interview process for software engineers
- Improving my ability and the ability of others to deliver software

What I look for in a workplace

- Feedback-friendly culture with short feedback cycles
- A working environment where asking questions is seen as a good thing
- Ability to use whatever tool is most suitable to solve a problem
- Standing desk (I've used standing desks exclusively since 2011)
- Relative quiet (vast, loud, open floorplans packed with people aren't for me)
- A robust software engineering practice using some kind of project management tool (e.g. Pivotal Tracker) or the desire to do so
- "Agile" development process; basically any process that includes short iterations, frequent production releases and quick feedback rather than a waterfall process