Sumeet Agarwal

sumeet.a@gmail.com
(937) 657-8614

Experience

Remind, San Francisco, CA

Software Engineer, 2015 – Current

Programming languages: Ruby, Go, Node, a little frontend JavaScript
Hacked a lot on the web API used by the web desktop, Android and iOS apps
Built scalable messaging backend used for teacher announcements and chats
Worked on a product team to build oversight and data synchronization tools for school and district leaders
Salesforce data loading (again)

Yelp, San Francisco, CA

Software Engineer, 2010 – 2014

Programming languages: Python, a little JavaScript
Hacked on the Python web backend, mostly rendering HTML pages for a multi-page web app. My team built one of the company's first microservices.
Led development on Yelp Deals, Gift Certificates and Yelp (Payment) Platform
Worked on bulk email infrastructure and performance, Yelp for Business Owners, Salesforce data loading

Sogeti, Dayton, OH

Consultant — Corporate IT, 2009 – 2010

Programming languages: Python, JavaScript, a little PHP
Rebuilt the company website and implemented a CMS
Administered Solaris and Linux servers
Built internal web applications and tools, often communicating with Windows or Oracle E-Business Suite to interact with employee data
Implemented a web-based single site authentication system that worked on top of Active Directory

Skills

I used to say my main language was Python because it's mostly what I used before and while working at Yelp. I use much more Ruby and Go at Remind these days, so these days it's probably better to say those three are my mains.

At Yelp, which was a larger engineering organization by the time I left at ~300 engineers, I spent much of my time working with product and engineering to break apart large projects, dividing and parallelizing work between engineers with overlapping skillsets.

At Remind, a much smaller org with ~40, I've typically been the only dedicated backend engineer on a team and have been less focused on organizing work for team members. Team collaboration is much more about discovering the business impact of our work, or coordinating with client devs on how the app will work, or how the API should look.

I value fast and thorough unit tests, refactoring, small methods, small objects, clear intention-revealing code and explicit up-front input validation. I am very sensitive to and try my best to fix or prevent slow tests, unwieldy and hard to understand code, bugs, application crashes and data corruption.

I'm very comfortable with my tools on a computer: I'm a 150 WPM typist and very fast at using Vim and the UNIX shell. I've been doing TDD for several years and am used to producing well-factored code quickly.

I wrote two testing libraries: a mocking and stubbing tool called expect (http://github.com/sumeet/expect) and a Python clone of VCR for Ruby for testing remote HTTP calls.

College

Boston University – Boston, MA, 2008

B.A. in Computer Science

Before College

I'm including this section to give some background on the deep connection I feel with the computer. It's some of my experience up to the end of high school.

I became interested in computers at age 5 after learning how to navigate the DOS command line by watching my cousin, because he would never explain what he was doing. I learned how to launch my favorite games and taught my dad how to launch WordPerfect.

I taught myself how to touch type when I was 8.

My first big software project was a peace-keeping Eggdrop botnet for EFNet IRC using Tcl. That's also where I learned about UNIX, DOS attacks and remote root vulnerabilities. If you're not familiar that world, think AOL chatrooms and punting, but the big leagues.

I implemented auto-away for Adium because I missed the feature from virtually every other IM client. My patch is still part of the project.

I worked on a PHP app with a team of Internet friends called VGMix. It was like SoundCloud for video game music remixes, built on 2003 technology. Musicians could upload their tunes and receive feedback on completed pieces or works in progress. It included a gamified reward and badge system for musicians and reviewers, music downloading, a chatterbox, and web forums.