Aaron Kuehler - Resume

> Always remember, our purpose is not to "write code", but rather add value to the businesses.

Skills / Experience Summary

- Web Application Development (Ruby/Rails, Elixir/Phoenix, and Standard Web Technologies - JavaScript, CSS, etc)
- API Development (Ruby/Rails, Elixir/Phoenix)
- Continuous Integration and Deployment Automation
- Team Growth and Leadership through agility, Lean Manufacturing Principles, and Extreme Ownership
- Individual Mentor / Mentee

Experience

Bauer Xcel Media

Technical Team Lead / Senior Software Developer

February 2015 - Present

At Bauer, Technical Team Lead is a role which augments a member of the development staff's role slightly. In addition to contributing to the development process, they are required to act as force multipliers. They are responsible for the augmentation of team effectiveness and efficiency. They inspire change in individual and team behavior to achieve this goal.

During my time as team lead we've reduced average Story cycle time by 34.37%. We've used pair programming as a means to build trust and rapport between teammates - removing waste in unnecessary review processes and improving morale. We've trimmed waste and refocused the team to reduce WIP and plan more appropriately to the constraints of our pipeline. We are able to react to unplanned work with minimal risk to commitments. We more predictably delivered on commitments which has built trust with stakeholders - planning and negotiations run more smoothly as a result.

During an internal "Hack-a-thon" in 2017, I rallied a team of 3 developers behind the idea of deployment automation. In two days we completed work which made on-demand application review instances - review instance per pull request - and continuous delivery to production a reality at Bauer. The work-products of these two days alone saved the company 50% in month-over-month hosting costs.

Sizmek (formerly PointRoll)

Software Developer

January 2015 - June 2016

In January of 2015, Pointroll was in the midst of a turbulent reorganization. I was brought on to shared my experience and ideas on becoming a more agile team; changing the mind-set away from delivering code to delivering value to the business. I helped shape development workflows and deployment pipeline to help facilitate the concurrent development by varied teams across a multitude of code bases. I helped create, contributed to, and ran the PointRoll Community or Practice. A bi-weekly lunch-and-learn type talk series designed to distribute knowledge and increase cross-team interpersonal relationships. Pointroll was acquired for its customer base by Sizmek in November of 2015 for $20 million. I was part of a skeleton crew who "kept the lights on" until June 2016 when the PointRoll systems were retired.

Hoopla

Software Developer

July 2012 - January 2015

In July 2012 I became the 3rd full-time developer hire at Hoopla. I was brought on to augment the development capacity of the team as the product grew into larger markets and acquired bigger clients like: LinkedIn, Zillow, Angie's List, and Ring Central. I would spend most of my first 2 years working on the core web application and external event processing integration with Salesforce. Later I would lead the development effort of the Native mobile version of the Hoopla platform.

Artisan (formerly AppRenaissance)

Senior Software Developer

July 2011 - July 2012

As the 3rd employee of, then, AppRenaissance I helped grow a small mobile-developers-for-hire into a small mobile products/platform company. I spent most of my time helping clients with product and business development. Later we would start to change focus to providing services and products for mobile developers.

Infor

Software Engineer

May 2009 - July 2011

In May 2009, Infor was in the midst of re-architecture of its ERP and logistics management software. This re-architecture saw an old monolithic, database driven application transformed into nearly two dozen independent, distributed services. As the member of many teams within the team, I adied in the definition and implementation of several of these components - from Requisition and Orders to Accounts Payable modules. Early on I would establish the "Brown Bag Club"; an opt-in lunchtime discussion forum and knowledge sharing opportunity.

Oracle (formerly AdminServer)

Application Engineer

February 2006 - June 2009

Initially worked on the flagship Life & Annuity insurance policy administration system. Eventually I was asked to join the technology skunk-works and performance teams to lay the foundation for the next-generation of this product. Later on I would be asked to lead the "backend" development of the next-generation of the policy administration system.

Open Source Contributions

heroku-cli-buildpack

`https://github.com/Thermondo/heroku-cli-buildpack`

Installs the Heroku toolbelt on a heroku dyno

keyword_parameter_matchers

`https://github.com/terryfinn/keyword_parameter_matchers`

RSpec matchers for method keyword parameters.

githug

`https://github.com/Gazler/githug`

Githug is designed to give you a practical way of learning git. It has a series of levels, each requiring you to use git commands to arrive at a correct answer.

Projects

futurist

`https://github.com/indiebrain/futurist`

An implementation of the future construct, inspired by Celluloid's block based futures, which uses process forking as a means of backgrounding work.

backbone-elasticsearch

`https://github.com/indiebrain/backbone-elasticsearch`

Adapters and Utilities to interface Backbone.js with ElasticSearch

OmniAuth Doximity OAuth2

`https://github.com/indiebrain/omniauth-doximity_oauth2`

An OmniAuth (`https://github.com/intridea/omniauth`) OAuth2 strategy for Doximity (`http://www.doximity.com/`)

Talks

Git Internals

`https://github.com/indiebrain/talks/blob/master/git_internals/git_internals.org`

Does git's user interface seem cryptic? Are you often confused about when you should use 'checkout' vs 'reset'? Does 'rebase' feel scary? This talk explains the inner workings of git and sheds a bit of light on how the internal structure of git as a data store influences its user interface.

Elsewhere

- `http://www.aaronkuehler.com`
- `http://www.github.com/indiebrain`
- `http://twitter.com/indiebrain`

Education

West Chester University of Pennsylvania

Bachelor of Science, Computer Science Informantion Assurance Minor

January 2006

Graduating Magna Cum Laude, I achieved the Dean's list in 2005 and 2006, was awarded the Honor of Academic Excellence in 2006.

Research

Small File Affects on Hadoop Distributed File System

- White Paper - `https://www.slideshare.net/slideshow/embed_code/key/S4XYiY0a4m0n8`
- Presentation - `https://www.slideshare.net/slideshow/embed_code/key/9oo7oSckHxMTHX`

The Hadoop Distributed File System is a high throughput distributed File system designed to accommodate large data sets; average file sizes in the gigabyte-terabyte range. However when a data set is composed of large amounts of small files, say in the kilobyte range, the storage system's semantics introduce hight amounts of overhead in terms of file system block storage and read latency. This paper explains the architectural attributes which cause these problems and examines techniques to mitigate their impact when working with data sets comprised of large numbers of small files.