Jan Nelson

• jn123456789@gmail.com • https://github.com/jn123456789

About Me

I am a seasoned veteran of the industry with a versatile skill set and a drive to always be learning more. I have a passion for tackling hard problems head-on and rooting out complexity wherever it resides. Whether adding a feature, tracking a bug, deploying infrastructure, or hammering out process with my colleagues, my focus is on developing a complete and sustainable solution to the challenge before me. I believe in the power of the team over the individual, and strive always to share knowledge and strengthen collaboration with my fellows. I have two goals in my professional life: to build excellent software systems, and to help everyone around me to be as excellent as they can possibly be.

Experience

Software Engineer at Minerva Project — Apr 2015 – Present

The Minerva Project is a university with a radical new approach to teaching that makes heavy use of web technology, both in teaching classes using the video conferencing environment known as the Active Learning Forum (ALF), and in its admissions process. My major contributions include the following:

- A distributed web service for recording classes conducted on the ALF. A master process (written in Node.js) communicates with clients via HTTP and launches EC2 instances to host worker processes that fulfill client requests. Worker processes (also written in Node.js) communicate with the master process via Amazon SQS, coordinate selenium, chromedriver, xvfb, and ffmpeg subprocesses to connect to the class and record it, then compress the resulting video and upload it to S3. The service has been used to record over 2,000 classes to date with a greater than 99% success rate.
- A Django/React application used by our faculty to review and grade the students' participation in class based on the aforementioned recordings. The application allows a user to play back the video from the class, jump around to parts where a particular student was heard speaking, enter grades for their participation, and write feedback comments for students to read.
- A Django/React application used by our admissions staff to evaluate videos submitted by prospective students. The application plays the video recorded by the student, and presents to the reviewer a decision tree in the form of a series of multiple-choice questions. The reviewer's decisions are used to compute a complex multi-dimensional score that is fed into the student's overall evaluation.
- A redesigned back end for our custom CMS for course materials, which our academic staff uses to author course syllabuses, lesson plans, assignments, etc. In its prototype form, the application was built on Sinatra (Ruby) and MongoDB. I rewrote it using Django and MySQL, built out additional functionality upon that foundation, and built a comprehensive suite of integration tests for it, as well as deployment scripts.
- In addition to application development, I carry substantial responsibilty for deployment and operation of all of our applications on AWS.
- Along with all of my colleagues, I'm directly involved in interviewing engineering candidates and making hiring decisions.

Director of Engineering at Learnist — Feb 2012 – Mar 2015

Learnist was a social knowledge-sharing company that produced a website as well as mobile applications. My major contributions included the following:

- Managed a team of 10–15 engineers at any given time, including mentoring the skills of junior engineers, facilitating technical discussions within the team, and making the final call on all hiring decisions.
- Served as the ultimate responsible individual for engineering issues as they affected people in other groups within the company.
- Personally oversaw all deployment and operation concerns in our AWS infrastructure. Authored a suite of user-friendly command-line tools for quickly and easily launching a new EC2 cluster with a copy of production data, scaling that cluster up or down, or terminating it. This enabled members of my team to do extremely rapid and decentralized iteration on their features, as they could deploy their code at any time to a private sandbox environment that looked just like production, with just a few simple commands.
- Authored large parts of a Redis-backed cache for ActiveRecord objects that seamlessly handled concerns such as automatic lazy caching of objects and automatic cache invalidation.
- Personally implemented large swaths of back end code for the website: API controllers, business logic modules, ActiveRecord models, Resque jobs, etc. Heavily optimized all of the above with the help of custom instrumentation and services like New Relic.
- Continually refactored code contributed by myself and others, discovering latent architectural patterns along the way, crystallizing those patterns into tight and cohesive interfaces, and sharing the learning with my team.

Director of Engineering at Grockit Jun 2008 – Jan 2012

Grockit produced an online environment to help students prepare for exams such as the SAT and GMAT in a fun and social game format. The Grockit trademark and operations were sold to Kaplan in 2013, and it is still in operation today. My major contributions included the following:

- Managed a team of 8-12 engineers, which entailed recruiting engineers, mentoring their skills, and facilitating group dynamics.
- Worked with our product team to define feature requirements, using the Agile Development method of expressing requirements in terms of "user stories".
- Managed our AWS infrastructure and the deployment of code thereto.
- Authored large amounts of code in the Rails 2.3 application that was our primary website.
- Contributed to our synchronous, multi-player game server, written in Java with Hibernate and Guice, and to its front-end code, which was written in a custom Javascript framework that did dynamic reactive rendering of HTML views, foreshadowing current technologies like Backbone and React.

Software Engineer at DataPower / IBM Apr 2000 – Jun 2008

I was the first employee at DataPower, which produced a line of network equipment for blazing-fast processing of XML Web Services protocols. From 2000 to 2005, the company grew from two people to nearly a hundred, and was eventually acquired by IBM. IBM continues to produce and market our products to this day. Memorable contributions:

- A highly optimized XML parser built for reading and writing directly to network sockets. This was written in C using extreme optimization techniques to avoid ever polling the network, thus avoiding any blocking behavior. Raw inline assembly code was used in certain key places to optimize the most critical operations.
- An optimizing compiler for the XSLT language. Interesting aspects of this project included an AST parser and optimizer (optimizations included constant folding, common subexpression elimination, and branch elimination), a single-static-assignment (SSA) intermediate representation supporting further optimizations (dataflow analysis, register allocation, etc.) and code generators for both x86 and SPARC targets.

Software Engineer at Art Technology Group (ATG) Sep 1998 – Apr 2000

ATG led the vanguard of companies doing e-commerce on the web in the late 90's. My role was initially to provide on-site support for customers, and later to implement some ancillary features, such as an adapter to connect our server application (ATG Dynamo) with a third-party CMS (Documentum). All in all I was too green at the time to contribute much to the success of ATG, but this is where I learned a great deal about professional software development, including techniques like source control, automated testing, and release engineering, but more importantly, making a conscious commitment to enabling customer success.

Skills

- Python/Django/Celery (1 year)
- Ruby/Rails/Resque (8 years)
- Javascript (8 years) / Node.js (1 year)
- SQL (15 years)
- Redis (6 years)
- AWS (EC2, S3, OpsWorks, SQS, etc.; 8 years)
- Unix administration, shell scripting (20+ years)
- Past experience: C, C++, Java, x86 assembler (12 years)
- Test-Driven Development
- Object-Oriented Domain Modeling
- Process Automation
- Continuous Integration / Continuous Delivery
- ETL (Extract/Transform/Load) techniques
- Recommendation Systems
- Compiler Design and Implementation
- Graph Algorithms

Education

Massachusetts Institute of Technology 1994 – 1998

S.B., Electrical Engineering and Computer Science

Areas of focus:

- Algorithms and Data Structures
- Parsers, Interpreters, and Compilers
- Recommendation Systems