

2014 in Review: A Year of Learning

Posted At : January 1, 2015 4:15 AM | Posted By : Cutter

Related Categories: Google Chrome, Bootstrap, IronWorker, Legacy Code, Development, CSS, AngularJS, Application Setup, NodeJS, JavaScript

2014 has been an outstanding year for me, in many ways, but perhaps one of the most important things (besides my family) has been continuing to do what I love. I'm passionate about development, and constantly working to learn new things. This is important for any developer, as our world changes so quickly today. New standards, new languages, new frameworks, it's a consistent onslaught of new ideas, and impossible to learn each and every one, but important to get exposure none-the-less.

The early part of the year I was still maintaining a large legacy application. We were in the final stages of migrating some very old pieces of the application into new framework architecture (**FW/1**) along with a new look (based on **Bootstrap 3**). When you're working on legacy applications, there are rarely opportunities to dive in to new things, so that project was a nice nudge to explore areas previously untouched. Back in April, though, I took on a new position that had me doing nothing but non-stop new product development. Not only was this a great switch, but the particular tasks I was given had me working with technologies with which I had little or no exposure, and often without a team peer who could mentor me, as many of the technologies were new for the company as well.

Historically, I'm a server-side programmer. But, over the last several years, I've spent a great deal of time honing my client-side skills. I'm no master, by any means, but I've consistently improved my front-end work over that time, and 2014 built upon that considerably as well.

One area I could get some mentoring in was **AngularJS**. This was a big shift for me, and while I am still learning more and more every day, it has been an exciting change up for me. Angular is extremely powerful and flexible, taking some hard things and making them dead simple (to be fair, it makes some simple things hard too ;)). Angular is one of those items I wished I had spent more time with a year or so back, because in hind-sight it could have saved me hundreds of hours of work. I'm really looking forward to working more with Angular.

From a software craftsmanship standpoint, I also had to dive in to a slew of technologies to assist in my day-to-day development. I now use **Vagrant** to spin up my local dev environment, which is a godsend. One quick command line entry, and I'm up and running with a fully configured environment. I went from playing around with **NodeJS** to working with it day in and day out, writing my own plugins (or tweaking existing ones), and to using (and writing/tweaking) both **Grunt** and **Gulp** task runners for various automation and build tasks. To take something as "source"

and convert it to "app" with a simple command line entry is the shiznit. How many hours did I waste building my own sprites, and compiling LESS in some app? Now it happens at the touch of a few keys.

Then there are the deep areas that some project might take you. I had to dust off year's old AS3 skills to refactor a Flash based mic recorder. There was some extreme study into cross-browser client-side event handling. **Iron.io** has a terrific product for queuing up remote service containers for running small, process intensive jobs in concurrency without taxing your application's resources. That lead into studies in **Ruby**, shell scripting, and Debian package deployment (not in any sort of order), as well as spinning up NodeJS http servers with **Express**.

Did you know that you could write automated testing of browser behavior by using a headless page renderer like **PhantomJS**? Load a page, perform some actions, and record your findings, it really is incredibly powerful. It also has some hidden 'issues' to contend with as well, but it's well worth looking into, as the unit testing applications are excellent. Then you might change direction and checkout everything there is to know about Aspect Ratio. It's something you should know more about, when thinking about resizing images or video.

(Did I also mention that I went from Windows to Mac, on my desktop, and Windows to Linux, on my dev server? Best moves I ever made!)

Speaking of video, I got the opportunity to go beyond the basics with **ffmpeg**'s video transcoding. For those unfamiliar with the process, you write a command line entry defining what you want. Basically it's one command with 200+ possible flags in thousands of possible combinations, and no clear documentation, by any one source, on how to get exactly what you want (read: a lot of reading and a lot of trial and error).

If that had been all of it, it would have been a lot, but then I really got to have fun, and got to rewrite a **Chrome Extension**. Now, I had the advantage that we already had an extension, but I was tasked with heavily expanding on it's capabilities, and it's been a blast. Going from something relatively simple to something much more complex is always a challenge, but doing so when you don't fully grasp the tech at hand is even more challenging. Google has created a brilliant triple tier architecture for interfacing the browser 'chrome' with the pages inside them, and developing advanced extensions with injected page applications has a lot of twists and turns along the way. I've learned enough along the way that I'm considering writing a presentation on this process for the upcoming conference season.

So, in retrospect, I went from maintaining a large legacy system to doing cutting edge custom development, learning something new *each and every day*. **Awesome!** Now, the downside to this sort of process is that you lose valuable productivity time getting

through the learning curve. It's difficult to make hard estimates on tasks that no one else has done before, and measuring success is taken in baby steps. But the upside is that you are constantly engaged, constantly motivated, and those skills will eventually translate into better products down the road. Products that *won't* incur that same learning curve (not to mention the new ideas that can come from exposure to so many different technologies). I can't claim mastery of any of it, yet, but did gain a solid foundation in most of it, and it just makes me hungry for more.

So, if I have one wish for 2015 for you, dear reader, as well as myself, it is only that you continue to learn every day. Maybe not to the levels described above (for I am a bit of a lunatic), but at least take the chance to branch out into one new area of development in the coming year.