Michael Winkler

Summer Internship Reflection Paper

This summer, I worked at Fidelity Investments as a software development intern. Fidelity is a financial services corporation. The core of its business involves managing mutual funds, but it also advises clients on how to invest their assets, how to plan for retirement, and how to manage their money. Generally speaking, I came away from my time at Fidelity impressed. I feel that one of the company's greatest strengths is its commitment to its customers. Every company pays lip service to how much it cares about its customers, of course, but I was surprised by the degree to which my coworkers were committed to ensuring Fidelity's customers were taken care of. I also think that Fidelity does a good job of investing in its employees; they offer great benefits and are investing large amounts of time and money in recruiting young talent, which is always a good sign. All that said, Fidelity isn’t without weaknesses. For example, because Fidelity is so large, it's hard to institute any kind of company-wide change. This issue is compounded by the fact that Fidelity is a conservative company in many ways, and so is somewhat resistant to change by nature. The result is that anyone who wants to make a change ends up fighting against a huge amount of inertia. I know that my boss in particular found this issue frustrating. Before I arrived, she instituted a new workflow methodology called agile for the teams that worked under her, but when she tried to sell its benefits to other managers, she faced pushback, because generally, the other managers preferred to do things the way they'd always been done in the past. Still, despite the fact that it’s somewhat conservative, Fidelity's culture is surprisingly laid-back for a massive financial institution. No one clocks in or out; employees can take breaks throughout the day as they see fit and are trusted to accurately report the time they spent working at the end of the week. The dress code was also more relaxed than I expected -- for the duration of the summer, all employees could wear jeans and t-shirts to work. In the department directly above where I worked, it was even common to wear flip-flops and shorts to work, which I would never have expected at Fidelity. That said, some of the relaxed culture was specific to my department, because I worked for an IT department, and the tech departments within Fidelity have a more relaxed atmosphere than their financially focused counterparts.

The challenges I had to overcome in order to complete my first two personal objectives (write a certain number of MARS operations and write a user interface to test those operations) were similar. In both cases, I had to learn a new technology: Spring Boot for the MARS operations, and AngularJS for user interface. I should clarify here what I mean by writing MARS operations: MARS is a financial services tool developed internally at Fidelity, and each operation I wrote was, essentially, one puzzle piece of the overarching program. The user interface I wrote was not the one that will eventually be used in the official release of MARS, it was only for my personal testing purposes. In writing both my MARS operations and my user interface, I had to make sure to write code that was both functional and also well-structured enough that it could be understood once I left. And in both cases, I had to write effective tests to make sure that everything was functioning properly. The challenges I faced in the process of completing my third objective, writing a white paper on EZPaaS, were different, but not completely dissimilar. EZPaaS is a proprietary service developed internally at Fidelity to help different departments standardize their workflow and the tools they use for projects. My team had a goal of beginning to use EZPaaS in the near future, but because it was developed by a different department, no one on my team was familiar with the tool or how it worked. My job was, by writing a paper on the service, to help my boss understand exactly how EZPaas worked and the changes that would be necessary to begin using it. Although I didn’t have to write any code to accomplish this task, it was similar to my other objectives in that it required me to learn and understand a new technology.

Over the course of the summer, I was able to develop both "hard" (i.e. technical) skills, as well as "soft", or more interpersonal skills. On the technical side, I gained valuable experience writing enterprise-level Java and JavaScript code. I also learned a lot about the how large projects are structured, and about how to construct a workflow that allows dozens of people to work on one project simultaneously. On the less technical side, I gave several presentations, which gave me an opportunity to work on my public speaking skills. I also learned about how to comport myself professionally, especially from Alan, the project manager who I sat next to. And finally, I got some experience working under serious pressure in the real world, because the first few weeks of my internship were extremely challenging. I was simultaneously learning several new technologies, getting up to speed on the project I was going to work on, and trying to understand the overall structure of Fidelity. As I progressed towards the middle and latter stages of my internship, the challenge decreased to a more manageable level. The biggest shift in difficulty came in early/mid-July, when I had several breakthroughs in my understanding of the MARS project. From that point on, my core work was significantly easier, though I still learned a lot in the second half of the summer. In retrospect, though, that first month was probably the most valuable time that I spent at Fidelity. Throughout the entire internship, but especially during that first month, I improved my ability to self-teach quickly and effectively. That ability is hugely important as a soon-to-be professional software developer, and is definitely one of the most valuable things I took away from my summer experience. My main contribution to Fidelity was the progress I made on the MARS project. Between John (the other intern on my team) and myself, we were able to complete all 36 operations that made up the entirety of MARS. In light of the fact that my manager only expected us to complete 5-10 operations at the beginning of the summer, I’m rather proud of what we accomplished.

Broadly speaking, my experience at Fidelity was a positive one. My coworkers were friendly and understanding, which I really appreciated as an intern who didn't know up from down for the first few weeks. I especially enjoyed working with my boss, who took a very hands-off approach and gave me the time and space to teach myself the things I needed to know. She also assigned me work that was very similar to what I'll be doing once I graduate, as opposed to giving me make-work to just kill time. I'm thankful for that. On the whole, my experience was in line with my expectations. As I mentioned before, the culture was slightly more relaxed than I expected, but that's a fairly minor difference. Other than that, Fidelity was similar to what I expected: a large, traditional company staffed by talented professionals. And I learned a lot from those professionals, both technical skills and how to, essentially, act like professional: how to interact with coworkers, bosses, and generally how to navigate the professional world. As far as my future’s concerned, my time at Fidelity was extremely relevant, because enterprise-level software development is what I plan to pursue as a career. So, writing enterprise-level software of Fidelity gave me a chance to test out my chosen career path and make sure it’s a good fit (it is) while also gaining pertinent experience that can go on my resume and help me land my first job. With regard to Elon’s curriculum, I actually feel like the computer science department has already done a good job of integrating the topics that I came across into the computer science major. I had already worked with several of the tools that I used on the job, and also had previously learned about good code-design practices that I could apply to the Job. Beyond that, I felt well-prepared to learn the things I didn’t already know.