Kevin Duke

A11668720

I have chosen to read on Adam Setters’ and Julia Len’s reflections on real world problems. In **@422** Victor talks about how no one on his team knew about how to fix a problem and no one was familiar with a certain architecture and that he was mostly all on his own. He reflects that this is different from UCSD which has tutors and professors to help. I can relate to him very well on this, when I first started working at Intel, my gut reaction was to go to my manager and ask for help on almost everything including how to do something, where to find files, etc. He made it very clear on day 2 until week 3 of the internship that I am “my own detective” and that I have to find everything out on my own. So what I did was ask the ticket submitter for clarification, read the wiki and documentation (albeit being really slow), and asking my co-workers and peers how to do specific things like use the source control software. But even then, I had many urges to want to ask my manager questions out of habit, and I’ve now broken out of that habit after weeks later. In **@1245**, Victor talks about making a tool for other employees to use and the short deadline associated with it, but he ended up finishing it on Friday. Project delays are a real world problem for sure, even on small individual ones. During my yearlong internship, I made several tools for other employees to use as well and some of them got delayed. The worst case was a few weeks of delay caused by unforeseen problems in the code or logistical problems. My way of handling it was to stay late at work and put over time, even though it was discouraged, or to allocate more time just to working on the project that was due even though it has lower priority.

In reading Adam Setters’ reflections on real world problems in **@432**, he reflects and says that he needs to complete tasks in the right way the first time and the preliminary steps are much more important. He says that in class, you can skip the planning stages and get right to the coding since unlike the job, it’s more important to have a full understanding of what you’re doing. While I agree with him that it is important to complete tasks the correct way and all preliminary steps are important, I felt that it is also subjective in that it really depends on the project or task you are doing. Just like sometimes for class assignments you need to do all the steps in the correct order if you don’t know what you’re doing. He also says that a good team can both make or break your experience, and this is totally true both in the real world and class experience. In **@845**, Adam talks about polishing up code and submitting it for review by his mentor. This is also a real world problem and I had the same issue at Intel. Whenever we make changes to the main branch, we are supposed to submit are code for review on SmartBear, but there were many times when a developer got lazy and thought their code worked for them. But then when the auto build process kicks in, the code breaks. I fell victim to this as well and many times didn’t submit my stuff up for review. It was annoying slightly how I would have my code up for review and it would not be reviewed for weeks on end and then my code would become out of date and need revisions. In **@1672**, Adam states that projects go much slower at work than at school, which is true and relatable. At school, professors need to move the class along and get through certain amount of material before the quarter ends, but at work, delays and slowdowns are commonplace. And depending on the project, many of the times I was blocked waiting for other people to finish stuff I depended on.

Comparing my fellow classmates with my own experiences, in **@1945** I talked about having to learn the new BAT process and that it was a bit overwhelming finding information on how to do it. But this is a real world problem. At work, you’re usually tasked with finding stuff out that you don’t know how to do, or pick up a project when it’s original members have moved on. And in **@2160**, I briefly touched on how differently your responsibility is from school vs work. Other things such as remaining professional and punctual are very important in the business workplace, while at school it’s not really emphasized at all.

Overall, I would say that I learned a lot about real world problems during my full year at Intel. From being in the classroom for about 22 years to experiencing real life work for 1 full year, I learned just how different and similar the two are from each other. Things like how I’m supposed to act, and demeanor were different. Intel is very business orientated so I was expected to uphold that stereotype simply because everyone else was doing it. The way I obtained knowledge was different as well: in the real world, no one is there to ensure you learn or hold your hand. I had to find out most of my things on my own, and even contact OEM’s and other developers if I didn’t know how to do things. Software such as the UTN software, no one know how to use, so it was on me to contact the devs in Germany, or email an Intel engineer to explain a bug report he filed and what it means. Even then, they would give me broad responses since they assume I know what they’re talking about, and most of the times one question would lead to another, and staying on top of it all was very different from what I was used to at school, or my entire life for that matter. By reading my peer’s work, I can see that many of them showed something similar to what I just described. To me, I much prefer being independent and going the work life solving problems on my own, though it can be daunting at times. Now that I’m back at UCSD, I still have some of that mindset, and even though we have a wealth of tutors and office hours, I’m trying to solve problems on my own. Sometimes it works out, but other times I waste hours when I could have just asked a TA. But nonetheless, this was such a great opportunity and I’m so glad to have taken it.