

COMP4800 Industrial Experience Report

I've held a variety of jobs over the last seven years of my life. From serving eight year olds at a video arcade, to dealing with junkies while working a graveyard shift in Civic, to handling angry crowds at Raiders and Brumbies games, and everything in between. I think all my jobs have helped me develop some skills that will be coming in useful for a career in software engineering. Below I'll discuss the ups and downs of three of them. Starting with my first job ever, Timezone.

Timezone (Assistant Manager):

I started work at Timezone when I was 15 years old through till it closed after the redevelopment of Woden Plaza. I always thought working at an arcade would be up there with working at a video store as a dream job for me as a child. Getting paid to hang out at a place that I would most likely be hanging out anyway, meeting a lot of people my age and (occasionally) playing video games while working. However, as it always does, the good came with the bad. Being slave to a bunch of 6 year olds having a children's party, following them around putting in credits to games...Is not fun. Having a child come up to the counter with what looks like a life savings all in silver coins and asks me to count it and give him dollar's instead...Is not fun, and believe it or not cleaning up a 'mess' after a child drank to much lemonade then decided to play a sit-down car racing game...Is not fun.

After a year or so though, I think my boss thought that I did enjoy these tasks because I got promoted to a supervisor. Given control of a \$3000 safe, given keys to the store, but more importantly given control of other employees. Sweet. I was given no formal training at that age about how to manage people and I decided to take the path where I would treat them how I would want to be treated myself. So instead of dictating orders the entire time, I still occasionally found myself cleaning up little 'messes'. I believe this approach to be successful when working with staff below you. By respecting your staff, they in turn respect you and when it comes time when orders need to be laid down they are more likely to do it for you. However, one issue I did have to deal with at work was when we suspected one of our staff was stealing from the till. I was the one who had to talk to him, which was not an easy task, as I had become friends with all my staff. I had to talk to him on a professional level not as a friend that day.

A while later I again got promoted to assistant manager, however the only noticeable difference was more paperwork every week and having to get up at 5am every second Wednesday to empty all the machines. However, though my experience at a video arcade, I learned managerial experience and professionalism that will be useful when working in any small groups. I also learned how to get the top score on Time Crisis.

Australian National University (Tutor):

When I was applying to become a tutor for a unit in E-Commerce I initially was a bit hesitant. I did the unit only the year before, and despite getting a good mark in it there were sections that I could barely remember myself, let alone teach. Still I applied thinking it would be a bit of a challenge, I got the position, and it turned out it was a challenge. The majority of the material itself wasn't difficult, however I was constantly paranoid that what I was telling my students was wrong. After the first few nervous sessions, I became more at ease with tutoring. I tried to learn all their names early on, and tried to remember a few things about each student (Where they worked, what hobbies they had, etc). It was a third year unit that I was teaching, and I knew that a lot of the students would be trying to show me up where they could. Learning names, and showing interest in what they did helped gain respect from the class. At the end of most tutorials I'd just sit and catch up with what they'd been doing. Again this respect for them meant that they had respect for me. When the section of the unit that I'd been dreading to try and teach came round I had no trouble telling my class that I wasn't confident with the section myself, and instead I had brought a 'guest tutor' with me (An honors student).

This general chitchat that I had amongst the class meant that they were also a lot more comfortable with me and with each other. This I found helped as a few of the tutorials involved discussion, and one thing I've noticed in my years at university is that many students have a fear of discussions in case they say the wrong thing. I think that because my students were more comfortable with each other, they added more to these discussions and in turn got more out of them. I think that this could translate to any area where group work is necessary. As long as everyone feels comfortable with each other, they will be more productive. Tutoring also helped my communication and listening skills. I had one student that was having trouble outside of university (he wife just divorced him) and a couple of times we met to just talk about things (non of which were related to the unit).

I guess the tutoring equivalent of cleaning up puddles on racing games, would be marking assignments. It was the one area that every one warned me would be bad however I think tedious would be a more appropriate word. I think if anything marking taught me one thing, that's there's little difference between an essay that got 17/20 and another that got 16/20. Also, again I had to take a professional attitude when it came to marking. I felt that I had become friends with most of the class. However, when it came to giving them grades, I had to be accurate with the marks.

Tutoring was an experience, sometimes nerve racking, sometimes fun, however overall rewarding. In coming in to university today for the first time in a couple of weeks I had received a couple of e-mails from students thanking me for teaching them however one in particular gave me that 'warm, fuzzy feeling' you only hear about on soap operas, and I quote:

“...have fun in Melbourne. You were a great tutor, the best I've had so far. You should look at a career in teaching”

It could have just been a student looking for a few extra marks, however the assignments had been marked and the exam was over. So I like to read it straight as what it says.

ATI Teacher Education And Learning Services (Software Developer):

My one job that is most relevant to my Software Engineering career would be the infamous ATI job. They say people learn from their mistakes, however after working at this job I'll extend that to say people learn from other people's mistakes. It is similar to one of those software projects that you would read about as a bad case study. I was hired as a coder, developing a web-based, planning tool. First day I went in, got introduced to all the people I'd be working under, (one of which was doing the unit that I was tutoring) and sat in on my first official software project meeting. I had gained an idea of the system, but before I was to start coding I asked to see the design document. Blank stares all around from management. They had designed parts of the system on paper, however nothing formal and I was stuck wading through scribbles and notes to try and make sense of it all. When I questioned people about the holes in the design, I was told that it was the requirements for those parts hadn't been decided yet, brilliant.

The full story was that Dan (of Team SCADAT fame) was working with them extracting requirements, however had been told by (as Mark Grundy would put it) the guys with the bags of money, that they wanted to see code. Enter myself, taken on as a coder but soon found out that I would be helping design parts of the system, not to mention given a leadership position and told to employ other developers to get a prototype out to show to various people. Employing people was as fun as it sounds, acting all professional as you question people about their experience. All the time you can see the look in their eyes wondering 'how much experience had we had?'

Maybe it's my experience or just my general nature, but I enjoy being a leader. I like working with people, explaining to them what to do. However, the problem with this position was that I didn't fully know myself what to do myself. No design document, and no formal requirements meant that the other coders would turn to me to tell them what to do, which meant that I would have to turn to Dan, who had to turn to the project manager, (or if he was away, which he frequently was, make something up). However, the situation got a lot worse when the project manager decided to leave (I guess he jumped ship at the right time) and a new manager was taken on. However this manager, despite being there for two months now, has shown no managerial qualities what so ever. He knows less about the overall system than Dan and I do, and doesn't really do much apart from check his e-mail all day. All this while eating up the budget at a rate of \$36 an hour. He should have been talking to the clients and extracting the remaining requirements. Recently we have been trying to reach a rather close deadline, when the clients would occasionally come down and see what we've done then say "wouldn't it be nice if it did this..." They were basically trying to give us new requirements a week before the first alpha release was due. To make matters worse, the "project manager" kept on saying yes to these people, it was Dan and I who had to go up and tell them that the extra functionality would have to wait.

The majority of functionality of the system is finished now. The first prototype was released and is currently waiting for feedback for the beta release. The deadline, however had been delayed and the budget over run. The reason could easily be traced to the above problems. Through working here I have learning more about the importance of formal documents, performing realistic estimates on workloads and sometimes having to say no to clients. I have also learnt the importance a good leader makes, and the effect a bad one can have on a team. Overall, I see this software project as an example of what we as software engineers are trying to put a stop to. Things taught to us over the years have come into context and are starting to make sense (who would have thought). I'm starting work on a new project for WXY and after experimenting with the extreme programming method with group project, a very ad hoc method with ATI, it's time to go back to a waterfall method, and start getting these requirements down on paper.