Placement Report

General Electric The Company

General Electric (GE) was founded in 1892 by the merging of two electric companies, one of which was founded by the very famous Thomas Edison. One-hundred and twenty one years later, the multinational conglomerate company is listed as the 14th most profitable company in the United States as well as 4th largest in the world as ranked by the Forbes Global 2000.

The hierarchy of the company is fairly complex and convoluted, which was clear in that most people in the company did not really seem to understand quite how everything fit together. It would appear that the company is split into four main sections each with their own subdivisions. The four sections being Technological infrastructure, Energy Infrastructure, Capital and NBC Universal. I worked in a subdivision of Energy infrastructure called Energy Management, whose main goal is to provide technology to improve the transmission, distribution and conversion of electrical energy. I worked in a particular segment of this subdivision, which specialises in Geospatial Information Systems (GIS) called Smallworld. Smallworld was a company founded in 1989 then later acquired by GE in 2000. Working mostly with GIS meant that our job was about providing information in any geographical context to GE customers, for instance overlaying all the elements of a power-grid including power-lines, transformers, generators, etc on a mapping service like Bing or Google maps (just as examples).

Smallworld has recently undergone restructuring where 40% of the staff have been made redundant due to lack of available funds for the sector. This kind of restructuring left the remaining staff with a very low opinion of the company, making employee spirit and moral fairly low throughout the placement. Indicated further by the fact that during my stay 2 members of my team arranged alternative employment.

The Team

My team consisted of 9 people: 5 developers, 1 designer, 1 business analyst, 1 scrum-master and our senior project manager. Of the developers 1 was what is called an Edison, which is the companies post grad recruitment scheme that involves doing 4, 6-month rotations at different GE branches over 2 years; who was on holiday for the first month of my placement. 1 was me and the other 3 were all highly experienced senior software engineers. These people made up the web applications team so our main responsibility was to make websites relating to GIS. Although we were running the agile scrum development model, we were really mostly doing research and development projects, meaning that we weren't ever releasing our work, somewhat offsetting many of the principles of the normal development cycle. This system suited me really very well, meeting every day to discuss exactly what each member of the team aimed to complete that day, what had been achieved yesterday, limiting factors and so on gave me clear goals every day. These daily meetings are also a really good forum for seeking help on any issue or pointing out potential bugs with the system, including being able to provide help to other members of the team when they explain their own issues.

Project Introductions

Upon arrival at the company I was instructed to simply spend a day getting familiar with their Industrial Internet Design System (IIDS), which was a method of standardising websites throughout all of GE. This was a way of firstly giving any website under the company banner a clear GE look and feel about it, secondly it enabled the supposedly easy redeployment of features created by any of the GE webapp teams. The IIDS was a series of useful tools such a bootstrap, datatables, high-charts, etc and then the GE twist on any of those tools including additional CSS and images. Any features implemented in this system were then documented with examples on an internal website, so that developers could pick and choose features to implement on any website they were building themselves.

One of my team's main goals was to create a component of the IIDS called the mappingDS, which was simply a collection of generic features you might want to implement involving maps and mapping data. Given that this component required fairly high levels of understanding of the Geographical mapping protocols being used and the fact that this was doing to be deployed throughout the entire company, I was mostly kept away from developing under this project. Although I did not do any development myself I was still able to contribute in our daily meetings about design choices etc.

In conjunction with the mappingDS we were making a website using many of the features implemented in in it, the main goal of which was to prove to GE executives how powerful HTML5 could be and the effect it could have on quick production of powerful websites and tools. This was called the WOWapp named for its intention to wow the aforementioned executives. The main objective of the WOWapp was to display important geographical data to a potential power company; this data included information such as power outages, weather conditions, the location of their service teams and the power grid.

My Contribution

WOWapp

For the first week of my placement I spent the vast majority of my time simply playing around with the source code of the WOWapp, changing figures and algorithms and taking note of its affect on the program. Having never really worked with JavaScript (nor any other functional language) before, this was at first slightly hard to follow, but after the first day or so I got familiar with the coding standards used by the team. In fact on my first day I managed to locate and fix a bug, but being too sure of my own lack of experience and inadequacy I was almost certain that what I perceived as a bug was probably intentional. It was only 2 weeks later at one of the scrum meetings that someone raised the issue, when I was very happy and surprised to say that I could fix it very quickly having already located the source of the problem. Before starting the placement I was fairly sure that my contribution would be minimal given my complete lack of experience, so even a small event like this really made me feel more like a proper value-adding part of the team with work to contribute.

Given that I was joining this project after a decent amount of work had gone into it already, I had to fairly quickly understand a large amount of other peoples code, which is an activity I am not actually all that familiar with. After a few weeks I could easily feel the difference in my own ability to comprehend large chunks of other peoples code. These new skills came to the test when I was set with my biggest task yet, most of the work I was given was editing, adding to or deleting work that the Edison worker had done (given his absence for the first month of my placement). At one stage I realised that I wanted to add a new feature to his code that really required bits and pieces of the code already implemented. This being the case I dug down deep into my “professional software development” brain and pulled out all my refactoring skills to restructure several hundred lines of his code. I was certain through the entire process that it was never going to work in a million years, because I really only understood about half of what it was doing so the chances of being able to rip it apart and stick it back together again felt minute. So it was with great surprise and satisfaction that after refactoring, adding in my new feature and removing the inevitable syntax errors that the program worked; needing only a few tweaks and alterations before being push-able code. After the completion of this refactoring and feature creation, I was in charge of doing all subsequent changes to this area of the project.

Most of the work I was doing involved displaying vectorised images to represent power outage data on a map, this involved manipulation and construction of SVG code to dynamically generate vector images. Any changes needing to be made to the appearance of outage data was my responsibility, given that this was the main part of the project and that the main purpose of the project was to WOW (look flashy to the executive), the appearance of the system was an extremely important task. The crux of the task revolved around using Openlayers (the tool we used to create overlays on mapping services) to cluster a number of outages in a single representation (in 2 sections confirmed and predicted, in a pie chart format), including size discrimination between different numbers of clustered outages. There were many considerations to be made when dealing with such a task, the big refactoring job I had to complete came about after realising that the pie charts were being creating using SVG lines and curves. As it happens SVG is incapable of creating a 360 degree curve (unless specified as a circle) so simply died if any of the would be pie charts was full circle (i.e. was either all confirmed or all predicted). I had originally been given the code created by the absent Edison that had not accounted for this issue, so I had to set about understanding his 700 odd lines of code so that I could rearrange and reuse as needed, catching full circle would-be pie charts and replacing them with correctly styled circles. Further considerations when specifying size of the representations in this type of work included making sure that none of the pies were overlapping at any resolution (as zooming in and out of the map will decrease and increase the data set shown respectively) and creating a minimum threshold to cluster (i.e. don’t bother clustering outage points if there are only 3).

Customer Database

At one stage in the placement, every member of my team except for the business analyst managed to get holiday for the same week. This left me without developer supervision, as it happens the business analyst had been working on a webapp solution to a customer database issue that he had identified in the building. He had noted that he along with the 3 other business analysts in the building each had their own list of customer details that they all kept and maintained separately. He decided that there should be a website hosting all of this information, that each could then be maintained easily; all updating the same table. Given that he had little developer experience he had simply (with a little help) ripped out the tools and structure from the WOWapp and made a few alterations to get a semi-working system. I was tasked with getting this from a semi-working minimal feature project to a usable system with as many features working as possible.

Because there were no developers to guide me through this process, simply the porting from his computer to mine was a huge task. Obviously all the files were easily transferred, but lots of his alterations were coupled to his computer, like directly finding the database using his computers network name. I also had no experience of accessing databases from a program before, so it was about 4 days before I even managed to get the program to the state that he had it in when he gave it to me. After days of reading up on how to use mongodb (a nonSQL database, which is completely inappropriate to use under these circumstances, but that was easier to use as lots of the infrastructure was already there), I eventually got to grips with the tool and managed to serve the data successfully.

Now to actually format the system so that it looks half decent and usable, this is my first fun dose of really getting to grips with the company's aforementioned IIDS. Very similarly to our PSD java interfaces the IIDS, although potentially useful it often just made everything many magnitudes harder. Many of the examples were very poorly documented, with the code given not matching the image of the results shown. They had also abstracted some of the tools they had included (such as datatables), by only giving access to the extensions of said tools. This occasionally meant functionality needed to be hunted down in order to be used, which sometimes took days to find. Obviously I didn't just sit around for days hunting down some function to make my life easier, but once finding this functionality it normally made several of my workarounds redundant.

I was told at the start of this project, that given that it was personal project started by the business analyst in his spare time and that it was only ever going to be used internally, that I had full rights to "hack about" as I saw fit. Although of course this is throwing out the window many of the really important software engineering principles that I would normally use, it also meant I could implement features (potentially less maintainable features) very quickly without worrying about if it was the most efficient or easily understandable code. This meant that I was able to very quickly poke around in the variety of tools I had to use, giving me greater insight into their potential and usability that I may not otherwise have gotten.

I am now really very comfortable using a number of tools I previously had no experience using such as: bootstrap for a good and consistent presentation, Node to easily create locally hosted servers in JavaScript, datatables for putting data into tabular format, mongo for storing and managing databases, bower for automated dependency maintenance and of course the variety of languages that are involved in web application development: JavaScript, HTML and CSS, as well as a small amount of experience in PHP.

Peer Programming

As my placement came to a close, the WOWapp parked, the customer database awaiting feedback, I found myself without any assigned work. In order to make myself useful I decided to help one of the other interns on my floor with his project that I was very familiar with having been popping by to offer advice and help on throughout my entire placement. For a week we partner programmed his project, making noticeably greater progress than if we had been programming alone. Both learning from each other, with our slightly different knowledge bases and understanding of the tools at hand. Given that we both had to use the IIDS I would often be able to gleam some new understanding of some feature, quickly run off to test a theory on my own system and return victorious. I had not really partner programmed since first year when none of us really had any idea what we were doing. This being the case it was really quite a learning experience, which could have more emphasis placed upon it in the university course. The only caveat of this story was that despite not having any work to do in my own team, because I had not informed me senior manager that I had moved to a different project I received a slap on the wrist for poor communication. Over this time I felt I was generally under utilised by the company, which is why I had tried to be proactive by helping in an alternative project.

Issues

Other than the daily set deadlines in the scrum meetings, our only big deadline that I was involved in was the meeting with the GE executive we were trying to "wow". As it happens there was an odd reversal of circumstances with this particular deadline. Usually come deadline time I've failed to get my work done in time, this time I was assigned 4 different tasks 2 days before this meeting, which I leapt to and completed to the best of my ability only to have the meeting cancelled without reason. Lesson one; executives can screw about the little man without much concern. I'd really never had a deadline I'd worked this hard to meet, only to then have it made redundant, by a cancellation especially without knowing what the reason was. As this deadline was fairly late in the placement, no rearrangement was made in the remaining time I was there. This being the case I never got to find out if the project was successful, which I feel would have made the placement considerably better.

Because GE is such a huge company with a lot of security and networking protocols, everything is routed through about 4 different proxies, which can often make getting things done very difficult. I spent about 2 days attempting to request data from a 3rd party server, until I eventually asked for help and discovered I was doing everything correctly except for the fact the buildings security was preventing the request from occurring.

The experience provided throughout university was fairly helpful in relation to version control systems, despite the fact that my team uses mercurial with the

MacHg interface (neither of which I've ever encountered before). Having a good understanding of how branches work and the need to be careful when pushing and pulling revisions was very important in order to not get a confused repository or 3 way merges. MacHg makes many functions much simpler than other version control systems that I have used in the past such as SVN and github, but is still in my opinion poorly designed. Version control seems to only be made for highly computer literate users anyway, so few systems seem to have felt the need to implement a very easy to use interface. Very shortly before our supposed product deadline, I managed to accidentally push my change set to the default branch of the repository, because of a very small error on my part. The lack of ability to roll back in a satisfactory way with this particular version control system meant manually unpicking the changes I had made in order to push it appropriately to a branch. All in all a good learning experience and demonstration for further care when dealing with team repositories. Working with a team with a proper project with correctly structured repositories was also just a generally good experience, seeing how a default branch should be left as a constantly working version, that branches should be merged frequently to prevent eventual 3 way merging. The final thing that working with version control in a business environment taught me is that that team communication of which bugs fixed/features have been implemented is very important. This was demonstrated when I fixed an issue which was clearly labelled in the repository, yet because I hadn't explained this to anyone on the team, it was fixed again (slightly differently) in a separate branch causing havoc on merge attempt.

I feel the placement could have been a greater learning environment if I had been working on projects that were either being pushed to market during my placement or could at least one day see the light of day. The reasons for this are 2-fold. Firstly I became an engineer because I enjoy the feeling of having a demonstrable result of my work, so to have work that other people can obviously benefit from. The second reason for this is that working in an R&D environment means that very little care has to be taken when considering using 3rd party code, this leaves me without some of the potentially boring documentation, it is also true that R&D work does not need to be sigma-six stable like most other work provided by the company. I feel that I might have enjoyed the experience less had it been out with R&D work, but that I may have learned some really important ideas about working in a commercial world that I will need at a later date.

Self Reflection

I believe that I generally out performed their expectations on this placement, it was questioned whether or not I could take the prize for most quickly getting up to speed with a new project within a few days of my arrival. I was quickly able to complete the tasks assigned to me, rarely needing assistance from the senior developers, I contributed well in scrum meetings and frequently found and fixed bugs before other members of the team. I made some errors for instance my faux pa involving the repository issues, my failure to communicate effectively with my senior manager and occasionally being overly informal in my code comments and familiarity with the team. My mistakes like usual were also the greatest moments for learning about this new environment and I'm sure will make me a better employee in the future

I was reassured that I had performed well within the placement when I had my exit talk with the senior manager. I received very good feedback commending my ability to quickly identify and solve problems, contribute effectively in meetings, ask appropriate questions, with the only negative was that I might be too confident and overly friendly/jokey with a tight knit team.

Company Reflection

Despite the low moral within the building everyone I came into contact with was a pleasure to work with and very helpful throughout the entire placement. I had a fantastic experience that I'm extremely grateful for. I was very rarely in need of help that went unprovided and the team were always very understanding when I made any mistakes affecting them adversely. It was a very relaxed working environment, going to the pub for a few beers during lunch was an event taking place at least once a week generally encouraged by senior management.

Although I would have liked some additional experiences such as working on projects that were going to see the market, I have learned a considerable amount about software engineering. I started off having never made any kind of interactive website, whereas now I am very comfortable starting from scratch on my own project. I've gained a whole host of new tools and understanding how the Internet works that has propelled me into a much better computer scientist.

During the last few weeks of placement I feel like I could have been utilised more effectively as I often had no work to do, so could generally find myself simply checking for bugs that may or may not have existed.

Conclusion

I've learned a great deal about web applications including a range of tools such as: Node, datatables, bootstrap, MacHg, mongodb, JavaScript (including jQuery, bootstrap and backbone), HTML5 and some CSS, as well as a considerable lesson in code comprehension. Beyond just technical skills, I've also developed other skills such as working in a team, partner programming, repository maintenance and gained some insight into the inner workings of a big corporation. I believe I have demonstrated to myself and my team that I can add value to a project, which gives me both confidence and the ability to justify myself to future employers.

I would have liked to experience some more aspects of a commercial world, such as working on a sellable product and real world deadlines, but regardless very much enjoyed my time at General Electric - Energy Management.