Jack Davey

**Software processes**

When I began the practical component of my prject. I needed to make decisions about how I would actually go about doing the practical work. I believe that a set of good software processes is essential for doing this so I set this out early on.

I decided against using a waterfall based development methodoly. This is because I have never completed a project of this size on my own before, and I am using a large number of technologies, which are unfamiliar to me. Because fo this, I did not feel that the waterfall approach would give me enough spce to move things around if difficulties hit my project.

I therefore chose the to sue an agile methodology with elements of Scrum built in. I chose an agile approach so that I could split the work up into manageable increments. . This had two major benefits, firstly, it allowed me to work on each feature In isolation, meaning that I only needed to focus on integrating into the main system at the end of each iteration. Another benefit of using an agile approach is that meant that I could easily keep an eye on the progress I was making, and make adjustments to deal with this. As an example, during my first iteration, I completed all the work two quickly, so I was able to plan to complete more work in the second week.

Once I made this decision, I created an iteration plan that detailed exactl how I would manage my time each iteration. Originally I had planned to use one week iterations and have each one self contained. After my first week working on the project however, it became clear that actually spreading this over two weeks would be beneficial. This helped me for two main reasons; firstly, it allowed me to vary the tasks that I completed each day. Secondly, it meant that I could give each task more than one or two days to do..

Another important consideration I had to take into accont ws the programming languge that I would be using. The first choice that sprang to mind for this project was Java, Java was the first programming langugge I was ever taught, so I felt that I would be able to get something going pretty quickly in Java.. The language can be used in most situations, because it has so many libraries and plugins available for it. On the other hand however, , I was concerned that the project might nvolve dealing with concurrency, and the model based on threads and locks that Java provides is notoriuly hard to get right. I also felt that I would be playing it safe a bit with Java, and I wanted to learn a language that I had never used before.

Another choice I considered was PHP, PHP is ideal for web based projects like mine. Thi is because you insert it directly into HTML code, and most web servers support it. On the one hand, I thought the fact that the language was easy to deploy to be a major selling point. On the other hand however, the fact that PHP is dynamically typed makes PHP programs harder to debug. To add to this, PHP has no built in support for concurrency. I therefore decided that the negatives of PHP outweight he positives for this project.

The third language I looked at was Scala. Scala was built ontop of the Java ecosystem, so all the benefits of programming in java were also true of Scala. Another major plus point for the language was that it supported the actor model fo concurrency. This involves not sharing any state at all, and communicating between different parts of the system through the use of immutable messages. I could see this approach working well for this sort of system. The reason being that it would be easy to translate real system full of servers communicating with each other to an actor based system.

These were not the only tools that I needed. One tool that I found I needed in partiularwas version control. Although I wasn’t working as part of a team, I did find that It would be usefulto keep track of old versions of documents. Another reasons as to why I chose version control was the fact that it allowed me to ensure my work was regularly backed up to a server to ensure that nothing was lost.

I therefore created an account on Github for this project. This allowed me to rpefrom the functions mentioned vove and also allowed me to easily share progress with my supervisor about my work.

Another tool that I found particularly helpful during my project was Jira. Jira is an issue tracking tool that allowed me to log and monitor the work I was doing as I was doing it. It also allowed me to provide weekly progress updates to my project supervisor, as I could indicate in a agrpahical way exactly how much work I had done.

In terms of rameworks, I chose the AKka and play framwoeks for the project. The akka framework enahanced the standard actor model capabilities provided by Scala. The play framework builds on top of Akka, and makes it very easy to worite both we services and web applications. I found osme of the play utilities, sucha s parsing JSon, particularly sueful.

Design was also n issue I had to consider. I tried initially drawing use case diagrams, but these did not work for several reasons, firstly, they were too genral engouth, and they ddidnt let specify the uses cases in enough detail. I used processdiagrams such as those taught I module CO545 to design the concurrency aspects of my solution.. Thse outlined thmajor compentnts, and illustrated how each component communicates with each other compenent. Fianly, for certain aspects of the problem, such as inheritance hierarchies, I used class diagrams.