Patrick Fischer

Senior Seminar

Work Reflection

Recently I completed an internship with a cytogenetics lab, Genetics Associates. This internship was the first opportunity that I have had to use the skills that I have learned at Lipscomb. The first project that I worked on was creating a large number of test cases for a new Laboratory Information System, NucleoLIS. The feature that I was testing was the case and patient management portion of the software. This involved creating a new case, a new patient, and then ordering a battery of tests for the patient. The tests performed by Genetics Associates involved chromosome analysis, so the test cases required for the software involved enumerating every possible test variation and result and checking for errors. This was somewhat tedious to do, as I could not use any programming to create the various sets of data required for the tests. Fortunately, I only had to work on a single section of the program.

After finishing the software testing project, I was given another project by my supervisor. Upper management used several reports that were generated from in-house software written by my supervisor. These reports worked well, but there were some problems with them. The first problem that I noticed was the complete lack of version control. Any backups of the programs were stored on an on-site server. Any previous iterations of the programs existed only in commented out code or in the sparse number of backups. This is not how programming is typically carried out, as normally a programmer will make changes to a program and commit the changes to their version control software of choice. This allows the programmer to quickly and easily revert any changes that have negatively impacted the program. It also allows the programmer to examine previous software versions in case older code needs to be used in their project. I rectified this situation using a popular version control software called Git. I use this software for my programming projects, as it is fairly easy to use after learning some of the commands. I created a repository on one of the servers and committed the current versions of the reporting programs to it. This solved the first problem, but there were a still some more issues to work through.

The second issue that I noticed was repetitive code usage. Code that is used more than once in a program is typically placed into a function to reduce the amount of clutter in the program. This allows the programmer to easily reference the code when required, and allows the code to be used in other programs if necessary. To fix this issue I created a number of separate files containing code for commonly used functions. These functions included connecting to a database, running SQL queries on the database, analyzing data, and performing mathematical calculations. This helped cut down on the size of the programs and allowed easier creation of new reports.

After fixing these issues upper management asked that more reports be created. Most of these requests did not involve pulling any new data, but merely consolidating some of the current reports. This involved pulling data used by multiple reports and then stitching it together in a single report for management to use. After finishing these reports my next project involved creating a web interface for viewing reports. I accomplished this by using various libraries for the R programming language. This was somewhat difficult, as I am not completely familiar with R. After some trial and error, I managed to create a web application that allowed users to view any report located in a directory on the server, and also to sort the reports and search for various results. I also created a number of utilities to generate images with graphs and plots that showed statistical information about the data contained in the reports. The application received positive feedback from management, and the project was then shelved for use in later intranet projects.

One of the final projects that I worked on involved translating the reporting programs from R to Python. This was done to make the programs easier to maintain in case future programmers needed to work on them. To accomplish this I researched a number of statistic and spreadsheet libraries, along with a library to interact with ODBC drivers to pull data from the SQL server. This project proceeded much more quickly than any of the previous projects, as I am more familiar with Python than with R. I was able to satisfactorily replicate the reports in Python after a few days of working. The web application that I created was not ported to Python, as it was determined that it would be prohibitively difficult to do so, especially since an already working solution was in place.

This internship was the first opportunity that I had to put into practice the skills that I learned at Lipscomb. Working at Genetics Associates was very different in comparison to how work is typically handled in classes. At school I am generally given a single project to work on for a class, and there is a hard deadline for completion. The work assigned at Genetics Associates was more fluid. One day I could be working on a project for one department only to have an urgent request come from another department. All the projects would be completed, but the deadlines would shift quite frequently to account for other work that arose during the day. However, other aspects of the job were similar to situations encountered in classes. Most of the programming that I did on the job was in an unfamiliar language, but I was given time and tutorials to learn it as in class. Overall, the internship was a great first exposure to working for a company, and I am quite thankful for the opportunity to work there.