# Web Script Programing Final Report

## The lifecycle

Not much consideration was given to the lifecycle. I used an agile and spiral approach as I usually do.This is a good fit for websites as new features can be added on to the existing functionality.

Agile had the benefit of keeping me focused on producing a functional product. I had the target of adding a new feature or positive change each coding session.

Spiral ensured I was constantly reviewing the design and testing the product.

## Scheduling

The Scheduling was awful. I was slow to start, slipping rapidly into complacency. The long deadline lead to prioritising other units.

I knew I worked better by coding for longer sessions. I planned early on to delay starting on the coursework until I had a better understanding of the material. With less knowledge I have to research more, which interrupts coding flow. I prepared by reviewing code from worksheets and creating gists for future reference.

My first concentrated attempt at producing the code was over the Christmas Holidays. This did not go as well as I had hoped.

The progress was slow, this was mainly due to not knowing how to debug web applications. It was only through one on one conversations with Rich that I discovered that Chrome had breakpoints and network traffic monitoring. Both were very valuable and I should have known about earlier.

I failed to appreciate that some of the code demonstrated was examples of how not to do it. This meant time was wasted time using document.write() and then having to debug missing content. However this lead to greater independence. I starting using developer.mozilla.org and php.net to improve my understanding and discover new knowledge. In addition to stackexchange.com to diagnose problems that have occured.

An unexpected issue during the Easter Holiday was not being able to use my laptop while it was being fixed. During this time I had to configure and use an older desktop to continue making progress. Due to my use of github and dropbox, no data was lost.

## What I have learnt

Complacency is dangerous, developing a web application is a significantly different experience from desktop applications and web sites. It takes time to adapt.

Code written quickly is poorly thought out, which leads to inconsistency and wastes time debugging. If it is designed is will be far more robust, better understood and repeated code can be separated out into libraries to prevent code duplication.

I have expanded my knowledge of HTML, CSS and PHP, which was weak at the start of the year.I am now reasonably confident with the languages. I have gone from no knowledge of javascript to a good level of competency.

I was introduced to JSON and PDO. Both have proved themselves useful, they are current technologies, so proficiency should improve my employability.

console.log is valuable but it is easy to overuse it, which can obscure important information. A different approach may be beneficial. Use of the .info and .warn could be beneficial as it separates the type of information given.

I have found dispatching custom events a convenient way of responding to data changes. Events provided the safety of ensuring functions weren't called before the data they required was available.

Along with InSE this unit has given me experience with GIT and github. A weakness I have worked on is vague commit messages. Making made small tweaks over many files means no single purpose to the commit. I cannot easily find previous versions. However I have been able to use revert and checkout to remove undesired changes.

At the start of the year I was using Vim as my prefered text editor. Rich recommended alternatives. I tried Sublime Text 3. It has been a good experience and there is a wide range of extensions to improve performance. Including vim emulation (taking advantage of existing muscle memory) and linters

I have been using jsHint through my text editor, this has provided valuable feedback. Highlighting errors and best practices. However I prefer indexing to the recommended dot notation. I will research reasons behind this later.

Build tools is something I intend to look into in the future. My attempts were not successful, but I had limited time to explore.

Additional testing with other people would have helped. By knowing what to do, I was able to avoid edge cases my system may not be able to handle.