Yeisol Woo

CS460 Senior Capstone Project 1

Homework #6

Review Software Configuration Management (SWEBOK KA-6)



The software configuration management knowledge area of SWEBOK deals with the configuration of software systems. Specifically, configuration management is the discipline of identifying the configuration of software at distinct points in time for the purpose of systematically controlling changes to the configuration and of maintaining the integrity and traceability of the configuration throughout the system life cycle. The knowledge area is further broken down in the following sub-topics: management of the SCM process, software configuration identification, software configuration control, software configuration status accounting, software configuration auditing, and software release management and delivery.

 Our project was implemented as a web-based system. As a result, much of the software configuration management process was simplified. For example, a desktop-based client server application requires that the client side application be kept up-to-date with changes made to the server side application. As such, updates to the client side application must be rolled out in conjunction to changes made in the server side application. In contrast, since our application is viewed through a web browser, it is updated (in effect) every time it is used. This greatly reduces the need to roll out new versions of client software down to possibly having to clear the browsers cache on updates to ensure that updated components are used.

 The software configuration management for our project consisted of tracking changes to either the front or back end of the system to ensure that the changes made to not adversely affect the other end of the system. For example, if the way data was processed on the backend was altered, we would have to alter the way that the front end presents said data to the back end. Furthermore, due to our web-based design, we could only support one version of the front and back ends at a time. This is in contrast to traditional client-server applications in which multiple versions of the client side application may be in use at the same time.