Yeisol Woo

CS460 Senior Capstone Project 1

Homework #5

Review Software Maintenance (SWEBOK KA-5)





Software maintenance is an integral part of a software life cycle. However, iThe software maintenance knowledge area of SWEBOK deals with making changes to a software system in order to patch bugs, add feature, etc. More specifically, maintenance is the phase of software lifecycle that starts upon delivery whose activities may start much earlier. In this life cycle phase, the code can be altered to fit changing needs of the client as well as fix any bugs that may arise. The knowledge area is further broken down into the following sub-topics: software maintenance fundamentals, key issues in software maintenance, maintenance process, and techniques for maintenance.

Our work on our project is done once we have the project complete (meaning that while our software system will eventually enter a maintenance phase, we won’t be working on it at that point). This does not mean, however, that our project does not relate to the software maintenance knowledge area. For example, while we won’t be working on the project once it enters the maintenance phase as a whole, each module of the system enters a pseudo maintenance phase after completion. Consider the database / persistence layer modules: upon completion, we found that while the modules met the original requirements, we wanted to add new features such as storing additional information about drivers and BOLO’s. To do this, we opened a separate branch in the SVN development tree to experiment with adding each feature. We then developed the feature in question and upon agreement that the feature was useful and did not disrupt any other requirements, we added the changes made in that development tree back into the original branch. Furthermore, upon completing integration tests on each of the modules and sub-modules, we found that while a module worked on its own, it may have problems interacting with its related modules. As a result, even though we had classified a module as ‘complete’, we had to go back and patch bugs that arose during further development. In this way, each of our modules went through its own pseudo maintenance phase.