Waterfall Life-Cycle Model

Our team decided to develop learning management system software using waterfall model which consisted straight forward process with each phase followed by maintenance and quality reassurance through documentation. Instead of completing each phase one by one, we decided to hard code the software first and analyze the code in each phase. Although this method was not recommended because of the lengthy process but having to go back and forth through the code made us realize our common mistakes and improved our overall programming skill. As we progressed through the phases of waterfall life-cycle model we were practically able to experience everything that we learned throughout the course regarding software development process.

Requirements Phase – The system required two profiles one for the student and other for the admin. Admin should have the right to see each and every information as well as control over the actions of the student’s functions. This meant there needed to be restrictions on what each profile could access. The system required inputs such as student’s name, student’s ID, registered courses, grades and GPA calculation. Users could only view this information and make no changes to it.

Requirements analysis – We discussed if any missing requirements could be easily added and edited beside the basic requirements.

Documentation – Basic needed features were agreed upon by the group.

Analysis Phase – Although using java was our first choice because we weren’t familiar with java and this could have been a great opportunity to learn new language. Later the team decided to go with visual basic and c ++ language to code the system because we all were familiar with the functions and classes required to create the requirements listed above.

Documentation – We documented the possible languages, database construction that could be used to create and satisfy the system requirements. We also discussed about how to create the web interface where the user would be able to login.

Design phase- Our team started designing learning system by creating classes and functions that closely followed the decisions made in both requirements and analysis phase. We used c++ language and mysql to program the system.

Documentation – Brief examinations of code bugs and if the code debugged properly where all functions required were working properly.

Implementation Phase – The system was implemented by connecting mysql to php in xaamp. We were able to add, modify and delete information in the database. Basically this phase was covered by both running and testing the code as well as database inventory.

Documentation – System was able to connect and make changes to the database.

Post-delivery maintenance – Post-delivery maintenance included adding new features and testing the whole functionality after each change. The changes were done in both the code and the database. Examples such as, adding extra classes for students, creating better outlook of the interface students would use and etc.