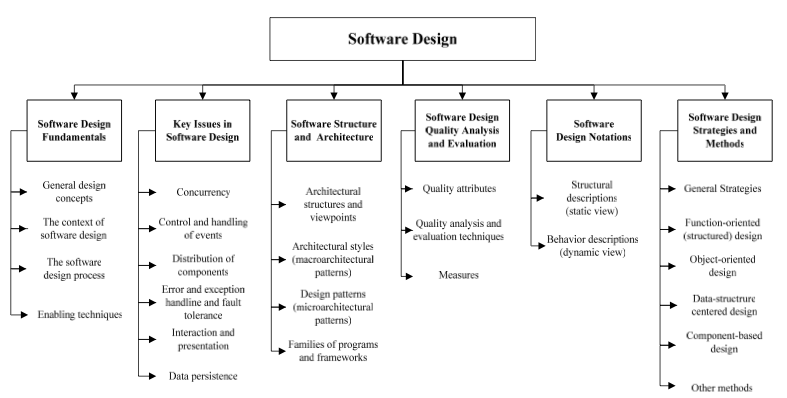
Yeisol Woo

CS460 Senior Capstone Project 1

Homework #2

Review Software Requirements (SWEBOK KA-2)



The software design knowledge area deals with the design of software systems. It can be viewed as a process in which software requirements are analyzed and translated into a description of the software system’s internal structure. This description can then be used during later software engineering life cycle activities in which it is translated into the actual software system. The design itself is defined as both ‘the process of defining the architecture, components, interfaces, and other characteristics of a system or component’ and the result of the aforementioned process. The knowledge area is further broken down into the following sub-topics: software design fundamentals, key issues in software design, software structure and architecture, software design quality analysis, software design notations, and software design strategies and methods.

As with any other software engineering project, part of our project’s process was to analyze the software requirements we came up with in the earlier engineering life cycle phases and translate them into a design of the overall software system. To begin, we took the final set of software requirements we had agreed upon earlier and analyzed each of them as well as any sub-requirements. The purpose of this analysis was to extract requirements for our design. For instance, if we had a software requirement that the system should store information about each driver’s insurance information, and then we could extract that the final system indeed needed to have a place to store such information. From there, we expanded on the extracted design hints in order to come up with a general outline of how we would handle each design feature. While this was generally implementation-agnostic, we did have to come up with a plan on how to meet each requirement. We continued to build upon each requirement’s design until we had met all of the software requirements. The end result was a combination of textual and graphical design requirements and descriptions. The set of graphical requirements and descriptions came in the form of UML diagrams that described not only what information would be stored in each module, but also how each module would communicate with one-another.