EECS 448

Team 16

Integration Strategy

Our team used sandwich integration to bring together all our software components into one cohesive unit. Integration is important to this project because it helps to combine the code of team members to create the final project. As we worked simultaneously, integration allowed us to maintain the software’s deliverability and functionality. More specifically, sandwich integration helped to maximize time spent on the project since we split up the tasks. As a result, our software was able to perform operationally from top-down and bottom-up.

This type of integration was well suited for our team’s performance. In the early discussions of the project, we decided to split the tasks into front-end and back-end. This ensured that integration would only take place if the work was functional. We worked top-down to develop the code by testing the software. By running our source code, it opened an avenue to detect any faults early on such as defects and bugs. Our top-down integration includes the back-end programming where we built the main framework for the website. We slowly built onto the skeleton from Project 3 by adding more requirements and functionality. As the top-down logic artifacts were constructed and tested, bottom-up integration could also be working at the same time. Here, our bottom-up integration includes designing and testing the operation artifacts. By working on the front-end and back-end together, it helped the team understand the look and feel of the software. Layering more code was successful because of this new understanding of how the project functions. Being able to test and get the website functioning while other members worked on a different area of the software was beneficial in our efficiency since each task would not affect one another. All in all, sandwich integration allowed the team to discover major design faults early on, test operational artifacts thoroughly, and work productively.