

Itr3 Retrospective

A component of the project that has not been implemented as well as we would have liked in previous iterations would be our test coverage. We did not employ test-driven development as significantly as we should have, and the effects of that were felt significantly throughout the rest of the development of the project. This meant that when we were refactoring and making changes to existing code, we often ran into problems of sudden crashes and broken code due to the limited test coverage. If we had employed test-driven development, and used more unit and acceptance tests, we could have likely spent less time trying to get our edits to line up properly with our existing code. It would have also encouraged us to make smaller, more easily testable methods and classes, which would have made the project overall more manageable.

Going forward in Iteration 3, we will work on employing more tests, and refactoring large classes and methods to make them more easily testable. This means that in the development of new methods, we will make sure to develop the tests before (or at the same time) as the method itself. And then, our core refactoring strategy for this iteration will be cleaning up the code so that we are being consistent in the way we handle and pass data (e.g. passing objects instead of large numbers of parameters).

Our success will be determined through a number of quantitative metrics. We would like to have a greater number of tests, a significantly higher proportion of code coverage, and a larger number of methods (but smaller average method size) than previous iteration. Greater numbers on all of these metrics would indicate that we were focusing more on testing, and we believe would end up with us spending less time on troubleshooting problems from editing.