SWE6733 – Emerging Software Engineering Processes

Assignment 2

Dr. Reza Parizi

Group 2:

Matt Decker

Devika Kandel Neupane

Ifeanyi Okoye

Cameron Page

**Assignment 2 Final Methods and Final Thoughts**

The following is a breakdown of how group 2 worked through assignment 2. We will outline our methods and how we approached the problem as well as our final thoughts on mob programming.

**Project Github:** <https://github.com/Decker-Matthew-R/Group2-GildedRose-Refactoring-Kata>

**F-2-F Tools:** For our mob programming sessions, we decided to use Teams video chat with our group. We met and pushed changes back and forth using git. We felt in-person, face to face sessions would be more effective, but due to group members locations out of state and constraints with our schedules, Teams was the only viable option to complete the assignment.

**Complete the Gilded Rose Refactoring Kata:**

1. Our final product is on our group github: <https://github.com/Decker-Matthew-R/Group2-GildedRose-Refactoring-Kata>
2. We have added test cases to our program. These are also located in the github repo and these tests (screenshots and test coverage statistics) will be addressed in the **Test Case** portion of this write up.
3. Our code coverage is addressed in the **Test Case** portion of this write up.
4. See a screenshot of our unit tests in the **Test Case** portion of this write up. A screenshot is also located in our github repo.
5. We used code smells and test cases to refactor our code during our mob session. Our mob session can be viewed here: <instert url>. A link is also posted on our github. We also have the refactored code on our github.
6. We also did a separate mob session to add updated features for the ConjuredItems, per the customer request. We have also included a test case for the conjured items. Here is a link to our second mob session: <insert link>. A link is also on our github. Please see the **Test Driven Development** section of this write up to see how we used TDD to drive the implementation of our new feature.

**Test Case**

**Test Driven Development**

**Final Thoughts**

This was a very good exercising in refactoring and mob programming. For our specific group, we found mob programming to be sort of clunky in terms of accomplishing work and this was due largely to our group having varying experience with refactoring, Java, and unit testing. These varying skillsets made our mob programming sessions long. As a group, our communication was excellent, and everyone contributed in different aspects to the assignment. We also found mob programming to be difficult given the length of time required to complete sessions and aligning everyone’s schedules. If we were doing this for our jobs, it would have likely been easier to find common blocks of time that could be shared for mob sessions, however given our job requirements, life requirements, and additional assignments/classes that we were juggling, it was difficult to find those common blocks of time to complete the assignment in a mob programming setting. Overall, this was a great opportunity to learn a new method of programming and sharpening some skills such as testing and