Jayme Rutkoski
SER 316
Assignment 7

Task 1

Size

(1). The Total Lines of Code in the project is 22,539.

(2). The largest code file is HTMLEditor.java and it's LOC is 2,144.

(3). The tool used the noteChanged method. The method goes through every noteListener and changes it with the note parameter.

Cohesion

(1). LCOM2 equals the percentage of methods that don't access a specific attribute averaged over all attributes in the class. The equation for LCOM2 is $LCOM2 = 1 - sum(mA)/(m * a)$, where m is the number of methods in the class, a is the number of attributes in the class, and mA is the number of methods that access an attribute.

(2). TaskListImpl.java has the highest level of Cohesion because it has a good number of methods that has relationships with other methods. Most of the class has Functional Cohesion.

Complexity

(1). The Cyclomatic Complexity is 16.

(2). The EventsManager.java has the worst Cyclomatic Complexity on average which is 16.

(3). I added a method getAttribute() to lower the complexity of the getEvent() method. The getAttribute() method is a helper method so that the getEvent() method doesn't have to do so much work. The complexity was lowered from a 5 to a 2.

Package-level Coupling

(1). Afferent Coupling is the number of classes in other packages that depend upon classes with the package is an indicator of the package's responsibility. Efferent Coupling is the number of classes in other packages that the classes in the package depend upon is an indicator of the package's dependence on externalities. The difference between the two is that Afferent Coupling are the incoming dependencies and Efferent Coupling are the outgoing dependencies.

(2). The package that has the worst Afferent Coupling is main.java.memoranda.util and it's value is 57.

(3). The package that has the worst Efferent Coupling is main.java.memoranda.ui and it's value is 49.
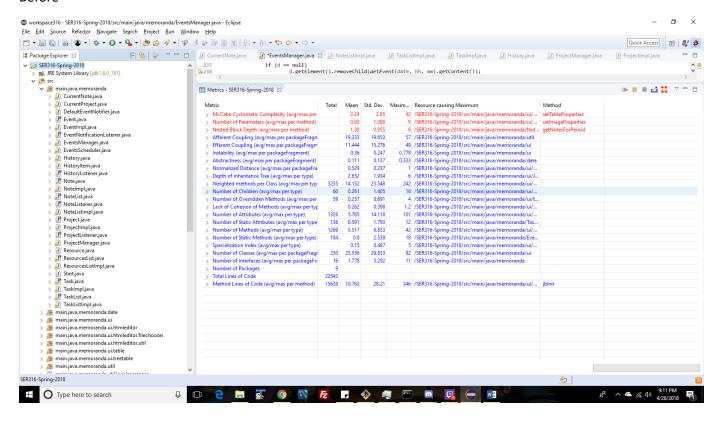
Worst quality

The EventsManager.java class has the worst quality. It has the highest Cyclomatic Complexity, and the highest Number of Parameters. Since the class has a high complexity it is harder to understand compared to the other classes which would take a longer time for another developer to understand. Plus, with the high number of parameters it naturally increases the complexity of the code.
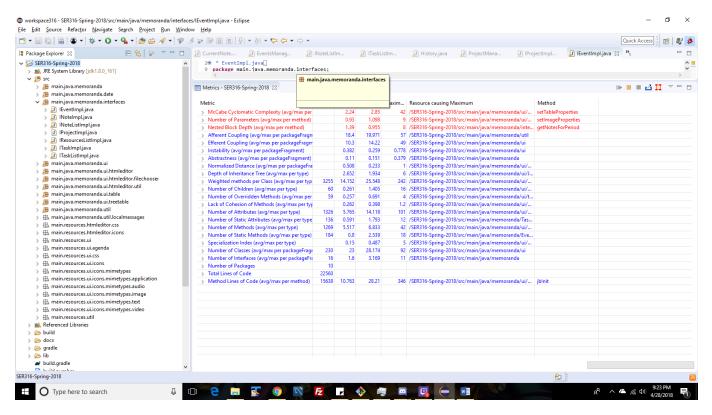
Jayme Rutkoski
SER 316
Assignment 7

Task 2

(Step 1)

Before

Jayme Rutkoski
SER 316
Assignment 7

(Step 7)

After

(Step 8)

The Afferent Coupling went down considerably because the package's responsibility got transferred to the new interface package so the overall responsibility of each package lowered in result.

Task 3

(Step 1)

I made the method 'getRepeatableEventsForDate' in the EventsManager.java class shorter because it was originally too long (Code Smell was long method). /SER316-Spring-2018/src/main/java/memoranda/EventsManager.java was the class being altered.

(Step 2)

I fixed the method 'createTask' in the ITaskListImpl.java class and in the TaskList.java class because there were too many parameters (Code Smell was too many parameters). /SER316-Spring-2018/src/main/java/memoranda/interfaces/ITaskListImpl.java and /SER316-Spring-2018/src/main/java/memoranda/TaskList.java were the classes being altered.

Jayme Rutkoski
SER 316
Assignment 7

(Step 3)

(after)