

Task 1

Size

1. 22539 LOC in the project.
2. HTMLEditor.java is the largest file at 2144 LOC.
3. It would appear that LOC is calculated by counting each statement as 1. (Method 1 from the lecture notes.)

Cohesion

1. LCOM2 calculation returns a value in the range [0,2] with values ≥ 1 indicating a very problematic class. Lower values mean a higher level of cohesion. From what I could find, LCOM2 is defined as the percentage of methods that do not access a specific variable averaged across all variables in the class. Basically, if all methods can access all variables in a class then there is high cohesion.
2. Assuming I'm reading the report right, PriorityQueue.java has the highest cohesion (lowest LCOM score). The class has only 2 variables and 4 methods. One variable is accessed by 3 of the 4 methods and the other variable is accessed by all 4 methods. This would indicate that all the pieces are working together very well for high cohesion.

Complexity

1. main.java.memoranda has a mean Cyclomatic Complexity of 1.746 with a maximum of 16 and a standard deviation of 1.547.
2. The Start.java class has the highest average CC at 3.5.
3. History.java started at a mean 2.429 CC. Slight changes brought the mean CC down to 2.357. The add method had a check for null and a check for equality in two nested if statements (Line 37 and 38). I consolidated it into a single if statement using an `&&`. Also, there was an if statement for setting the size of `_list` (A vector). Because Vectors resize dynamically I couldn't find a reason to keep it in the context of the code so I commented it out (Line 39 and 40). Granted, there could have been a good reason for that check that I'm not aware of buried somewhere else in the code, but for now an explicit call to `setSize` on a Vector did not seem necessary.

Package-level Coupling

1. Afferent Coupling refers to the dependency of other classes on the classes in the subject package. In other words, if I am a package then afferent coupling refers to other packages relying on me. Afferent coupling signals inward.

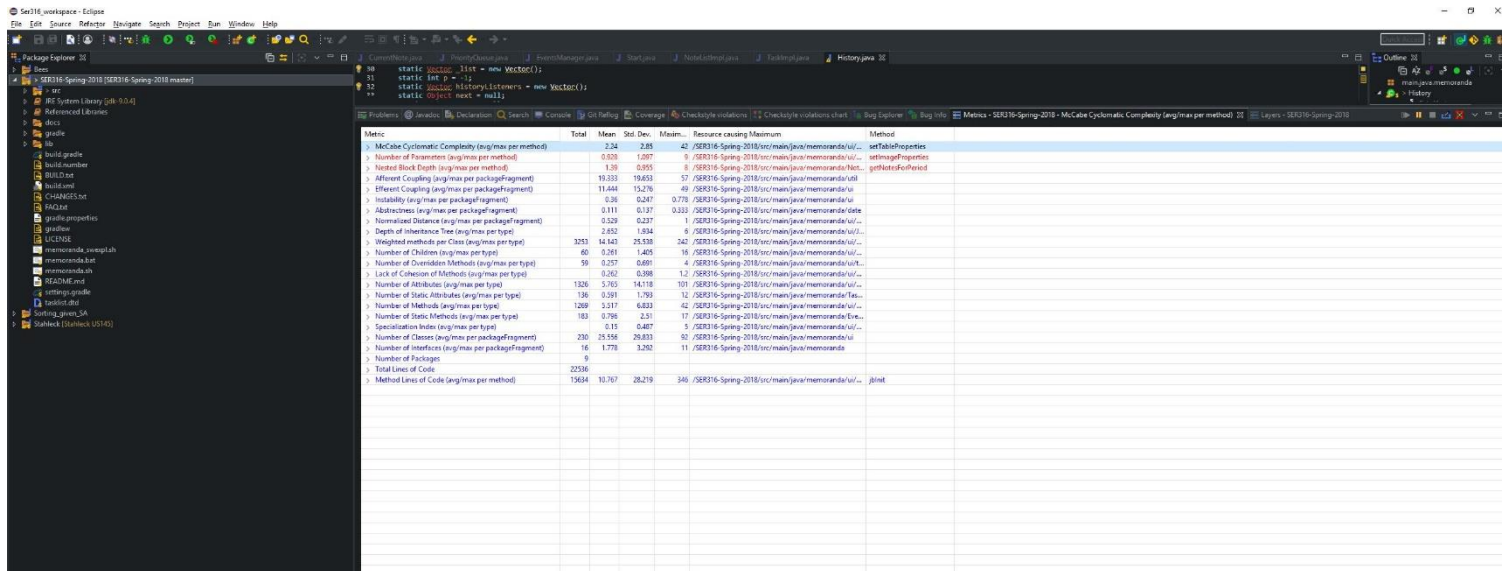
Efferent Coupling refers to the dependency of the subject package on classes from other packages. In other words, if I am a package then efferent coupling refers to me relying on other packages. Efferent coupling signals outward.

2. main.java.memoranda.util has the highest Afferent Coupling with a value of 57.
3. main.java.memoranda.ui has the highest Efferent Coupling with a value of 49.

Worst Quality

I would place both **HTMLEditor.java** and **AltHTMLWriter.java** (Both in main.java.memoranda.ui.htmleditor) as tied for worst class in the project. I base my decision primarily on Cyclomatic Complexity. HTMLEditor has a mean CC of 3.562 and AltHTMLWriter has a mean of 4.292. There is actually a class that has a higher mean CC (ProjectPackager.java @ Mean=5.667) but the clincher for my picks is that both classes have an extreme spike in complexity according to their respective Maximum values. HTMLEditor at maximum has a CC of 42 and AltHTMLWriter has a maximum CC of 23. ProjectPackager only maxes out at a CC of 10, so even though the mean is higher, there is not such a dramatic spike in complexity. Additionally, at maximum value for LCOM both HTMLEditor and AltHTMLWriter come close to a value of 1 (.976 and .9 respectively), just barely which is another indicator that there could be some improvements in the code.

Task 2



Metric	Total	Mean	Std. Dev.	Mean...	Resource causing Maximum	Maximum
M McCabe Cyclomatic Complexity (avg/max per method)	2,34	2.85	42	/SER316-Spring-2018/src/main/java/memoranda/ui/...	setFahisProperties	
> Number of Parameters (avg/max per method)	0.030	1.087	9	/SER316-Spring-2018/src/main/java/memoranda/ui/...	setImageProperties	
> Nested Block Depth (avg/max per method)	1.38	0.955	6	/SER316-Spring-2018/src/main/java/memoranda/ui/...	getFahisProperties	
> Afferent Coupling (avg/max per package/fragment)	19,333	19,653	57	/SER316-Spring-2018/src/main/java/memoranda/ui/...		
> Efferent Coupling (avg/max per package/fragment)	11,444	15,276	49	/SER316-Spring-2018/src/main/java/memoranda/ui/...		
> Instability (avg/max per package/fragment)	0.36	0.247	0.776	/SER316-Spring-2018/src/main/java/memoranda/ui/...		
> Abstraction (avg/max per package/fragment)	0.111	0.137	0.333	/SER316-Spring-2018/src/main/java/memoranda/ui/...		
> Normalized Distance (avg/max per package/fragment)	0.329	0.237	1	/SER316-Spring-2018/src/main/java/memoranda/ui/...		
> Depth of Inheritance Tree (avg/max per type)	2.651	1.834	6	/SER316-Spring-2018/src/main/java/memoranda/ui/...		
> Weighted methods per Class (avg/max per type)	3213	14,543	25,538	242	/SER316-Spring-2018/src/main/java/memoranda/ui/...	
> Number of Children (avg/max per type)	60	0.261	1,405	18	/SER316-Spring-2018/src/main/java/memoranda/ui/...	
> Number of Overridden Methods (avg/max per type)	99	0.257	0.891	4	/SER316-Spring-2018/src/main/java/memoranda/ui/...	
> Lack of Cohesion of Methods (avg/max per type)	0.262	0.398	1.2	/SER316-Spring-2018/src/main/java/memoranda/ui/...		
> Number of Attributes (avg/max per type)	1326	5,763	14,718	101	/SER316-Spring-2018/src/main/java/memoranda/ui/...	
> Number of Static Attributes (avg/max per type)	136	0.381	1,793	12	/SER316-Spring-2018/src/main/java/memoranda/ui/...	
> Number of Methods (avg/max per type)	1269	5,317	6,833	42	/SER316-Spring-2018/src/main/java/memoranda/ui/...	
> Number of Static Methods (avg/max per type)	183	0.796	2,511	17	/SER316-Spring-2018/src/main/java/memoranda/ui/...	
> Specification Index (avg/max per type)	0.11	0.487	5	/SER316-Spring-2018/src/main/java/memoranda/ui/...		
> Number of Classes (avg/max per package/fragment)	230	15,556	29,833	92	/SER316-Spring-2018/src/main/java/memoranda/ui/...	
> Number of Interfaces (avg/max per package/fragment)	16	1,778	3,252	11	/SER316-Spring-2018/src/main/java/memoranda/ui/...	
> Number of Packages	9					
> Total Lines of Code	22,536					
> Method Lines of Code (avg/max per method)	13634	10,767	28,219	346	/SER316-Spring-2018/src/main/java/memoranda/ui/...	print

The screenshot displays the IntelliJ IDEA IDE with three panes:

- Left Pane (Project Structure):** Shows the package hierarchy starting from `main`, including `main.java.memoranda` and `main.java.memoranda.ui`.
- Center Pane (Source Code):** Displays the implementation of `EventImpl.java`. It includes imports for `java.util.Calendar` and `java.util.Date`, and defines the `EventImpl` class implementing the `Event` interface.
- Right Pane (Metrics):** Shows the 'Metrics' tab for the project 'SER316-Spring-2018'. It lists various code quality metrics calculated by McCabe's Cyclomatic Complexity tool, such as McCabe Cyclic Complexity, Number of Parameters, and Weighted Methods per Class.

Metric	Total	Max	Std. Dev.	Maximum	Resource Causing Maximum	Method
MCCabe Cyclic Complexity (avg/max per method)	2.28	235	42	SER316-Spring-2018/src/main/java/memoranda/ui/...		getLabProperties
Number of Parameters (avg/max per method)	0.820	1.097	9	SER316-Spring-2018/src/main/java/memoranda/ui/...		setLabProperties
Nested Block Depth (avg/max per method)	1.39	0.955	8	SER316-Spring-2018/src/main/java/memoranda/ui/...		getNetOfPeriod
Affected Coupling (avg/max per package/fragment)	21.6	25.011	57	SER316-Spring-2018/src/main/java/memoranda/ui/...		
Different Coupling (avg/max per package/fragment)	10.4	14.203	49	SER316-Spring-2018/src/main/java/memoranda/ui/...		
Instability (avg/max per package/fragment)	0.335	0.243	0.779	SER316-Spring-2018/src/main/java/memoranda/ui/...		
Abstractness (avg/max per package/fragment)	0.172	0.301	1	SER316-Spring-2018/src/main/java/memoranda/ui/...		
Normalized Distance (avg/max per package/fragment)	0.251	0.522	1	SER316-Spring-2018/src/main/java/memoranda/ui/...		
Depth of Inheritance Tree (avg/max per type)	2.652	1.954	6	SER316-Spring-2018/src/main/java/memoranda/ui/...		
Weighted methods per Class (avg/max per type)	3259	14.143	242	SER316-Spring-2018/src/main/java/memoranda/ui/...		
Number of Children (avg/max per type)	80	0.281	1.405	16	SER316-Spring-2018/src/main/java/memoranda/ui/...	
Number of Overridden Methods (avg/max per type)	59	0.257	0.891	4	SER316-Spring-2018/src/main/java/memoranda/ui/...	
Lack of Cohesion of Methods (avg/max per type)	0.282	0.398	1.2	SER316-Spring-2018/src/main/java/memoranda/ui/...		
Number of Attributes (avg/max per type)	1326	5.763	14.118	101	SER316-Spring-2018/src/main/java/memoranda/ui/...	
Number of Static Attributes (avg/max per type)	136	0.591	1.793	12	SER316-Spring-2018/src/main/java/memoranda/ui/...	
Number of Methods (avg/max per type)	1259	5.517	6.833	42	SER316-Spring-2018/src/main/java/memoranda/ui/...	
Number of Static Methods (avg/max per type)	18	0.796	2.51	17	SER316-Spring-2018/src/main/java/memoranda/ui/...	
Specialization Index (avg/max per type)	0.15	0.487	5	SER316-Spring-2018/src/main/java/memoranda/ui/...		
Number of Classes (avg/max per package/fragment)	230	23	25.174	82	SER316-Spring-2018/src/main/java/memoranda/ui/...	
Number of Interfaces (avg/max per package/fragment)	16	1.6	3.169	11	SER316-Spring-2018/src/main/java/memoranda/ui/...	
Number of Packages	22					
Total Lines of Code	22583					
Method Lines of Code (avg/max per method)	1534	10.767	28.279	346	SER316-Spring-2018/src/main/java/memoranda/ui/...	jshk

`createRepeatableEvent()` on line 122 had a list of 8 parameters. This was a smell of a long parameter

2.2 Small Between Classes

Both the main issue memoranda, *ClassesNotelist.html* issue and *EventsManager.html* issue both contained

Final Capture

The screenshot shows an IDE with a Java project structure on the left and a metrics table on the right. The project structure includes a package explorer with a tree view of the project hierarchy. The metrics table displays various code quality metrics and their values.

Metric	Total	Mean	Std. Dev.	Maxim...	Resource causing Maximum	Method
> McCabe Cyclomatic Complexity (avg/max per method)	2.233	2.849	42	/SER316-Spring-2018/src/main/java/memoranda/ui/...	setTableProperties	
> Number of Parameters (avg/max per method)	0.923	1.081	9	/SER316-Spring-2018/src/main/java/memoranda/ui/...	setImageProperties	
> Nested Block Depth (avg/max per method)	1.39	0.955	8	/SER316-Spring-2018/src/main/java/memoranda/Not...	getNotesForPeriod	
> Afferent Coupling (avg/max per packageFragment)	21.8	20.252	57	/SER316-Spring-2018/src/main/java/memoranda/util		
> Efferent Coupling (avg/max per packageFragment)	11.1	14.529	49	/SER316-Spring-2018/src/main/java/memoranda/ui		
> Instability (avg/max per packageFragment)	0.342	0.244	0.778	/SER316-Spring-2018/src/main/java/memoranda/ui		
> Abstractness (avg/max per packageFragment)	0.172	0.301	1	/SER316-Spring-2018/src/main/java/memoranda/inte...		
> Normalized Distance (avg/max per packageFragment)	0.516	0.248	1	/SER316-Spring-2018/src/main/java/memoranda/ui/...		
> Depth of Inheritance Tree (avg/max per type)	2.667	1.936	6	/SER316-Spring-2018/src/main/java/memoranda/ui/...		
> Weighted methods per Class (avg/max per type)	3247	14.241	23.637	242	/SER316-Spring-2018/src/main/java/memoranda/ui/...	
> Number of Children (avg/max per type)	60	0.263	1.411	16	/SER316-Spring-2018/src/main/java/memoranda/ui/...	
> Number of Overriden Methods (avg/max per type)	59	0.259	0.694	4	/SER316-Spring-2018/src/main/java/memoranda/ui/...	
> Lack of Cohesion of Methods (avg/max per type)	0.268	0.401	1.2	/SER316-Spring-2018/src/main/java/memoranda/ui/...		
> Number of Attributes (avg/max per type)	1331	5.838	14.17	101	/SER316-Spring-2018/src/main/java/memoranda/ui/...	
> Number of Static Attributes (avg/max per type)	136	0.596	1.8	12	/SER316-Spring-2018/src/main/java/memoranda/inte...	
> Number of Methods (avg/max per type)	1271	5.575	6.902	42	/SER316-Spring-2018/src/main/java/memoranda/inte...	
> Number of Static Methods (avg/max per type)	183	0.803	2.52	17	/SER316-Spring-2018/src/main/java/memoranda/Eve...	
> Specialization Index (avg/max per type)	0.151	0.489	5	/SER316-Spring-2018/src/main/java/memoranda/ui/...		
> Number of Classes (avg/max per packageFragment)	228	22.8	28.131	92	/SER316-Spring-2018/src/main/java/memoranda/ui	
> Number of Interfaces (avg/max per packageFragment)	16	1.6	3.169	11	/SER316-Spring-2018/src/main/java/memoranda/inte...	
> Number of Packages	10					
> Total Lines of Code	22577					
> Method Lines of Code (avg/max per method)	15611	10.737	28.215	346	/SER316-Spring-2018/src/main/java/memoranda/ui/...	jblinit

I can't say that my small bits of refactoring did anything of consequence.

Mean CC decreased by .007. A decrease in complexity is good, but this was a negligible decrease.

Mean LCOM increased by .006. Mean Afferent coupling increased by .2 and mean efferent coupling increased by .5.