Weixiang Zhang & Xinkai Wang & Huijing Liang

Javier Gonzalez-Sanchez

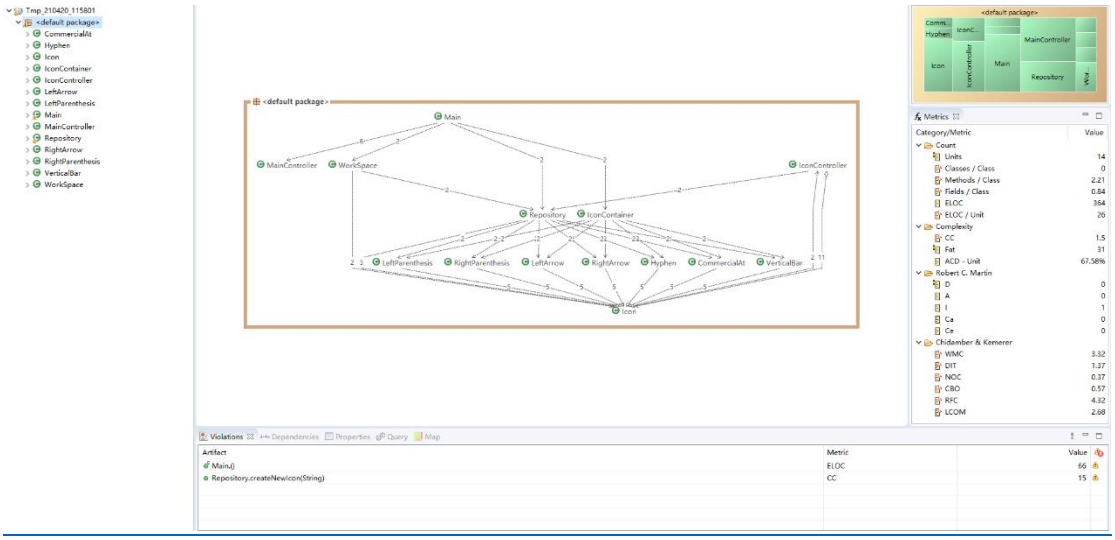
SER 516

20 Apr 2021

Structural Metrics

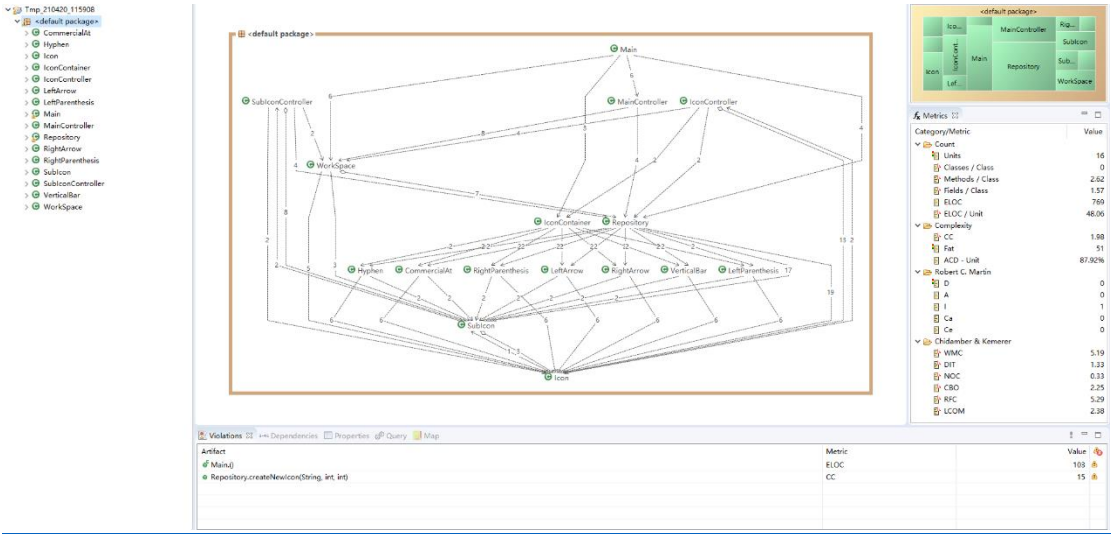
NOTE: The screenshots maybe too large to fit in a word document.

Sprint One: Report generated from STAN



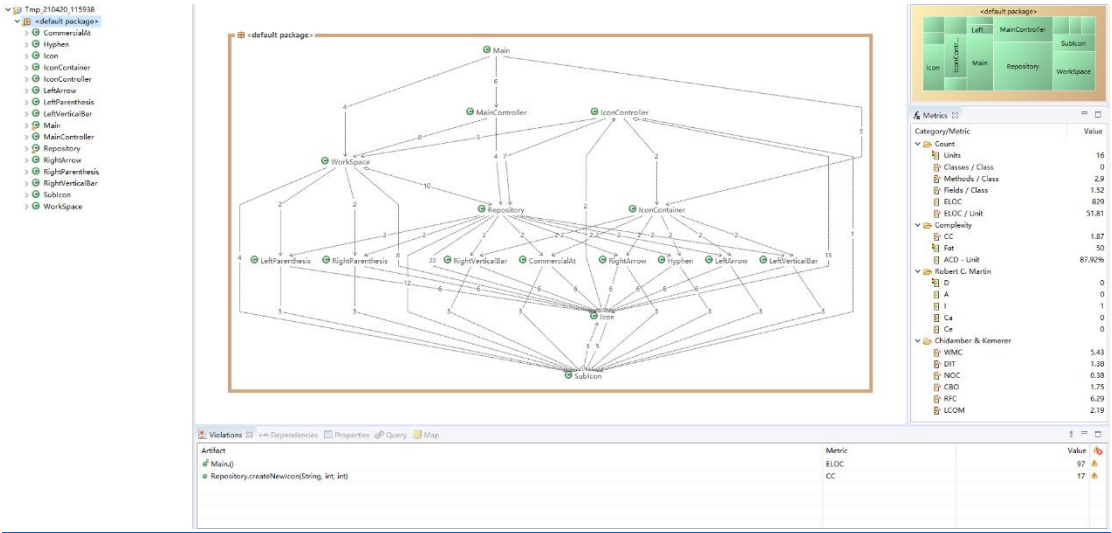
In the sprint 1, there are two main things we would like to point out: 1. There are 66 Effective Lines Of Codes in the constructors of the Main.java class, and 2. In another class – Repository.java, the Cyclomatic Complexity of one of the methods is 15. The ELOC for the constructor is not that bad but can still be split into two methods.

Sprint Two: Report generated from STAN



In the sprint 2, there are two main things we would like to point out: 1. There are 103 Effective Lines Of Codes in the constructors of the Main.java class, and 2. In another class – Repository.java, the Cyclomatic Complexity of one of the methods is 15.

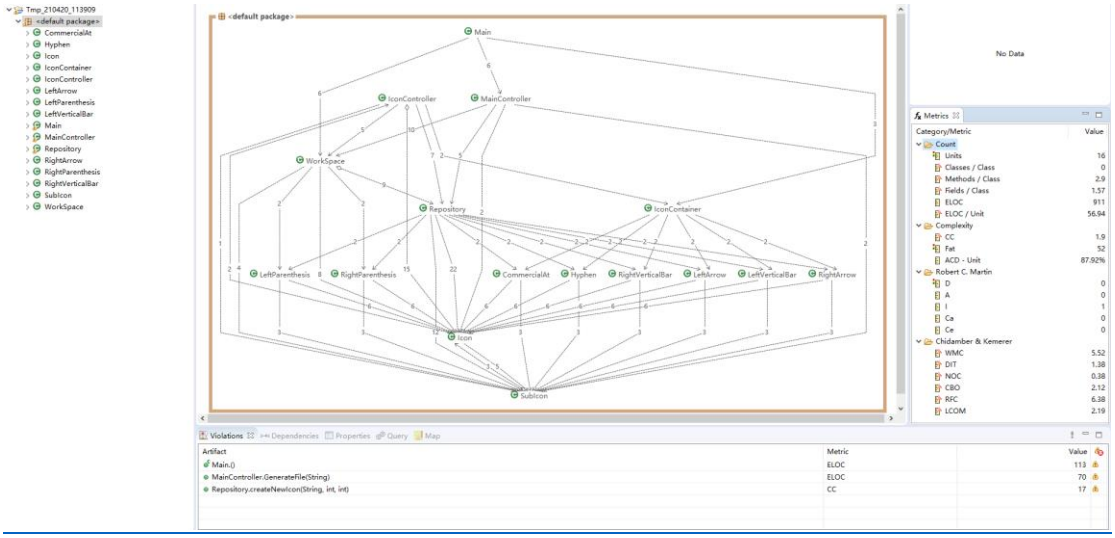
Sprint Three: Report generated from STAN



In the sprint 3, there are two main things we would like to point out: 1. There are 97 Effective Lines Of Codes in the constructors of the Main.java class, and 2. In another class – Repository.java, the Cyclomatic Complexity of one of the methods is

17.

Sprint Four: Report generated from STAN



In the sprint 4, there are three main things we would like to point out: 1. There are 113 Effective Lines Of Codes in the constructors of the Main.java class, 2. One of the methods in the MainController.java, there are 70 Effective Lines Of Codes within it, and 3. In another class – Repository.java, the Cyclomatic Complexity of one of the methods is 17.

Overall, among four sprints, there are two major problems: the ELOC and CC of some methods. To resolve the high ELOC of a method, we can simply review the method and try to split it into two if possible. Meanwhile, we used around 10 if-else statements to determine which button the mouse clicks on, causing the high CC among four sprints. Alternatively, we can employ Event-driven programming for each button so that the internal logic can help us with the determination rather than using redundant if-else statements.

The class Repository.java in the version of sprint 4 “grew” into a Big Fat Model

in terms of the MVC design pattern due to the increasing number of requirements. We think it is inevitable.