**Complexity metrics**

The complexity metric, also known as cyclomatic complexity is used to indicate the program's complexity. This may be applied to several parts of a program, such as individual methods, classes, packages, modules, or even the whole project. The complexity also greatly impacts the cost of maintaining a program. And then, the main goal of this metric is to modularize a more maintainable and testable program.

The metric is based on a quantitative measure of the number of linearly independent paths through a program's [source code](https://en.wikipedia.org/wiki/Source_code). It’s computed using the control-flow graph of the program. The nodes of the graph correspond to basic blocks of program commands. The directed edge connected between two nodes indicates that the second one will be immediately executed after the first command. Another way to measure the program complexity is by counting the line of codes (LOC).

This metric evaluates the class complexity on three parameters:

- **Average operation complexity** (OCavg).

- **Maximum operation complexity** (OCmax).

- **Weighted Methods per Class** (WMC): this measures the sum of complexities of methods defined in a class.

As we can see on the collected class metrics diagram, there´s an extreme WCM value in the class “net.sourceforge.ganttproject.task.TaskManagerImpl”. The indicated value is 173.00 when the class average WCM value is 13.61. That´s a trouble spot, showing that there´s a class with many high-complexity methods with too many lines of code. It, therefore, represents the complexity of a class as a whole and then causes a “large class” code smell. The class will take on more and more responsibilities, making it get larger and larger.

Another trouble spot is on the class “net.sourceforge.ganttproject.

GanttOptions.GanttXMLOptionsParser”, which has an extreme OCmax and OCavg value compared with both average values. That means the class has the most complex method in the program codebase. This method is responsible for more tasks than it should be, turning the method very large and complex, which can cause code smells such as long methods. It makes the code hard to maintain and hard to read.

**Bibliografia:**

<https://en.wikipedia.org/wiki/Cyclomatic_complexity>

<https://www.geeksforgeeks.org/complexity-metrics/>

<https://thevaluable.dev/complexity-metrics-software/>