**Complexity Metrics**

Martim Costa 64901

Brief Introduction

In this report I will analyse the Complexity Metrics of the GanttProject, in methods, classes, packages, modules and in the entire project. The metrics are:

- Cogc: Cognitive Complexity

- ev(G): Essential Cyclomatic Complexity

- iv(G): Design Complexity

- v(G): Cyclomatic Complexity

- OCavg: Average Operation Complexity

- OCmax: Maximum Operation Complexity

- WMC - Weighted Method Complexity

- v(G)avg - Average Cyclomatic Complexity

- v(G)tot - Total Cyclomatic Complexity

Cognitive, Essential Cyclomatic, Design and Cyclomatic Complexities

Explaining it simply:

- Cognitive Complexity (Cogc) is a measure of how difficult a unit of code is to intuitively understand or in other words tells you how difficult your code will be to read and understand. At a method level, 15 is a recommended maximum.

- Cyclomatic Complexity (v(G)) effectively measures the number of possible independent paths through a method or function. This tells us how complex the method is to test.

- Essential Cyclomatic Complexity (ev(G)) tells how much complexity is left once we have removed the well-structured complexity. Essential Cyclomatic Complexity and Cyclomatic Complexity are related. Programs with a level below 10 for both Cyclomatic Complexities are considered to be within acceptable range.

- Design Complexity (iv(G)) is related to how interlinked a methods control flow is with calls to other methods. It also represents the minimal number of testes necessary to exercise the integration of the method with the methods it calls.



**Table 1-Methods**

As you can see, at the level of methods (more than 6500) the average for all complexities is under 2, which is very a good average, way below the values ​​considered minimum.

Bellow are the tables that are related to Packages, Modules and the whole Project. As you can see he values of Cyclomatic Complexity ​​presented referring to the methods are represented in the other tables, which have simular values. v(G)avg is Average Cyclomatic Complexity and v(G)tot is the Total Cyclomatic Complexity.



**Table 2- Packages**



**Table 3 - Modules**



**Table 3- Whole Project**

Average Operation, Maximum Operation, Weighted Method Complexities

- Average Operation Complexity (OCavg) is the Average Cyclomatic Complexity of all non-abstract methods in each class. Inherited methods are not counted.

- Maximum Operation Complexity (OCmax) is the Maximum Cyclomatic Complexity of all non-abstract methods in each class. Inherited methods are not counted.

- Weighted Method Complexity (WMC) is the total cyclomatic complexity of the methods of each class.

Uma imagem com mesa

Descrição gerada automaticamente

**Table 5 – Classes**

As this table shows the Average of the Average Cyclomatic Complexity of the methods of each class don’t differ much to those shown above in the **Table 1**. Even the Average of the Maximum Cyclomatic Complexity value is around 3 so even the methods with higher values have values that are considered in the acceptable range.

Conclusion

Analising, and according to this data, I can conclude that this Project in general in terms of understanding and reading it is easy to do so. I can also say that about the dificulty to test this Project the values of Cyclomatic Complexity are bellow the value that are considered acceptable(usually 10 is the maximum value to be considered that).

Of course some improvements can be done, some methods have high values of Cognitive Complexity and Cyclomatic Complexity. Some ways to reduce Cognitive Complexity are :

-Avoiding multiple if-else or nested if-else statements.

-Move repeated Code/nested if else to a separate function.

-Reduce number of parameters of the method. It is always good to have max 2–3 parameters. If it are exceeds you can wrap all of then into a class and pass the object.

About Cyclomatic Complexity:

- Use small methods. Try reusing code wherever possible and create smaller methods which accomplish specific tasks.

- Reduce if/else statements.