Metrics Analysis review (Complexity)

By Tiago Fernandes 57677

My colleague begins to explain correctly what are cyclomatic complexity metrics, what they do and what they can be used for. He explains the different ranges for low, medium and high values of complexity and also states that these metrics determine if the code is stable and trustworthy.

* Project-wise: 1.79 average cyclomatic complexity metrics (acceptable);
* Package-wise: 4,49 (acceptable)

## Methods:

As my colleague said, in the method view we have 4 complexity metrics:

* Cognitive complexity
* essencial cyclomatic complexity
* design complexity
* cyclomatic complexity

### Cognitive complexity and Design complexity:

The cognitive complexity measures the difficulty to intuitively understand a certain unit of code and the design complexity measures how complex is the design.

I agree with my teammate when he states that these metrics have both very high values for some methods and I also consider his solutions for these high values correct.

One solution to lower these high cognitive complexity values is to cut down the nesting by deleting unecessary conditions and also some switch case can eliminate some code smells (Switch Statementes code smell).

For the design complexity, we can fix the Long method code smell by lowering the methods size.

Regarding the cyclomatic complexity, my colleague suggests that we should eliminate all code smells that affect the legibility of the methods. I agree with both this solution and the perceptive relation that he found between the complexity of the methods and the complexity of the rest of the class, package and project.

[[1]](#endnote-1)

1. I only fixed a minor spelling error in my teammate document [↑](#endnote-ref-1)