# Complexity metrics

## **Cyclomatic Complexity**

Cyclomatic complexity is a metric used to indicate the complexity of a program. It calculates several [linearly independent](https://en.wikipedia.org/wiki/Linearly_independent) paths that can generate all possible paths through a method.

Class Metrics:

### Average operation method

This metric calculates the average Cyclomatic Complexity of all non-abstract methods in each class. Inherited methods are not counted.

For example, the class GanttXMLOptionsParser has “long method” code smells. Specifically the method startElement() has more than 200 lines with a lot of if statements which indicates that the method is very complex(has higher cyclomatic complexity values) and should be divided into different methods. ~~The average cyclomatic complexity value is exceeded.~~

### Maximum operation method

This metric calculates the maximum Cyclomatic Complexity of the non-abstract methods in each class. Inherited methods are not counted.

The class mentioned above has a high cyclomatic complexity value (it is the maximum complexity of a method calculated in the class), which means the code is more complex, increase the risk to modify and is harder to comprehend

### Weighted method

This metric calculates the total cyclomatic complexity of the methods in each class.

A class with a low cyclomatic complexity is better, ensures good readability and maintainability of the code. Reducing the complexity of the class makes writing test cases easier. If the class has high cyclomatic complexity probably is a long class and is doing too many things therefore should be split.

Leticia Silva 60221