**Complexity Metrics**

MetricsReloaded Plugin

Complexity metrics are measured based on cyclomatic complexity. The complexity of a module is the number of independent cycles in the flow graph (all the paths that can be traversed during a program execution). This metric correlates complexity with maintenance effort, meaning the more complex a module is the harder it is to maintain.

|  |  |  |
| --- | --- | --- |
| package | Average Cyclomatic Complexity | Total Cyclomatic Complexity |
| |  |  |  | | --- | --- | --- | | org.jabref.gui |  |  | | 2,01 | 595 |
| |  |  |  | | --- | --- | --- | | org.jabref.logic |  |  | | 1,18 | 13 |
| |  |  |  | | --- | --- | --- | | org.jabref.migrations |  |  | | 2,56 | 100 |
| |  |  |  | | --- | --- | --- | | org.jabref.model |  |  | | 1,96 | 106 |
| |  |  |  | | --- | --- | --- | | org.jabref.preferences |  |  | | 1,44 | 511 |

According to this table, on average every package has only 2 independent paths except for preferences and logic that have 1 path.

**Cognitive Complexity**

This complexity tells us how hard it is for a person to understand a method.

The package with the most cognitive complex methods is java/org/jabref/logic such as the one located in jabref/src/main/java/org/jabref/logic/bst/BibtexCaseChanger.java which the team has already pointed out in the code smells.

**Essential Cyclomatic Complexity**

This complexity shows how much complexity is left once we have removed the well structured complexity (i.e. a for loop which we know when it is going to finish). Methods with lower Essential complexity are easier to break into smaller methods, on the other hand methods with higher Essential complexity are more difficult to understand, maintain and test.

Gui and Logic are the packages with higher values of Essential Complexity. The method *org.jabref.gui.fieldeditors.FieldNameLabel.getDescription(Field)* has the highest Essential complexity because it has a big case with many returns in it.

**Design Complexity**

The Design Complexity is related to how interlinked a methods control flow is with calls to other methods.

The packages Gui and Logic have the most methods with the highest Design Complexity meaning it is harder to understand at once their interconnections. *org.jabref.logic.importer.fileformat.RisImporter.importDatabase(BufferedReader)* is the method with the worst design because it is long and calls many other methods, having many interconnections.

**Cyclomatic Complexity**

This complexity calculates how many independent paths there are in a method, thus how many tests are necessary.  
Once again, Gui and Logic are the packages with the highest complexity. *org.jabref.logic.layout.format.RTFChars.transformSpecialCharacter(long)* is the method with the highest complexity due to the amount of ifs there are that increment the number of independent paths

In conclusion, the packages Gui and Logic are the ones with methods that have the highest complexities, therefore the ones we should focus on improvements.