Method Metrics

References

- jqassistant
- Neo4j Python Driver

Effective Method Line Count

Table 1a - Effective method line count distribution

This table shows the distribution of the effective method line count per artifact. For each artifact the number of methods with effective line count = 1,2,3,... is shown to get an overview of how line counts are distributed over methods.

Only the 15 artifacts with the highest method count and their effective method line count distribution (limited by 40)is shown here. The whole table can be found in the CSV report `Effective_Method_Line_Count_Distribution`.

Have a look below to find out which packages and methods have the highest effective lines of code.

artifactName	effectiveLineCount

Table 1b - Effective method line count distribution (normalized)

The table shown here only includes the first 40 rows which typically represents the most significant entries. Have a look below to find out which packages and methods have the highest effective lines of code.

artifactName	effectiveLineCount

Table 1b Chart 1 - Effective method line count distribution (normalized)

```
No data to plot
```

Table 1c - Top 30 packages with highest effective line counts

The following table shows the top 30 packages with the highest effective lines of code. The whole table can be found in the CSV report `Effective_lines_of_method_code_per_package`.

artifactName	fullPackageName	linesInPackage	methodCount	maxLinesMethod	maxLinesMethodName

Table 1d - Top 30 methods with the highest effective line count

The following table shows the top 30 methods with the highest effective lines of code. The whole table can be found in the CSV report `Effective_lines_of_method_code_per_package`.

index	artifactName	fullPackageName	maxLinesMethodType	maxLinesMethodName	maxLinesMethod	linesInPackage

Cyclomatic Complexity

Table 2a - Cyclomatic method complexity distribution

This table shows the distribution of the cyclomatic complexity of methods per artifact. For each artifact the number of methods with the cyclomatic complexity = 1,2,3,... is shown to get an overview of how cyclomatic complexity is distributed over methods.

Only the 15 artifacts with the highest method count sum and their cyclomatic method complexity distribution (limited by 40) is shown here. The whole table can be found in the CSV report `Cyclomatic_Method_Complexity_Distribution`.

Have a look below to find out which packages and methods have the highest effective lines of code.

artifactName	cyclomaticComplexity

Table 2b - Cyclomatic method complexity distribution (normalized)

The table shown here only includes the first 40 rows which typically represents the most significant entries. Have a look below to find out which packages and methods have the highest effective lines of code.

artifactName	cyclomaticComplexity

Table 2b Chart 1 - Cyclomatic method complexity distribution (normalized)

```
No data to plot
```

Table 2c - Top 30 packages with highest cyclomatic complexity

The following table shows the top 30 packages with the highest cyclomatic complexity. The whole table can be found in the CSV report `Effective_lines_of_method_code_per_package`.

artifactName	fullPackageName	complexityInPackage	methodCount	maxComplexity	maxComplexityMethod

Table 2d - Top 30 methods with highest cyclomatic complexity

The following table shows the top 30 packages containing the methods with the highest cyclomatic complexity. The whole table can be found in the CSV report `Effective_lines_of_method_code_per_package`.

index	artifactName	fullPackageName	maxComplexityType	maxComplexityMethod	maxComplexity