

## **Cyclomatic Complexity**

Cyclomatic complexity, that being the measure developed by McCabe (1976) that uses the number of linearly independent paths within code to assess its complexity, is a metric that is still used today despite the changes within the software development landscape. It is worth assessing, however, whether it should still be considered as a viable metric to assess the security and complexity of code.

One thing that supports the idea that cyclomatic complexity should still be used today is the fact that it still is. Developers and analysts still use this to assess their code, therefore it can be assumed that there is some merit in this. As per Garcia (2024), “measuring cyclomatic complexity is straightforward”, which allows a wider number of people to use this. As such, both developers and customers can understand the working of this. Another benefit of cyclomatic complexity comes from it being quantitative. This allows it to be non-subjective, with two different pieces of code being easily comparable without further analysis being needed. It is easy to compare a block of code with four independent paths with one that has ten independent paths, with the results of this being easy to understand.

One drawback to using cyclomatic complexity as a metric for code quality is that it can drive developers to oversimplify their code to reduce its score, even if this may make the code worse. Codacy affirms this, stating that “developers may be tempted to reduce Cyclomatic Complexity by introducing overly simplistic solutions that sacrifice readability or maintainability”.

To conclude, it may be beneficial to retain cyclomatic complexity in order to assess the quality of code, but this shouldn't be an exclusive tool and should be paired with rigid testing to ensure that solutions are effective.

## **References**

Codacy (2024) Cyclomatic Complexity: A Complete Guide. Available from: <https://blog.codacy.com/cyclomatic-complexity#AreThereAnyDisadvantagesToMeasuringCyclomaticComplexity%C2%A0> [Accessed 20 October 2024].

Garcia, D., (2024) Cyclomatic Complexity in Software Development and Its Impact on Cybersecurity. Available from: <https://blog.cr0hn.com/cyclomatic-complexity-in-software-development-and-its-impact-on-cybersecurity/> [Accessed 20 October 2024].

McCabe, T.J., (1976). A Complexity Measure. IEEE Transactions on Software Engineering, SE-2(4), pp.308-320. doi:10.1109/tse.1976.233837.