**To what extent is cyclomatic complexity relevant when developing object-oriented code?**

Cyclomatic complexity offers us simple yet effective approach to asses the complexity of the code and to see possible improvements. If the complexity is higer, we can “generally” assume, that the piece of code is more complicated to understand, consequently it requires in-depth testing. On the other hand, code, that has lower complexity number does not require much effort to grasp its functionality. Looking from the perspective of object-oriented code, it helps developers to determine the complexity of the program through the execution paths and consequntly reveal possible improvements (Kalagara, 2020). However higher cyclomatic complexity does not necessarily mean that the code is harder to understand and same goes in the opposite direction. This is esspecialy common when the methods rely on the side effects. Nevertheless it is important to bear in mind, that we shold always hold the complexity level as low as possible, while taking into account the performance of the program. If the latter is less important it is therefore better to have lower complexity and lower performance, than higher complexity with higher performance. Moreover the time required to properly test the program will also be higher. In order to minimize the cyclomatic complexity we should be aware of the few most common approaches, which are: Taking the functions length small and concise, avoiding flag parameters, appropriate use of design patterns and use of already invented functionalities where possible (Schulz, 2021).

**References:**

Kalagara, S. (2020). Cyclomatic Complexity in Software Development. International Journal of Engineering Research & Technology (IJERT). Available from: <https://www.ijert.org/research/cyclomatic-complexity-in-software-development-IJERTCONV8IS16011.pdf> [Accessed 25 March 2023]

Schults, C. (2021). Cyclomatic Complexity Defined Clearly, With Examples. LinearB. Available from: <https://linearb.io/blog/cyclomatic-complexity/> [Accessed 25 March 2023]