The Cyclomatic Complexity is commonly considered in modules on testing the validity of code design today. However, in your opinion, should it be? Does it remain relevant today? Specific to the focus of this module, is it relevant in our quest to develop secure software?

Cyclometric Complexity is a metric designed by Thomas McCabe in 1976 to analyse the complexity of code. Lower complexity is thought to be a positive indication that the code is higher quality, easier to maintain and more reliable. However, it is a controversial metric with academics and those in industry disagreeing over its relevance (Elbert ,2016).

Alenezi & Zarour (2020) concluded that there is a correlation between increased complexity, and an increase in the number of security flaws. However, they did concede that this relationship is vague. An increase in complexity makes the code more complicated and therefore more difficult to understand and maintain, leading to security vulnerabilities being missed. Furthermore, testing is more challenging as the complexity increases and the attack surface of the software also increases.

Based on this, it is my opinion that a reduction in the complexity is likely to increase the security of software. Therefore, cyclometric complexity is still an important metric, as minimising complexity should remain a goal for programmers.

References

Alenezi, M. & Zarour, M. (2020) On the relationship between software complexity and security. *International Journal of Software Engineering & Applications* 11(1)

Ebert, C., Cain, J., Antoniol, G., Counsell, S. & Laplante, P. (2016) Cyclomatic Complexity. *IEEE Software* 33(6): 27-29.