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? Justify all opinions which support your argument and share your responses with your team.

Cyclomatic Complexity is an important metric for software development and testing. It helps measure the risk of a bug according to a specific function. The more decision points a function has, the higher the risk. In project management and software engineering, we must manage these risks to deliver the product on time and ensure it is secure and bug-free. Cyclomatic Complexity is still relevant in today’s world more than ever since the software is getting exponentially larger and more complex. In a way, Cyclomatic Complexity does help developers create secure software. Since bugs are one way for a hacker to exploit software. The few bugs there are, the better and more secure the software is. However, this does not mean we should solely rely on Cyclomatic Complexity because it does not measure how secure a specific function is. It only measures the likely hood of a function having a bug. For example, a function could be written with a very low Cyclomatic Complexity score but with a significant bug like a buffer overflow.

References:

Beningo, J. (2022) *3 reasons to monitor cyclomatic complexity*, *Embedded.com*. Available at: https://www.embedded.com/3-reasons-to-monitor-cyclomatic-complexity/ (Accessed: December 12, 2022).