Relevance of Cyclomatic Complexity when Developing Object-Oriented Code

Schultz (2021) explains Cyclomatic Complexity and examples. The examples provided elaborate on the depth or path complexity inside code to support the calculation of complexity.

```
public static string IntroducePerson(string name, int age)
{
    var response = $"Hi! My name is {name} and I'm {age} years old.";

    if (age >= 18)
        response += " I'm an adult.";

    if (name.Length > 7)
        response += " I have a long name.";

    return response;
}
```

Fig 1. Example code snippet

The example in Fig 1 depicts a simple method but from this, we can calculate and demonstrate the level of complexity. The calculation from a code perspective can be summarised by evaluating the executions inside the function. The code contains 3 separate executions. Starting with an assignment using the parameters, followed by 2 logical branches that provide a response based on the inputs and concatenate a response value and finally return the assigned value of the variable. From a calculation point of view, this would be a 3 in complexity because of the paths within the function.

References:

Schultz, C. (2021) Cyclomatic Complexity Defined Clearly, With Examples. Available from https://linearb.io/blog/cyclomatic-complexity/ [Accessed 26 March 2023].