Code smell and cyclomatic complexity

Code smell describes a scenario in which the code contains errors which may prevent it from running optimally. It may be discovered during testing either prior to deployment or during an issue after deployment. Thomas Mcabe came up with the idea of cyclomatic complexity, which “rates” the code based on a number of factors. A higher rating indicates that the code is “smelly: and may not operate optimally.

In my opinion, this issue may not be as important as it once was due to the development of useful IDE’s, such as PyCharm. An error in coding will be immediately noticeable, and may be rectified during coding, thus reducing the need to run a Mcabe cyclomatic complexity assessment. This does not mean that it is not useful, it merely runs in the background, while a developer is coding, reducing the developer’s reliance on running the test at the end of a coding sprint.