Relevance of Cyclomatic Complexity when Developing Object-Oriented Code

Cyclomatic complexity certainly becomes more relevant in the industry. Ebert, Cain, Antoniol, Counsell & Laplante (2016) refer to and discuss Thomas McCabe's metric and how it applies to the benefits of software projects and predictions of defect rates. What is interesting is the view that CC is more aligned with the industry and how it can often fail to consider CC and development practicalities. What is interesting in their paper is the view is split from both perspectives. James Cain offers compelling arguments based on his experience as a professional programmer and how CC is widely employed in the industry or at least interpretations or perhaps a selective approach to certain aspects of the CC model. What is apparent and most relevant is the approach of the model itself in how it can provide a meaningful way of communicating between stakeholders as a metric to measure software delivery. When you have a metric that considers the level of complexity, inheritance and paths inside code between objects and methods in OOP it provides an effective way to communicate with those responsible for software development and its quality.

References:

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