1. Cyclomatic complexity is a measure of the complexity of a program, “based on the number of linearly independent paths through the program's source code”. Corporations may use cyclomatic complexity as a measure of code quality because it can provide insight into the maintainability and testability of the code. However, academia may not place as much emphasis on cyclomatic complexity because it does not necessarily reflect the real-world performance or functionality of the program. https://www.cqse.eu/en/news/blog/mccabe-cyclomatic-complexity/

2. Three software defect models are:

- The defect-based model, which focuses on identifying and fixing defects in the software.
- The defect-density model, which measures the number of defects per unit of software size (e.g. lines of code).
- The defect-potential model, which estimates the likelihood of defects based on factors such as the complexity of the code and the skill level of the developers.

I would recommend using the defect-based model to my manager because it focuses on identifying and fixing actual defects in the software, which can help improve the quality and reliability of the final product. The other models, while useful for identifying trends and patterns in defects, may not necessarily address the root cause of the defects and may not be as effective at improving the overall quality of the software.