Compare the effectiveness of each tool in defining and identifying code quality. What can you conclude about the effectiveness of each approach?

Pylint

Pylint is taken the longest to run but produced the most detailed results. Pylint prefixes each violation with a character to indicate the type of issue. R for good practice metric violation that needs to be Refactored, C for coding standard violation, W for warning or minor issues, E for errors causing important issues and F for Fatal errors

PyFlakes

PyFlakes doesn't identify stylistic errors such as missing docstrings. It instead focuses on potential errors. Since it does not produce as detailed results and the results aren't organised it is significantly faster than Pylint. Due to the format of the results, it can be harder to interpret.

Pycodestyle

Pycodestyle tests code against the PEP8 styling guideline. It doesn't check naming conventions or docstrings. Pycodestyle categories the results for improved readability. Pycodestyle can be configured to ignore certain PEP8.

Pydocstyle

Pydocstyle is similar to Pycodestyle however it only checks docstrings against the PEP257 conventions. Pydocstyle also categories the results for improved readability.

Effectiveness of linters

Pylint, PyFlakes, Pycodestyle and Pydocstyle are all useful tools in enabling you to write clean code that conforms to the relevant styling guidelines and best practises. While they are extremely useful the value gained will largely depend on the developer’s expertise as if they have already written high quality code then the linters won’t provide many insights. As the saying goes a fool with a tool is still a fool. As they all do similar things it’s ultimately the developer’s decision to decide which tools to use as all the tools have pros and cons and should probably be used in conjunction with each other to produce the best results.