Cerca Trova

Checkliste Code Review

1. Code formatting

    1.1. Using alignements (left margin) and proper white space,
    block starting point and ending point are identifiable easily

    1.2. Proper naming conventions

    1.3. Code should fit in standard 14 inch laptop screen -> no horizontal scroll needed

    1.4. Sufficient code description / explanatory comments

2. Architecture

    2.1. Code lies in appropriate folders

3. Coding best practice

    3.1. Group similar values under an enumeration

    3.2. Sufficient code description / explanatory comments, workarounds, temporary fixes, pending tasks (TODO's)

    3.3. Avoiding multiple if/else blocks

    3.4. Using framework features, wherever possible instead of writing custom code

4. Non-functional requirements

    4.1. Maintainability (App should require the least amount of effort to support in near future, should be easy to identify and to fix)

        4.1.1. Readability (Code should be self-explanatory, using appropriate names for variables, functions and classes)

        4.1.2. Testability (Code should be easy to test)

        4.1.3. Debuggability (Provide support to log the flow of control, parameter data and exception details to find the root cause easily)

4.1.4. Configurability (Keep the configurable values in place so that no code changes are required, if the data is changed frequently)

4.2. Reusability

4.2.1. DRY (Do not repeat yourself): same code should not be repeated more than twice

4.2.2. Consider reusable services, functions and components

4.3. Reliability

4.3.1 No not-used functions in the code

4.4. Extensibility

4.4.1. Easy to add enhancements with minimal changes to the existing code, one component should be easily replaceable by a better component

4.5. Performance

4.5.1. Use a data type that best suits the needs

Sources:

Static analysis tools for python: https://github.com/mre/awesome-static-analysis

https://www.codacy.com/blog/review-of-python-static-analysis-tools/:

Pylint
Pyflakes
MyPy