



# Code quality for grand with SonarQube

JM. Colley (JMC), LPNHE



# Plan

- Quality tools used
- Quality profile
- Quality gate
- Quality workshop

# Quality tools used

- Static analysis
  - Pylint
  - Mypy
- Dynamics analysis
  - Coverage.py apply on directory grand/tests
- SonarQube dashboard
  - Disable sonarlint for python
- Code formatter
  - black

# Quality tools used

- Grand package, for local
  - Directory « grand/quality » with requirement to install quality tools in local
  - Set of scripts quality for grand without parameter
    - grand\_quality\_xxxx.bash
- Github continuous integration, for each push
  - Check Pylint result
  - Check Mypy result
  - Check the percentage of code tested
  - Push report result on SonarQube
    - on grand\_master project for master branch
    - on commiter branch « user\_<github login> » for others

# Development cycle with sonarQube



# Quality profile

- Actually
  - Pylint : check only FATAL and ERROR level
    - No : WARNING, CONVENTION ([PEP8 level](#)), REFACTORING
    - No PEP8 is currently tested
    - Defined in grand/quality/pylint.conf
  - Sonarlint in SonarQube : totally disable for issue => just only pylint error message but get other metrics like code duplication
  - Mypy : disable some false positive
    - Defined in grand/quality/mypy.conf
- We check only grand/grand directory (not examples)

# Quality gate v.1

- With github CI and SonarQube, failed if
  - Pylint message > 0
  - Mypy message > 0
  - Percent coverage < 80 %
- Specific sonarqube « grand\_quality\_gate\_level\_1 »

grand_quality_gate_level_1		
Conditions ⓘ		
Conditions on Overall Code		
Metric	Operator	Value
Coverage	is less than	80.0%
Duplicated Lines (%)	is greater than	3.0%
Maintainability Rating	is worse than	A
Major Issues	is greater than	0
Reliability Rating	is worse than	A

# Quality gate v.1

- Master passed quality gate v1 and contents continuous integration mechanism
  - Branch grand\_coordinates developed by Ramesh (coordinate calculus without astropy) passed the quality gate v.1 so merged with master



# Quality gate v.2

- Next step, to discuss later
  - Certainly contents some PEP8 recommendations like naming convention ?
  - Include new criteria on number of comment ?
  - Increase test coverage to 85 % ?

# Workshop quality applied to grand

- Plan

- Set local quality tools in grand package
- Use framework pytest to improve coverage
- Solve some pylint issues
- Solve some mypy issues
- Create your personal project on sonarqube server
- Solve some sonarlint issues

# Workshop quality applied to grand

- Set local quality tools in grand package
  - git clone <https://github.com/grand-mother/grand.git>
  - cd grand
  - git checkout workshop
  - git checkout -b <name\_branch>
  - git push origin <name\_branch>
  - source env/setup.sh
  - python -m pip install -r quality/requirements.txt
  - grand\_quality\_analysis.bash
  - See [Sonarqube server](#)
  - See github grand branch <name\_branch>
- Use framework pytest to improve coverage
  - grand\_quality\_test\_cov.bash
  - Solve problem
  - Git commit
  - git push origin <name\_branch>
  - See github grand branch <name\_branch>
  - See [Sonarqube server](#)