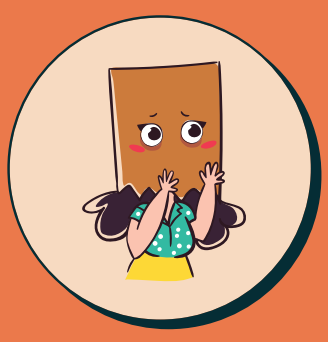
CREATED BY: @SimonMolinsky

SCIENTIFIC (OPEN) SOFTWARE DEVELOPMENT RULES



DON'T BE ASHAMED!

Scientific Software development is not reserved for scientists! Do you know how to code and your software is a solution for some scientific problem? It's enough! Share your work - that's the best way to find collaborators and users.

EVOLUTION NOT REVOLUTION

Did you rewrite functionalities from one language to the other? Did you implement **only** one abstract model? It's enough!

COMPLETENESS

Don't leave loose ends related to the main functionalities. Software should do what it's expected to do. Use examples to prove it.

WELL DOCUMENTED

- As much as possible examples,
- Comments in the code,
- Every function with a docstring,
- API reference,
- Bibliography!
- How to cite your software?

TESTED... BUT NOT TOO MUCH

- Unit tests,
- Test by examples,
- But don't focus on this aspect, let the users test your software.

BIBLIOGRAPHY:

- Lee BD (2018) Ten simple rules for documenting scientific software. PLoS Comput Biol 14(12): e1006561. https://doi.org/10.1371/ journal.pcbi.1006561

- Neidhardt A (2017) Writing Code for Scientific Software. In: Applied Computer Science for GGOS Observatories: Communication, Coordination and Automation of Future Geodetic Infrastructures. Pages: 13-130. Springer International Publishing. https://doi.org/10.1007/978-3-319-40139-3_2