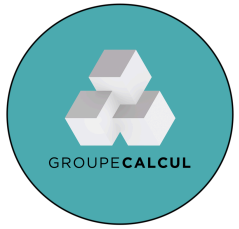

Quelques éléments bibliographiques

Loïc Gouarin, Matthieu Haefele

Paris, Décembre 2023

Publish your computer code: it is good enough

Arguments against publishing :

- It is not common practice
- **=> this should change**
- People will pick holes and demand support and bug fixes
- **=> Publishing code may see you accused of sloppiness. Not publishing can draw allegations of fraud. Which is worse?**

- The code is valuable intellectual property that belongs to my institution.
- **=> Really, that little MATLAB routine to calculate a two-part fit is worth money? And for larger software, without scientist expertise, it is not software any more, it is abandonware**
- It is too much work to polish the code
- **=> if your code is good enough to do the job, then it is good enough to release — and releasing it will help your research and your field**

Publish your computer code: it is good enough, *Barnes, Nick*, Nature, 2010, DOI 10.1038/467753a

Achieving Quality in Open Source Software

Table 1

Quality management in open source and closed-source software development

Closed source	Open source
Well-defined development methodology	Development methodology often not defined or documented
Extensive project documentation	Little project documentation
Formal, structured testing and quality assurance methodology	Unstructured and informal testing and quality assurance methodology
Analysts define requirements	Programmers define requirements
Formal risk assessment process—monitored and managed throughout project	No formal risk assessment process
Measurable goals used throughout project	Few measurable goals
Defect discovery from black-box testing as early as possible	Defect discovery from black-box testing late in the process
Empirical evidence regarding quality used routinely to aid decision making	Empirical evidence regarding quality isn't collected
Team members are assigned work	Team members choose work
Formal design phase is carried out and signed off before programming starts	Projects often go straight to programming
Much effort put into project planning and scheduling	Little project planning or scheduling

Achieving Quality in Open-Source Software, *Aberdour, Mark*, IEEE Software, 2007, DOI 10.1109/MS.2007.2

Achieving Quality in Open Source Software

- Open Source Software produces good quality code
- Same steps as closed source soft but done differently
- Spontaneous contributions, architecture management comes after
- A bit like the collective moment today
- Community involvement
- Rewards for developers for code quality to get a vote or approuval of the community
- Rewards for fast bug fix

Warning: cited projects are very large (linux kernel, mozilla, apache...), research software might be different

Other interesting references

On the usage, co-usage and migration of CI/CD tools: A qualitative analysis, *Pooya Rostami Mazrae, Tom Mens, Mehdi Golzadeh, Alexandre Decan*, Empir Software Eng 28, 2023, DOI 10.1007/s10664-022-10285-5

Open-Source Software in the Sciences: The Challenge of User Support, *Jason Swarts*, 2019, Journal of Business and Technical Communication, 33(1), 60-90. DOI 10.1177/1050651918780202.

Ten simple rules on writing clean and reliable open-source scientific software, *Hunter-Zinck H, de Siqueira AF, Vásquez VN, Barnes R, Martinez*, PLOS Computational Biology 17. DOI 10.1371/journal.pcbi.1009481.

Open Source Software Development Process: A Systematic Review, *Bianca Minetto Napoleão and Fabio Petrillo and Sylvain Hallé*, 2020, 2008.05015, arXiv.