

authors: Jurriaan H. Spaaks
Jason Maassen
Frank J. Seinstra
date: October 8, 2018
doi: 10.5281/zenodo.1451751

Netherlands eScience Center Software Sustainability Protocol

Today, software plays a crucial role in advancing and accelerating state-of-the-art academic research. It is therefore important to adhere to proven best practices when developing research software, as it will help avoid errors, improve maintenance and sustainability, while accelerating the overall development process. Also, it will help other researchers to better understand the intricacies of the software, and the analysis performed with it. This is an important prerequisite for reproducibility of scientific results, and it will allow other researchers to adopt the software into their own workflows, possibly even contributing to the software or expanding it.

To promote the practice of Open Science, and the principles of FAIR¹ data and software, the Netherlands eScience Center actively participates in the National Platform Open Science². In line with this initiative, the eScience Center strives to make the software developed in the projects it participates in to become freely and sustainably available, as much as possible, for reuse by other researchers. To this end, the Principal Investigator of every project (co-)funded by the eScience Center is required to submit a *Software Sustainability Plan*, i.e. a document that makes explicit the researcher's foreseen actions regarding software sustainability.

Software Sustainability Plan

The questions in the following sections are meant to make transparent the expected sustainability and impact of software developed as part of an eScience Center project. The questions are classified into three groups: 1) *minimum effort*, 2) *recommended practices*, and 3) *long-term aspects*.

Naturally, not all software is equally viable when it comes to potential reuse; research software typically consists of some generic components which could in principle be reused in other projects, which are then combined with other, non-reusable components which tailor the software to the problem at hand.

Generally speaking, at least the minimum effort should be made for the complete body of code, i.e. both the reusable and non-reusable parts (see section *Minimum Effort* below). For the potentially reusable parts, higher standards should be set (see section *Recommended Practices* below).

Note that the eScience Research Engineers part of your research team can provide advice when compiling your Software Sustainability Plan and can help with its implementation during the project.

The Software Sustainability Plan should be submitted within 4 months after the project has been awarded funding. The eScience Center will approve the plan as soon as possible thereafter. Approval of the Software Sustainability Plan by the eScience Center is a condition for disbursement of the funding. The Software Sustainability Plan can be adjusted during the research.

¹ <https://www.go-fair.org/fair-principles>

² <https://www.openscience.nl>

1. Minimum Effort

From the start of the project,

- a) Will the software be available under a permissive license such as Apache-2.0 (preferred), MIT, BSD, or GPL?

Yes
 No (warrants explanation)

- b) Will the software be developed in a publicly accessible repository such as GitHub (preferred), GitLab, or BitBucket?

Yes
 No (warrants explanation)

- c) Will you adhere to the FORCE11 recommendations³ for citing software?

Yes
 No (warrants explanation)

As soon as possible after the projects starts,

- d) Will a persistent identifier such as a DOI issued by Zenodo be added to the software?

Yes
 No (warrants explanation)

2. Recommended Practices

As soon as possible after the project starts,

- a) Will the software have its own entry in one or more software repositories such as a Research Software Directory⁴, KBLab⁵, BioTools⁶, or any other relevant directory?

Yes
 No (warrants explanation)

- b) Will the software have documentation targeting new users, illustrating the software's intended usage?

Yes
 No (warrants explanation)

³ <https://doi.org/10.7717/peerj-cs.86> - section: *What software to cite*

⁴ <https://software.esciencecenter.nl>

⁵ <http://lab.kb.nl>

⁶ <https://bio.tools>

c) Will you fill out the CII Best Practices Badge Program⁷ checklist or an equivalent one?

⇒ *The checklist mainly serves as guidance on what aspects of your software could be better organized. Also, some sections of the checklist may be more relevant than others: use your best judgement in choosing which rules to comply with. While more compliance is better, a 100% score is not usually expected.*

- Yes
 No (warrants explanation)

d) Will you prominently post the results from the abovementioned checklist, for example as a 'badge' in the README?

- Yes
 No (warrants explanation)

3. Long-term Aspects

During the project, the eScience Research Engineer(s) part of your research team will help lay the groundwork for making the software sustainable, for example by implementing software engineering best practices. The sustainability of the software beyond the project's end date, however, ultimately is the responsibility of the project's Principal Investigator (PI). The PI may delegate it to, for example, the research team, or the research community at large. Due to its funding model, the Netherlands eScience Center cannot typically sustain the developed software past a project's end date.

This section of your Software Sustainability plan must focus on the generic question: *What will you do during the project to ensure that the software lives on past the project's end date?*

Below are a few suggestions to get you started. Feel free to interpret the above question in the broadest sense.

a) What efforts are being undertaken to create the critical mass necessary to start building a community of users, promoters, or otherwise interested parties around the software. For example:

- through publications in mainstream media such as blogs, newspaper articles, YouTube videos, tweets, etcetera.
- through organizing workshops, hands-on user trainings, etcetera.

b) Is there additional funding (in-kind or cash) that will be used to support and maintain the project's software outputs?

c) If the software is made available as a service, for how long will the service be offered? Which party will host the service? How will this be made possible financially?

⇒ *For example, some services may be transferred to external institutes such as SURFsara, DANS, or other national or international parties, either during the project or shortly after the project ends.*

d) Describe other aspects that promote the software's longevity.

⁷ <https://bestpractices.coreinfrastructure.org/en>

Contact, Submission, Review and Publication

For more information on the contents of your Software Sustainability Plan, please contact the eScience Coordinator or eScience Research Engineer(s) as part of your research team.

For submitting your Software Sustainability Plan, please send it by email to the eScience Coordinator in your project team. After submission, the eScience Coordinator will ask the software sustainability experts in the eScience Center to review your Software Sustainability Plan. They will either approve it, or propose adaptations, if necessary. The final approved Software Sustainability Plan will be included in the set of project deliverables, and kept with all project administration. It will also be used as a reference in the Annual Project Review.

The eScience Center stimulates and strongly advocates the publishing of the approved Software Sustainability Plan, also on the project pages part the eScience Center website. Making the Software Sustainability Plan publicly available is the responsibility of the Principal Investigator.