Five recommendations for “FAIR software”

nationaal programma open science

The five recommendations

01/ Use a publicly accessible repository with version control
02/ Add a License
03/ Register your code in a community registry
04/ Enable citation of the software
05/ Use a software quality checklist

nationaal programma open science

Use a publicly accessible *repository* with version control

/01

nationaal programma open science

Use a publicly accessible repository with version control

Why is this important

Why public

Developing scientific software in publicly accessible repositories enables early involvement of users, helps build collaborations, contributes to the reproducibility of results generated by the software, facilitates software reusability, and contributes to improving software quality. Taken together, this ensures that your software has the best chance of being used by as many people as possible while promoting transparency.

Why version control

Using a version control system allows you to easily track changes in your software, both your own changes as well as those made by collaborators. There are many flavors of version control systems, ranging from older systems such as CVS and Subversion to more modern ones such as Git, Mercurial, and Bazaar. By configuring your version control system to use GitHub, GitLab or Bitbucket, you'll even have backups of every version of the software you ever made. Additionally, those platforms offer collaboration tools such as an issue tracker and project management tools, and you'll be able to use third-party services such as code quality checkers, correctness checkers, and a lot more.

Use a publicly accessible repository with version control

Help me choose

Git is the version control system which is most feature-rich, most modern and most popular by a good margin, and we heartily recommend you use it for all you version control needs. To get the best out of Git, use it in combination with GitHub.com, Bitbucket.org, or GitLab.com.

Recommended links

- ☑ How to get started with git
- ☑ Choosing a platform for your software project
- ☑ GitHub.com
- ☑ BitBucket.org
- ☑ GitLab.com

Add a *license*

/02

Add a license

Why is this important

Any creative work (including software) is automatically protected by copyright. Even when the software is available via code sharing platforms such as GitHub, no one can legally use it unless they are explicitly granted permission. This is done by adding a software license, which defines the set of rules and conditions for people who want to use the software.

Therefore even a small analysis script should be covered by a license.

Usage license

You should also consider the usage license of packages you are using. Make sure you respect the conditions set by the license of software packages you use.

Add a license

Help me choose

We recommend you stick to one of the more popular licenses. Because these are typically written by lawyers, the license text is precise in expressing its terms. While that carries the unfortunate side effect of being difficult to understand, the widespread use of the more popular licenses means that there is a larger number of people who understand how the letter of the law should be interpreted.

Some of our favorite licenses are the Apache-2.0 and MIT licenses. These permissive licenses have very few restrictions, allowing others to easily reuse your work. We recommend you use choosealicense.com to find out which license is best for your purposes. If you just want to check what is and what is not allowed under a given license, visit tldrlegal.com to find out.

Recommended links

- ☑ choosealicense.com
- ☑ tldrlegal.com
- ☑ How to add a license to your GitHub repository

Register your code in a community *registry*

/03

nationaal programma open science

Register your code in a community registry

Why is this important

For others to make use of your work, they need to be able to find it first. Community registries are like the yellow pages for software; registering your software makes it easier for others to find it, particularly through the use of search engines such as Google. Community registries typically employ metadata to describe each software package. With metadata, search engines are able to get some idea of what the software is about, what problem it addresses, and what domain it is suited for. In turn, this helps improve the ranking of the software in the search results – better metadata means better ranking.

Register your code in a community registry

Help me choose

Community registries come in many flavors. Choosing the one that is best suited for your needs can be tricky. Here are some things to think about:

- How much traffic does the community registry get?
- Is the community registry targeting the audience you are trying to reach?
- What metadata does the community registry offer? This is sometimes described in the documentation of the registry, but you can also see for yourself by installing a tool like the OpenLink Structured Data Sniffer.

Finally, ask a couple of colleagues which registries they would use if they were looking for software like yours.

Recommended links

- [View a list of software registries]

Enable *citation* of the software

/04

Enable citation of the software

Why is this important

Citation is an integral part of scientific accountability and reproducibility, but accurately citing software is inherently more difficult than citing a paper.

Citation metadata format

CodeMeta and the Citation File Format were specifically designed to enable citation of software and will likely meet your needs.

Enable citation of the software

Help me choose

Initialize your CITATION.cff files here.

Regarding archiving copies of your software, look for services that store their own copy of a snapshot of your software, such that whatever persistent identifier you get (DOI, URN, ARK, etc) points to a specific version of the software, and will continue to resolve to exactly that version for the foreseeable future. Ideally, storing snapshots of your code should be as easy as possible: either at the push of a button, or automatically, for example each time you make a new release of your software.

Archiving services:

- ☑ Zenodo
- ☑ FigShare
- ☑ Software Heritage Archive

Use a software quality checklist

/05

nationaal programma open science

Use a software quality checklist

Why is this important

Checklists help you write good quality software. What exactly constitutes 'good quality' depends on the specific application of the software, but typically covers things like documenting the source code, using continuous testing, and following standardized code patterns.

There are many checklists available. We find that the most useful checklist are those that

1. Allow for a granular evaluation of a software package, as opposed to just pass or fail
2. Explain the rationale behind each item in the checklist
3. Explain how to get started with implementing each item in the checklist

Use a software quality checklist

Help me choose

We recommend that you include the checklist as part of the README, for example as a badge or by including the checklist as a Markdown table. The point is decidedly not to show perfect compliance, but rather to be transparent about the state of the code while providing the necessary guidance on which aspects could be improved.

One of our favorite checklists that meets these criteria is the Badge Program developed by the Core Infrastructures Initiative, but there are many other checklists to choose from.

Here is a list of some candidates:

- ☑ Core Infrastructures Initiative (online, interactive)
- ☑ DLR Class 1, Class 2, Class 3 (MarkDown)
- ☑ SSI's software evaluation checklist (Google form)
- ☑ CLARIAH checklist (PDF page 38-42)
- ☑ EURISE (MarkDown)

Endorsers of these recommendations

netherlands eScience center

Data Archiving and Networked Services DANS

Radboudumc university medical center

KB nationale bibliotheek

bioexcel Centre of Excellence for Computational Biomolecular Research

VERSEN

NWO

elixir

DTL

4TU.Centre for Research Data

uni.lu LCSB

SURF

nationaal programma open science

CODE REFINERY

ifb INSTITUT FRANÇAIS DE BIOINFORMATIQUE

CWI Centrum Wiskunde & Informatica

RSE SOCIETY OF RESEARCH SOFTWARE ENGINEERING

Software Sustainability Institute

About FAIR Software

The FAIR principles are a concept which originated in data management. The acronym stands for Findable, Accessible, Interoperable and Reusable. They have served as a flagship for promoting good data management practices, but until recently they were not directly applicable to software. We hope that the FAIR principles will have a positive effect in research software development.

These slides are an adjusted version of the content from the **https://fair-software.eu/** website, which is a collaboration between the Netherlands eScience Center and DANS.

https://fair-software.eu/

nationaal programma open science