



Share your software -SOFTWARE DOCUMENTATION AND CITATION CHECKLIST

Shelley Stall, Alison Specht, Margaret O'Brien, Jeaneth Machicao,
Pedro-Luiz-Pizzigatti Corrêa, Romain David, Rorie Edmunds,, Nobuko
Miyairi,, Yasuhiro Murayama,, Solange Santos,, et al.

► To cite this version:

Shelley Stall, Alison Specht, Margaret O'Brien, Jeaneth Machicao, Pedro-Luiz-Pizzigatti Corrêa, et al.. Share your software -SOFTWARE DOCUMENTATION AND CITATION CHECKLIST. AGU; ERINHA. 2023, <https://doi.org/10.5281/zenodo.7841908>. hal-04179562

HAL Id: hal-04179562

<https://hal.science/hal-04179562>

Submitted on 10 Aug 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Share your software – SOFTWARE DOCUMENTATION AND CITATION CHECKLIST

Welcome to “Your Open Science Journey”!

Plan, manage and share your software. Use this checklist to improve your software management and sharing practices.

Target Audience

Primary: Team or project researcher

Secondary: Team or project lead

This checklist is generalized and will need to be adjusted based on your institution, lab, research team, and/or funder requirements.

A. PLAN AND MANAGE YOUR SOFTWARE DURING YOUR RESEARCH PROJECT

1. Plan the Format and Organize Your Software Files:

- a. **Use** open source software tools where possible.
- b. **Choose descriptive and useful file names** that both humans and machines can read. Examples of ideas to improve filename convention may include, 1) Using deliberate delimiters such as “-” to connect terms together and “_” to separate different information; 2) Avoiding special characters (e.g., \$), spaces, and punctuation; 3) Creating meaningful names and metadata that explain the content; 4) Using methods that facilitate default ordering such as YYYY-MM-DD date format (an international standard).
See Jenny Bryan’s [How to name files](#)
- c. **Determine hierarchical folder and file structure for the project.** Be consistent, use one directory for one project, separate by function (e.g., data, code) and by output (e.g., figures, results). Ensure that original, raw data is separate, cannot be edited, and copies from the raw data are made for editing. Reference the [compendium “file-structure”](#) developed by the Turing Way.
- d. **Create one or more README files.** Capture information that will be vital to describing and citing your software later including authors (contributors), project/software name, releases (versions), coding standards, dependencies, frameworks, tests, and sustainability. –
- Depending on your file structure and complexity you may want a README for each type of software (re: data cleansing, analytics) as well as one to describe the entire file structure. A helpful tool is <https://readme.so>. Keep your README file updated.

2. Manage Your Software More Efficiently with Versioning and Computational Notebooks:

- a. **Use a version control system to keep track of file changes** or the history of a project (e.g., software updates). Platforms such as [GitHub](#), [GitLab](#), or [Open Science Framework \(OSF\)](#), provide additional collaborative and online backup capabilities.
- b. **Use a computation notebook (e.g., Jupyter, R Markdown)** to make your research easier to understand and more reproducible.

- c. **Be Aware of and adhere to the usage license(s) for software created by others.**

3. Store and Backup Your Software Files:

- a. **Keep your files safe by ensuring they are automatically backed up** by a trusted system. Take advantage of your institution's backup services if offered. It is preferred that the backup system be in a different geographic location (e.g., cloud storage service).

B. SHARE YOUR SOFTWARE OPENLY

Use these steps when your research completes a milestone and before publication.

1. Select Your Software Preservation Repository

- a. **Choose a repository.** Software development platforms such as GitHub are not long term preservation solutions. Look ahead to when you will preserve and cite your software and identify a repository you will use (domain, institutional, general). [SciCodes membership](#) includes commonly used domain and general repositories.
- b. **Determine what metadata** you will need. E.g. Making your code citable via Zenodo and GitHub and using Zenodo's Sandbox allows you to see additional metadata that you will need including ORCIDs for authors, related identifiers (e.g. paper DOIs), funding information, etc. and the CodeMeta file that is created.
- c. **For simulation software/models**, generally the model and configuration should be preserved. The EarthCube model community [developed a framework to guide](#) what to preserve from your research and is easy to adapt.

2. Document Your Software Prior to depositing - README files, Metadata, and Licenses.

a. Prepare your metadata. (recommended)

Many preservation repositories provide guidance for the type of documentation necessary to understand your data. When this guidance is not available, follow the [CodeMeta](#) guidance.

b. License your Software (required)

Select and use an open-source license. Use the "[Choose an open-source license](#)" site to decide.

3. Deposit Your Software with your Selected Repository

- a. **Add instructions** to your development platform (e.g., GitHub) on how to cite your software. Use the [Citation File Format](#). FORCE11 Software Citation Implementation Working Group has prepared detailed guidance for [developers](#) and [authors](#).
- b. Ensure that the repository clearly displays your preferred citation.

C. CITE ALL SOFTWARE

- 1. **Include your Software Availability Statement** if it has a significant impact on your research and/or your research is not reproducible without it. The statement should include:
 - a. A very brief description of what the software is used for
 - b. The repository DOI/PID for your preserved software (e.g. Zenodo)

- c. Version of the software used and preserved
 - d. (Open source) license and other access conditions
 - e. Development platform link (e.g. GitHub repository)
 - f. In the methodology section of your paper, describe how your software works as it pertains to your research
 - g. Include the software citation(s) in your references using the preserved software's repository DOI/PID.
2. Use [the availability statement and citation checklist](#) to ensure your citation and availability statements are complete. See [Data and Software Sharing Guidance for Authors](#) for more details.

References

Chue Hong NP, Allen A, Gonzalez-Beltran A, et al. (2019). Software Citation Checklist for Developers (0.9.0). Zenodo. <https://doi.org/10.5281/zenodo.3482769>

Chue Hong NP, Allen A, Gonzalez-Beltran A, et al. (2019). Software Citation Checklist for Authors (0.9.0). Zenodo. <https://doi.org/10.5281/zenodo.3479199>

Quick Links to related checklists

- [Your Digital Presence](#)
- [Data Documentation and Citation Checklist](#)

To recommend updates, please email datahelp@agu.org. Include the name and DOI for the checklist in your email.

To cite this checklist: Stall, Shelley, Specht, Alison, O'Brien, Margaret, Machicao, Jeaneth, Corrêa, Pedro Luiz Pizzigatti, David, Romain, Miyairi, Nobuko, Murayama, Yasuhiro, Santos, Solange, Wyborn, Lesley, Vellenich, Danton Ferreira, & Mabile, Laurence. (2022). Software Documentation and Citation Checklist. Zenodo. <https://doi.org/10.5281/zenodo.7062413>