

## Foundations of Research Software Publication

**Reproducibility for Matter Research, Online, 17.03.2022**

Tobias Schlauch <Tobias.Schlauch@DLR.de>

Institute for Software Technology

German Aerospace Center (DLR)

<http://www.dlr.de/sc>



# HIFIS

HELMHOLTZ  
FEDERATED  
IT SERVICES

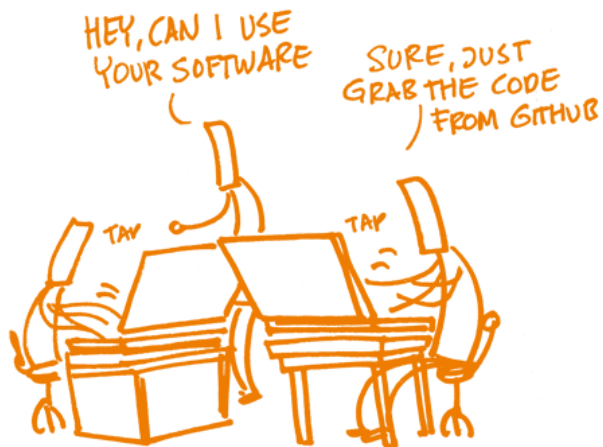
- Reproducibility of scientific results and enabling others to review them are vital in research.
  - Making scientific results reproducible is **context-specific** and can be **hard in the concrete case**.
  - Involved research software introduces a separate domain of potential reproducibility issues.
- In the following we consider **minimum practices** when publishing research code to improve reproducibility of scientific results.

# Making research code openly available helps. But is there more to consider?

OPEN SOURCE

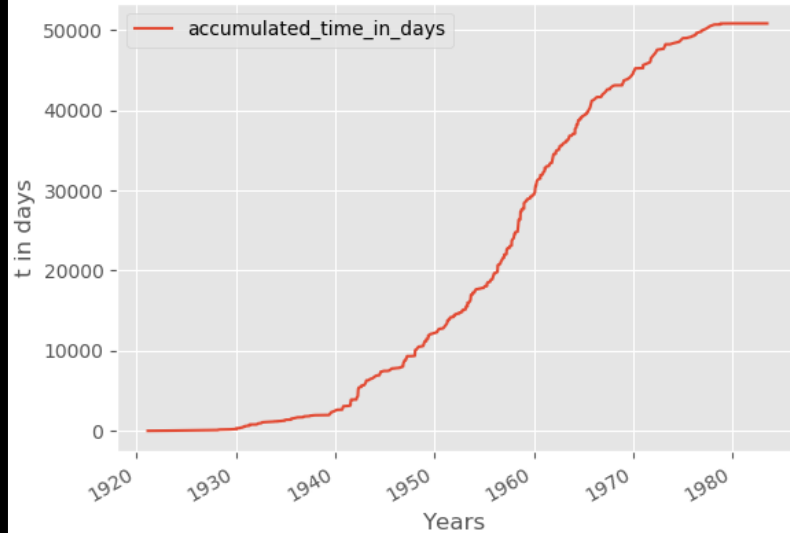
VS

CLOSED SOURCE

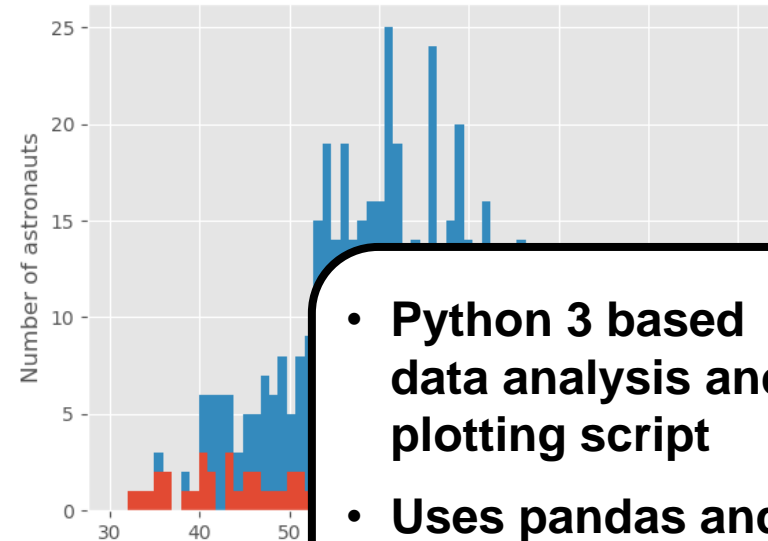


# Example: Astronaut Analysis

Total time humans have spend in space



Dead vs. Alive astronauts



- Python 3 based data analysis and plotting script
- Uses pandas and matplotlib 3

# How should I publish my analysis code to enable others to reproduce or check my results?

---

1. Put your code under version control
2. Make sure that your code is in a sharable state
3. Add essential documentation
4. Add a license file
5. Make your code citable
6. Release your code

# 1. Put your code under version control



---

## Key points

- Version control helps you to prepare the code for sharing.
- Make sure to put all relevant artifacts into the repository.
- **.gitignore** helps you to specify things that you do not want to share.

## 2. Make sure that your code is in a sharable state



### Key points

- Make sure that others can (re-)**use** your code
- Do not share internals and secrets with your code
- Strive for understandable and idiomatic code
- Think about introducing test automation

#### Changes:

- **PEP8 code style**
- **Code comments**
- **Relative file paths**
- **Functions**
- **Better directory structure**

### 3. Add essential documentation

---

#### Key points

- Provide documentation for relevant target groups
- Add a README file as a minimum documentation artifact to your repository

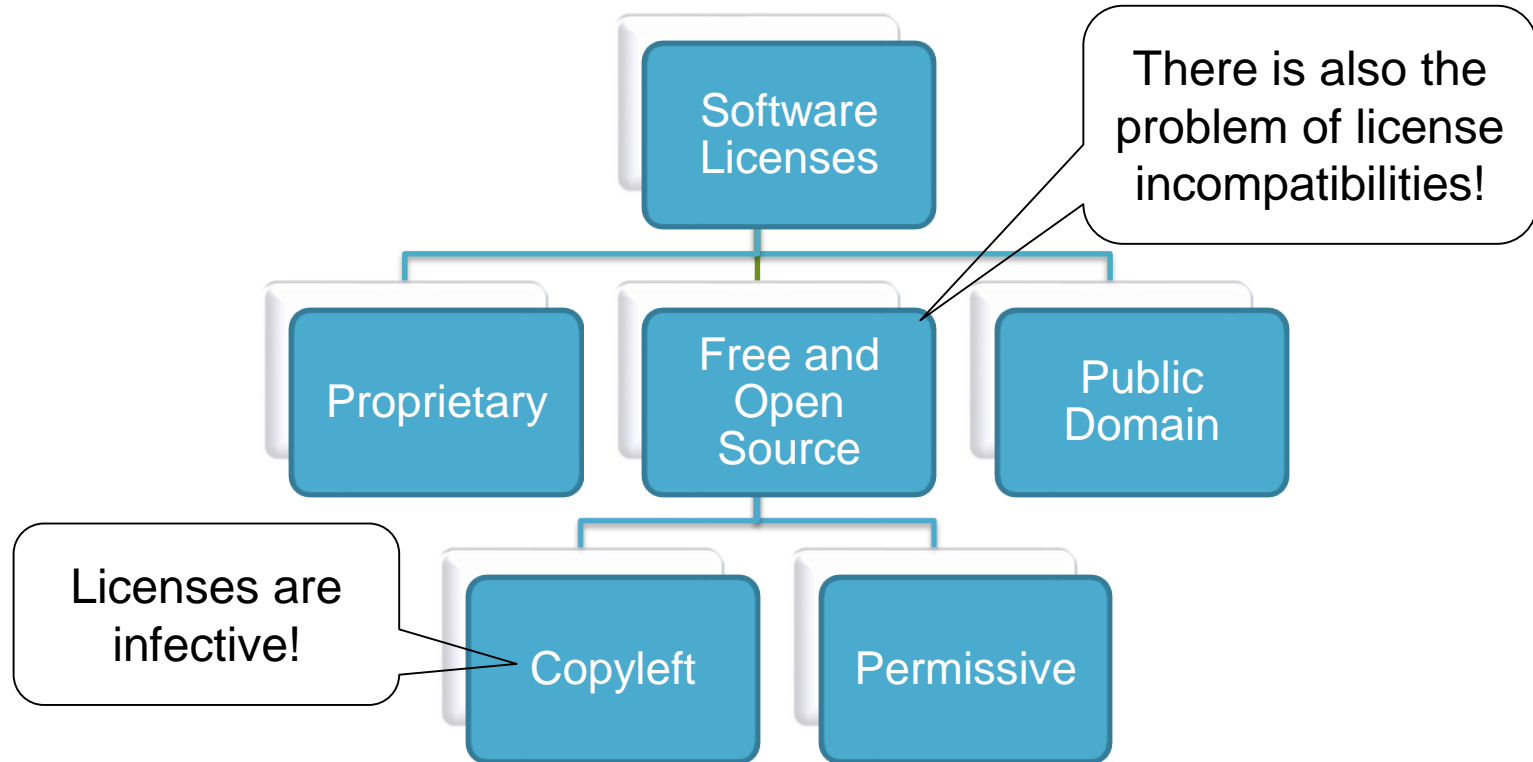


## 4. Add a license

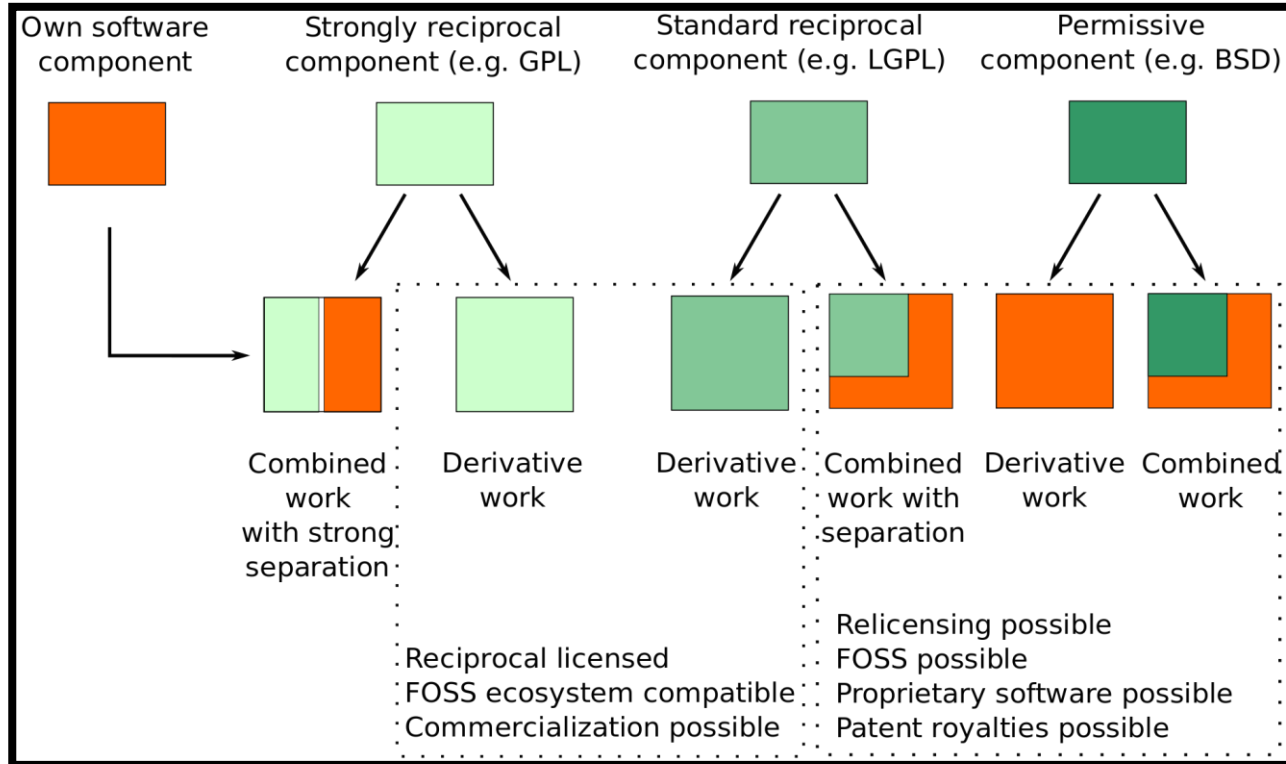
- 
- Excursus on (open source) licenses...

- Copyright:
  - protects the expression of an idea
  - grants exclusive rights to the copyright holder
- Everyone contributing to a software is considered a copyright holder of the software
  - All contributors are considered as copyright holders
  - A company paying an employed developer obtains most of the exclusive rights





# Combining Modules under Different Licenses



**Take care when  
combining  
code under  
different  
licenses!**

## 1. Choose a license

- Analyze / comply with licenses of third-party dependencies

## 2. Ask your boss for permission

## 3. Prepare your code

- Minimum: Add a license file and state the copyright holder(s)
- Recommended: Follow REUSE Specification

**Ask for legal  
advice if you  
are unsure!**

# Add the License File

Adapt the  
copyright hint



```
MIT License Copyright (c) 2013 German Aerospace Center (DLR)
```

```
Permission is hereby granted, free of charge, to any person obtaining a copy  
of this software and associated documentation files (the "Software"), to deal  
in the Software without restriction, including without limitation the rights  
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell  
copies of the Software, and to permit persons to whom the Software is furnished  
to do so, subject to the following conditions:
```

```
The above copyright notice and this permission notice (including the next  
paragraph) shall be included in all copies or substantial portions of the  
Software.
```

```
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS  
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS  
OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,  
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF  
OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

Do not  
change the  
license  
text!

# Document the Licenses of your Dependencies

## Install

The script requires Python  $\geq 3.6$  and uses the libraries `pandas` (*BSD 3-Clause License*) as well as `matplotlib` (*Matplotlib License*).

Please clone this repository and install the `required dependencies`. We suggest that you install them in a virtual environment.

The script has been used on CentOS 7 and Windows 7 with Python 3.6.6.



 **requirements.txt** 32 Bytes 

```
1 matplotlib==3.0.0
2 pandas==0.23.4
```

- 
- **Minimum:** Add a license file and state the copyright holder(s)
  - **Recommended:** Follow REUSE Specification
  - Consider third-party licenses from the very beginning



## 5. Make your code citable

- 
- Excursus in software citation...

# How to cite Software?

## Example

*“The data sets and the notebook containing the analysis details have been published separately [11].”*

### References:

[11] Schlauch, Tobias, & Haupt, Carina. (2019).  
Analysis of the DLR Knowledge Exchange Workshop  
Series on Software Engineering (Version 1.2.0).  
Zenodo. <https://doi.org/10.5281/zenodo.3403991>

- **Authors**
- **Software Name**
- **Software itself**
- **Source code version**
- **Exact version**
- **Publication date**
- **PID**

---

- **Basic steps:**

- Provide citation metadata
- Obtain a persistent identifier
- Add a citation hint

- **Practical approaches:**

- Archive release in an (open) digital repository (e.g., Zenodo)
- Add machine-readable citation metadata and archive via Software Heritage

# Example 1: Archive via Zenodo

## Installation

- Install Python  $\geq 3.6$
- Install the dependencies via: `pip install -r requirements.txt`

## Run the Jupyter notebook(s)

- Start Jupyter: `jupyter notebook`
- Your browser should start and you can select the corresponding analysis notebook.

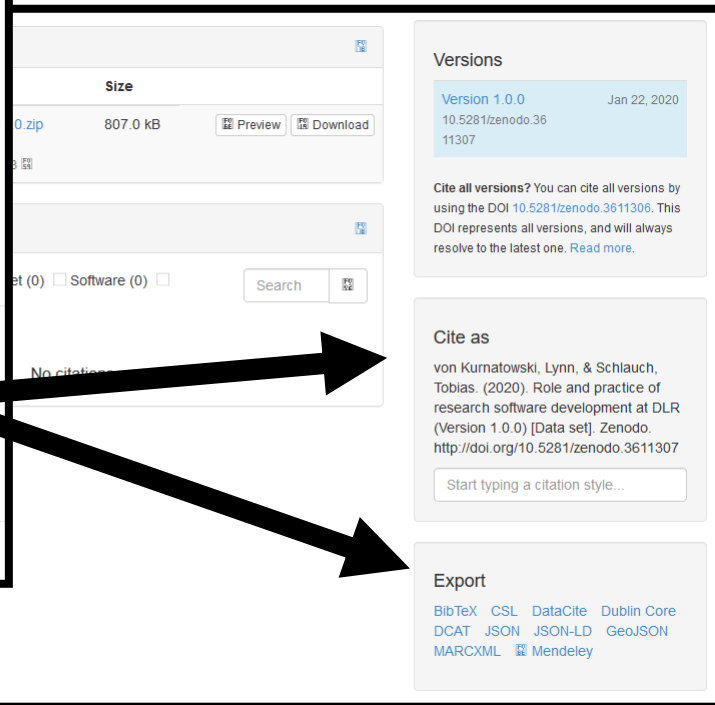
## Citation

If you use this work in a research publication, please cite the specific version that you used using the citation metadata Zenodo `DOI 10.5281/zenodo.3611306`.

You can find an overview about the different versions of the material in the [changelog](#).

## License

Copyright © 2020 German Aerospace Center (DLR)



The screenshot displays the Zenodo archive page for a dataset. It includes a table of files, a 'Versions' section, a 'Cite as' section, and an 'Export' section.

File Name	Size	Preview	Download
0.zip	807.0 kB	<a href="#">Preview</a>	<a href="#">Download</a>

**Versions**

Version 1.0.0 Jan 22, 2020  
10.5281/zenodo.3611306  
11307

**Cite all versions?** You can cite all versions by using the DOI 10.5281/zenodo.3611306. This DOI represents all versions, and will always resolve to the latest one. [Read more](#).

**Cite as**

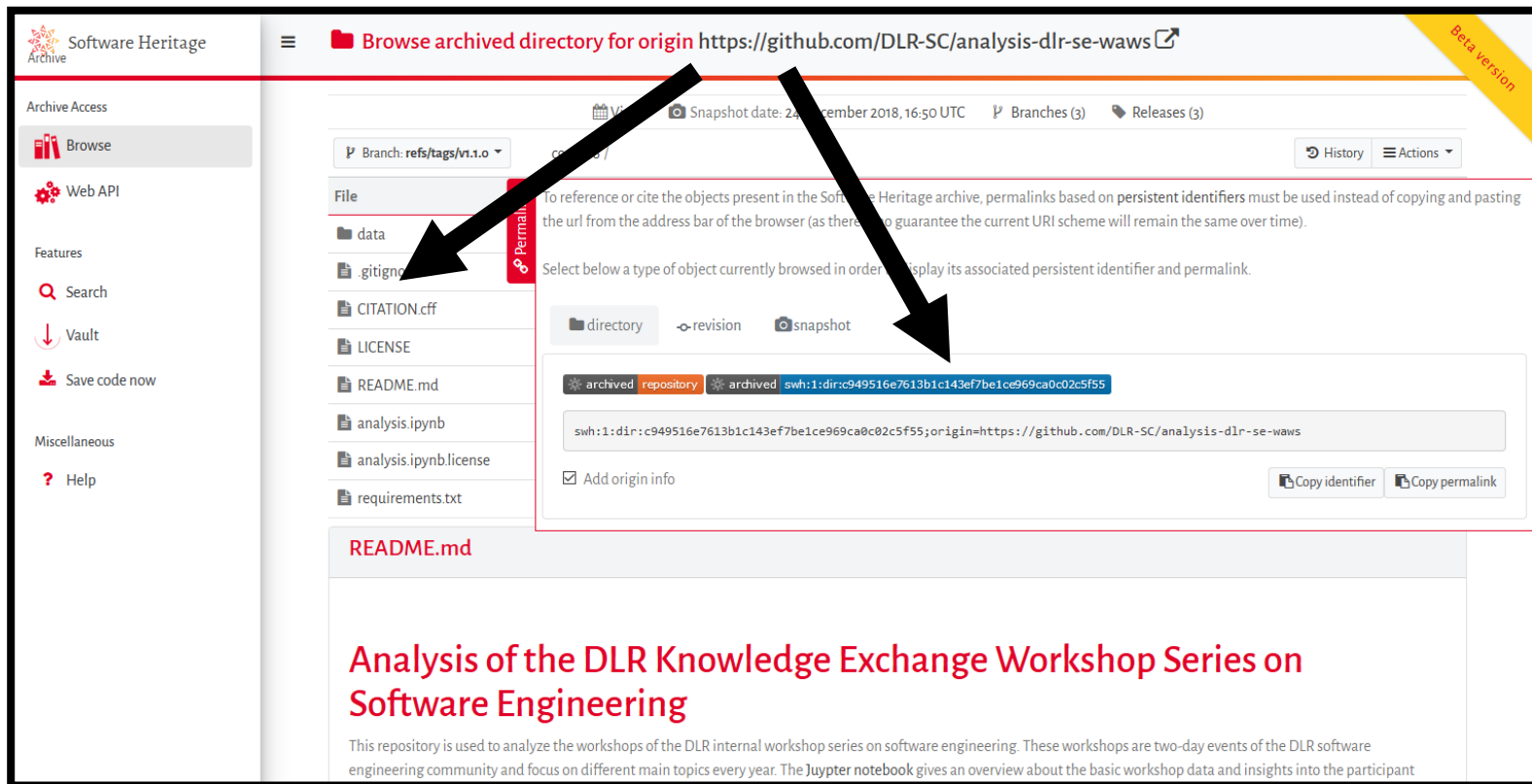
von Kurnatowski, Lynn, & Schiauch, Tobias. (2020). Role and practice of research software development at DLR (Version 1.0.0) [Data set]. Zenodo. <http://doi.org/10.5281/zenodo.3611307>

[Start typing a citation style...](#)

**Export**

[BibTeX](#) [CSL](#) [DataCite](#) [Dublin Core](#) [DCAT](#) [JSON](#) [JSON-LD](#) [GeoJSON](#) [MARCXML](#) [Mendeley](#)

# Example 2: Archive via Software Heritage



Software Heritage Archive

Browse archived directory for origin <https://github.com/DLR-SC/analysis-dlr-se-waws>

Beta version

Archive Access

- Browse
- Web API

Features

- Search
- Vault
- Save code now

Miscellaneous

- Help

Branch: refs/tags/v1.1.0

Snapshot date: 24 September 2018, 16:50 UTC

Branches (3)

Releases (3)

History

Actions

File

- data
- gitignore
- CITATION.cff
- LICENSE
- README.md
- analysis.ipynb
- analysis.ipynb.license
- requirements.txt

To reference or cite the objects present in the Software Heritage archive, permalinks based on persistent identifiers must be used instead of copying and pasting the url from the address bar of the browser (as there is no guarantee the current URI scheme will remain the same over time).

Select below a type of object currently browsed in order to display its associated persistent identifier and permalink.

directory revision snapshot

archived repository archived swb:1:dir:c949516e7613b1c143ef7be1ce969ca0c02c5f55

swb:1:dir:c949516e7613b1c143ef7be1ce969ca0c02c5f55;origin=https://github.com/DLR-SC/analysis-dlr-se-waws

☒ Add origin info

Copy identifier Copy permalink

README.md

## Analysis of the DLR Knowledge Exchange Workshop Series on Software Engineering


This repository is used to analyze the workshops of the DLR internal workshop series on software engineering. These workshops are two-day events of the DLR software engineering community and focus on different main topics every year. The [Jupyter notebook](#) gives an overview about the basic workshop data and insights into the participant


# Increase Chances to receive Credit? Publish in a Software Journal


The Journal of Open Source Software is a developer friendly, open access journal for research software packages.

Committed to publishing quality research software with zero article processing charges or subscription fees.

**Submit a paper to JOSS**

 **Volunteer to review**

 [Explore Papers](#)

 [Documentation](#)

 [Learn More](#)

## Recommended options:

- Journal of Open Source Software (JOSS)
- Journal of Open Research Software (JORS)

# Key Points

- 
- Cite all relevant software packages as good as possible in your academic work
  - Provide relevant information and encourage citation

## 6. Release your code

- 
- Excursus on release management...



- 
- A **release** is a specific working software version.
  - The **release number** uniquely identifies the release (e.g., 1.0.1 or 2022-03-17)
  - A user uses the **release package** to install and use the released software:
    - Contains code + documentation
    - Simplest form: snapshot of your source code repository packaged as Zip file


---

## 1. Prepare your code for release

- a) Define the release number
- b) Update the documentation and citation metadata

## 2. Check your code

## 3. Publish and archive the release

- a) Mark the release in the source code repository using a tag
  - b) Create the release package and make it available
  - c) Archive the release package
- 
- A decorative horizontal bar at the bottom of the slide, featuring a blue background with a green and yellow geometric shape on the right side.

- 
- Mark used, working software versions using release numbers
  - Document important changes in a changelog
  - Archive the release package

- 
1. Put your code under version control
  2. Make sure that your code is in a sharable state
  3. Add essential documentation
  4. Add a license file
  5. Make your code citable
  6. Release your code

- All content is licensed under [Attribution 4.0 International \(CC BY 4.0\)](#) with the following exceptions:
  - HIFIS logos / slide layout, © Helmholtz Association of German Research Centres. All rights reserved.
  - Open Source vs. Closed Source, slide 2, image by Patrick Hochstenbach, [CC0 1.0 Universal \(CC0 1.0\)](#), source: [https://raw.githubusercontent.com/OpenScienceMOOC/Module-5-Open-Research-Software-and-Open-Source/master/content\\_development/images/open\\_research\\_software\\_open\\_source.png](https://raw.githubusercontent.com/OpenScienceMOOC/Module-5-Open-Research-Software-and-Open-Source/master/content_development/images/open_research_software_open_source.png)
  - Copyright logo, slide 9, public domain, source: <https://commons.wikimedia.org/wiki/File:Copyright.svg>
  - License compatibility, slide 11, image by Mikko Välimäki, public domain, source: <https://commons.wikimedia.org/wiki/File:Software-license-compatibility-graph.svg>
  - Philae landing on comet 67 P/Churyumov-Gerasimenko, slide 32: DLR, [CC BY 3.0](#)

# Thank you!

Questions?

[Tobias.Schlauch@dlr.de](mailto:Tobias.Schlauch@dlr.de)

[www.DLR.de/sc](http://www.DLR.de/sc) | Twitter: [@TobiasSchlauch](https://twitter.com/TobiasSchlauch)