Sustainable Research Software Through Software Management Plans



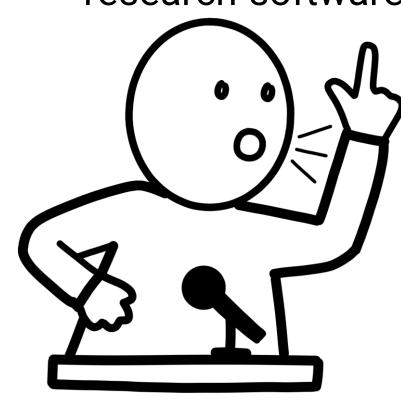
Michael Franke¹, franke@mpdl.mpg.de, https://orcid.org/0000-0002-2661-8242

Yves Vincent Grossmann¹, grossmann@mpdl.mpg.de, https://orcid.org/0000-0002-2880-8947

¹ Max Planck Digital Library, Collections, Landsberger Straße 346, 80687 Munich, Germany

1. CENTRAL OBJECTIVES FOR SMPs

- Support scientists with the reproducibility and sustainability of research software
- Offer service for scientists and research software engineer
- Management tool to promote explicit use of research software



2. NORMATIVE FRAMEWORK

- Software is often necessary for the reproducibility of scientific results
- Reproducibility of results is in terms with the Good
 Scientific Practice, so it should be available
- DFG Guideline 13: "Software programmed by researchers themselves is made publicly available along with the source code."
- Internal guidelines and/or general regulations might require or recommend the publication of software
- Increasing focus on software by third-party funders as an important project outcome

3. SMPs IN THE DISCUSSION

- Chue Hong et al. (2014): "Writing and using a software management plan", https://www.software.ac.uk/resources/guides/software-management-plans.
- Alves et al. (25.10.2021): "ELIXIR Software Management Plan for Life Sciences", https://doi.org/10.37044/osf.io/k8znb.
- Martinez-Ortiz et al. (27.10.2022): Practical guide to Software Management Plans, https://doi.org/10.5281/zenodo.7248877.
- Giraldo et al. (31.05.2023): Workshop machineactionable Software Management Plans. https://doi.org/10.5281/zenodo.8087357.
- and more...

4. DEFINITION OF AN SMP

"According to a DINI/nestor AG Forschungsdaten definition, software management plan (SMP) includes general and technical information about the software project, information about quality assurance, release and public availability, as well as legal and ethical aspects affecting the software."

https://forschungsdaten.info/praxis-kompakt/glossar/#c822402

5. ADVANTAGES THROUGH SMP USE

- Transforming implicit knowledge into explicit knowledge
- Tool for project management with focus on research software
- Use for consulting services, e.g. by local IT, scientific computing unit, third-party funding office
- Quality management and assurance
- Can be used for third-party funding applications
- Better overview of which software projects are run ning in parallel at an institution
- ...

6. TARGET GROUPS

- Scientists, who:have not yet dealt much with software management
- would like to achieve quality in research software with little time investment

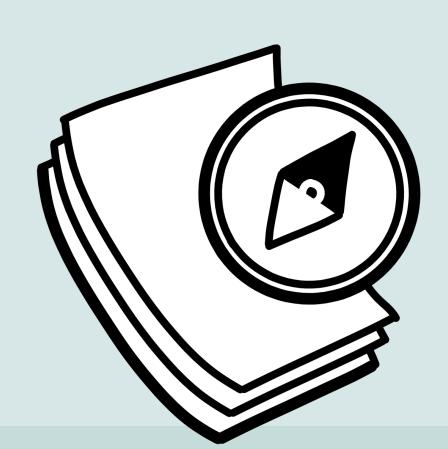
Also colleagues from:

- IT, Scientific Core Unit
- Third-party funding applications
- Project and quality management
- Research coordination
- **.**..

s nt •••

7. CONTEXT OF THE PROJECT

- https://rdmo.mpdl.mpg.de
- Team from MPDL Collections
 Period July 2022 to December 2022
- RDMO as a technical basis
- Result: CC0 push of an SMP catalogue as a contribution to the RDMO community



8. DETAILS ON THE RDMO CATALOGUE

- Title: Software Management Plan for Researchers
- in German and English
- 50 questions in total
- 1 additional RDMO condition

Publicly Availability

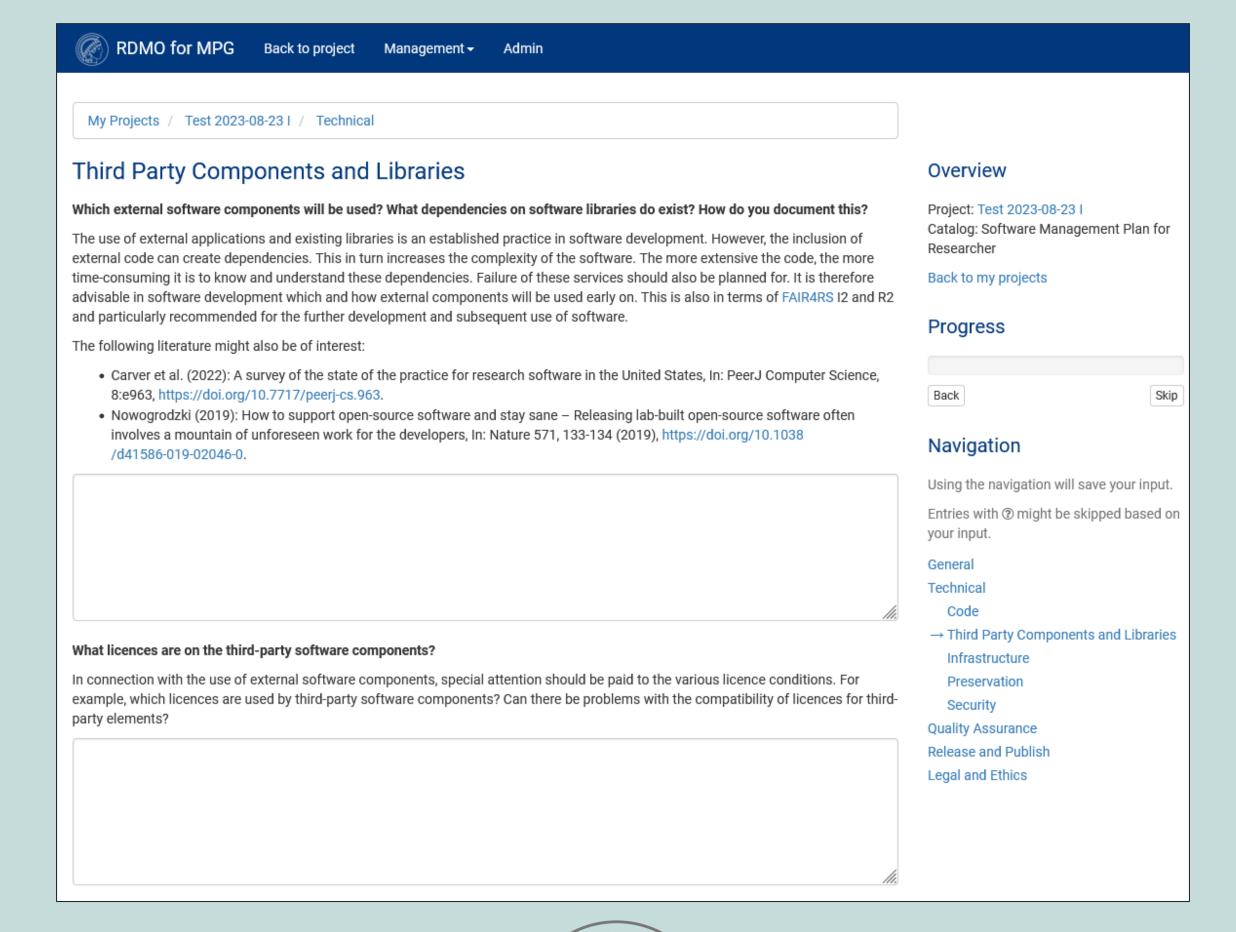
- 44 new RDMO attributes
- 2250 lines xml
- under CC0: https://github.com/rdmorganiser/
 rdmo catalogue available
- all questions are additionally available as .docx at https://doi.org/10.17617/2.3481986

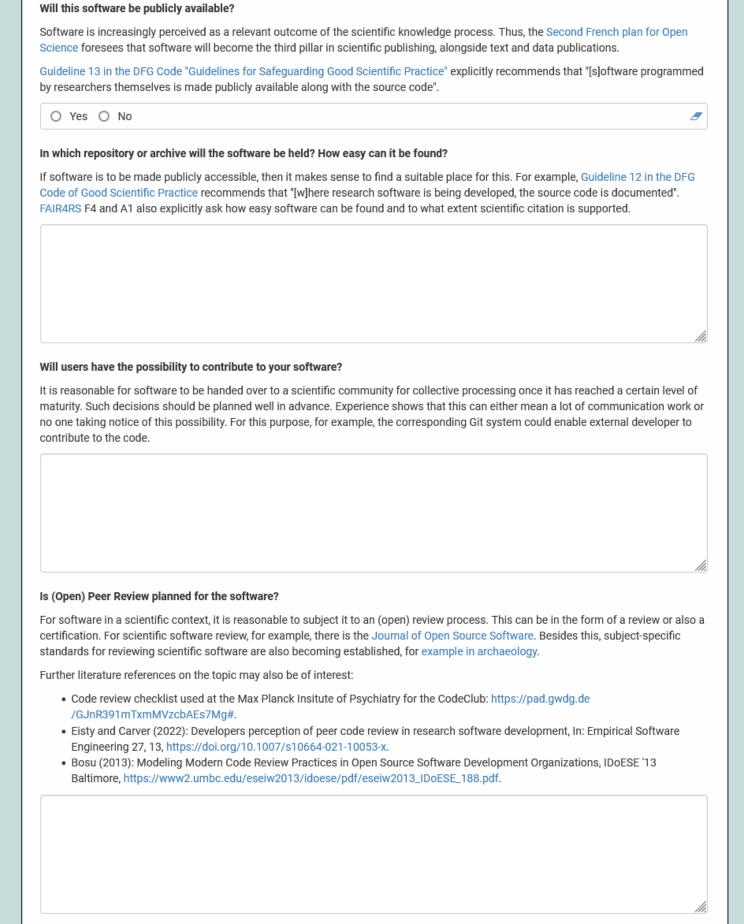
1. General

9. STRUCTURE

- a.o. persons involved, resources
- 2. Technical information
- i.a. code, infrastructure, security
- 3. Quality assurance
- Testing, documentation, etc.
- 4. Release and public availability
- a.o. releases, metadata
- 5. Legal and ethical issues
- i.a. copyright, licences, dual use

10. SCREENSHOTS





11. FAIR4RS-VIEWER

- Presentation of own answers according to the FAIR4RS principles
- Quick way to FAIRify your research software
- Free available on GitHub as well

Findability	Options
F: Software and its associated metadata are easy for both humans and machines to find	Back to project overvi
 F1. Software is assigned a globally unique and persistent identifier F1.1. Components of the software representing levels of granularity are assigned distinct identifiers ① The single software components can be identified implicitly via the version number and the source file name. 	Export
 F1.2. Different versions of the software are assigned distinct identifiers ① Versioning will be handled as follows: 	PDF Rich Text Format
F2. Software is described with rich metadata ① •	Open Office Microsoft Office HTML
F3. Metadata clearly and explicitly include the identifier of the software they describe	Markdown mediawiki LaTeX
 F4. Metadata are FAIR, searchable and indexable ⊕ Searchability of the software metadata is granted by the platform where the software is stored. 	
Accessibility	
A: Software, and its metadata, is retrievable via standardized protocols	
• A1. Software is retrievable by its identifier using a standardized communications protocol ①	
 A1.1. The protocol is open, free, and universally implementable ① The communication protocol is determined by the platform where the software is stored. 	
 A1.2. The protocol allows for an authentication and authorization procedure, where necessary ① The communication protocol is determined by the platform where the software is stored. 	
 A2. Metadata are accessible, even when the software is no longer available ① Long-term accessibility of the software metadata is granted by the platform where the software is stored. 	
Interoperability	
l: Software interoperates with other software by exchanging data and/or metadata, and/or through interaction via application programming interfaces (APIs), described through standards	
 I1. Software reads, writes and exchanges data in a way that meets domain-relevant community standards ① The following coding standards and guidelines are followed: 	
I2. Software includes qualified references to other objects ①	



The software relies on the following external services: