

HeFDI Data Talk



Datum	Thema	Referent*in(nen)
24. Februar 2023	Nachhaltige Forschungssoftware mit Softwaremanagementplänen	Dr. Yves Vincent Grossmann (Max Planck Digital Library)

Abstract:

Software wird zunehmend als wissenschaftliche Publikation und eigenständige Leistung geschätzt. Gleichzeitig ist der Umgang mit und die Wertschätzung von Forschungssoftware (noch) nicht derart entwickelt wie bei Textpublikationen. Die Max Planck Digital Library unterstützt hierbei mit einem eigenen Fragenkatalog für den Research Data Management Organizer (RDMO), dem „Software Management Plan for Researchers“. Zielgruppe sind Wissenschaftler*innen ohne tiefere Kenntnisse im Software Engineering sowie autodidaktische Wissenschaftler*innen, die selbst Software entwickeln (müssen). Durch vorgegebene Fragen mit Hilfstexten werden diese darin unterstützt, das Management ihrer entstehenden Forschungssoftware umfänglich, aber auch zeitschonend, durchzuführen.

Zu den HeFDI Data Talks:

Die HeFDI Data Talks sind eine zweiwöchentliche offene Informations- und Diskussionsveranstaltung rund um das Thema Datenmanagement im Wissenschaftskontext, in deren Rahmen sich einschlägige NFDI-Konsortien sowie Forschungsdatenmanagements-Dienste vorstellen. Die Reihe diskutiert aktuelle Themen und stellt zahlreiche – auch lokale und regionale – Tools und Services vor. Angeboten werden die HeFDI Data Talks von der Landesinitiative HeFDI - Hessische Forschungsdateninfrastrukturen, welche vom Hessischen Ministerium für Wissenschaft und Kunst (HMWK) finanziert wird.

DOI-Link: <https://doi.org/10.5281/zenodo.7736092>; Lizenzinformation: Creative Commons Attribution 4.0 International ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/))



Nachhaltige Forschungssoftware mit Softwaremanagementplänen

HeFDI Data Talk

24. Februar 2023

Dr. Yves Vincent Grossmann
grossmann@mpdl.mpg.de
Max Planck Digital Library



MAX PLANCK

digital library

Agenda

1. Kurze Vorstellung
2. Forschungssoftware
3. Softwaremanagementpläne
4. Mehrwerte
5. Umsetzung mit RDMO
6. Fragen und Diskussion

Kurze Vorstellung

Referent

- Yves Vincent Grossmann
- <https://orcid.org/0000-0002-2880-8947>
- Referent für Forschungsdatenmanagement seit Oktober 2020 an der Max Planck Digital Library
- u.a. Service Lead für [Edmond](#), [Labfolder](#) und [RDMO](#)
- Promotion über die Sozialgeschichte des (west-)deutschen Industriedesign von 1959-1990
- Kontakt: grossmann@mpdl.mpg.de

Max-Planck-Gesellschaft

- 84 Forschungsinstitute
 - auch Italien, Luxemburg, Niederlande und USA (Florida)
- Max Planck Digital Library und lokale Bibliotheken an den Instituten
- Drei Rechenzentren
- Generalverwaltung in München
- Rund 24.000 Mitarbeiter
- Budget von 2 Milliarden € im Jahr 2020
- 30 Nobelpreisträger

https://www.mpg.de/zahlen_fakten
<https://www.mpg.de/preise/nobelpreis>



https://www.mpg.de/institute_karte

Max Planck Digital Library

- Amalienstraße 33, 80799 München
- Informationsservices seit 2007
- Seit 2012 eigenständige Serviceeinheit
- rund 80 Mitarbeiter_innen aus Software-Entwicklung, Bibliothek, Wissenschaftsmanagement und Verwaltung
- www.mpd.l.mpg.de
- Abteilung Collections u.a. für das Thema Forschungsdaten zuständig



Das einzige Bücherregal in der MPDL

Max Planck Digital Library

- MPDL versteht sich als eine der vier zentralen IT-Service-Einrichtungen der Max-Planck-Gesellschaft (zusammen mit MPCDF, IKT und GWDG)
- MPDL bildet gemeinsam mit den Institutsbibliotheken das Bibliothekswesen der Max-Planck-Gesellschaft (Subsidiaritätsprinzip!)



Blick auf das Gebäude mit der MPDL, Ecke Amalienstraße und Theresienstraße in München

Forschungssoftware

Drei Säulen wissenschaftlicher Ergebnisse

Text

- Zeitschriftenartikel
- Buch
- Poster-Präsentation
- ...

Daten

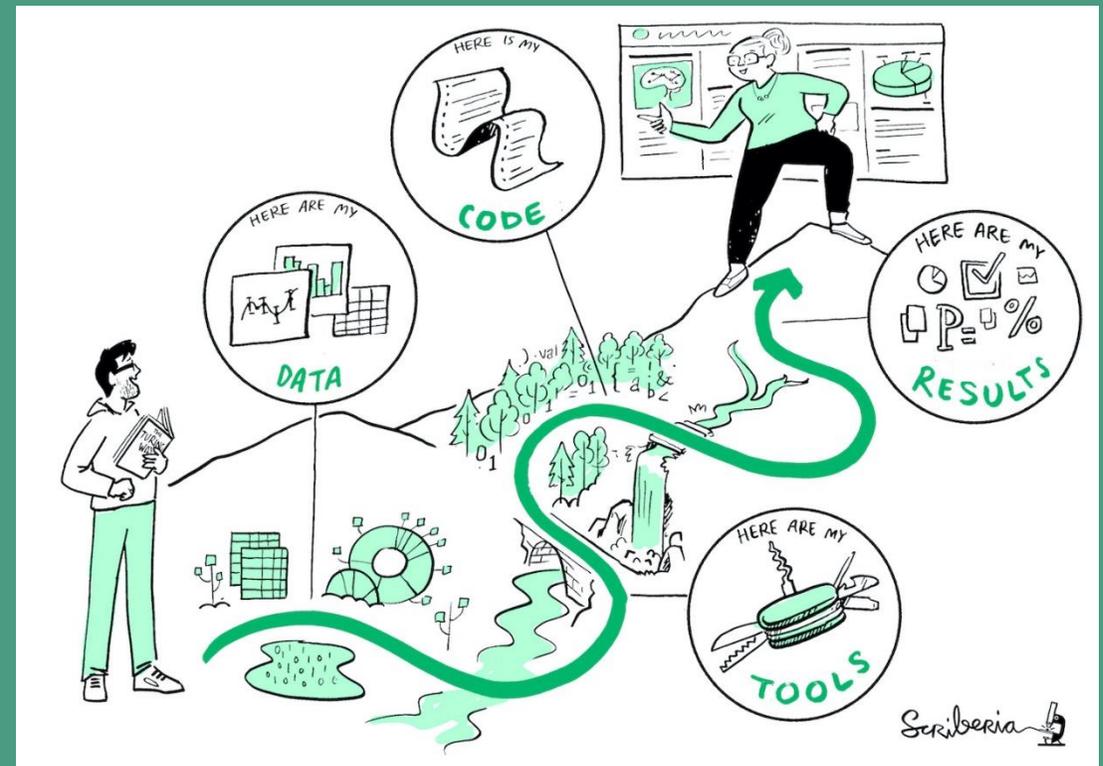
- Forschungsdaten
- Labor-Daten
- ...
- ...

Code

- Software
- Software-Dokumentation
- ...

Reproduzierbarkeit und Zugänglichkeit

- Software ist häufig notwendig für die Reproduzierbarkeit von Ergebnissen
- im Sinne einer GWP müsste sie verfügbar sein
- interne Richtlinien/Policies und/oder Fördergebern verlangen/empfehlen die Veröffentlichung von Software
 - „Selbst programmierte Software wird unter Angabe des Quellcodes öffentlich zugänglich gemacht.“
Leitlinie 13: Herstellung von öffentlichem Zugang zu Forschungsergebnissen im Kodex "Leitlinien zur Sicherung guter wissenschaftlicher Praxis" der DFG



The Turing Way Community, & Scriberia. (2022). Illustrations from The Turing Way: Shared under CC-BY 4.0 for reuse. Zenodo. <https://doi.org/10.5281/zenodo.6821117>.

Second French Plan for Open Science

Path Three “Opening up and promoting source code produced by research”

French Ministry of Higher Education, Second French Plan for Open Science. Generalizing open science in France 2021-2024, July 2021, https://www.ouvri.lascience.fr/wp-content/uploads/2021/10/Second_French_Plan-for-Open-Science_web.pdf , S. 1 and S. 16-19, CC BY 4.0.



Nutzung von Software Repositories

- Anderes Publikationsverhalten bei Software gegenüber Daten
- Beobachtung: Wenn eigene Software in einer Veröffentlichung erwähnt wird, ist diese häufig über GitHub, GitLab usw. zugänglich
- „In 2021, one out of five publications in the arXiv corpus included a URI to GitHub“ (S. 1)



Emily Escamilla, Martin Klein, Talya Cooper, Vicky Rampin, Michele C. Weigle, Michael L. Nelson: The Rise of GitHub in Scholarly Publications, 9. August 2022, <https://doi.org/10.48550/arXiv.2208.04895>, CC BY-NC-SA 4.0.

Forschungssoftware und -daten

Gemeinsamkeiten von Software und Forschungsdaten im Forschungsprozess:

- beides spielt eine immer bedeutendere Rolle in der Forschung
- beides ist notwendig, um Forschungsergebnisse reproduzierbar zu machen
- beides stellt potentiell einen Wert für Nachnutzer dar
- beides wird bisher nicht (oder nur in geringem Maße) als eigenständige Forschungsleistung anerkannt

Forschungssoftware und -daten

Gemeinsamkeiten von Software und Forschungsdaten im Management:

- beides kann mit Metadaten versehen und archiviert werden
- beides muss kuratiert werden, um die Nutzbarkeit über längere Zeiträume zu gewährleisten
- bei beidem fehlt oft das notwendige Wissen für ein adäquates Management

Forschungssoftware und -daten

Unterschiede von Software zu Forschungsdaten:

- die Kuratierung von Software ist wesentlich aufwändiger
- Metadaten für Software sind wesentlich homogener als die für Forschungsdaten
- bei Software gibt es viel Vorerfahrung mit offenen/kommerziellen Lizenzen
- eine genaue Versionierung spielt bei Software eine wesentlich größere Rolle

FAIR Principles for Research Software (FAIR4RS Principles)

- Adaption der FAIR Data Principles für Forschungssoftware
- Sommer 2022
- von der Community unterstützte Grundsätze
- Wie die Prinzipien mit „Leben“ füllen?

FAIR Principles for Research Software
(FAIR4RS Principles)

RDA Recommendation 

DOI: 10.15497/RDA00068

Authors: Neil P. Chue Hong*, Daniel S. Katz*, Michelle Barker*, Anna-Lena Lamprecht, Carlos Martinez, Fotis E. Psomopoulos, Jen Harrow, Leyla Jael Castro, Morane Gruenpeter, Paula Andrea Martinez, Tom Honeyman, Alexander Struck, Allen Lee, Axel Loewe, Ben van Werkhoven, Catherine Jones, Daniel Garijo, Esther Flomp, Françoise Genova, Hugh Shanahan, Joanna Leng, Maggie Hellström, Malin Sandström, Manodeep Sinha, Mateusz Kuzak, Patricia Herterich, Qian Zhang, Sharif Islam, Susanna-Assunta Sansone, Tom Pollard, Udayanto Dwi Atmojo, Alan Williams, Andreas Czerniak, Anna Niehues, Anne Claire Fouilloux, Bala Desinghu, Carole Goble, Céline Richard, Charles Gray, Chris Erdmann, Daniel Nüst, Daniele Tartarini, Elena Rangelova, Harwig Anzt, Ilian Todorov, James McNally, Javier Moidon, Jessica Burnett, Julián Garrido-Sánchez, Khalid Belhajjame, Laurents Sesink, Lorraine Hwang, Marcos Roberto Tovani-Palone, Mark D. Wilkinson, Mathieu Servillat, Matthias Liffers, Merc Fox, Nadica Miljković, Nick Lynch, Paula Martinez Lavanchy, Sandra Gesing, Sarah Stevens, Sergio Martinez Cuesta, Silvio Feromi, Stian Soiland-Reyes, Tom Bakker, Tovo Rabemanantsoa, Vanessa Sochat, Yo Yehudi, [FAIR4RS WG](#)

(* lead authors with equal contributions)

Published: 24th May 2022

Version: 1.0

Abstract: To improve the sharing and reuse of research software, the FAIR for Research Software (FAIR4RS) Working Group has applied the FAIR Guiding Principles for scientific data management and stewardship to research software, bringing together existing and new community efforts. Many of the FAIR Guiding Principles can be directly applied to research software by treating software and data as similar digital research objects. However, specific characteristics of software — such as its executability, composite nature, and continuous evolution and versioning — make it necessary to revise and extend the principles.

This document presents the first version of the FAIR Principles for Research Software (FAIR4RS Principles), and includes explanatory text to aid adoption. It is an outcome of the FAIR for Research Software Working Group (FAIR4RS WG) based on community consultations that started in 2019.

The FAIR for Research Software Working Group was jointly convened as a Research Data Alliance (RDA) Working Group, FORCE11 Working Group, and Research Software Alliance (ReSA) Task Force. Going forward, the RDA Software Source Code Interest Group is the maintenance home for the principles. Concerns or queries about the principles can be raised at RDA plenary events organized by the SSC IG, where there may be opportunities for adopters to report back on progress. The full maintenance and retirement plan for the principles can be found on the RDA website.

Chue Hong et al.
(2022): FAIR
Principles for
Research Software
(FAIR4RS Principles,
Zenodo, CC BY 4.0,
<https://doi.org/10.15497/RDA00068>).

Unterschiede FAIR Data ↔ FAIR4RS

Findable: Software, and its associated metadata, is easy for both humans and machines to find

- (=) F1 Software is assigned a globally unique and persistent identifier
 - (new) F1.1 Components of the software representing levels of granularity are assigned distinct identifiers
 - (new) F1.2 Different versions of the software are assigned distinct identifiers
- (=) F2 Software is described with rich metadata
- (=) F3 Metadata clearly and explicitly include the identifier of the software they describe
- (=) F4 Metadata are FAIR, searchable and indexable

Accessible: Software, and its metadata, is retrievable via standardised protocols

- (=) A1 Software is retrievable by its identifier using a standardised communications protocol
 - (=) A1.1 The protocol is open, free, and universally implementable
 - (=) A1.2 the protocol allows for an authentication and authorization procedure, where necessary
- (=) A2 metadata are accessible, even when the software is no longer available

Interoperable: Software interoperates with other software by exchanging data and/or metadata, and/or through interaction via application programming interfaces (APIs), described through standards

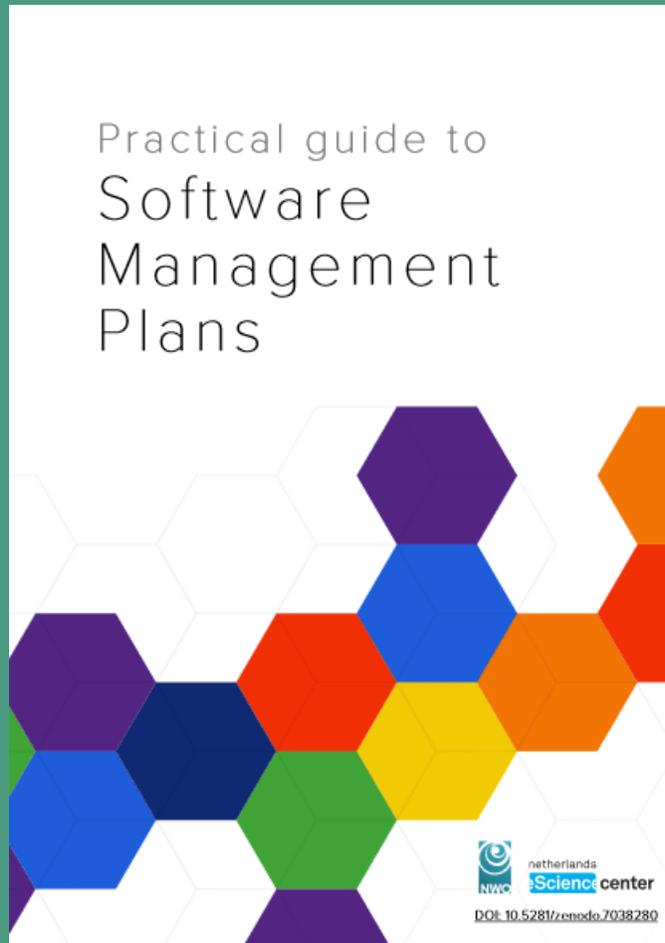
- (≠) I1. Software reads, writes and exchanges data in a way that meets domain-relevant community standards
- ~~I2. (meta)data use vocabularies that follow FAIR principles~~
- (=) I2.3. Software includes qualified references to other objects

Reusable: Software is both usable (can be executed) and reusable (can be understood, modified, built upon, or incorporated into other software)

- (=) R1. Software is described with a plurality of accurate and relevant attributes
 - (=) R1.1. Software is given a clear and accessible license
 - (=) R1.2. Software is associated with detailed provenance
- (new) R2 Software includes qualified references to other software
- (=) R1.3. Software meets domain-relevant community standards

Softwaremanagementpläne

SMP aktuell in Diskussion



Practical guide to
Software
Management
Plans

Martinez-Ortiz et al. (27.10.2022):
Practical guide to
Software
Management
Plans, v1.0,
<https://doi.org/10.5281/zenodo.7248877>,
CC BY 4.0.

DFG: **Call for Proposals to Increase the Usability of Existing Research Software** durch „Research Software – Quality Assured and Re-usable“, (Information für die Wissenschaft Nr. 85 | 8. November 2022)
http://www.dfg.de/en/research_funding/announcements_proposals/2022/info_wissenschaft_22_85.

Definition eines Software-Management-Plans

„Laut Definition der [DINI/nestor AG Forschungsdaten](#) beinhaltet ein Softwaremanagementplan (SMP) allgemeine und technische Informationen zum Softwareprojekt, Angaben zur Qualitätssicherung, zum Release und zur öffentlichen Verfügbarkeit sowie rechtliche und ethische Aspekte, die die Software betreffen.

Der SMP fasst Informationen zusammen, die die Erstellung, Dokumentation, Speicherung, Versionierung, Lizenzierung, Archivierung und/oder Veröffentlichung der in einem Projekt erzeugten oder verwendeten Software hinreichend beschreiben und dokumentieren. Dazugehörige Hardware und notwendige andere Ressourcen, aber auch damit verbundene weitere Software und Softwarebibliotheken, Text- und Datenpublikationen sind ebenfalls zu beschreiben und stellen eine Besonderheit des SMP dar.

Zweck eines SMPs ist zunächst die Nachvollziehbarkeit sowie ggf. die langfristige Nutzbarkeit der Software (zur direkten Anwendung sowie zur Weiterverarbeitung) zu unterstützen und den Support der Nutzer*innen bei Rückfragen zu erleichtern. Der SMP dient folglich auch der Qualitätssicherung (vgl. hierzu [FAIR4RS Principles](#)).“

Perspektiven der Drittmittelgeber

Deutsche Forschungsgemeinschaft

- SMP ist nicht notwendig
- Kodex Leitlinie 13 „Herstellung von öffentlichem Zugang zu Forschungsergebnissen“: *„die eingesetzte Software verfügbar zu machen und Arbeitsabläufe umfänglich darzulegen. Selbst programmierte Software wird unter Angabe des Quellcodes öffentlich zugänglich gemacht.“*

ERC

- SMP ist nicht notwendig
- Aber Software wird meist zusammen mit Daten im „Information for ERC grantees“ (v4.1) genannt

Perspektiven der Drittmittelgeber

Horizon Europe

- SMP ist nicht notwendig
- Software wird im General Model Grant Agreement (v1.1) kaum explizit erwähnt
- aber: *„the beneficiaries must provide (digital or physical) access to data or other results needed for validation of the conclusions of scientific publications, to the extent that their legitimate interests or constraints are safeguarded“* S. 112

Wissenschaftler_innen schreiben wissenschaftliche Software

Beobachtungen:

- häufig keine ausgebildeten Software-Entwickler_innen, eher Autodidakten
- Funktionalität geht vor Dokumentation geht vor langfristiger Erhaltung
- erst Textpublikation, dann lange nichts, und dann vielleicht Datenpublikation und Software-Publikation
- Software wird von Generation zu Generation „vererbt“
- ...

Unterschiede zwischen SMP und DMP

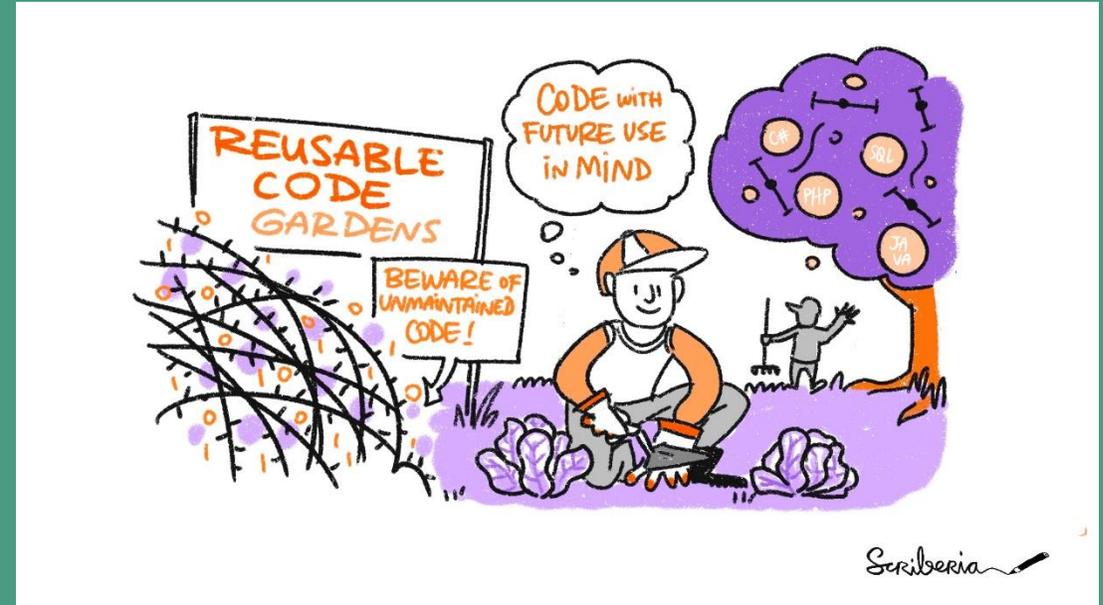
- die Einbindung von externen Komponenten findet häufiger statt
- die Kuratierung von Software ist gegenüber Forschungsdaten wesentlich aufwändiger
- Testing und Dokumentation folgen anderen Standards und Zielen
- eine genaue Versionierung spielt bei Software eine wesentlich größere Rolle
- Metadaten für Software sind meist homogener als die für Forschungsdaten
- bei Software gibt es viel Vorerfahrung mit offenen und kommerziellen Lizenzen

Mehrwerte

Nachnutzbarkeit von Forschungssoftware

bewusster Umgang steigert die Chance der Nachnutzbarkeit, durch:

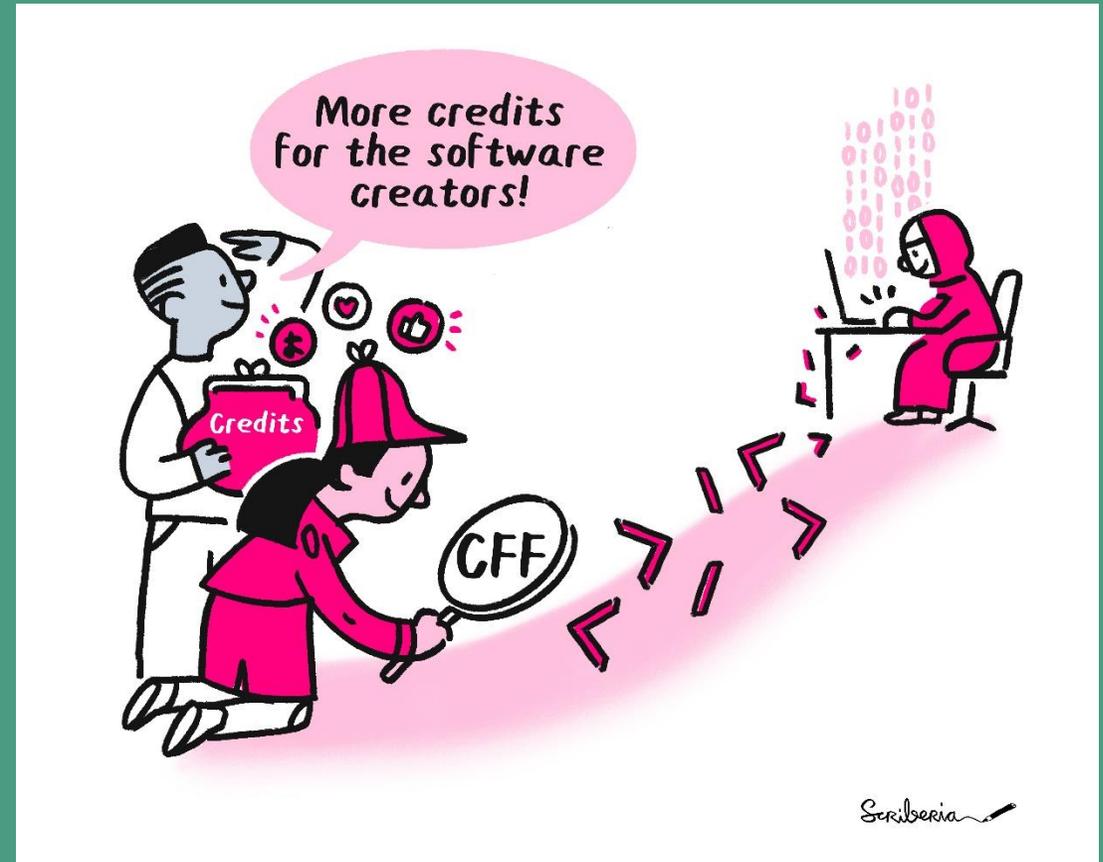
- die gestiegene Wahrscheinlichkeit einer Software-Publikation
- eine explizite Lizenzvergabe
- klarere Codestruktur und eine reflektiertere Nutzung von Drittbibliotheken
- eine gezielte Auseinandersetzung mit Archivierung
- ...



The Turing Way Community, & Scriberia. (2022). Illustrations from The Turing Way: Shared under CC-BY 4.0 for reuse. Zenodo. <https://doi.org/10.5281/zenodo.6821117>.

Anerkennung von Forschungssoftware

- fachinterne Anerkennung für Forschungssoftware
- Anerkennung durch Fördergeber
- institutionsinterne Credits für Software-Publikation
- Policies
 - rechtliche Rahmen zur einfachen Software-Publikation
 - klares Statement, dass Software-Publikationen erwünscht sind



The Turing Way Community, & Scriberia. (2022). Illustrations from The Turing Way: Shared under CC-BY 4.0 for reuse. Zenodo. <https://doi.org/10.5281/zenodo.6821117>.

Beispiel für Software Policies

Software Policy

MAX-PLANCK-INSTITUT
FÜR METEOROLOGIE

CODE @ MPI-M

Software Licensing and Copyright Policy for Research Software CODE @ MPI-M

1. Preamble

The Max Planck society is determined to promote Open Access to research data, and Open Science (Berliner Erklärung¹).

MPI for Meteorology in Hamburg (MPI-M) has developed, partly on its own, partly in collaboration with partners, various Research Software, for example the ICON Model Code², the Climate Data Operators cdo³, and others.

MPI-M believes that for the benefit of science such Research Software should be released as Open-Source Software.

2. Rules

This policy treats the issues of copyright and licensing. It is applicable and restricted to Research Software Source Code developed at MPI-M (CODE). CODE must either be copyright of MPI-M alone (i.e. new code for, or code developed earlier at MPI-M under the copyright of the institute) or code contributed by third parties to MPI-M code which has been licensed to MPI-M under a permissive license like MIT/X11, (2- or 3-Clause) BSD or Apache 2.0 under the Contributor License Agreement (CLA⁴) or where unlimited copyrights have been transferred to MPI-M by other means under similar conditions.

a. Contributor License Agreement (CLA)

Every developer of CODE at MPI-M, being an employee or in any other connection to MPI-M (freelancer, guest, post-doc, scholarship etc.) (CONTRIBUTOR) must sign a CLA to be allowed to contribute to an MPI-M project in the field of Research Software development. To clarify: MPI-M can agree upon co-operations with other institutions where this policy is not applicable.

b. License

CODE shall be licensed under the BSD-3-Clause License⁵, also see attachment. For any other open source license, you must consult with the MPI-M person responsible for licenses⁶.

¹ <https://openaccess.mpg.de/Berliner-Erklärung>
² <https://mpimet.mpg.de/en/science/models>
³ <https://code.mpimet.mpg.de/projects/cdo/>
⁴ Link t.b.d. on MPI-Met Internet site.
⁵ See <https://opensource.org/licenses/BSD-3-Clause>, SPDX short identifier: BSD-3-Clause
⁶ Currently: The person responsible for strategic IT partnerships, or the head of the MPI-M administration. mailto:licenses@mpimet.mpg.de

Authors: Reinhard Budich, MPI-M; Maximilian Funk, MFS 1 / 7 Version: V3.8.2, Date: 2022-07-04; License: CC BY 4.0

Reinhard Budich und
Maximilian Funk:
Software Licensing and
Copyright Policy for
Research Software
CODE @ MPI-M, v3.8.2
vom 4. Juli 2022,
<https://mpimet.mpg.de/fileadmin/staff/budichreinhard/MPI-M-SW-Policy.pdf>,
CC BY 4.0.

TU Delft Guidelines on Research Software Licensing, Registration and Commercialisation



TU Delft

Bazuine et al. (2021): TU Delft
Guidelines on Research
Software: Licensing,
Registration and
Commercialisation, CC BY 4.0,
<https://doi.org/10.5281/zenodo.4629635>.

Zielgruppe für SMPs

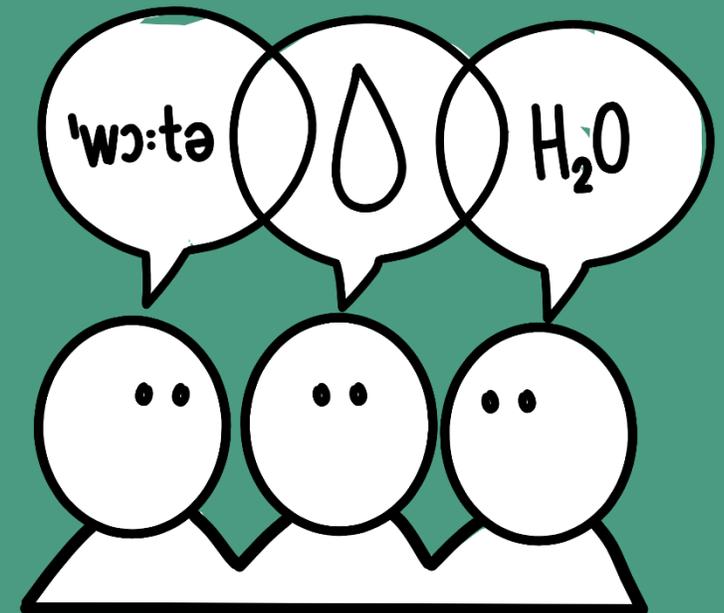
- Wissenschaftler_innen
- Kolleg_innen aus IT, Scientific Core Unit, ... zur Beratung
- Kolleg_innen aus dem Bereich Drittmittelanträge
- Kolleg_innen aus dem Bereich Projektmanagement und Forschungscoordination
- ...



<https://doi.org/10.5281/zenodo.2581783>

Wissenschaftler_innen schreiben SMPs

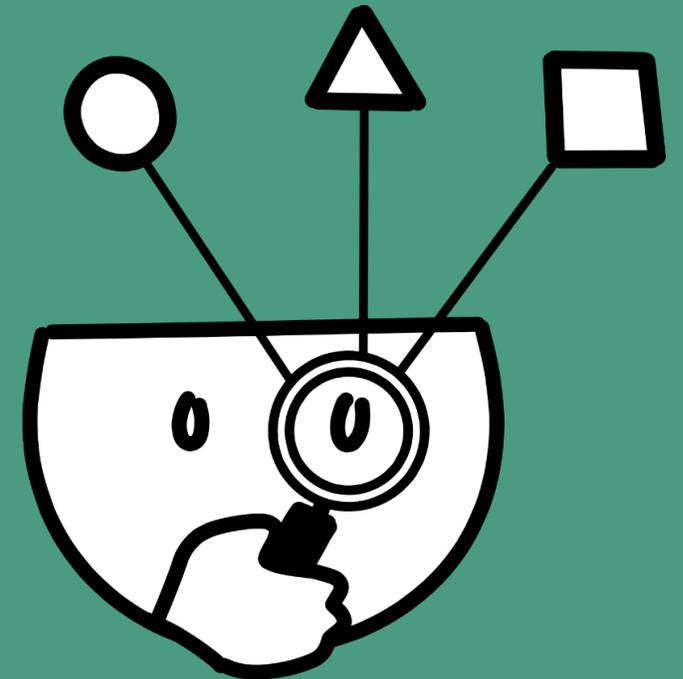
- für sich selbst!
- zusammen mit IT/Scientific Computing Unit/... um das Software-Projekt besser zu konzipieren
- für einen Drittmittelantrag
- für die interne Planung
- im Vorfeld einer Beratung (z.B. durch IT, Scientific Computing Unit,...)
- für Nachhaltigkeit und eine mögliche Archivierung (siehe GWP)
- zur Qualitätssicherung
- ...



<https://doi.org/10.5281/zenodo.3674561>

Vorteile beim institutionellen Nutzen

- Unterstützung für Software-Entwickelnde
- (mehr) Informationen über startende/laufende Software-Projekte
- bessere Bedarfsplanung
- bessere Qualitätssicherung
- bessere Archivierungsmöglichkeiten
- bessere Nachnutzungsmöglichkeiten
- ...



<https://doi.org/10.5281/zenodo.3674561>

Umsetzung mit RDMO

Rahmendaten

- Team der Max Planck Digital Library, Abteilung Collections
 - vor allem Michael Franke, Yves Vincent Grossmann und Jan Matthiesen
- Zeitraum Juli 2022 bis Dezember 2022
- RDMO als technische Basis
- Ergebnis: CC0-Push eines SMP-Katalogs als Beitrag zur RDMO Community

SMP als RDMO-Katalog

- Titel: „Software-Management-Plan für Forschende“
- in Deutsch & Englisch
- insgesamt 49 Fragen
- 1 zusätzliche Bedingung
- 44 neue Attribute
- 2250 Zeilen xml
- unter CC0 auf <https://github.com/rdmorganiser/rdmo-catalog> verfügbar
- FAIR4RS-Viewer in Vorbereitung

Struktur

1. Allgemein
 - u.a. beteiligte Personen, Projektmanagement
2. Technische Informationen
 - u.a. Code, Infrastruktur, Sicherheit
3. Qualitätssicherung
 - u.a. Testing, Dokumentation
4. Release und öffentliche Verfügbarkeit
 - u.a. Releases, Metadaten
5. Rechtliche und ethische Fragen
 - u.a. Urheberrecht, Lizenzen, Dual Use

Screenshots

The screenshot displays the RDMO for MPG web interface. At the top, a dark blue navigation bar contains the logo, 'RDMO for MPG', and links for 'Zurück zum Projekt', 'Management', and 'Admin'. On the right, it shows 'Sprache' and the user name 'Yves Vincent Grossmann'. Below this, a sidebar on the left contains a search bar, a list of items, and a 'Zurück zum Projekt' link. The main content area is titled 'Technische Informationen' and 'Metadaten'. It includes a section 'Wie vergeben Sie Metadaten für Ihre Software?' with explanatory text and a large text input field. Below that is 'Geben Sie einen dauerhaften Identifikator für Ihre Software an?' with another text input field. At the bottom of the main area are buttons for 'Zurück', 'Überspringen', 'Sichern', and 'Sichern und fortfahren'. On the right side, there is an 'Übersicht' section with project details and a progress bar, and a 'Navigation' section with a list of links.

Zu welchem(n) RDMO for MPG Zurück zum Projekt Management Admin Sprache Yves Vincent Grossmann

Die Liste der Dis

Bitte nutzen Sie blauen Kreuz (x)

Welche Programm

Die Softwaresprac dokumentieren, für

Eintrag hinzuf

Was ist der bea

Die Intention für Anwendung der auch deutlich v

Zur Orientierung können:

- Anzt et al open cha
- Gardner et al bioinform
- Katerbow /10.5281
- Lee, et al /j.patter.2
- Wuttke et Research
- Wilson et https://dc

Es liegt in der N Unterstützung b Diese können Ih

Technische Informationen

Metadaten

Wie vergeben Sie Metadaten für Ihre Software?

Für die Veröffentlichung von Software ist es empfohlen die Software mit Metadaten zu beschreiben (siehe FAIR4RS F1 bis F2 und A2). Die Leitlinie 7 im DFG-Kodex "Leitlinien zur Sicherung guter wissenschaftlicher Praxis" schreibt ebenfalls vor, dass "[d]er Quellcode von öffentlich zugänglicher Software [...] persistent zitierbar und dokumentiert sein" muss. Falls keine Publikation geplant ist, kann es dennoch sinnvoll sein, die Software mit Metadaten anzureichern. Besonders für die langfristige Aufbewahrung am Institut kann dies hilfreich sein, um ein späteres Umgehen mit der Software zu erleichtern.

Beispielsweise mit einer .cff-Datei können Sie vergleichsweise einfach Metadaten in menschen- und maschinenlesbarer Art zu Ihrer Software zur Verfügung stellen. Mit der Software Description Ontology können Sie auch eine Ontologie zur formalen Beschreibung der Software verwenden. Diese Handreichung zur Relevanz von Software-Zitationen sammelt nochmals die wesentlichen Punkte zu diesem Themenkomplex.

Geben Sie einen dauerhaften Identifikator für Ihre Software an?

Für die langfristige Verfügbarkeit und die klare Zuordnung ist es empfehlenswert, die Software(-version) mit einem persistenten Identifikator zu veröffentlichen. Diese Empfehlung geht einher mit FAIR4RS F3 und der Leitlinie 7 im DFG-Kodex "Leitlinien zur Sicherung guter wissenschaftlicher Praxis". Ein Beispiel für einen persistenten Identifikator ist etwa der Digital Object Identifier (DOI).

Übersicht

Projekt: Test 2022-12-09 I
Katalog: Software-Management-Plan für Forschende

Zurück zu meinen Projekten

Fortschritt

Zurück Überspringen

Navigation

Bitte beachten Sie, dass bei der Verwendung der Navigation alle nicht gespeicherten Eingaben verworfen werden.

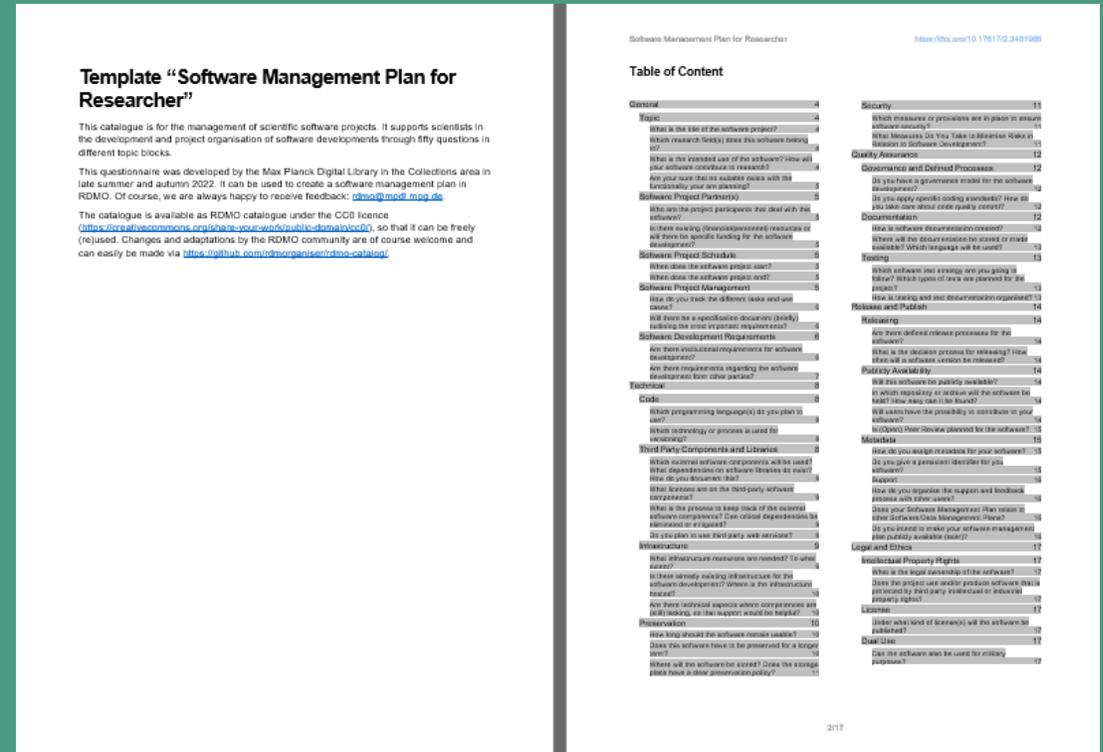
Einträge mit Ⓞ können aufgrund Ihrer Eingabe übersprungen werden.

Allgemein
Technische Informationen
Qualitätssicherung
Release und öffentliche Verfügbarkeit
Release
Öffentliche Verfügbarkeit
→ Metadaten
Support
Rechtliche und ethische Fragen

Zurück Überspringen Sichern Sichern und fortfahren

Fragen und Hilfstexte auch als .docx

- zusätzliches .docx-Dokument
- englische Fragen und Hilfstexte
- CC0 → eigene Adaption möglich
- <https://doi.org/10.17617/2.3481986>



FAIR4RS-Viewer

- Darstellung der eigenen Antworten aufgegliedert nach den FAIR4RS-Prinzipien
- schneller Weg zur FAIRheit der eigenen Forschungssoftware
- exportierbar in verschiedene Dateiformate
- Entwicklung durch Jan Matthiesen (MPDL)
- kommt im März als Pull request auf GitHub

The screenshot displays the FAIR4RS-Viewer interface. At the top, there is a navigation bar with the logo 'RDMO for MPG', links for 'Zurück zum Projekt', 'Management', and 'Admin', and a language selector 'Sprache' set to 'Yes'. The main content area is titled 'FAIR4RS' and includes the following information:

- Project title:** Test 2023-01-31 |
- Start date:**
- End date:**

On the right side, there are two menu sections: 'Optionen' with a link 'Zurück zum Projekt', and 'Export' with options for PDF, Rich Text Format, Open Office, Microsoft Office, HTML, Markdown, mediawiki, and LaTeX.

The main content is organized into sections for the FAIR4RS principles:

- Findable**
F: Software, and its associated metadata, is easy for both humans and machines to find
 - F1. Software is assigned a globally unique and persistent identifier**
 - F1.1. Components of the software representing levels of granularity are assigned distinct identifiers:*
 - F1.2. Different versions of the software are assigned distinct identifiers*
 - F2. Software is described with rich metadata**
 - F3. Metadata clearly and explicitly include the identifier of the software they describe**
 - F4. Metadata are FAIR, searchable and indexable**
- Accessible**
A: Software, and its metadata, is retrievable via standardized protocols
 - A1. Software is retrievable by its identifier using a standardized communications protocol**
 - A1.1. Software is retrievable by its identifier using a standardized communications protocol*
 - A1.2. The protocol allows for an authentication and authorization procedure, where necessary*
 - A2. Metadata are accessible, even when the software is no longer available**
- Interoperable**
I: 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**
 - I2. Software includes qualified references to other objects**
- Reusable**
R: Software is both usable (can be executed) and reusable (can be understood, modified, built upon, or incorporated into other software)
 - R1. Software is described with a plurality of accurate and relevant attributes**
 - R1.1. Software is given a clear and accessible license*
 - R1.2. Software is associated with detailed provenance*
 - R2. Software includes qualified references to other software**
 - R3. Software meets domain-relevant community standards**

Take Home Message

1. Relevanz und Anerkennung von Forschungssoftware wird (weiter) steigen
2. Explizite Umgang mit Forschungssoftware bringt Mehrwerte
3. Softwaremanagementpläne sind eine Werkzeug zum expliziten Umgang mit Forschungssoftware

Dank

an das MPDL-Collections Team, die UAG-DMP der DINI/nestor-AG
Forschungsdaten und an die RDMO-Community

**Danke für die
Aufmerksamkeit**

grossmann@mpdl.mpg.de