

Softwaremanagementpläne mit RDMO

Netzwerktreffen Forschungsdaten Berlin-Brandenburg

2. Mai 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

- 85 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



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.mpdl.mpg.de
- Abteilung Collections u.a. für das Thema Forschungsdaten zuständig



Das einzige Bücherregal in der MPDL

Forschungssoftware

Drei Säulen wissenschaftlicher Ergebnisse

Text

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

Daten

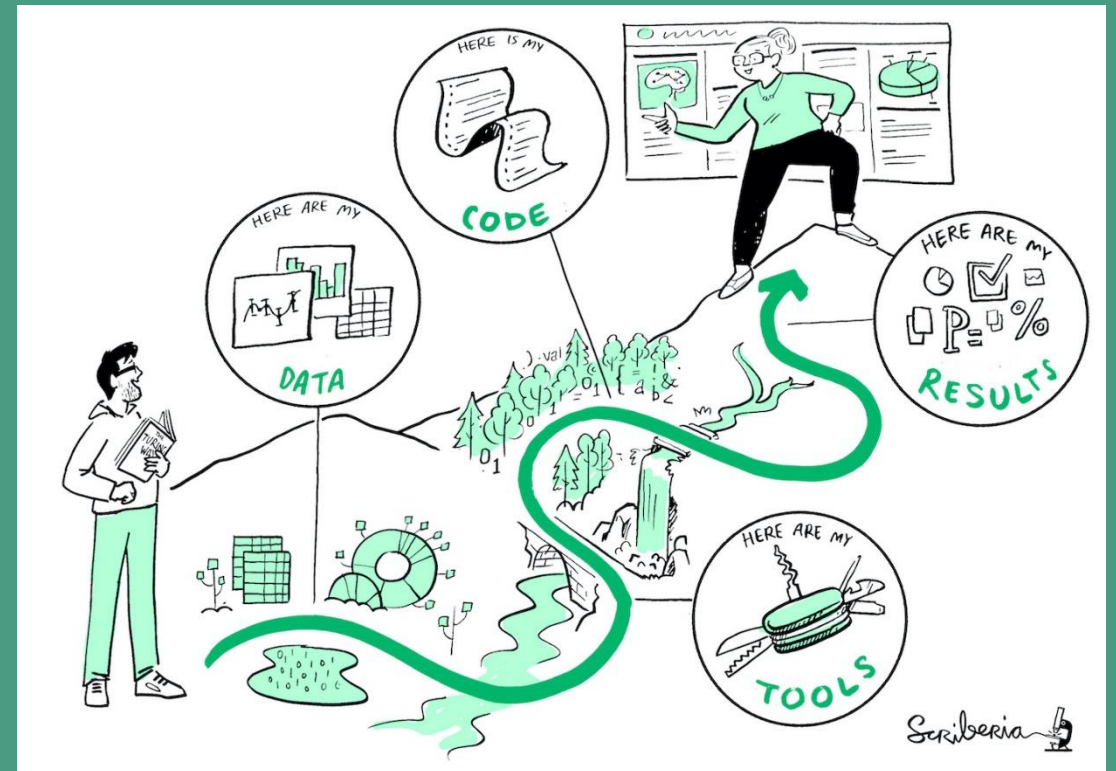
- Forschungsdaten
- Labor-Daten
- Bilder
- ...

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>.

Zielgruppe für SMPs

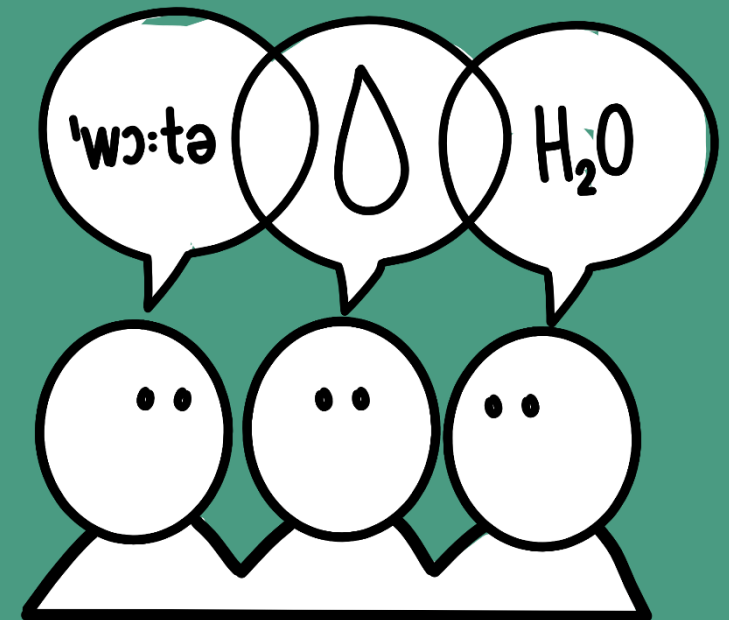
- Wissenschaftler_innen
- Kolleg_innen aus IT, Scientific Core Unit, ... zur Beratung
- Kolleg_innen aus dem Bereich Drittmittelanträge
- Kolleg_innen aus dem Bereich Projektmanagement und Forschungscoordination
- ...



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

Wissenschaftler_innen schreiben SMPs

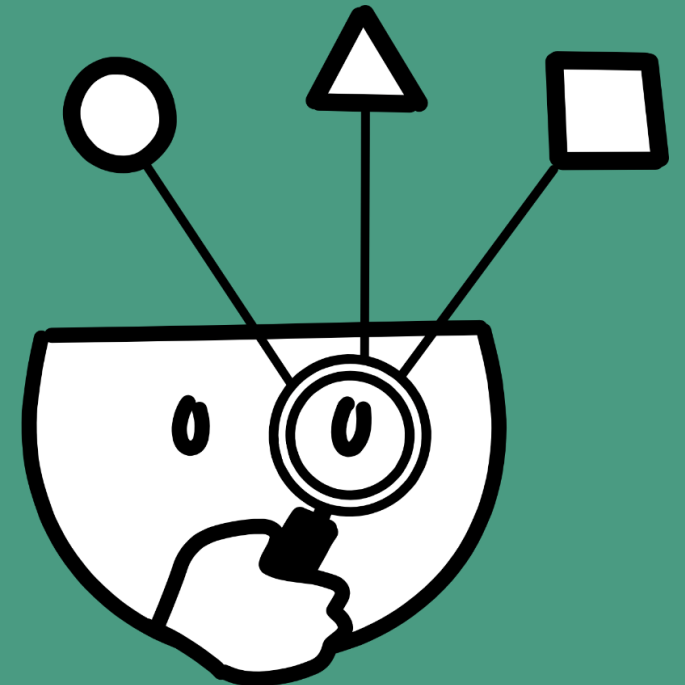
- für sich selbst!
- zusammen mit IT/Scientific Computing Unit/... um das Software-Projekt besser zu konzipieren
- für einen Drittmittelantrag
- für die interne Planung
- im Vorfeld einer Beratung (z.B. durch IT, Scientific Computing Unit,...)
- für Nachhaltigkeit und eine mögliche Archivierung (siehe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

 **RDMO for MPG** Zurück zum Projekt Management Admin Sprache Yves Vincent Grossmann

Zu welchem(n) Forschungsbereich(en) gehört diese Software?

Die Liste der Disziplinen entspricht der [Fachsystematik der Deutschen Forschungsgemeinschaft \(DFG\)](#).

Bitte nutzen Sie für die Einträge jeweils eine eigene Zeile. Sie können weitere Einträge mit dem grünen Button hinzufügen und mit dem blauen Kreuz (x) entfernen.

x

[Eintrag hinzufügen](#)

Was ist der beabsichtigte Verwendungszweck der Software? Wie wird Ihre Software zur Forschung beitragen?

Die Intention für die Entwicklung von Software im wissenschaftlichen Kontext kann vielfältig sein. Gleiches gilt für die konkrete Anwendung der Software und dem Beitrag zum wissenschaftlichen Erkenntnisgewinn. Je nach Fachkultur können sich diese Punkte auch deutlich voneinander unterscheiden.

Zur Orientierung und als Unterstützung bieten wir hier einige Literaturempfehlungen, die bei der Beantwortung der Fragen hilfreich sein können:

- Anzt et al. (2021): deRSE Position Paper: "An environment for sustainable research software in Germany and beyond: Current state, open challenges, and call for action", <https://doi.org/10.12688/f1000research.23224.2>.
- Gardner et al. (2022): Sustained software development, not number of citations or journal choice, is indicative of accurate bioinformatic software, In: Genome Biology 23, 56, <https://doi.org/10.1186/s13059-022-02625-x>.
- Katerbow, & Feulner (2018): Recommendations on the development, use and provision of Research Software, <https://doi.org/10.5281/zenodo.1172988>.
- Lee, et al. (2021): Barely sufficient practices in scientific computing, In: Patterns, 2(2), 100206, <https://doi.org/10.1016/j.patter.2021.100206>.
- Wuttke et al. (2022): Guidelines for collaborative development of sustainable data treatment software, In: Journal of Neutron Research, vol. 24, no. 1, pp. 33-72, <https://doi.org/10.3233/JNR-220002>.
- Wilson et al. (2017): Good enough practices in scientific computing, In: PLOS Computational Biology, 13(6), e1005510, <https://doi.org/10.1371/journal.pcbi.1005510>.

Es liegt in der Natur der Sache, dass diese Literaturhinweise nicht vollumfänglich sind. Wenn Sie bei der Beantwortung konkrete Unterstützung benötigen, dann sprechen Sie am besten mit Ihren lokalen Kolleg_innen von der IT, der Scientific Computing Unit usw. Diese können Ihnen weiterhelfen.

Übersicht


Projekt: Test 2022-11-10 I
Katalog: Software-Management-Plan für Forschende (Beta-Testmodus)

[Zurück zu meinen Projekten](#)

Fortschritt

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
→ Thema

- Software-Projektpartner
- Zeitplan für Softwareprojekte
- Software-Projekt-Management
- Anforderungen an die Softwareentwi...

Technische Informationen
Qualitätssicherung
Release und öffentliche Verfügbarkeit
Rechtliche und ethische Fragen

Screenshots

The screenshot displays the RDMO for MPG interface. The top navigation bar includes the RDMO logo, 'RDMO for MPG', and links for 'Zurück zum Projekt', 'Management', and 'Admin'. On the right, it shows 'Sprache' and the user 'Yves Vincent Grossmann'. The main content area is titled 'Technische Informationen / Code' and contains two sections: 'Welche Programmiersprache(n) planen Sie zu verwenden?' and 'Welche Technologie oder welches Verfahren wird für die Versionierung verwendet?'. The first section includes a text area for documentation. The second section includes a list of links to external Git systems. The right sidebar contains sections for 'Übersicht', 'Fortschritt', and 'Navigation', with a list of links for 'Allgemein', 'Technische Informationen', and 'Code'.

Technische Informationen / Code

Welche Programmiersprache(n) planen Sie zu verwenden?

Die Softwaresprachen gehen einher mit unterschiedlichen Methoden, Schwerpunkten und benötigten Fähigkeitsgraden. Hier können Sie dokumentieren, für welche Sprache(n) Sie sich entscheiden und weshalb. Beispiele für Softwaresprachen während Java, Python, R, usw.

An dieser Stelle bietet es sich zusätzlich an, zu dokumentieren, ob technische Standards für die Entwicklung von Software relevant sind. Beispiele hierfür sind etwa der [IEC Standard 62304](#) für Software bei medizinischen Anwendungen, oder [ECSS-E-ST-40C](#) bei Software für Anwendungen im Weltraum oder [DICOM](#) für den Austausch von medizinischen Bilddaten.

Welche Technologie oder welches Verfahren wird für die Versionierung verwendet?

Das Versionieren von Code bei der Entwicklung ist dringend zu empfehlen. Code-Versionierungen auf Basis der Git-Software sind weitverbreitet.

Verschiedene Max-Planck-Institute pflegen eigene Git-Systeme. Gleichzeitig gibt es auch mehrere zentrale Git-Systeme, die für Max-Planck-Wissenschaftler_innen zugänglich sind:

- Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen: <https://gitlab.gwdg.de>
- Max Planck Computing and Data Facility: <https://gitlab.mpcdf.mpg.de>
- Max-Planck-Institut für molekulare Genetik: <https://github.molgen.mpg.de> (als Max-Planck-Wissenschaftler_in können Sie über deren Helpdesk ein Konto erhalten (helpdesk@molgen.mpg.de))

Übersicht

Projekt: [Test 2022-11-10 I](#)
Katalog: [Software-Management-Plan für Forschende \(Beta-Testmodus\)](#)

[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

→ **Code**

- [Externe Komponenten und Bibliothek...](#)
- [Infrastruktur](#)
- [Langfristige Erhaltung](#)
- [Sicherheit](#)
- [Qualitätssicherung](#)
- [Release und öffentliche Verfügbarkeit](#)
- [Rechtliche und ethische Fragen](#)

Screenshots

 RDMO for MPG [Zurück zum Projekt](#) [Management](#) [Admin](#) [Sprache](#) [Yves Vincent Grossmann](#)

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).

Zurück Überspringen Sichern Sichern und fortfahren

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.

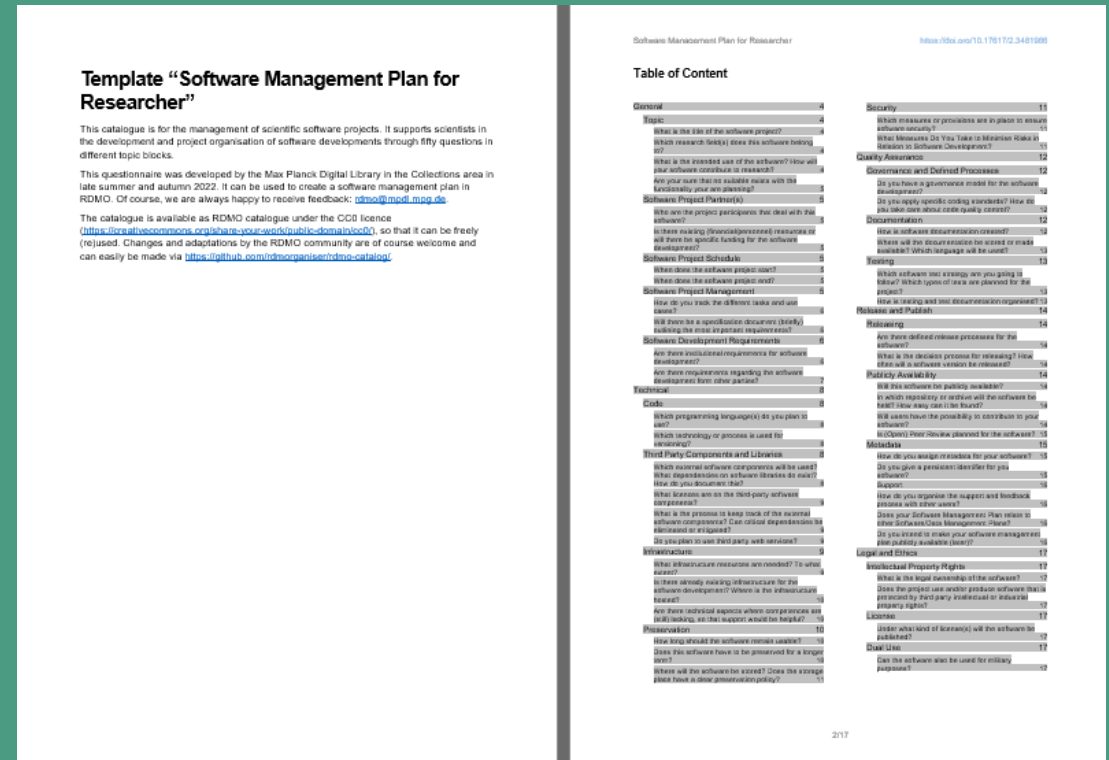
Einträge mit  können aufgrund Ihrer Eingabe übersprungen werden.

- [Allgemein](#)
- [Technische Informationen](#)
- [Qualitätssicherung](#)
- [Release und öffentliche Verfügbarkeit](#)
 - [Release](#)
 - [Öffentliche Verfügbarkeit](#)
- [→ Metadaten](#)
- [Support](#)
- [Rechtliche und ethische Fragen](#)

19

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 web 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 shows project information: 'Project title: Test 2023-01-31', 'Start date:', and 'End date:'. On the right side, there are two vertical menus: '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', 'Accessible', 'Interoperable', and 'Reusable'. Each section includes a general description and a list of specific requirements (F1-F4, A1-A2, I1-I2, R1-R3).

FAIR4RS

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

Optionen
Zurück zum Projekt

Export
PDF
Rich Text Format
Open Office
Microsoft Office
HTML
Markdown
mediawiki
LaTeX

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**
 - A.1.1. Software is retrievable by its identifier using a standardized communications protocol*
 - A.1.2. 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**

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