

# Machine-Actionable Metadata for Software and Software Management Plans for NFDI



Olga Giraldo, Danilo Dessi, Stefan Dietze,  
Dietrich Rebholz-Schuhmann, **Leyla Jael Castro**



DOI [10.5281/zenodo.8349183](https://doi.org/10.5281/zenodo.8349183)



# From LOD to FAIR to Linked Open Science



Image taken from <https://www.w3.org/DesignIssues/LinkedData>

- ▶ Linked Open Data → RDF standards, common ground, openness, linking
- ▶ FAIR → minimum metadata, enriched metadata, community agreement
- ▶ Open Science → sharing as much as possible
  
- ▶ **Linked Open Science** → (lightweight) LOD + FAIR + Open Science

# Metadata – key to Linked Open Science

As open as possible as close as necessary

**But...** we still need some minimum information on that that is closed  
(same on that that is open)

- Metadata help us describe research outcomes: publications, data, software
- Structured metadata enables bridges and also \*ilities
- Structured metadata plays a key role in **FAIR** and makes things easier for machines and humans
- Big community effort, cultural change

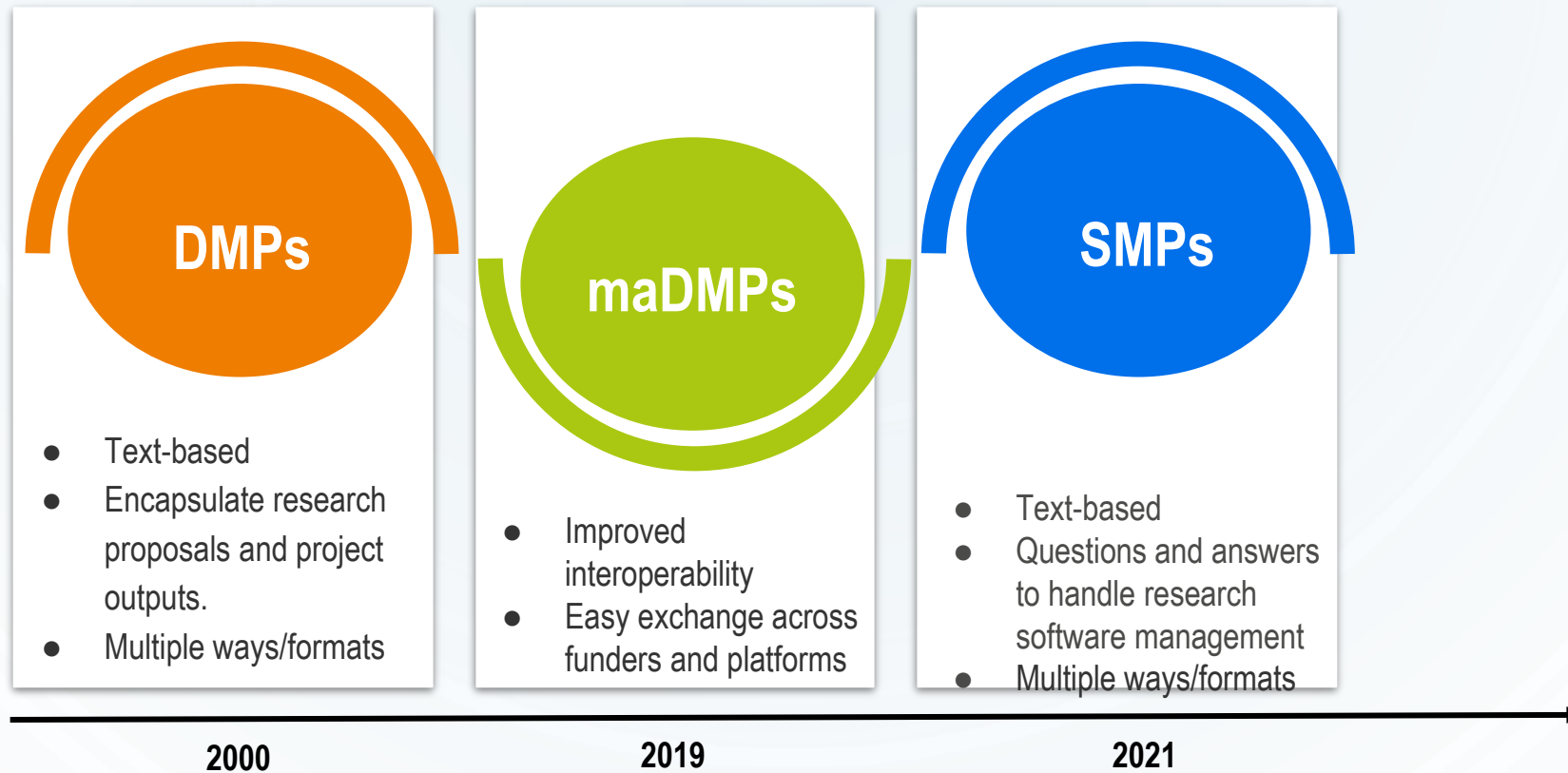


A decorative background featuring stylized circuit board traces in the corners, consisting of thin black lines and small white circles. The main content is enclosed in a large rectangular box with a red-to-orange gradient background.

Metadata to improve research  
and research management

Use Case: Software and  
Software Management Plans

# From Data to Software Management Plans



# Software Management Plans



- ▼ Software Management Plan
  - ▼ Accessibility & License
    - What is the name of the software?
    - ▶ ○ How can the software be accessed by third parties?
    - ▶ ○ Does your software have a license?
  - ▼ Documentation
    - ▶ ○ What type of documentation is available, provided with the software?
    - ▶ ○ Is the purpose of the software stated in the documentation?
    - ▶ ○ Does the documentation describe how to use the software?
  - ▼ Testing
    - ▶ ○ What type of testing do you use?
    - ▶ ○ Are sample data and/or parameters that can be used to test the software?
  - ▼ Interoperability
    - ▶ ○ Do you use well-established standard input/output formats?
    - What programming languages are you using in your project?
  - ▼ Versioning
    - ▶ ○ Do you use a version control system?
    - ▶ ○ Do you use Semantic Versioning?

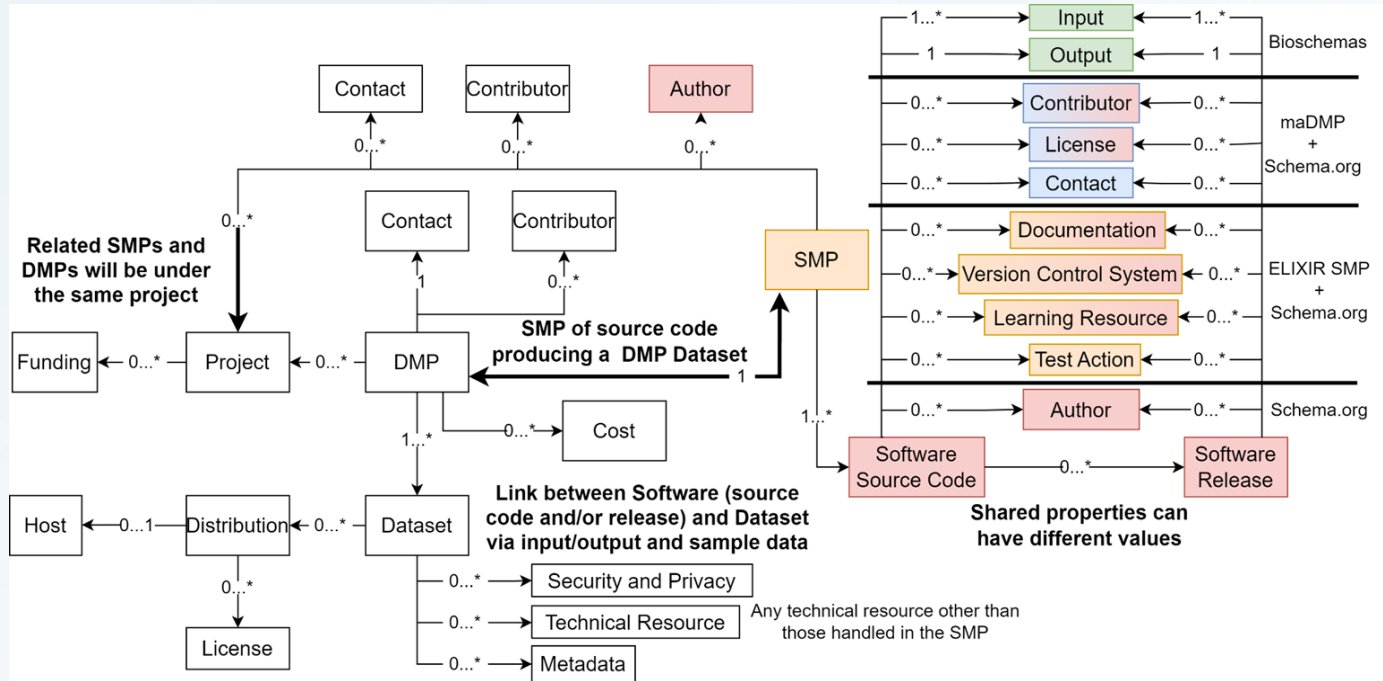
Source: <https://smw.ds-wizard.org/>

Core requirement (Section 5.1)	Example SMP question(s) (Section 6.1)
Purpose	Please provide a brief description of your software, stating its purpose and intended audience.
Version control	How will you manage versioning of your software?
Repository	How will you make your software publicly available? If you do not plan to make it publicly available you should provide a justification.
User documentation	How will your software be documented for users? Please provide a link to the documentation if available. How will you document your software's contribution guidelines and governance structure?
Software licencing and compatibility	What licence will you give your software? How will you check that it respects the licences of libraries and dependencies it uses?
Deployment documentation	How will the installation requirements of your software be documented? Please provide a link to the installation documentation if available.
Citation	How will users of your software be able to cite your software? Please provide a link to your software citation file (CFF) if available.
Developer documentation	How will your software be documented for future developers?
Testing	How will your software be tested? Please provide a link to the (automated) testing results.
Software Engineering quality	Do you follow specific software quality guidelines? If yes, which ones?
Packaging	How will your software be packaged and distributed? Please provide a link to available packaging information (e.g. entry in a packaging registry, if available).
Maintenance	How do you plan to procure long term maintenance of your software?

Source: <https://doi.org/10.5281/zenodo.7248877>

# Machine-actionable Software Management Plans

- An overview of concepts used in the metadata model for maSMPs is available at: <https://github.com/zbmed-semtec/maSPMs>
- The first version of maSMP ontology is available at: [10.5281/zenodo.8089518](https://zenodo.org/record/8089518)



RDA DMP Common Standard -maDMP



CodeMeta

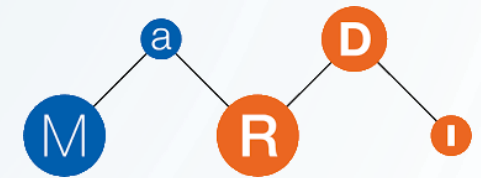
# Software metadata in maSMPs

	Software Source Code (aka SoftwareSourceCode in schema.org)	Software Release (aka SoftwareApplication in schema.org)
	Property name	Possible values (range)
	identifier	PropertyValue, Text, URL
	name	Text
	description	Text
	license	Text, URL
	author	Organization or Person
	contributor	Organization or Person
	citation	CreativeWork, Text, URL
From schema.org	conditionsOfAccess	Text
	isAccessibleForFree	Boolean
	codeRepository	URL
	programmingLanguage	ComputerLanguage, Text
	targetProduct (aka Software Release)	SoftwareApplication
	archivedAt	URL
	discussionURL	URL
	usageInfo	CreativeWork, URL
	version (i.e., semantic version)	Text
From maDMP	hasContact	Organization or Person
From Bioschemas	input	FormalParameter, Dataset
	output	FormalParameter, Dataset
From maSMP (New elements)	hasAPIDocumentation	Documentation
	hasDeveloperDocumentation	Documentation
	hasUserDocumentation	Documentation
	hasLearningResource	LearningResource
	hasVersionControlSystem	SoftwareApplication
	hasReadme	URL
	testedWith	TestAction



# Software-related metadata at NFDI

- ▶ Working group at NFDI-Metadaten on metadata for RSE
- ▶ MaRDI → Algorithms ontology
- ▶ Base4NFDI
  - nfdi4software
  - Jupyter4NFDI
  - Terminology lookup services



# Software-related metadata at NFDI4DS

## ► NFDI4DS

- maSMPs
  - Better support for testing and actual plan
  - Validation of resulting software metadata against initial plan
  - Integration to RDMO
  - Alignment to SMPs by Max Planck Digital Library
- Additional metadata supporting data science and artificial intelligence use case
  - Software + training datasets + models + evaluation
  - Use of metadata to assess first level of reproducibility
- Metadata extraction: from FAIRness evaluators to enablers (building up on top of OEG and GESIS)



MAX PLANCK  
digital library



gesis  
Leibniz-Institut  
für Sozialwissenschaften

# Machine Learning Metadata

- ▶ Software
  - Machine Learning algorithms
  - Software to train the model → hyperparameters, hardware, training time
- ▶ Data
  - Pre-processing → cleaning, missing values, normalization
  - Training dataset
- ▶ Machine learning Model



# Next steps

- ▶ Mini-hackathons in November @ ZB MED Köln
  - FAIRification Game for Software
  - Metadata for Machine Learning
    - Machine Learning Lifecycle
    - Machine Learning Lifecycle - Visualization
    - Metadata for Machine Learning
  - Machine-actionable Software Management Plans
- ▶ Metadata schemas as Bioschemas profiles
  - BioHackathon Germany
- ▶ Proof-of-concept using RO-Crates and SignPosting

**schema.org**

 Bioschemas

**deNBI**  
GERMAN NETWORK FOR BIOINFORMATICS INFRASTRUCTURE

**RO-Crate**

Thanks!  
Danke!

