



# Veröffentlichung von Forschungsdaten

**Prof. Dr.-Ing Stefan Gudenkauf**

Ein Leitfaden mit Markdown für GitHub und Zenodo

Dieses Dokument wurde gefördert durch das Niedersächsische Ministerium für Wissenschaft und Kultur mit Mitteln aus dem Programm zukunft.niedersachsen der VolkswagenStiftung.  
Projekt Data Driven Health (DEAL)

v1.0.0

Institut für Wirtschaftsinformatik (IfW)  
Fachbereich Management, Information, Technologie  
Jade Hochschule

8. Februar 2025



# Inhaltsverzeichnis

<b>Abstract</b>	<b>1</b>
<b>1 Der Weg zur Datenveröffentlichung</b>	<b>3</b>
1.1 Daten vorbereiten . . . . .	4
1.2 GitHub-Repository einrichten . . . . .	5
1.3 Zenodo mit GitHub verbinden . . . . .	6
1.4 Daten auf Zenodo veröffentlichen . . . . .	7
1.5 Veröffentlichung bekannt machen . . . . .	7
<b>2 Ein einfacher Datenmanagementplan</b>	<b>9</b>
<b>3 FAIR-Prinzipien berücksichtigen</b>	<b>11</b>
3.1 Findable: DOI und Metadaten ergänzen . . . . .	11
3.2 Findable: Suchmaschinen unterstützen . . . . .	11
3.3 Accessible: Langfristige Zugänglichkeit sicherstellen . . . . .	12
3.4 Interoperable: Datenformate und Standards beschreiben . . . . .	12
3.5 Reusable: Dokumentation und Lizenzierung beschreiben . . . . .	12
3.6 README-Datei und Metadaten . . . . .	13
<b>4 Semantic Versioning für Forschungsdaten</b>	<b>15</b>
4.1 Übertragung von SemVer auf Forschungsdaten . . . . .	15
4.2 Vorgehen für die Versionierung . . . . .	15
<b>5 Zusammenfassung</b>	<b>17</b>
5.1 Anleitung zur Datenveröffentlichung . . . . .	17
5.2 Datenmanagementplan (DMP) erstellen und pflegen . . . . .	17
5.3 Versionierung für Forschungsdaten anpassen . . . . .	17
<b>6 Anhang</b>	<b>19</b>
6.1 README-Datei für GitHub-Repository: README.md . . . . .	19
6.2 Datenmanagementplan: DMP.md . . . . .	19
6.3 Checkliste der FAIR-Prinzipien: FAIR.md . . . . .	21
6.4 Beispiel für ein Changelog: CHANGELOG.md . . . . .	22
6.5 Verwendete Softwarewerkzeuge . . . . .	22
<b>Literaturverzeichnis</b>	<b>25</b>



# Abbildungsverzeichnis

1.1	GitHub Desktop im Überblick. . . . .	6
1.2	GitHub-Integration in Zenodo. . . . .	7
1.3	Hauptseite eines Repositorys auf der GitHub Webseite. . . . .	8
1.4	Erstellen eines Release in einem GitHub-Repository. . . . .	8



# Tabellenverzeichnis

1.1	Zieldefinition gem. <i>Goal Question Metric</i> (GQM) von Basili u. a. (1994). . . . .	3
1.2	Werkzeuge zur Veröffentlichung von Forschungsdaten. . . . .	4
1.3	Ausgewählte Empfehlungen zur Dateibenennung. . . . .	4





# Abstract

Dieses Dokument richtet sich an Sie als Forschende/r einer Einrichtung mit begrenzten Mitteln für Forschungsdatenmanagement. Entweder stehen Sie direkt vor der Veröffentlichung eines neuen Datensatzes oder Sie möchten die Daten zu einer bereits bestehenden Veröffentlichung nachträglich zugänglich machen. Vielleicht sind sie auch im Rahmen einer Förderung durch Dritte dazu angehalten, Ihre Daten zu veröffentlichen. Dieses Dokument möchte Ihnen helfen, dieses Vorhaben mit möglichst geringem persönlichen Aufwand professionell umzusetzen.

This document is intended for researchers at institutions with limited resources for research data management. You may be about to publish a new dataset or looking to make data from an existing publication publicly accessible. Perhaps you are required to share your data as part of a third-party funding agreement. This document aims to help you achieve this professionally with minimal personal effort.

**Keywords:** research data management, data management plan, semantic versioning, fair principles, rdm tools



# 1 Der Weg zur Datenveröffentlichung

Dieses Dokument richtet sich an Sie als Forschende/r einer Einrichtung mit begrenzten Mitteln für Forschungsdatenmanagement. Es soll Ihnen helfen, dieses Vorhaben mit möglichst geringem persönlichen Aufwand umzusetzen (Tabelle 1.1).

Tabelle 1.1: Zieldefinition gem. *Goal Question Metric* (GQM) von Basili u. a. (1994).

Aspekt	Beispiel
Zweck	Verringerung
Problem	des persönlichen Aufwands
Objekt oder Prozess	zum Management von Forschungsdaten
Sichtpunkt	aus der Sicht einzelner Forschende/r

Bevor wir die einzelnen Schritte zur Veröffentlichung von Forschungsdaten durchgehen, benennen wir die grundlegenden Fragen:

1. Wo speichere ich meine Daten?
2. Wie verwalte ich neue Versionen meines Datensatzes?
3. Wie mache ich meinen Datensatz öffentlich und referenzierbar?
4. Wie beschreibe ich meinen Datensatz und dessen Metadaten sinnvoll?
5. Wie handhabe ich die FAIR-Prinzipien um?

Die Wahl der Werkzeuge beeinflusst sehr, wie aufwendig die Beantwortung dieser Fragen ist. Für diese Anleitung haben wir die Auswahl nach folgenden Kriterien getroffen:

1. So wenig Werkzeuge wie möglich
2. Robustheit und langfristige Verfügbarkeit
3. Hohe fachübergreifende Verbreitung
4. Keine zwingenden Abhängigkeiten von Verwaltungsstrukturen innerhalb der eigenen Einrichtung

Tabelle 1.2 beschreibt die Werkzeuge, die wir zur Veröffentlichung von Forschungsdaten verwenden werden. Schauen Sie sich ruhig schon einmal an, bevor wir mit den weiteren Schritten fortfahren.

Tabelle 1.2: Werkzeuge zur Veröffentlichung von Forschungsdaten.

Werkzeug	Beschreibung	Fragen
<a href="#">GitHub</a>	Internetdienst zum kollaborativen Erstellen, Speichern, Verwalten und Teilen von Code. Basiert auf dem kostenlosen und quelloffenen verteilten Versionskontrollsystem <a href="#">Git</a> . Unterstützt die Versionierung mittels <i>Releases</i> .	1, 2
<a href="#">Zenodo</a>	Offene europäische Plattform zur Veröffentlichung wissenschaftlicher Ergebnisse. Jede Einreichung erhält einen dauerhaften digitalen Objektbezeichner (DOI, siehe auch <i>DOI Handbook</i> (2023)).	3
<a href="#">Markdown</a>	Stark vereinfachte Auszeichnungssprache für die Erstellung von formatierten Text. Kann leicht in verschiedene Zielformate übersetzt werden.	4, 5

## 1.1 Daten vorbereiten

### 1. Bereiten Sie Ihre Daten auf:

- Verwenden Sie eine strukturierte, verständliche Dateibenennung. Es empfehlen sich die Best Practices der MIT Libraries, siehe Malinowski (2020). In Tabelle 1.3 stellen wir Ihnen ausgewählte Empfehlungen vor.
- Entfernen Sie sensible oder personenbezogene Daten.
- Dokumentieren Sie Ihr Vorgehen zur Datenerhebung und -aufbereitung in einer separaten Datei, z. B. `README.md` oder `metadata.txt`.

### 2. Erstellen Sie eine geeignete Ordnerstruktur:

- Erstellen Sie ein Hauptverzeichnis mit klar strukturierten Unterordnern, z. B. `/raw_data`, `/processed_data`, `/scripts`.

#### Hinweis

In Kapitel 6.1 finden Sie das Beispiel einer README-Datei in einfachem [Markdown](#)-Textformat. Es beinhaltet Textbausteine, die für die Ablage in einem GitHub-Repository geeignet sind.

Tabelle 1.3: Ausgewählte Empfehlungen zur Dateibenennung.

Thema	Empfehlungen
Dateinamen	nicht mehr als 32 Zeichen

Thema	Empfehlungen
Trennzeichen	nur Unterstrich <code>_</code> und <i>camelCase</i> als Trenner, z. B. <code>Handout_dateiBenennung.pdf</code> keine Leerzeichen
Sonderzeichen	keine Punkte außer direkt vor der Dateiendung keine sonstigen Sonderzeichen wie z. B. <code>&amp;</code> , <code>*</code> , <code>%</code> , <code>#</code> , <code>;</code> , <code>!</code> , <code>@</code> , <code>~</code> , <code>'</code> , <code>[</code> , <code>]</code> , <code>{</code> , <code>}</code> , <code>?</code> , <code>&lt;</code> , <code>&gt;</code>
Datumsangaben	möglichst konsistent, um das Suchen und Sortieren zu erleichtern <code>YYYYMMDD</code> (Jahr, Monat, Tag) ist ein sehr gutes Format, z. B. <code>20240828_Protokoll.pdf</code>
Nummerierungen	<b>führende Nummern</b> zur Abbildung fester Sequenzen Sequenzen von 1 bis 10: <code>01_bis 10_</code> Sequenzen von 1 bis 100: <code>001_bis 100_</code>

## 1.2 GitHub-Repository einrichten

GitHub basiert auf dem kostenlosen und quelloffenen Versionskontrollsystem [Git](#). Grundsätzlich können Sie auf der Kommandozeile Ihres Computers mit Git-Befehlen arbeiten, um Ihren Datensatz in GitHub zu pflegen (z. B. `git init`, `git add`, `git commit`). Dasselbe können Sie aber auch mit der leicht zu bedienenden Desktop-Anwendung [GitHub Desktop](#) erreichen.

Im Rahmen dieser Anleitung empfehlen und verwenden wir GitHub Desktop (Abbildung 1.1). Einen guten Einstieg in die Arbeit mit GitHub Desktop finden Sie in der offiziellen [GitHub-Dokumentation](#).

### 1. GitHub-Konto erstellen:

- Registrieren Sie sich auf [GitHub](#).
- Nach der Registrierung können Sie sich auf GitHub anmelden.
- Installieren Sie [GitHub Desktop](#).

### 2. Neues Repository erstellen:

- Melden Sie sich auf [GitHub](#) an und erstellen Sie ein neues öffentliches Repository.
- Alternativ können Sie das Repository auch mit GitHub Desktop erstellen: `File > New repository...` (Menüleiste oben links in Abbildung 1.1)
- Benennen Sie das Repository sinnvoll und passend zu `README.md` (z. B. `projectname-dataset`).
- Ist das Repository erstellt, gleicht GitHub Desktop das lokale Repository-Verzeichnis auf Ihrem Rechner mit dem öffentlichen Repository auf GitHub ab. Änderungen, die Sie am lokalen Ordner vornehmen, können hinzugefügt (`git add`), festgeschrieben (`git commit`) und anschließend in das öffentliche Repository übertragen werden (`git push`).

### 3. Daten in das Repository übertragen:

## 1 Der Weg zur Datenveröffentlichung

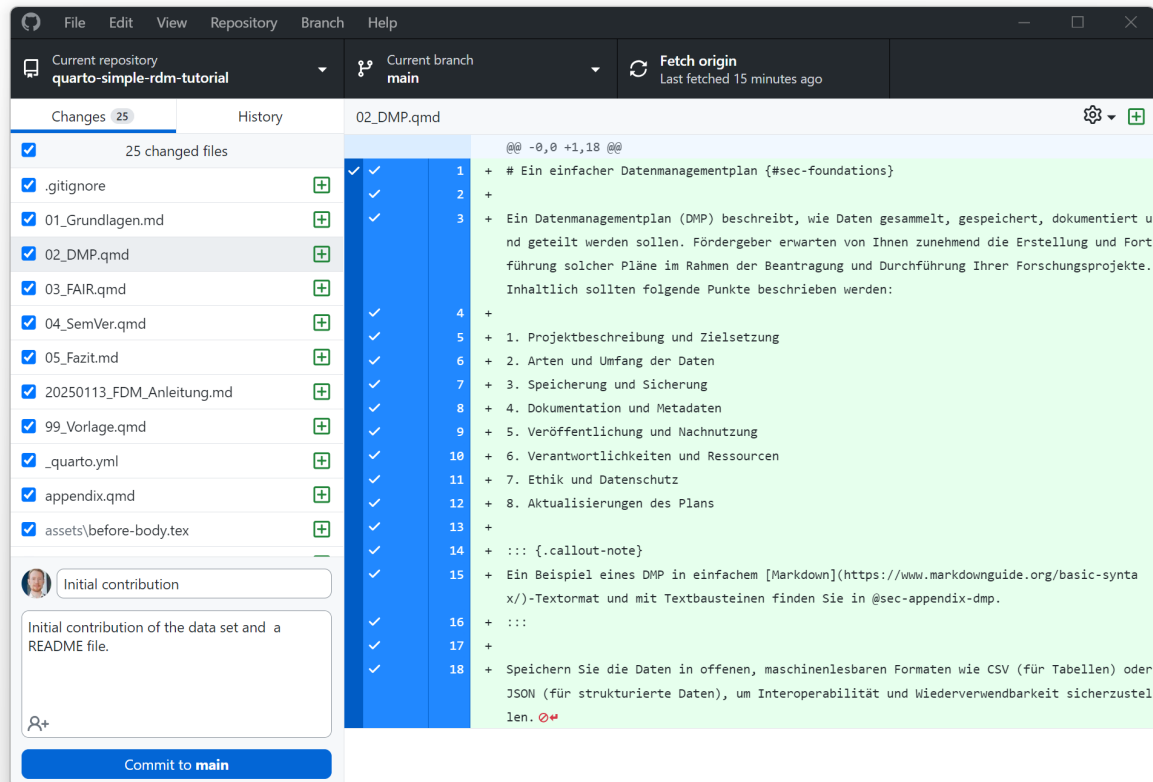


Abbildung 1.1: GitHub Desktop im Überblick.

- Legen Sie Ihre Daten und die README-Datei in das lokale Repository-Verzeichnis auf Ihrem Rechner.
- Geben Sie Ihre Änderungen am lokalen Repository-Verzeichnis an das öffentliche Repository auf GitHub weiter: **Commit to main** (unten links in Abbildung 1.1), dann **Push origin** (erscheint nach dem **Commit** oben rechts statt **Fetch origin** in Abbildung 1.1)
- Arbeiten Sie mit mehreren Personen gemeinsam am Repository, sollten Sie vor jedem **Commit** zunächst den aktuellen Stand einholen und mit Ihren Änderungen vergleichen: **Fetch origin** (oben rechts in Abbildung 1.1)

## 1.3 Zenodo mit GitHub verbinden

### 1. Zenodo-Konto erstellen:

- Registrieren Sie sich auf [Zenodo.org](https://zenodo.org). Nach der Registrierung können Sie sich z. B. mit Ihrer [ORCID ID](#) anmelden.

### 2. GitHub-Integration aktivieren:

- Gehen Sie nach der Anmeldung zu **“Settings”** und verbinden Sie Ihr GitHub-Konto mit Zenodo: **My account > GitHub** (Abbildung 1.2)
- Wählen Sie das Repository aus, das mit Zenodo veröffentlicht werden soll.

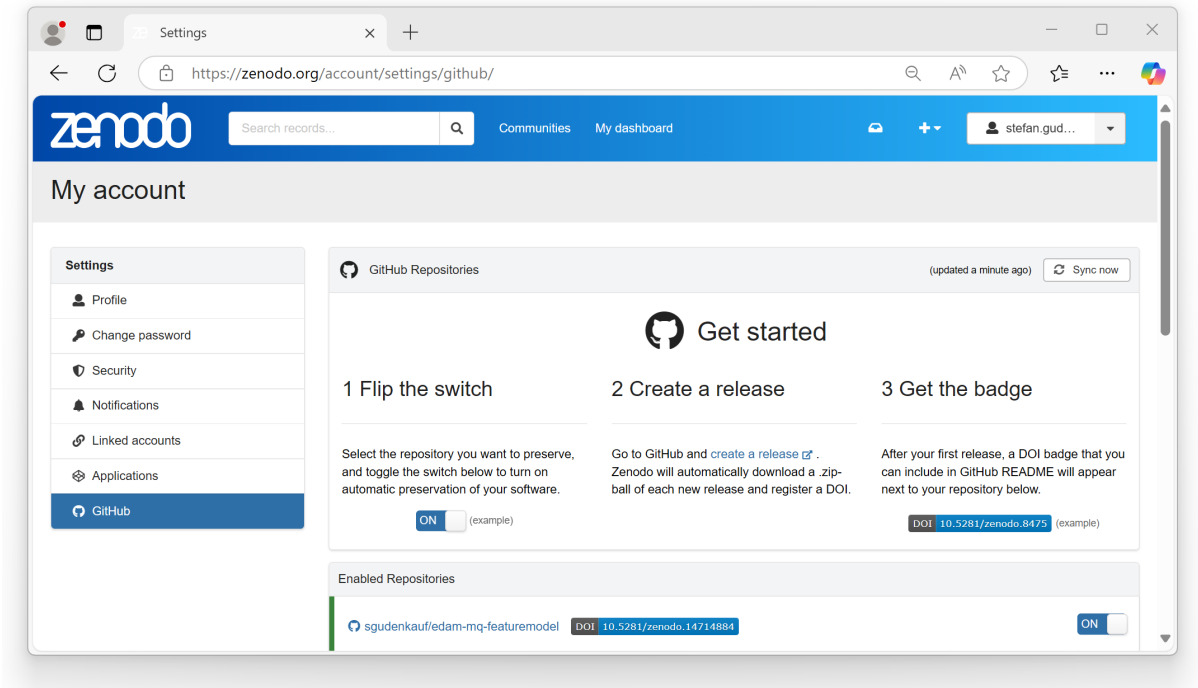


Abbildung 1.2: GitHub-Integration in Zenodo.

## 1.4 Daten auf Zenodo veröffentlichen

### 1. GitHub-Release erstellen:

- Melden Sie sich auf der GitHub-Webseite an und öffnen Sie die Hauptseite Ihres Repositories.<sup>1</sup>
- Erstellen Sie ein Release im GitHub-Repository: Releases > Draft a new release (unten rechts in Abbildung 1.3)
- Geben Sie als *Tag* eine passende Versionsnummer (z. B. v0.1.0) und eine kurze Beschreibung für das Release an (Abbildung 1.4).
- Eine detaillierte Anleitung zum Erstellen eines Releases finden Sie in der [GitHub-Dokumentation](#).

### 2. Automatische Zenodo-DOI erstellen:

- Zenodo generiert automatisch einen DOI für das Release.
- Fügen Sie den DOI zur README-Datei hinzu.

## 1.5 Veröffentlichung bekannt machen

- Verweisen Sie im Paper auf die DOI des Datensatzes.
- Teilen Sie den Zenodo-Link und den DOI mit Ihrem Netzwerk.

<sup>1</sup>GitHub Desktop unterstützt das Erstellen von Releases momentan nicht.

## 1 Der Weg zur Datenveröffentlichung

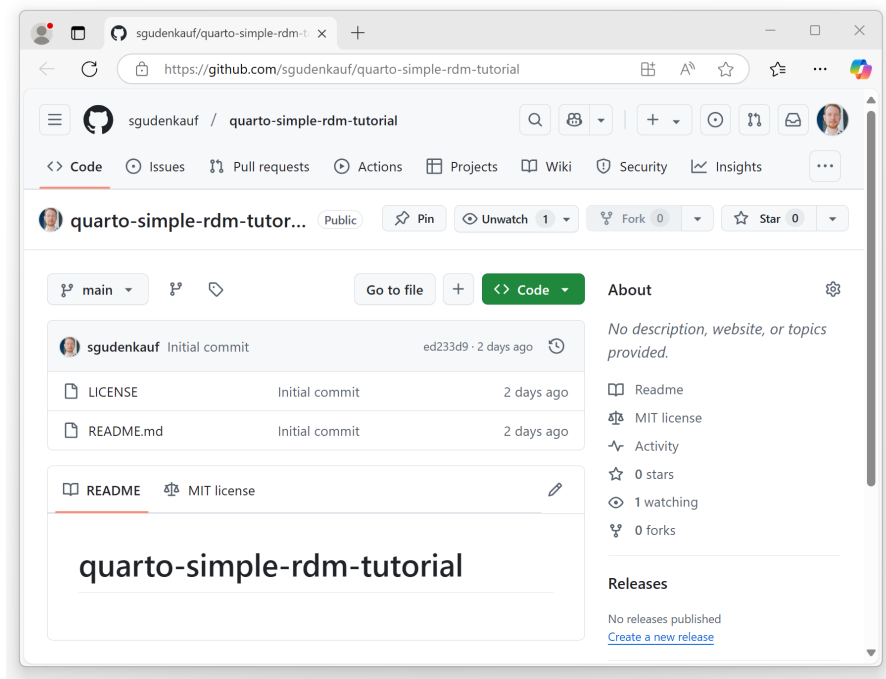


Abbildung 1.3: Hauptseite eines Repositories auf der GitHub Webseite.

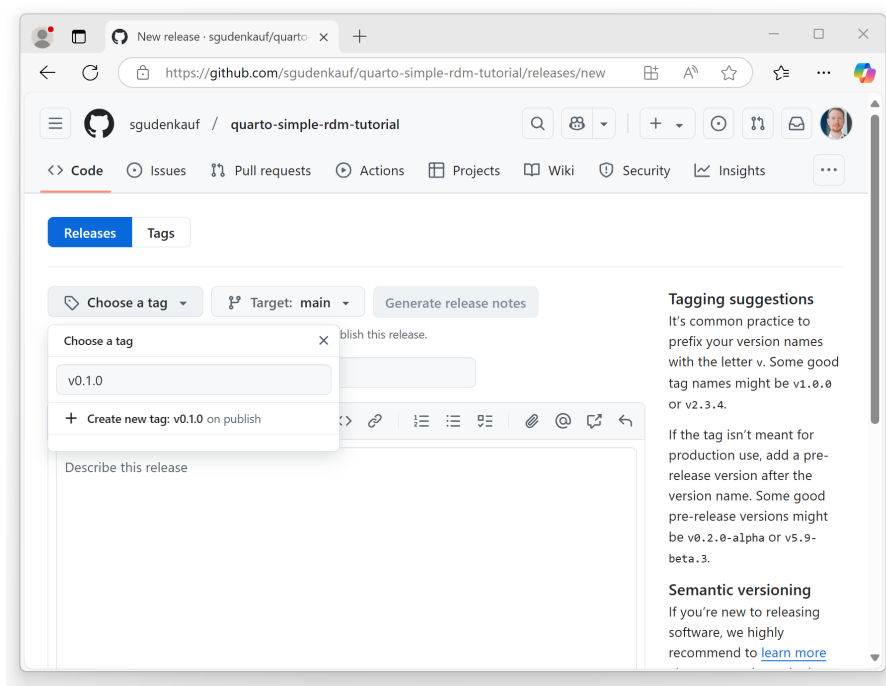


Abbildung 1.4: Erstellen eines Release in einem GitHub-Repository.



## 2 Ein einfacher Datenmanagementplan

Ein Datenmanagementplan (DMP) beschreibt, wie Daten gesammelt, gespeichert, dokumentiert und geteilt werden sollen. Fördergeber erwarten von Ihnen zunehmend die Erstellung und Fortführung solcher Pläne im Rahmen der Beantragung und Durchführung Ihrer Forschungsprojekte. Inhaltlich sollten folgende Punkte beschrieben werden:

1. Projektbeschreibung und Zielsetzung
2. Arten und Umfang der Daten
3. Speicherung und Sicherung
4. Dokumentation und Metadaten
5. Veröffentlichung und Nachnutzung
6. Verantwortlichkeiten und Ressourcen
7. Ethik und Datenschutz
8. Aktualisierungen des Plans

Speichern Sie die Daten in offenen, maschinenlesbaren Formaten wie CSV (für Tabellen) oder JSON (für strukturierte Daten), um Interoperabilität und Wiederverwendbarkeit sicherzustellen.

### Hinweis

In Kapitel [6.2](#) finden Sie das Beispiel eines Datenmanagementplans in einfachem [Markdown-Textformat](#). Er beinhaltet Textbausteine, die für die Ablage in einem GitHub-Repository geeignet sind.



## 3 FAIR-Prinzipien berücksichtigen

Die FAIR-Prinzipien sind Richtlinien für die Speicherung und Veröffentlichung wissenschaftlicher Daten. Einen guten Einstieg bieten „Data models to GO-FAIR“ (2017) und Wilkinson u. a. (2016).

Daten sollen auffindbar, zugänglich, interoperabel und wiederverwertbar sein. Diese vier Eigenschaften werden auf 15 [FAIR-Prinzipien](#) abgebildet. Auf der Ebene der vier grundlegenden Eigenschaften bietet Ihnen dieses Kapitel aber einen ersten Einstieg.

### Hinweis

In Kapitel [6.3](#) finden Sie eine Checkliste der 15 FAIR-Prinzipien in einfachem [Markdown-Textformat](#).

### 3.1 Findable: DOI und Metadaten ergänzen

Die Verwendung von Zenodo zur Vergabe eines digitalen Objektbezeichners (DOI, siehe auch *DOI Handbook* (2023)) und zur Nutzung von Metadaten sollte im Datenmanagementplan dokumentiert sein. Um die Auffindbarkeit der Daten noch weiter zu verbessern, können Sie folgende Punkte adressieren:

- Hinterlegen Sie Metadaten in einem standardisierten Format. Bekannte Formate sind z. B. [DataCite](#) oder [Dublin Core](#).
- Machen Sie Angaben zu üblichen Schlagwörtern und standardisierten Vokabularen zu Ihrem Fachgebiet.

**Ergänzen Sie bei Bedarf im DMP:**

```
1  **Auffindbarkeit der Daten:**
2    - DOI wird über Zenodo vergeben.
3    - Metadaten nach DataCite-Standard: Titel, Autor, Datum, Lizenz, Beschreibung
4    - Schlagwörter: [Themenrelevante Begriffe, z. B. "Machine Learning",
   ↪ "Umweltdaten"]
```

### 3.2 Findable: Suchmaschinen unterstützen

- GitHub-Projekte werden in der Regel von GitHub automatisch indexiert, so dass sie besser gefunden werden können.
- Zusätzlich sollten Sie keine unnötigen `.gitignore`-[Einstellungen](#) verwenden.

### 3.3 Accessible: Langfristige Zugänglichkeit sicherstellen

- Ergänzen Sie im Datenmanagementplan explizit Angaben zur langfristigen Speicherung über Zenodo und dass die Daten öffentlich zugänglich sind ([Open Access](#)).
- Zeitliche Einschränkungen des Zugriffs können über die den so genannten [Embargo-Status](#) in Zenodo geregelt werden.

**Ergänzen Sie bei Bedarf im DMP:**

```
1  **Zugänglichkeit:**
2  - Die Daten sind über Zenodo öffentlich zugänglich (Open Access).
3  - DOI garantiert die dauerhafte Verfügbarkeit.
4  - Einschränkungen des Zugriffs werden über Embargo-Status Optionen von Zenodo
   ↪ geregelt.
```

### 3.4 Interoperable: Datenformate und Standards beschreiben

- Verwenden Sie für Ihre Daten Dateiformate, die eine möglichst einfache maschinelle Verarbeitung erlauben.
  - Es bieten sich offene Formate wie z. B. CSV gemäß Hausenblas u. a. (2014) und JSON gemäß Charollais an.
  - Verzichten Sie auf proprietäre Formate wie z. B. Microsoft Excel (.xlsx-Dateien).
- Nennen Sie die verwendeten Standards für Daten und Metadaten wie z. B. [DataCite](#) und [JSON-Schema](#).
- Nennen Sie Ontologien oder kontrollierte Vokabulare zur Standardisierung von Fachbegriffen, z. B. [MeSH](#), [AGROVOC](#).

**Ergänzen Sie bei Bedarf im DMP:**

```
1  **Interoperabilität:**
2  - Daten werden in offenen Formaten gespeichert: CSV (Tabellen), JSON
   ↪ (strukturierte Daten)
3  - Metadaten-Standards: DataCite für Beschreibung und DOI-Referenzierung
4  - Ontologien oder kontrollierte Vokabulare zur Standardisierung von Begriffen:
   ↪ Verwendung von [z. B. MeSH, AGROVOC]
```

### 3.5 Reusable: Dokumentation und Lizenzierung beschreiben

Ohne Angabe einer Lizenz können Ihre Daten nicht weiterverwendet werden. Weit verbreitet und üblich sind [CC-BY 4.0](#) für Dokumente und die [MIT-Lizenz](#) für ausführbaren Programmcode und Softwareprojekte.

Als recht freizügige Lizenz bietet die MIT-Lizenz ein geringes Risiko im Zusammenspiel mit anderen Lizenzen. Beispielsweise erlaubt sie die Wiederverwendung von Code in proprietärer Software. Creative-Commons-(CC)-Lizenzen sind dagegen nicht für die Nutzung in Software oder Hardware empfohlen (Bals (2024)).

Da Veröffentlichungen von Forschungsdaten häufig auch ausführbaren Code beinhalten, bevorzugen wir im Zweifelsfall die MIT-Lizenz.

**Ergänzen Sie bei Bedarf im DMP:**

```

1  **Wiederverwendbarkeit:**
2    - Lizenz: MIT (erlaubt die Nachnutzung auch in proprietärer Software unter der
    ↪ Bedingung der Namensnennung)
3    - Vollständige Dokumentation der Daten: Methoden, Kontext, verwendete Software (z.
    ↪ B. Python-Skripte mit spezifischen Bibliotheksversionen)
4    - README-Datei enthält Anleitungen für die Nachnutzung.
```

## 3.6 README-Datei und Metadaten

- Falls Metadaten generiert werden, kann dies in der README-Datei ergänzt werden (z. B. `metadata.txt` wird automatisch erzeugt).
- Python-Skripte oder die Zenodo-API könnten Ihnen dabei helfen.

**Ergänzen Sie bei Bedarf in der README-Datei:**

```

1  ## Metadata
2  The dataset is described using the DataCite metadata schema. Key elements include:
3    - **Title**: Title of the dataset.
4    - **Author(s)**: Name(s) of the contributors.
5    - **Keywords**: Relevant keywords for discoverability.
6    - **Description**: Detailed dataset description, including methodology and tools.
7    - **License**: MIT
8    - **DOI**: [Zenodo DOI link].
```



## 4 Semantic Versioning für Forschungsdaten

*Semantic Versioning* ([SemVer](#)) ist ein Standard für die Vergabe von Versionsnummern bei der Entwicklung von Software. Es besteht aus drei Komponenten: MAJOR.MINOR.PATCH. Eine typische Versionsnummer nach SemVer kann z.B. so aussehen: v1.0.1.

### Hinweis

In Kapitel [6.4](#) finden Sie ein Beispiel für ein Changelog in einfachem [Markdown-Textformat](#).

### 4.1 Übertragung von SemVer auf Forschungsdaten

SemVer eignet sich aus unserer Sicht auch sehr gut für die Veröffentlichung von Forschungsdaten. Hierzu müssen wir die Semantik der Komponenten geringfügig anpassen:

- MAJOR: Erhöhen bei grundlegenden Änderungen, die die Abwärtskompatibilität der Forschungsdaten brechen (z. B. neue Variablenstruktur, neue Datensätze).
- MINOR: Erhöhen bei Hinzufügungen, die abwärtskompatibel sind (z. B. neue Datenpunkte, zusätzliche Dokumentation).
- PATCH: Erhöhen bei kleinen Korrekturen, die abwärtskompatibel sind (z. B. Fehlerkorrekturen in den Daten).

### 4.2 Vorgehen für die Versionierung

1. **Initiale Version:** Starten Sie mit v1.0.0 bei der ersten Veröffentlichung.
2. **Dokumentation von Änderungen:** Dokumentieren Sie sämtliche Änderungen zwischen den Versionen in einem Changelog (z. B. CHANGELOG.md).
3. **Release-Management:**
  - Verwenden Sie [GitHub-Releases](#), um neue Versionen zu kennzeichnen.
  - Die Zenodo-Integration erzeugt automatisch neue DOIs für jede Version.





# 5 Zusammenfassung

## 5.1 Anleitung zur Datenveröffentlichung

- **Plattformen:**
  - [Markdown](#) als Textauszeichnungssprache zur einfachen Dokumentation
  - [GitHub](#) für die Versionierung
  - [Zenodo](#) für DOI-Vergabe und Archivierung
- **Vorgehen:**
  1. Daten strukturieren
  2. GitHub-Repo erstellen
  3. Zenodo verknüpfen
  4. Release veröffentlichen
- **README:** Strukturierte Dokumentation der Daten und Nutzungshinweise

## 5.2 Datenmanagementplan (DMP) erstellen und pflegen

- **Metadaten:** Standardformate (z. B. [DataCite](#)), offene Formate ([CSV](#), [JSON](#))
- **Lizenz:** [CC-BY 4.0](#) oder [MIT-Lizenz](#) für die maximale Nachnutzung
- **FAIR-Prinzipien:** Daten auffindbar, zugänglich, interoperabel, und wiederverwendbar gestalten

## 5.3 Versionierung für Forschungsdaten anpassen

- **Semantic Versioning (SemVer):**
  - MAJOR.MINOR.PATCH für Änderungen
  - Änderungen dokumentieren in `CHANGELOG.md`
- **GitHub-Releases:** Neue Versionen in GitHub erzeugen automatisch neue DOIs in Zenodo



## 6 Anhang

### 6.1 README-Datei für GitHub-Repository: README.md

```
1 # Project Name - Dataset
2
3 ## Overview
4 This repository contains the dataset used in the research article:
5 "Title of the Paper", published in Journal Name.
6 DOI of the article: [Link to Paper DOI]
7
8 ## Dataset Description
9 - Raw Data: `/raw_data/` contains the original data collected during the study.
10 - Processed Data: `/processed_data/` contains cleaned and analyzed data.
11 - Scripts: `/scripts/` includes scripts used for data analysis.
12
13 ## How to Use
14 1. Clone this repository: `git clone
   ↪ https://github.com/username/projectname-dataset.git`
15 2. Navigate through the folders for relevant files.
16 3. Refer to `metadata.txt` for detailed descriptions of the dataset and methodology.
17
18 ## Citation
19 If you use this dataset, please cite:
20 - Dataset: [DOI provided by Zenodo]
21 - Paper: [Paper DOI]
22
23 ## License
24 This dataset is licensed under [LICENSE NAME]. See `LICENSE` file for details.
```

### 6.2 Datenmanagementplan: DMP.md

```
1 # Datenmanagementplan (DMP)
2
3 ## 1. Projektbeschreibung
4 Projektname: [Titel des Forschungsprojekts]
5 Verantwortliche*r Forschende*r: [Ihr Name]
6 Institution: [Name der Hochschule oder Forschungseinrichtung]
7 Förderorganisation (falls zutreffend): [Förderorganisation, z. B. DFG]
8
9 ### Ziel des Projekts
10 [Kurze Beschreibung des Projekts, z. B.:
```

## 6 Anhang

```
11 "Dieses Projekt untersucht [Thema]. Die generierten Daten dienen als Grundlage für
   ↳ die Publikation im Journal [Name] und werden gemäß den FAIR-Prinzipien
   ↳ öffentlich zugänglich gemacht."
12
13 ## 2. Arten und Umfang der Daten
14 **Datentypen**:
15 - [Primärdaten, z. B. experimentelle Messdaten, Umfrageergebnisse, Bilddaten]
16 - [Sekundärdaten, z. B. aus Literatur oder externen Datenquellen]
17 - [Metadaten, z. B. Dokumentation, Kontextbeschreibungen]
18
19 **Umfang der Daten**:
20 - Geschätzte Dateigröße: [z. B. 500 MB]
21 - Anzahl der Dateien: [z. B. 30 CSV-Dateien]
22
23 **Dateiformate**:
24 - Rohdaten: [z. B. CSV, TIFF]
25 - Analysedaten: [z. B. XLSX, JSON]
26 - Dokumentation: [z. B. Markdown, PDF]
27
28 ## 3. Speicherung und Sicherung
29 **Speicherorte**:
30 - **Primäre Speicherung**: GitHub-Repository ([Link hinzufügen])
31 - **Backup-Strategie**: Automatische Sicherung auf lokalen Systemen, Cloud-Diensten
   ↳ (z. B. [Google Drive, OneDrive]), o.a.
32
33 **Versionierung**:
34 - Git wird zur Versionierung und Nachverfolgung von Änderungen verwendet.
35
36 **Zugriffsrechte während des Projekts**:
37 - Nur Projektbeteiligte haben während der Erstellung Schreibrechte.
38 - GitHub ermöglicht eine öffentliche Einsicht nach Abschluss des Projekts.
39
40 ## 4. Dokumentation und Metadaten
41 **Metadatenformat**:
42 - [z. B. DataCite, Dublin Core]
43
44 **Beschreibung der Daten**:
45 - Jede Datei wird durch eine README-Datei dokumentiert, die Informationen über:
46   - Den Inhalt
47   - Die Erhebungsmethodik
48   - Verwendete Software oder Tools enthält.
49
50 **DOI-Zuweisung**:
51 - Die Veröffentlichung erfolgt mit einem DOI über Zenodo.
52
53 ## 5. Veröffentlichung und Nachnutzung
54 **Geplante Veröffentlichung**:
55 - Veröffentlichung auf GitHub und Zenodo nach Akzeptanz des zugehörigen Papers.
56
57 **Lizenzierung**:
58 - CC-BY 4.0 empfohlen für maximale Nachnutzung
59 - (Alternativen: CC-BY-NC 4.0, CC0, MIT-Lizenz)
```

```

60
61 **Zugänglichkeit**:
62 - Vollständig öffentlich.
63 - Einschränkungen: Keine.
64
65 **Langfristige Archivierung**:
66 - Zenodo garantiert die Archivierung für mindestens 20 Jahre.
67
68 ## 6. Verantwortlichkeiten und Ressourcen
69 **Datenverantwortliche/r**:
70 - [Ihr Name]
71
72 **Zusätzliche Ressourcen**:
73 - Genutzte Software: [z. B. Python, R]
74 - Infrastruktur: GitHub, Zenodo.
75
76 ## 7. Ethik und Datenschutz
77 **Datenschutzanforderungen**:
78 - Keine personenbezogenen Daten enthalten.
79 - Sensible Daten wurden entfernt oder anonymisiert.
80
81 **Ethikfreigaben**:
82 - Nicht erforderlich. (Alternative: "Wurde durch die Ethikkommission genehmigt am
  ↳ [Datum].")
83
84 ## 8. Aktualisierungen des Plans
85 **Versionierung des DMP**:
86 - Initiale Version: [Datum]
87 - Geplante Überprüfung: [z. B. alle 6 Monate oder nach Projektmeilensteinen]

```

## 6.3 Checkliste der FAIR-Prinzipien: FAIR.md

```

1 # FAIR-Checkliste für GitHub & Zenodo
2
3 ## Findable (Auffindbar)
4 - [ ] **F1**: Der Datensatz hat eine eindeutige und dauerhafte **DOI** (z. B. von
  ↳ Zenodo).
5 - [ ] **F2**: Metadaten enthalten einen klaren, beschreibenden **Titel** und
  ↳ **Schlüsselwörter**.
6 - [ ] **F3**: Der Datensatz ist in einem **offenen Repository** (GitHub & Zenodo)
  ↳ gespeichert.
7 - [ ] **F4**: Metadaten sind **durchsuchbar** (z. B. über Zenodo-Suchfunktionen).
8
9 ## Accessible (Zugänglich)
10 - [ ] **A1**: Der Datensatz ist über eine **öffentliche URL** abrufbar.
11 - [ ] **A1.1**: Daten und Metadaten sind auch nach Ablauf des Projekts verfügbar
  ↳ (Zenodo sichert dies).
12 - [ ] **A1.2**: **Metadaten bleiben abrufbar**, auch wenn die Daten entfernt wurden.
  ↳
13 - [ ] **A2**: Falls Zugriffsbeschränkungen bestehen, sind diese klar dokumentiert.

```

```
14
15 ## Interoperable (Interoperabel)
16 - [ ] I1: Datensatz und Metadaten nutzen offene, standardisierte Formate (z.
    ↳ B. CSV, JSON, Markdown).
17 - [ ] I2: Metadaten enthalten kontrollierte Vokabulare oder
    ↳ standardisierte Begriffe (z. B. ORCID für Autoren).
18 - [ ] I3: Der Datensatz verlinkt relevante Publikationen, Software oder andere
    ↳ Daten.
19
20 ## Reusable (Nachnutzbar)
21 - [ ] R1: Metadaten enthalten eine klare Lizenz (z. B. CC-BY 4.0 oder MIT
    ↳ für Code).
22 - [ ] R1.1: Der Datenmanagementplan (DMP) beschreibt den Datensatz präzise.
23 - [ ] R1.2: Der Datensatz enthält eine detaillierte README.md zur Nutzung.
24 - [ ] R1.3: Die Datenquelle, Entstehung und Verarbeitungsschritte sind
    ↳ dokumentiert (z. B. in der README oder im DMP).
25
26 Tipp: Halte die README.md und Metadaten aktuell, um die FAIR-Prinzipien
    ↳ langfristig zu gewährleisten.
```

### 6.4 Beispiel für ein Changelog: CHANGELOG.md

Legen Sie die Datei CHANGELOG.md auf derselben Ordnerbene wie README.md an:

```
1 # Changelog
2
3 ## v1.1.0 - 2025-01-10
4 - Hinzugefügt: Neue Datenpunkte für [Experiment XYZ].
5 - Geändert: Erweiterung der README mit zusätzlichen Analyseschritten.
6
7 ## v1.0.1 - 2024-12-20
8 - Behoben: Fehler in den Variablenbeschreibungen.
9
10 ## v1.0.0 - 2024-12-01
11 - Initiale Veröffentlichung des Datensatzes.
```

### 6.5 Verwendete Softwarewerkzeuge

Zur Erstellung dieses Leitfadens haben wir verschiedene Softwarewerkzeuge verwendet. Deren Gebrauch dokumentieren wir gemäß Baresel u. a. (2024). Genannt werden der Name des Werkzeugs, Quelle, Verwendungszweck und Anwendungsbereich.<sup>1</sup> Die Angaben sollten den **Empfehlungen** des APA-Stils für die Angabe von Softwareversionen entsprechen.

1. OpenAI. (2023). **ChatGPT** (Version 14. März) [Großes Sprachmodell]. <https://chat.openai.com><sup>2</sup>

<sup>1</sup>Wir verzichten auf die zusätzliche Nennung der "genutzten Funktion" gemäß Baresel u. a. (2024), da sie sich aus unserer Sicht mit dem Verwendungszweck doppelt. Siehe auch Tabelle 2 in Baresel u. a. (2024).

<sup>2</sup>Die genaue Versionsnummer von ChatGPT ist nicht öffentlich zugänglich. Wir verwenden daher das Datum der letzten Aktualisierung als Versionsbezeichnung.

- Übersetzung deutsch/englisch (Abstract)
  - Generieren von Vorschlägen für Markdown-Vorlagen (Anhang, Kapitel 3)
  - Generieren von Vorschlägen zu Vorgehen und Struktur (gesamtes Dokument)
2. Microsoft Corporation. (2024). **Visual Studio Code** (Version 1.96.2) [Quelltext-Editor]. <https://code.visualstudio.com>
- Editor zum Schreiben der Quelldateien (gesamtes Projekt)
3. Posit, PBC. (2024). **Quarto** (Version 1.118.0) [Dokumentations- und Publishing-Software]. <https://quarto.org>
- Übersetzen der Quelldateien in Webseite und PDF-Dokument, siehe Verzeichnis docs/ (gesamtes Projekt)
4. Posit, PBC. (2024). **Quarto Extension for Visual Studio Code** (Version 1.6.39) [Software-Erweiterung]. <https://quarto.org/docs/tools/vscode.html>
- Komfortable Bedienung von Quarto aus Visual Studio Code heraus (gesamtes Projekt)
  - Quarto-Syntaxunterstützung (gesamtes Projekt)





# Literaturverzeichnis

[Data models to GO-FAIR](#). In: *Nature Genetics* Bd. 49 (2017), Nr. 7, S. 971–971

[DOI Handbook](#), 2023

Bals, Fred: [Top open source licenses and legal risk for developers](#), 2024

Baresel, Kira ; Eube, Cornelia ; Knorr, Dagmar ; Lutter, Ly ; Nys, Jasmin de ; Röben, Marieke: [KI-Gebrauch im Studienkontext dokumentieren](#) (2024), S. 304839 b. — Artwork Size: 304839  
b Medium: application/pdf Publisher: Medien- und Informationszentrum, Leuphana Universität Lüneburg Version Number: 1

Basili, Victor R. ; Caldiera, Gianluigi ; Rombach, H. Dieter: [Goal Question Metric Paradigm](#). In: *Encyclopedia of Software Engineering - 2 Volume Set* : John Wiley & Sons, Inc., 1994 — ISBN \#1-54004-8, S. 528–532

Charollais, Patrick: ECMA-404, 2nd edition, December 2017

Hausenblas, M. ; Wilde, E. ; Tennison, J.: [URI Fragment Identifiers for the text/csv Media Type](#) ( Nr. RFC7111) : RFC Editor, 2014

Malinowski, Christine: [File Naming Best Practices](#) : MASSACHUSETTS INSTITUTE OF TECHNOLOGY, 2020

Wilkinson, Mark D. ; Dumontier, Michel ; Aalbersberg, IJsbrand Jan ; Appleton, Gabrielle ; Axton, Myles ; Baak, Arie ; Blomberg, Niklas ; Boiten, Jan-Willem ; u. a.: [The FAIR Guiding Principles for scientific data management and stewardship](#). In: *Scientific Data* Bd. 3 (2016), Nr. 1, S. 160018

