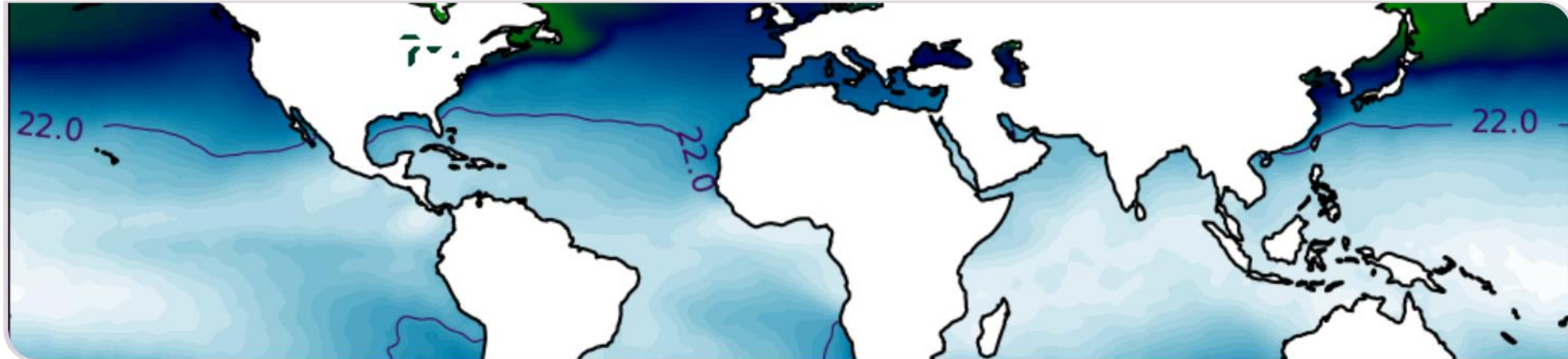


M.Sc. Modul Geodatenanalyse 1

Softwareentwicklung, -lizensierung, Datenethik

Gabriel Rau

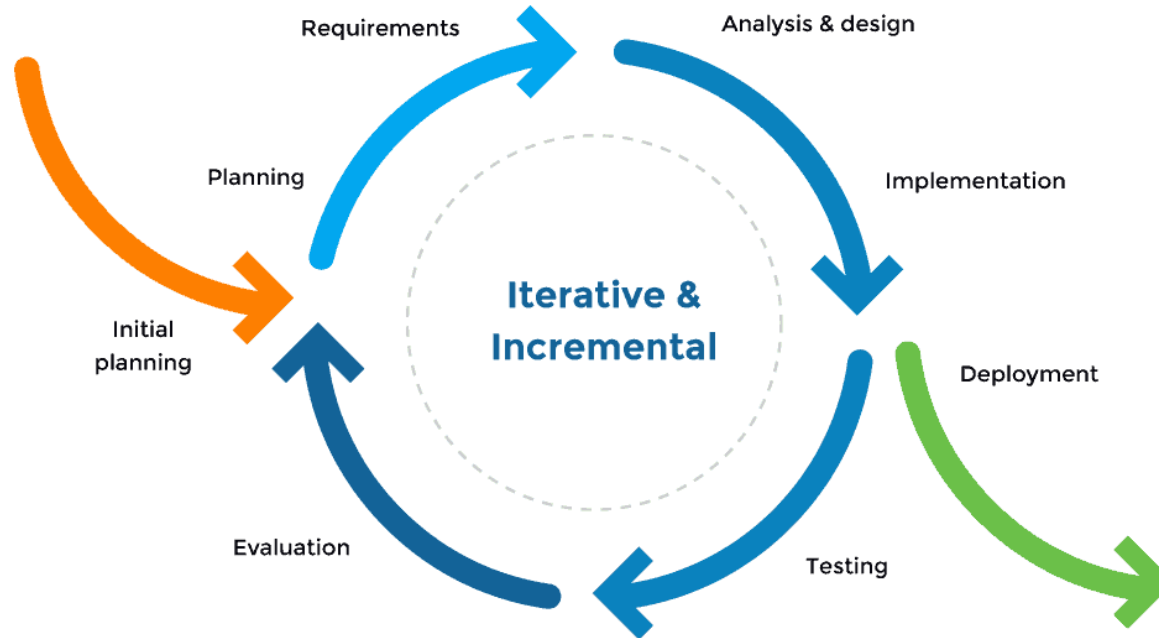


Softwareentwicklung

Entwicklung von Code

- Während dieses Moduls habt ihr die Entwicklung von Code selbst erlebt
- Dazu gehört auch „Trial-and-Error“
- Während der Entwicklung der Logik habt ihr Schritte vorwärts und rückwärts gemacht
- Dabei habt ihr gemerkt, dass die Entwicklung von Code **iterativ** ist
- Die Entwicklung von Software basiert auf der iterativen Entwicklung von Code

Überblick Softwareentwicklung



[Quelle Plutora](#)

Die Schwierigkeit der Entwicklung ...

- Welche Änderungen wurden wann und von wem gemacht?
- Das Problem: Kleine Änderungen können Code oder Software unbrauchbar machen
- Software besteht meistens aus sehr vielen Text-Dateien, welche Code enthalten
- Wie kann man diese am besten archivieren?
- Was passiert, wenn man Code auf verschiedenen Rechnern entwickelt?
- **Weil Code-Entwicklung iterativ, inkrementell und dezentral sein kann, braucht es eine robuste Versionskontrolle**

Zentrale Systeme

Beispiele

- MediaWiki
- SCCS
- RCS
- CVS
- Subversion (SVN)

Dezentrale Systeme

Beispiele

- Bazaar
- BitKeeper
- Darcs
- Fossil
- Git (wird hier vorgestellt)
- GNU arch
- Mercurial
- Monotone

Git als Versionskontrolle

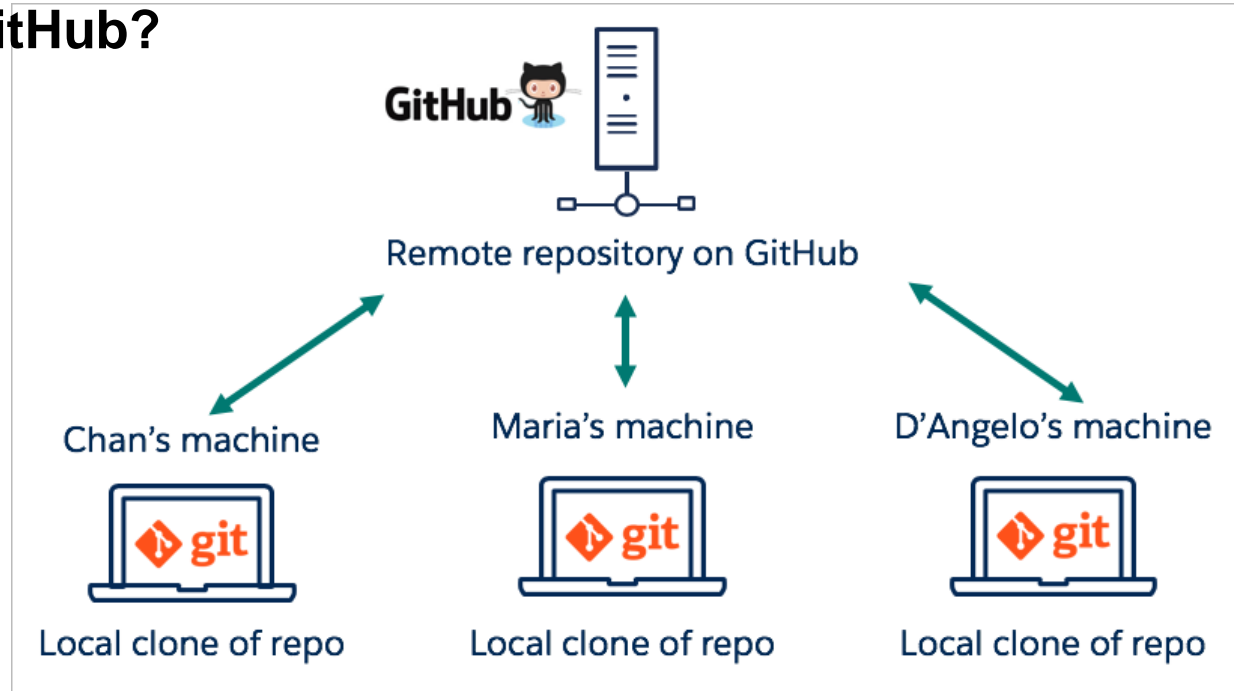
Was ist Git?

- Git ist ein kostenloses, verteiltes Versionskontrollsystem für Softwareprojekte
- Das Programm ermöglicht es mehreren Entwicklern, unabhängig von ihrem Aufenthaltsort gleichzeitig an einem Projekt zu arbeiten
- Die Versionskontrolle macht es einfach, Änderungen eigenständig und von überall aus dem Projekt hinzuzufügen
- Die Änderungen werden protokolliert
- Git ist plattformunabhängig und lässt sich somit in nahezu jeder Umgebung nutzen

Wir verwenden Git für dieses Modul

GitHub und Versionskontrolle

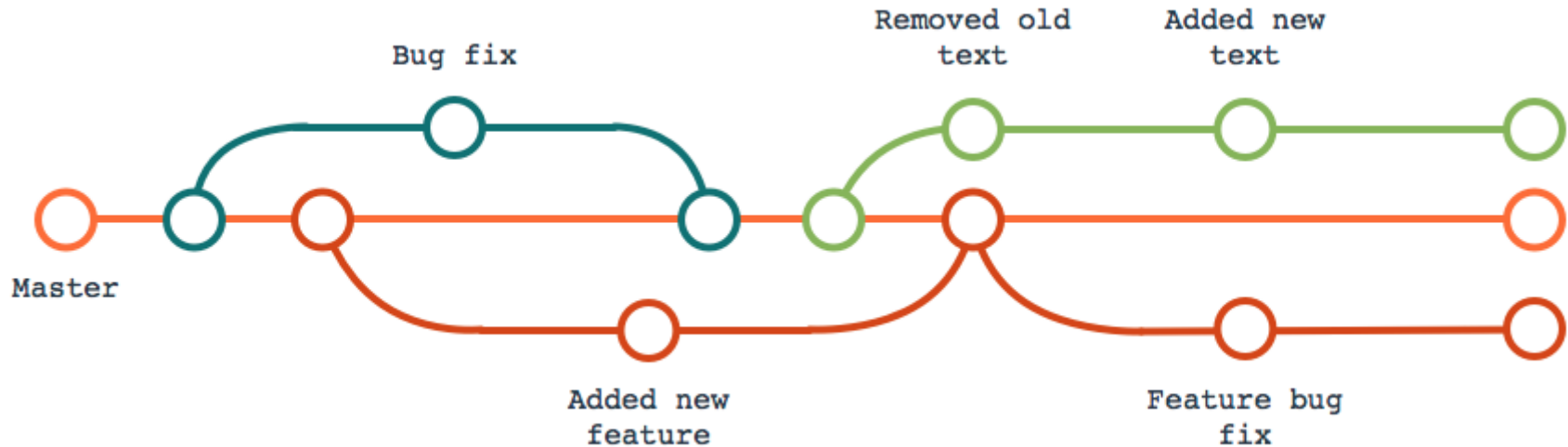
Warum GitHub?



Quelle: [Trailhead](#)

GitHub und Versionskontrolle

Übersicht über Den Workflow



Quelle: [cPanel](#)

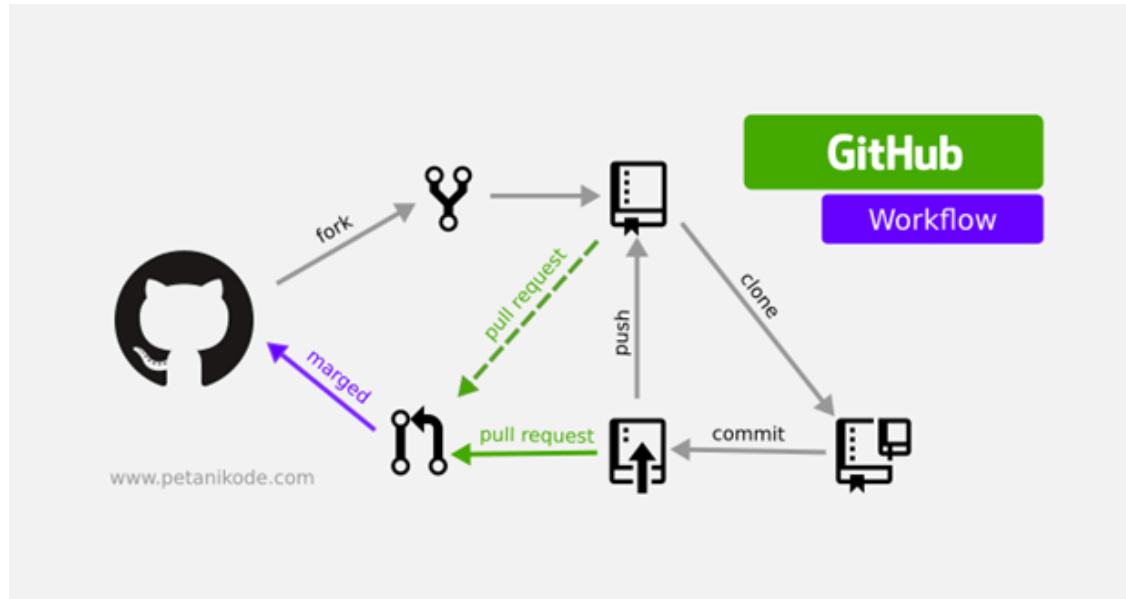
GitHub und Versionskontrolle

Änderungen werden angezeigt

```
...  
784 784         if ns is None:  
785 785             return  
786 786         if not isinstance(ns, (tuple, list)):  
787 -             raise Exception("namespace must be tuple")  
787 +             raise Exception("namespace must be a list or a tuple")  
788 788         if ns[0] is None:  
789 789             self.expns = ns[1]  
790 790         else:  
...
```

GitHub und Versionskontrolle

Übersicht über die Möglichkeiten



Quelle: [C#corner](#)

Softwarelizenzierung

Geistiges Eigentum

■ Was ist überhaupt geistiges Eigentum?

*„Unter dem Begriff "geistiges Eigentum" – international als "intellectual property (IP)" bezeichnet – fallen Eigentumsrechte an Schöpfungen des menschlichen Intellekts (beispielsweise Erfindungen, Know-how, **Software**). Der Begriff "gewerbliche Schutzrechte" bezeichnet die Gesamtheit der Rechte, die diese individuellen geistigen Leistungen schützen, wie das Patent- und Gebrauchsmusterrecht in Bezug auf Erfindungen oder das Urheberrecht in Bezug auf Werke der Wissenschaft, Literatur und Kunst (**einschließlich Software**).“*

[Deutsches Patent- und Markenamt](#)

Warum Lizenzierung?

Entstehung von geistigem Eigentum

- Bei der Analyse von Geodaten entsteht Code, welcher Logik enthält
- Diese Logik ist durch Denken entstanden und somit als geistiges Eigentum zu werten
- Oftmals sind es projektspezifische, individuelle Arbeitsflüsse
- Manchmal aber sind es allgemeingültige oder übertragbare Konzepte die entstehen
- Diese könnte man in einer Software zusammenfassen

Der Schutz geistigen Eigentums wird durch die Lizenzierung geregelt!

Warum Lizenzierung?

Verwendung von geistigem Eigentum anderer

- Meistens wird das geistige Eigentum anderer verwendet, z.B. Python und seine Pakete bei der Geodatenanalyse!
- Bei der Verwendung muss man unbedingt beachten, wie dieses Eigentum geschützt wurde
- Eine Verwendung könnte z.B. für bestimmte Bedingungen begrenzt sein
- Hier sollte man unbedingt nachprüfen welche Nutzungsbedingungen gegeben sind

Vor der Verwendung von geistigen Eigentum muss der Schutz geprüft werden

Software ohne Lizenz

Was passiert bei Software ohne Lizenz?

- Das Fehlen einer Lizenz bedeutet, dass die generellen Urheberrechtsgesetze gelten
- Hier greift dann das Gesetz über Urheberrecht und verwandte Schutzrechte (Urheberrechtsgesetz) ein
- Da Software geistiges Eigentum ist bedeutet dies, dass der/die Entwickler/in der Software (oder sein/ihr Arbeitgeber/in) alle Rechte am Code haben

Software ohne Lizenz

Gesetz über Urheberrecht und verwandte Schutzrechte (Urheberrechtsgesetz)

§ 2 Geschützte Werke

(1) Zu den geschützten Werken der Literatur, Wissenschaft und Kunst gehören insbesondere:

1. Sprachwerke, wie Schriftwerke, Reden und **Computerprogramme**;
2. Werke der Musik;
3. pantomimische Werke einschließlich der Werke der Tanzkunst;
4. Werke der bildenden Künste einschließlich der Werke der Baukunst und der angewandten Kunst und Entwürfe solcher Werke;
5. Lichtbildwerke einschließlich der Werke, die ähnlich wie Lichtbildwerke geschaffen werden;
6. Filmwerke einschließlich der Werke, die ähnlich wie Filmwerke geschaffen werden;
7. Darstellungen wissenschaftlicher oder technischer Art, wie Zeichnungen, Pläne, Karten, Skizzen, Tabellen und plastische Darstellungen.

(2) Werke im Sinne dieses Gesetzes sind nur persönliche geistige Schöpfungen.

Quelle: https://www.gesetze-im-internet.de/urhg/_2.html

Software ohne Lizenz

Gesetz über Urheberrecht und verwandte Schutzrechte (Urheberrechtsgesetz)

§ 69a Gegenstand des Schutzes

- (1) Computerprogramme im Sinne dieses Gesetzes sind Programme in jeder Gestalt, einschließlich des Entwurfsmaterials.
- (2) Der gewährte Schutz gilt für alle Ausdrucksformen eines Computerprogramms. Ideen und Grundsätze, die einem Element eines Computerprogramms zugrunde liegen, einschließlich der den Schnittstellen zugrundeliegenden Ideen und Grundsätze, sind nicht geschützt.
- (3) Computerprogramme werden geschützt, wenn sie individuelle Werke in dem Sinne darstellen, daß sie das Ergebnis der eigenen geistigen Schöpfung ihres Urhebers sind. Zur Bestimmung ihrer Schutzfähigkeit sind keine anderen Kriterien, insbesondere nicht qualitative oder ästhetische, anzuwenden.
- (4) Auf Computerprogramme finden die für Sprachwerke geltenden Bestimmungen Anwendung, soweit in diesem Abschnitt nichts anderes bestimmt ist.
- (5) Die Vorschriften der §§ 32d, 32e, 36 bis 36c, 40a und 95a bis 95d finden auf Computerprogramme keine Anwendung.

Quelle: https://www.gesetze-im-internet.de/urhg/_69a.html

Software ohne Lizenz

Wie kann ich unlizenzierte Software verwenden?

- **Bitten Sie den Urheber höflich, eine Lizenz hinzuzufügen.** Wenn die Software keine starken Hinweise auf das Gegenteil enthält, ist das Fehlen einer Lizenz wahrscheinlich ein Versehen. Wenn die Software auf einer Seite wie GitHub gehostet wird, öffnen Sie ein Ticket mit der Bitte um eine Lizenz und fügen Sie einen Link zu dieser Seite hinzu. Wenn Sie mutig sind und es ziemlich offensichtlich ist, welche Lizenz am besten geeignet ist, öffnen Sie eine Pull-Anfrage um eine Lizenz hinzuzufügen - siehe "Diese Lizenz vorschlagen" in der Seitenleiste der Seite für jede Lizenz auf dieser Seite (z.B. MIT).
- **Verwenden Sie die Software nicht.** Finden oder erstellen Sie eine Alternative, die unter einer Open-Source-Lizenz steht.
- **Verhandeln Sie eine private Lizenz.** Bringen Sie Ihren Anwalt mit.

Quelle: <https://choosealicense.com/no-permission/>

Lizenzierungsmöglichkeiten

- Generell könnte jeder Urheber eine eigene Lizenz aufsetzen
- Allerdings ist das Thema und das zugehörige Recht dazu kompliziert
- Vor allem wenn man international denkt
- Eine einfache Möglichkeit bieten Standardlizenzen
- Diese werden mit verschiedenen Möglichkeiten angeboten

Standardlizenzen

- Generell könnte jeder Urheber eine eigene Lizenz aufsetzen
- Allerdings ist das Thema und das zugehörige Recht dazu kompliziert
- Vor allem wenn man international denkt
- Eine einfache Möglichkeit bieten Standardlizenzen
- Diese werden mit verschiedenen Möglichkeiten angeboten

Auswahl einer Lizenz

Choose an open source license

An open source license protects contributors and users. Businesses and savvy developers won't touch a project without this protection.

Which of the following best describes your situation?



I need to work in a community.

Use the [license preferred by the community](#) you're contributing to or depending on. Your project will fit right in.

If you have a dependency that doesn't have a license, ask its maintainers to [add a license](#).



I want it simple and permissive.

The [MIT License](#) is short and to the point. It lets people do almost anything they want with your project, like making and distributing closed source versions.

[Babel](#), [.NET Core](#), and [Rails](#) use the MIT License.



I care about sharing improvements.

The [GNU GPLv3](#) also lets people do almost anything they want with your project, *except* distributing closed source versions.

[Ansible](#), [Bash](#), and [GIMP](#) use the GNU GPLv3.

What if none of these work for me?

My project isn't software.

[There are licenses for that](#)

I want more choices.

[More licenses are available.](#)

I don't want to choose a license.

[Here's what happens if you don't](#)

<https://choosealicense.com>

Was ist Open Source?

- Open Source ist Quellcode, der für mögliche Änderungen und Weiterverbreitung frei verfügbar gemacht wird
- Produkte beinhalten die Erlaubnis, den Quellcode, die Designdokumente oder den Inhalt des Produkts zu verwenden
- Der Begriff bezieht sich meist auf das Open-Source-Modell, bei dem Open-Source-Software oder andere Produkte als Teil der Open-Source-Software-Bewegung unter einer Open-Source-Lizenz veröffentlicht werden
- Der Begriff wurde ursprünglich für Software verwendet, hat sich aber über den Softwarebereich hinaus auf andere offene Inhalte und Formen der offenen Zusammenarbeit ausgeweitet

Quelle: [Wikipedia](#)

■ Findable

The first step in (re)using data is to find them. Metadata and data should be easy to find for both humans and computers. Machine-readable metadata are essential for automatic discovery of datasets and services, so this is an essential component of the **FAIRification process**.

- F1. (Meta)data are assigned a globally unique and persistent identifier
- F2. Data are described with rich metadata (defined by R1 below)
- F3. Metadata clearly and explicitly include the identifier of the data they describe
- F4. (Meta)data are registered or indexed in a searchable resource

■ Accessible

Once the user finds the required data, she/he needs to know how can they be accessed, possibly including authentication and authorisation.

- A1. (Meta)data are retrievable by their 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 authorisation procedure, where necessary
- A2. Metadata are accessible, even when the data are no longer available

■ **Interoperable**

The data usually need to be integrated with other data. In addition, the data need to interoperate with applications or workflows for analysis, storage, and processing.

- I1. (Meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.
- I2. (Meta)data use vocabularies that follow FAIR principles
- I3. (Meta)data include qualified references to other (meta)data

Wilkinson et al. (2016) The FAIR Guiding Principles for scientific data management and stewardship, *Scientific Data*,
<https://doi.org/10.1038/sdata.2016.18>

FAIR Prinzipien

■ Reusable

The ultimate goal of FAIR is to optimise the reuse of data. To achieve this, metadata and data should be well-described so that they can be replicated and/or combined in different settings.

- R1. (Meta)data are richly described with a plurality of accurate and relevant attributes
 - R1.1. (Meta)data are released with a clear and accessible data usage license
 - R1.2. (Meta)data are associated with detailed provenance
 - R1.3. (Meta)data meet domain-relevant community standards

Datenethik

Datenethik

■ Was ist Datenethik?

„Datenethik, ist ein neuer Zweig der Ethik. Sie untersucht und bewertet moralische Probleme im Zusammenhang mit Daten (einschließlich Generierung, Aufzeichnung, Kuratierung, Verarbeitung, Verbreitung, Austausch und Nutzung), Algorithmen (einschließlich künstlicher Intelligenz, künstlicher Agenten, maschinellern Lernen und Robotern) und entsprechenden Praktiken (einschließlich verantwortungsvoller Innovation, Programmierung, Hacking und Berufskodizes), um moralisch gute Lösungen (z. B. richtige Verhaltensweisen oder richtige Werte) zu formulieren und zu unterstützen. Die Datenethik baut auf den Grundlagen der Computer- und Informationsethik auf, verfeinert aber gleichzeitig den Ansatz, indem sie die Abstraktionsebene ethischer Fragestellungen von der Informations- zur Datenzentriertheit verschiebt.“

Floridi and Taddeo (2016) <https://doi.org/10.1098/rsta.2016.0360>

■ Warum Datenethik?

„Die Datenwissenschaft bietet enorme Möglichkeiten, das private und öffentliche Leben sowie unsere Umwelt zu verbessern (man denke an die Entwicklung von Smart Cities oder die Probleme, die durch Kohlenstoffemissionen entstehen). Leider sind solche Möglichkeiten auch mit erheblichen ethischen Herausforderungen verbunden. Die umfassende Nutzung von immer mehr Daten - oft persönlichen, wenn nicht gar sensiblen Daten (Big Data) - und die zunehmende Abhängigkeit von Algorithmen zu deren Analyse, um Entscheidungen zu treffen (einschließlich maschinellem Lernen, künstlicher Intelligenz und Robotik), sowie die allmähliche Verringerung der menschlichen Beteiligung oder sogar der Aufsicht über viele automatische Prozesse, werfen unter anderem drängende Fragen der Fairness, Verantwortung und Achtung der Menschenrechte auf.“

Floridi and Taddeo (2016) <https://doi.org/10.1098/rsta.2016.0360>

Datenethik Beispiele

■ Google Street View

- Frage: Ist es OK, wenn man ungefragt Aufnahmen von Personen macht und diese öffentlich zugänglich bereit stellt?

■ Fernerkundung

- Frage: Ist es OK, wenn Daten von in Flugzeugen oder Satelliten befindlichen Sensoren persönliche Informationen enthalten oder verfügbar machen?

■ Geodaten

- Frage: Ist es OK, Geodaten aus kommerziellen Interessen unter Verschluss zu halten?

Datenethikkommission

Bundesministerium des Innern, für Bau und Heimat



- **Auftrag:** *“...auf der Basis wissenschaftlicher und technischer Expertise ethische Leitlinien für den Schutz des Einzelnen, die Wahrung des gesellschaftlichen Zusammenlebens und die Sicherung und Förderung des Wohlstands im Informationszeitalter entwickeln.“*
- Die Datenethikkommission hat am 4. September 2018 ihre Arbeit aufgenommen

Datenethik für Geodaten

- Was sagt die Wissenschaft?
- Es gibt nur sehr wenig Literatur zu diesem Thema
- Aus dem Jahr 2012:

„Der Artikel kommt zu dem Schluss, dass es keine absolute ethische Position in Bezug auf die Erdbeobachtung gibt, sondern eine Abhängigkeit von der Perspektive des Beobachters.“

Logan Cochrane, Jon Corbett, Mike Evans, Mark Gill. (2017) Searching for social justice in GIScience publications. Cartography and Geographic Information Science 44:6, pages 507-520, <https://doi.org/10.1080/01431161.2012.718466>

