

**„Command not found“  
Der steinige Weg zu Reproduzierbarkeit  
in der quantitativen Biogeographie**

Wolfgang Traylor

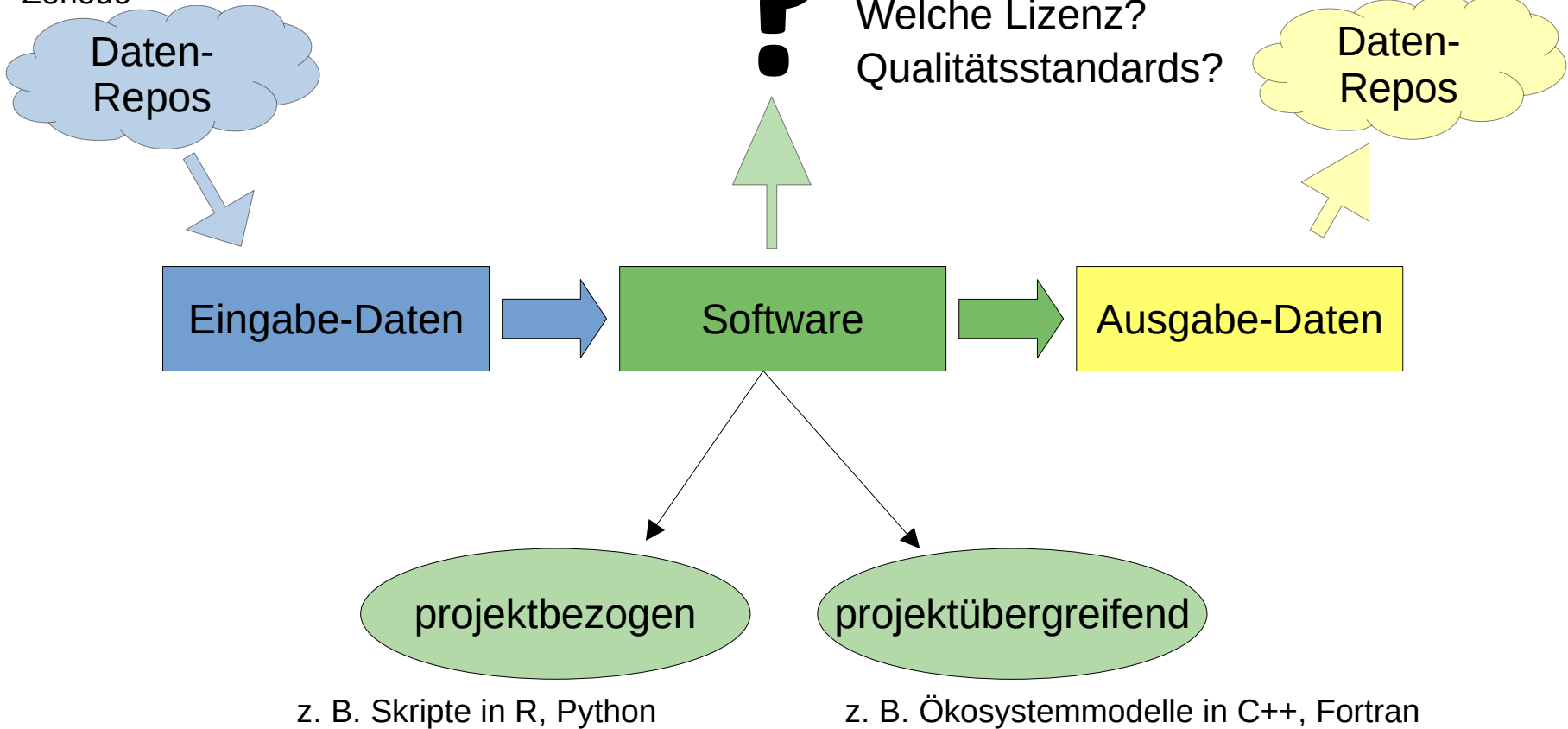
Senckenberg Biodiversitäts- und Klimaforschungszentrum  
Frankfurt am Main

```
bash-5.1$ ./run_simulation.sh
./run_simulation.sh: line 3: python2: command not found
bash-5.1$ █
```

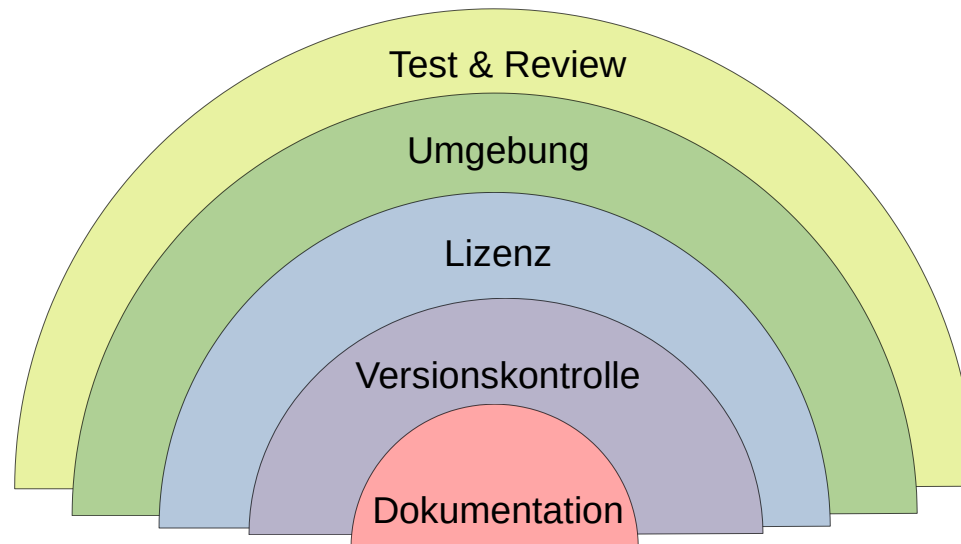
## Wie sieht ein Projekt aus?

z. B. PANGAEA, EPD,  
DataDryad, FigShare,  
Zenodo

Wo archivieren?  
Welche Struktur?  
Wie reproduzierbar?  
Welche Lizenz?  
Qualitätsstandards?



## Vorschlag für ein Rahmenwerk



## Basic Documentation

### README.md

```
# SUSTAIN Niche Modeling

## Authors

...

## Overview

...

## Project Structure

...

## Usage (How to Reproduce)

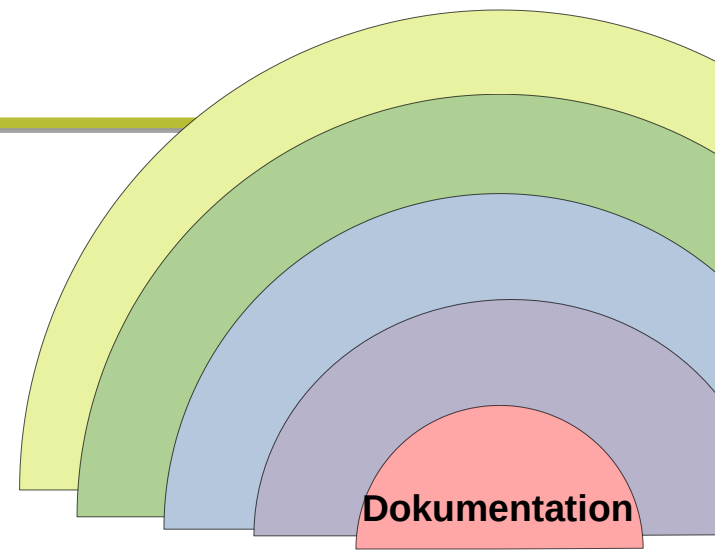
...
```

### JOURNAL.txt

```
2022-09-15:
...

2022-09-16:
...

2022-09-25:
...
```



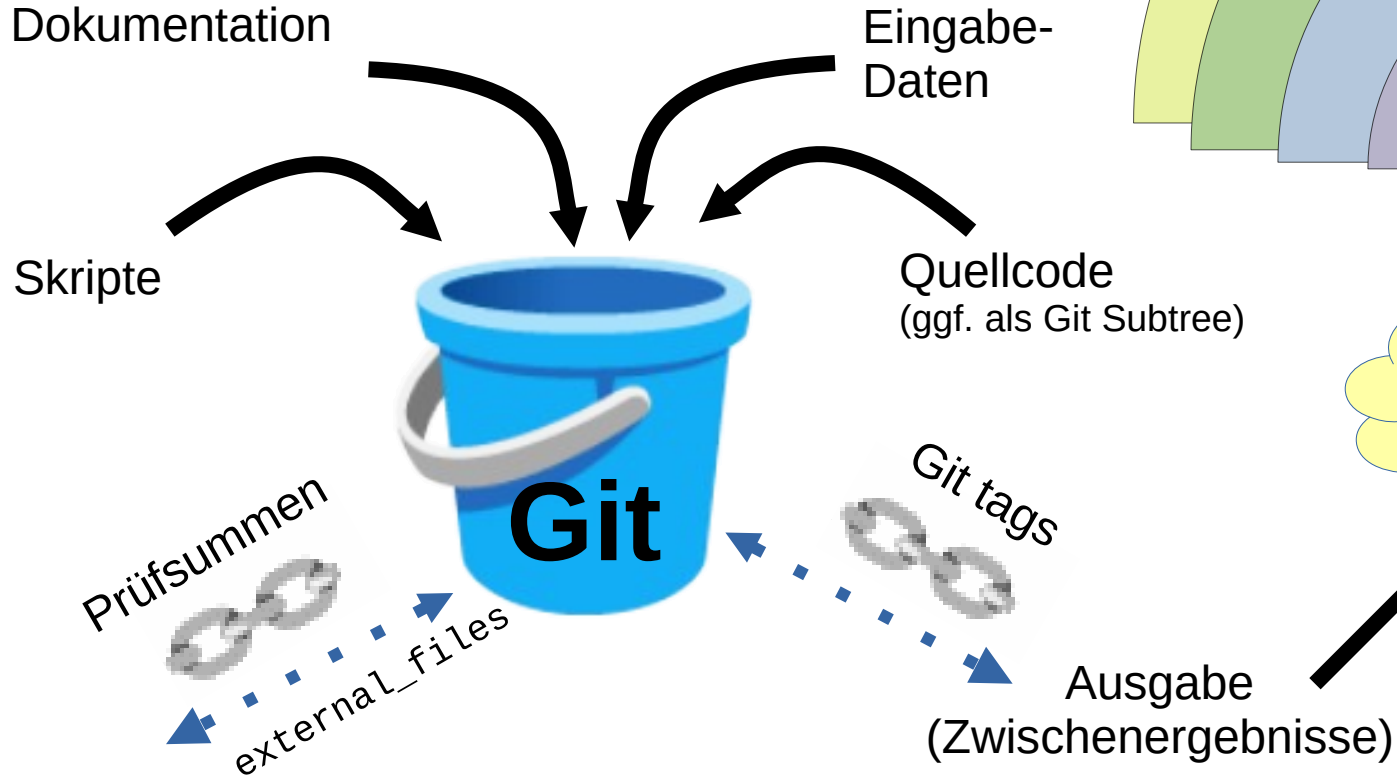
### codemeta.json

```
{
  "@context": "https://doi.org/10.5063/schema/codemeta-2.0",
  "@type": "SoftwareSourceCode",
  "license": "https://spdx.org/licenses/LGPL-3.0-or-later",
  "codeRepository":
    "git+https://github.com/wtraylor/modular_megafauna_model.git",
  "contIntegration":
    "https://gitlab.com/wtraylor/modular_megafauna_model/pipelines",
  "datePublished": "2021-04-26",
  "dateModified": "2022-07-01",
  "issueTracker":
    "https://github.com/wtraylor/modular_megafauna_model/issues/",
  "name": "Modular Megafauna Model (MMM)",
  "version": "1.1.4",
  "identifier": "10.5281/zenodo.4710254.",
  ...
}
```

### CITATION.cff

```
cff-version: 1.1.0
message: If you use this software, please cite it as below.
authors:
  - family-names: Traylor
    given-names: Wolfgang
    affiliation: Senckenberg Biodiversity and Climate Research
      Centre
    orcid: https://orcid.org/0000-0002-4813-1072
  title: "Modular Megafauna Model"
  version: 1.1.4
  doi: 10.5281/zenodo.4710254
  date-released: 2022-07-01
  license: LGPL-3.0-or-later
  repository-code:
    https://github.com/wtraylor/modular_megafauna_model
```

Alles in einen Pott!



große Dateien,  
Dateien ohne freie Lizenz

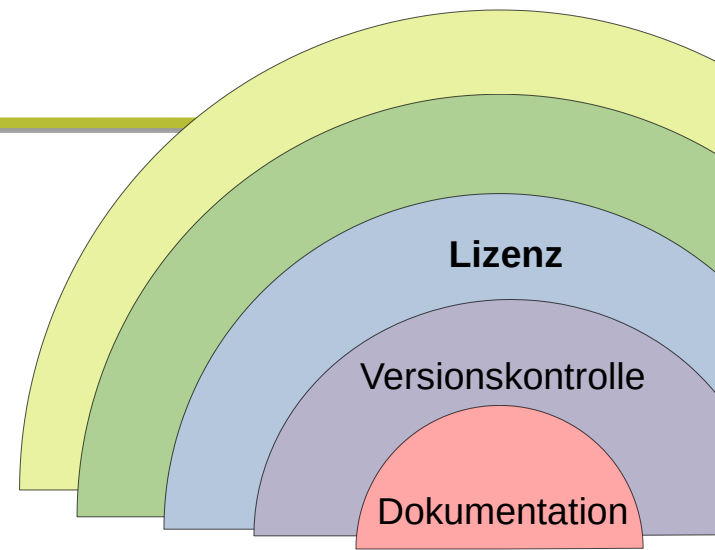
## Lizenzierung

- Vielfalt an Dateien  
⇒ Individuelle Lizenz-Angaben  
*pro Datei*
- Meine Empfehlung:
  - CC-BY für Texte, Bilder, usw.
  - MIT für Skripte
  - (L)GPL für projektunabhängige Software
  - CC0 für alles ohne besonderen „Wert“

### REUSE-Kopfzeile

```
<!--
SPDX-FileCopyrightText: 2022 Wolfgang Traylor
<wolfgang.traylor@senckenberg.de>

SPDX-License-Identifier: CC0-1.0
-->
```



**REUSE**  
SOFTWARE

<https://reuse.software/>

## Reproduzierbare Ausführungsumgebung



Komplettes Betriebssystem  
virtualisieren

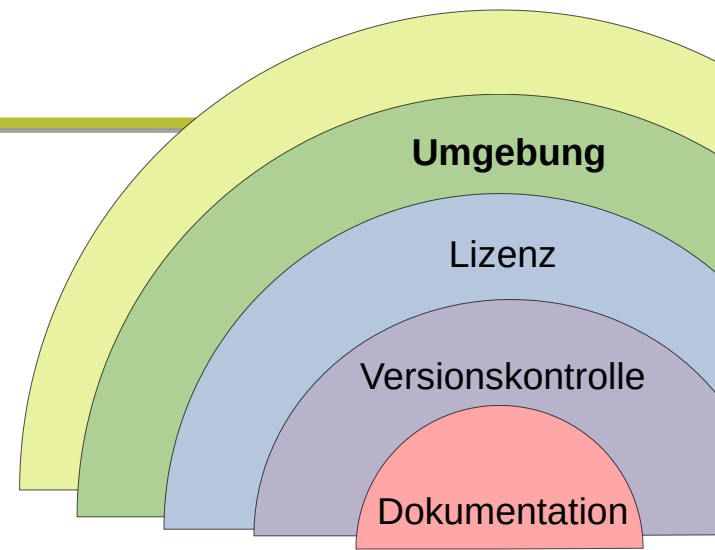
(z. B. Apptainer, Docker, ...)

Umgebung automatisch  
wiederherstellen

(z. B. Packrat, Conda, ...)

Abhängigkeiten dokumentieren

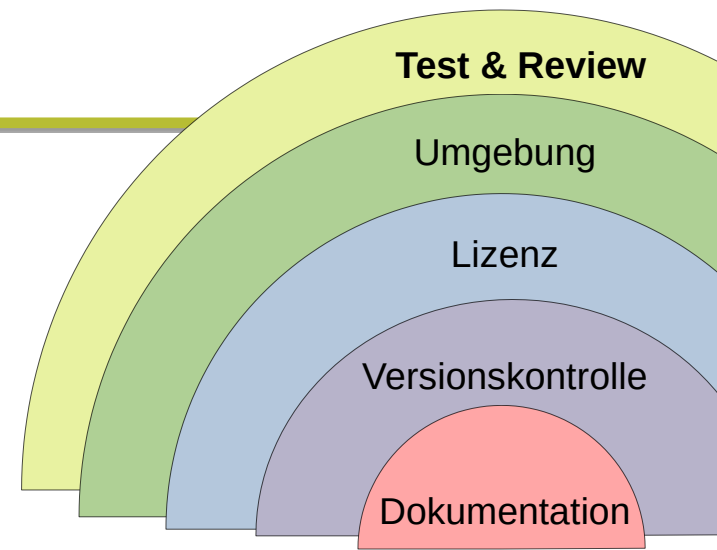
(z. B. sessionInfo(), inxi, ...)

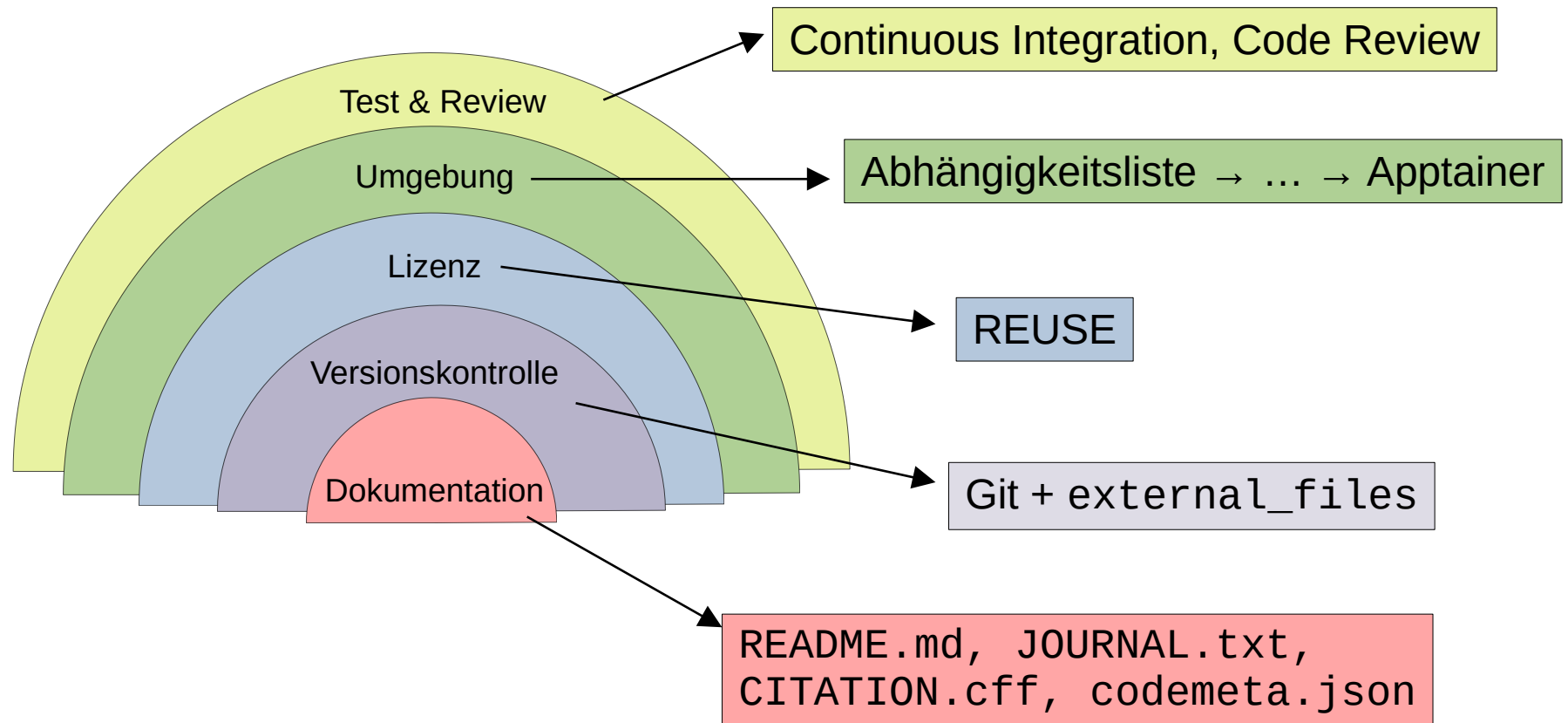




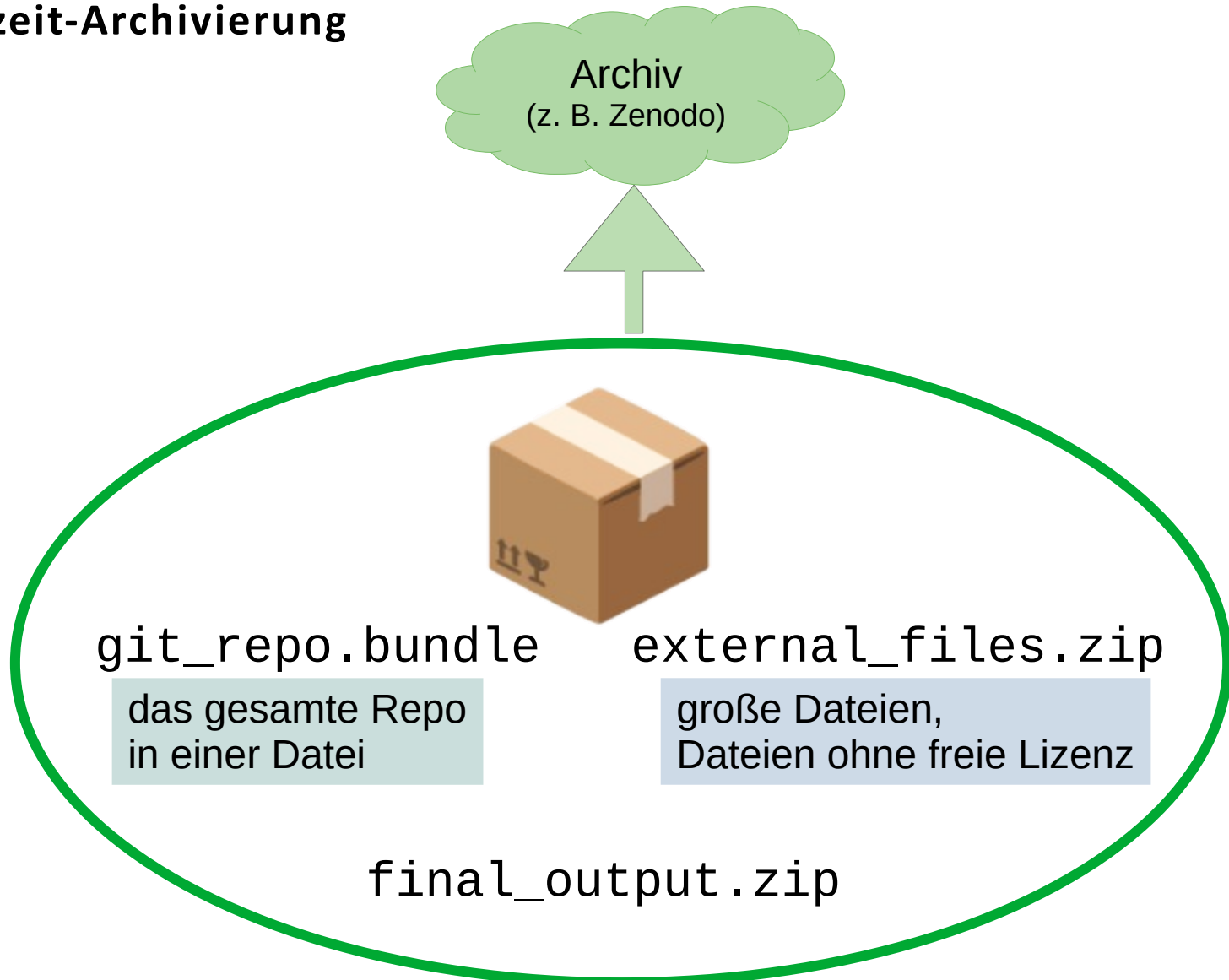
## Programmierfehler finden

- Code Review findet bei uns quasi nicht statt.
- Continuous Integration zur automatischen Fehlersuche





## Zum Ende des Projekts: Langzeit-Archivierung



## Steine im Weg

zu wenig  
Teamarbeit?

Wasserfall statt  
Agilität?

zu steile  
Lernkurve?

Reproduzierbarkeit  
überbewertet?



keine Vorbilder,  
kein „Gruppenzwang“?

wenig Anreiz?  
⇒ keine Zeit!

keine Standards  
⇒ Verwirrung?

Danke für die  
Aufmerksamkeit!