Publikation und Zitation von Forschungssoftware

Stephan Druskat
Deutsches Zentrum für Luft- und Raumfahrt (DLR)
Institut für Softwaretechnologie
Sustainable Software Engineering Group
Open Up! Webinar 31.05.2022
HdM Stuttgart

License CC-BY-4.0 International
DOI 10.5281/zenodo.6598444

DLR

Wer/was bin ich?

- 1999-2019: • Geisteswissenschaften
- 2009-2022: • Research Software Engineering (Linguistik)
- Seit 2016: • Nachhaltigkeit von Forschungssoftware
- Seit 2017: • Softwarezitierung
- Seit 2019: • Software Engineering

- Fellow des Software Sustainability Institute
- Mitbegründer von de-RSE e.V. – Gesellschaft für Forschungssoftware
- Project Lead im Citation File Format Projekt
- Principal Investigator HERMES
- Doktorand (Informatik), DLR und Humboldt-Universität zu Berlin

Überblick

- Einführung
  - Nachhaltigkeit von Forschungssoftware
  - Forschungssoftware definieren und identifizieren
- Zitierung von Forschungssoftware
- Publikation von Forschungssoftware

Hintergrund: Nachhaltigkeit von Forschungssoftware

Nachhaltigkeit**en**:

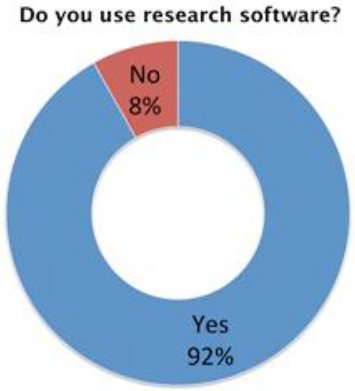
- Software und die 17 Sustainable Development Goals der UN (u.a. [1])
- **Nachhaltigkeit von (Forschungs-) Software**
- Nachhaltigkeit von (Forschungs-) Softwareinfrastrukturen

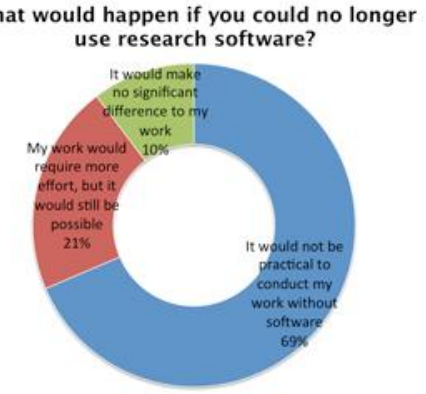
The capacity of the software to endure and to continue to be available in the future, on new platforms, meeting new needs.

Daniel S. Katz

Warum Nachhaltigkeit von Forschungssoftware?

- Software ist zentraler Bestandteil von Forschung! [2]



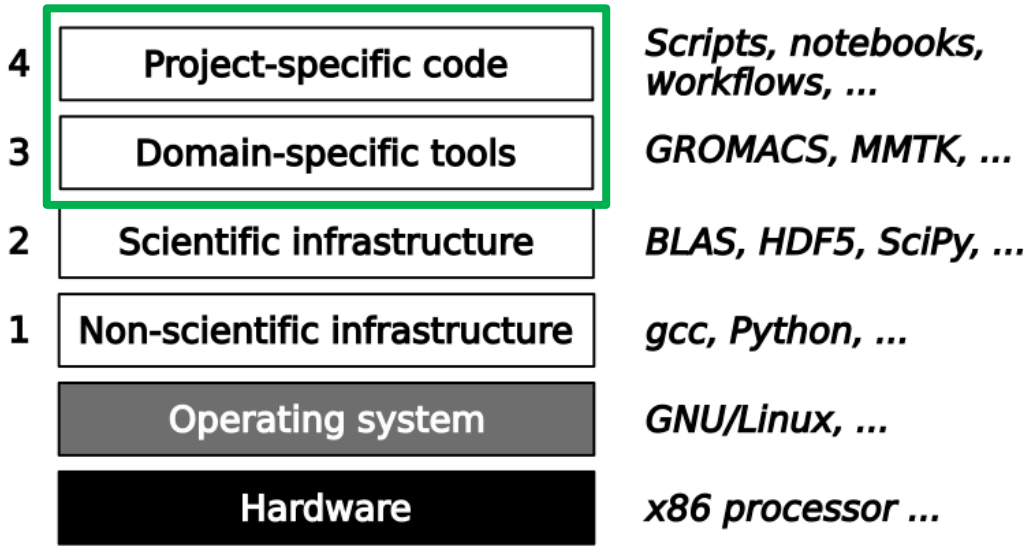


2014, aus [3], 417 Befragte an 15 Russel Group Universitäten, UK.

- Reproduzierbarkeit
- Forschungsperformanz, Ressourcen
- Nachhaltigkeit für wen?

Forschungssoftware?

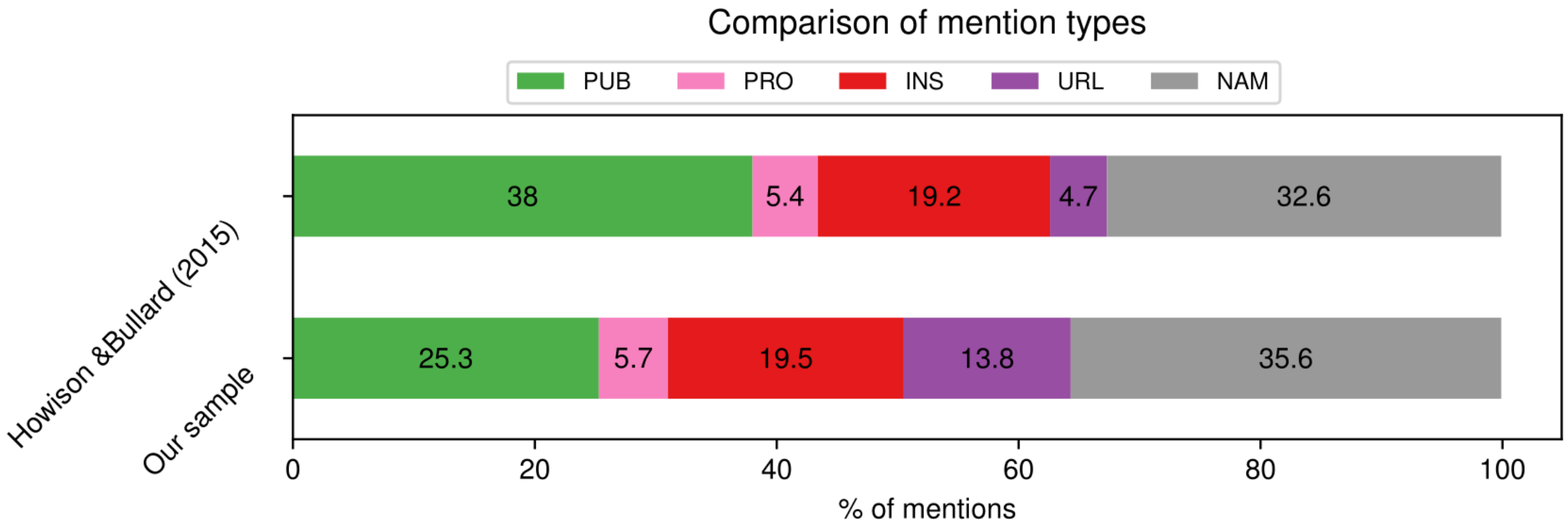
- Teil des „scientific software stack“ [4]



- Mindestens: „**Software, die von Forschenden (direkt) verwendet wird**“

- Möglichkeiten des Nachweises
  - Bereitstellung durch Autor*innen (Publikation)
  - Kuratierte Verzeichnisse (Katalogisierung)
    - Disziplinspezifische Kataloge (ASCL, swMATH)
    - Institutionelle Kataloge (Research Software Directory)
    - Generische Sammlungen (Research Software Encyclopedia)
  - Data Mining und Analyse
    - Zitationsanalyse (offene Zitationen)

Identifizierung durch Zitierung von Forschungssoftware



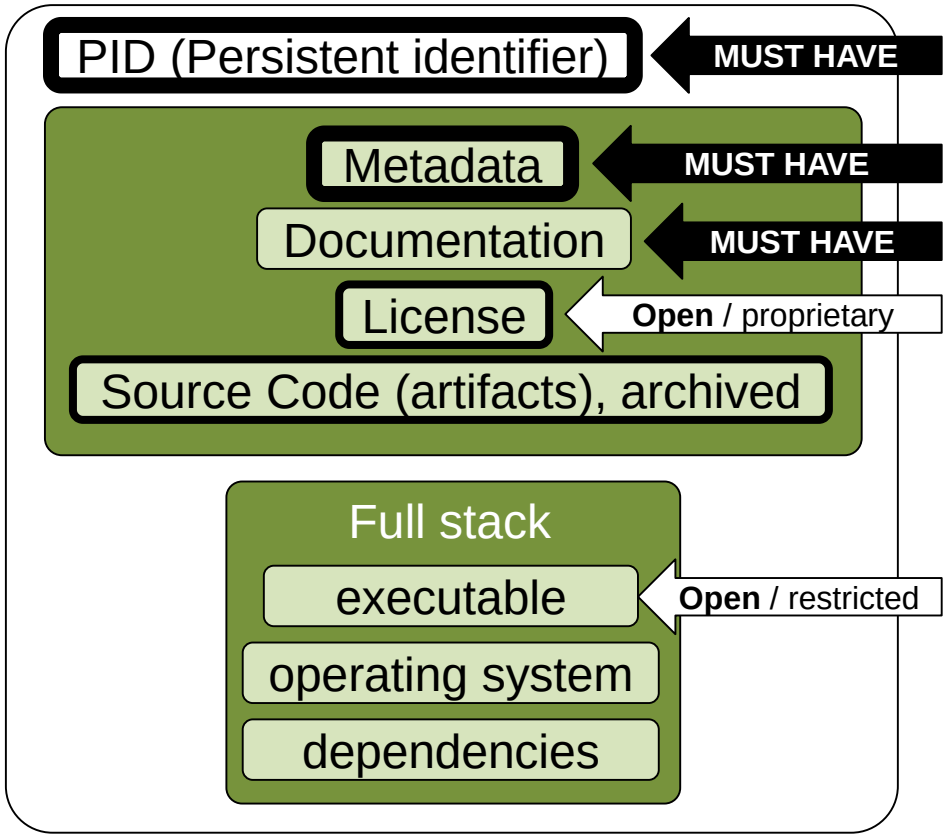
Aus [6], forthcoming. Nennungstypen von Software in wissenschaftlichen Pubblikationen in %, in Daten aus [5] und eigenen Daten.

Identifizierung durch FAIRe Softwarepublikation

- **F**indability
- **A**ccessibility
- **I**nteroperability
- **R**eusability

- Nutzen für Nachhaltigkeit
- Nutzen der „Marke“ FAIR

FAIR Software, open source, nachhaltig

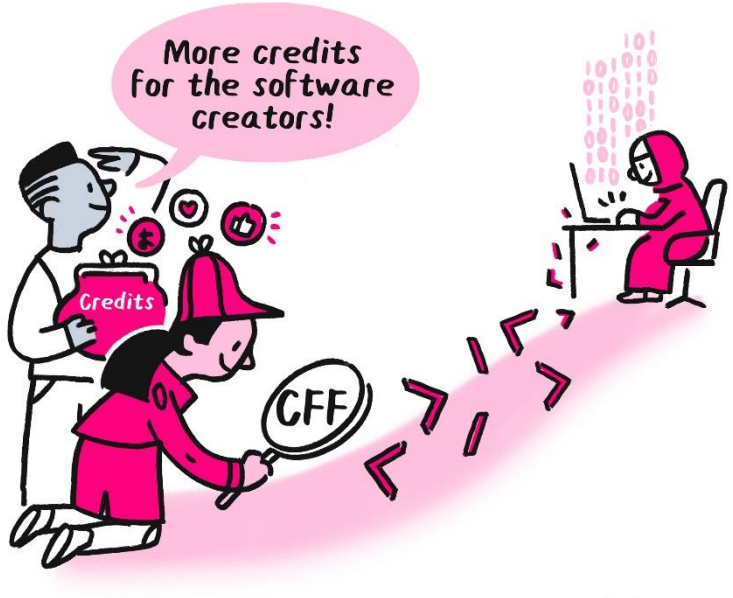


Nach [7], Erratum.

Zitierung von Forschungssoftware

Identifizierung und mehr: Zitierung von Forschungssoftware

- Attribution und Anerkennung für Research Software Engineers
- potenziell bessere Nachhaltigkeit von Forschungssoftware [8]
- bessere Reproduzierbarkeit von Forschungsergebnissen



More credits for the software creators. The Turing Way project illustration by Scriberia. Zenodo. https://doi.org/10.5281/zenodo.3332807
License: CC BY-4.0

Prinzipien der Softwarezitierung [9]

PeerJ Computer Science

Software citation principles

Arfon M. Smith[1,*], Daniel S. Katz[2,*], Kyle E. Niemeyer[3,*] and FORCE11 Software Citation Working Group

[1] GitHub, Inc., San Francisco, California, United States
[2] National Center for Supercomputing Applications & Electrical and Computer Engineering Department & School of Information Sciences, University of Illinois at Urbana-Champaign, Urbana, Illinois, United States
[3] School of Mechanical, Industrial, and Manufacturing Engineering, Oregon State University, Corvallis, Oregon, United States
* These authors contributed equally to this work.

ABSTRACT

Software is a critical part of modern research and yet there is little support across the scholarly ecosystem for its acknowledgement and citation. Inspired by the activities of the FORCE11 working group focused on data citation, this document summarizes the recommendations of the FORCE11 Software Citation Working Group and its activities between June 2015 and April 2016. Based on a review of existing community practices, the goal of the working group was to produce a consolidated set of citation principles that may encourage broad adoption of a consistent policy for software citation across disciplines and venues. Our work is presented here as a set of software citation principles, a discussion of the motivations for developing the principles, reviews of existing community practice, and a discussion of the requirements these principles would place upon different stakeholders. Working examples and possible technical solutions for how these principles can be implemented will be discussed in a separate paper.

Subjects Digital Libraries, Software Engineering
Keywords Software citation, Software credit, Attribution

SOFTWARE CITATION PRINCIPLES

The main contribution of this document are the software citation principles, written fairly concisely in this section and discussed further later in the document (see Discussion). In addition, we also motivate the creation of these principles (see Motivation), describe the process by which they were created (see Process of Creating Principles), summarize use cases related to software citation (see Use Cases), and review related work (see Related Work). We also lay out the work needed to lead to these software citation principles being applied (see Future Work).

1. **Importance:** Software should be considered a legitimate and citable product of research. Software citations should be accorded the same importance in the scholarly record as citations of other research products, such as publications and data; they should be included in the metadata of the citing work, for example in the reference list of a journal article, and should not be omitted or separated. Software should be cited on the same basis as any other research product such as a paper or a book, that is, authors should cite the appropriate set of software products just as they cite the appropriate set of papers.

Submitted 24 June 2016
Accepted 23 August 2016
Published 19 September 2016

Corresponding author
Daniel S. Katz, d.katz@ieee.org

Academic editor
Silvio Peroni

DOI 10.7717/peerj-cs.86

Copyright
2016 Smith et al.

Distributed under
Creative Commons CC-BY 4.0

OPEN ACCESS

How to cite this article Smith et al. (2016), Software citation principles. Peerj Comput. Sci. 2:e86; DOI 10.7717/peerj-cs.86

1. **Importance**

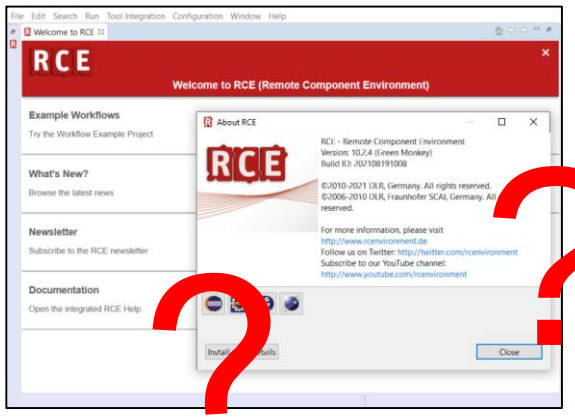
   Software wird genauso zitiert wie ein Paper.
2. Credit and attribution
3. Unique identification
4. Persistence
5. **Accessibility**

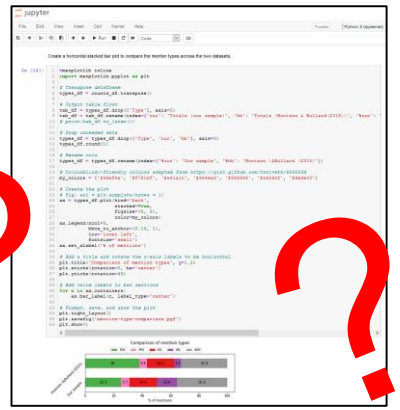
   Zitierung ermöglicht den Zugang zu Software und ihren Metadaten.
6. **Specificity**

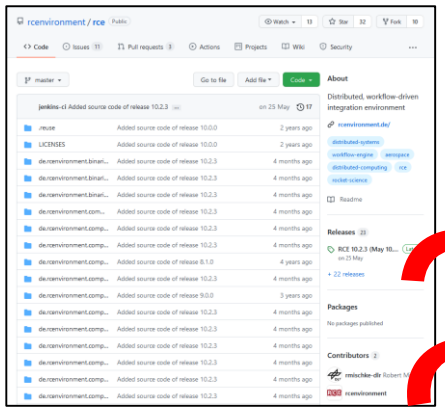
   Zitation identifiziert die verwendete Softwareversion.

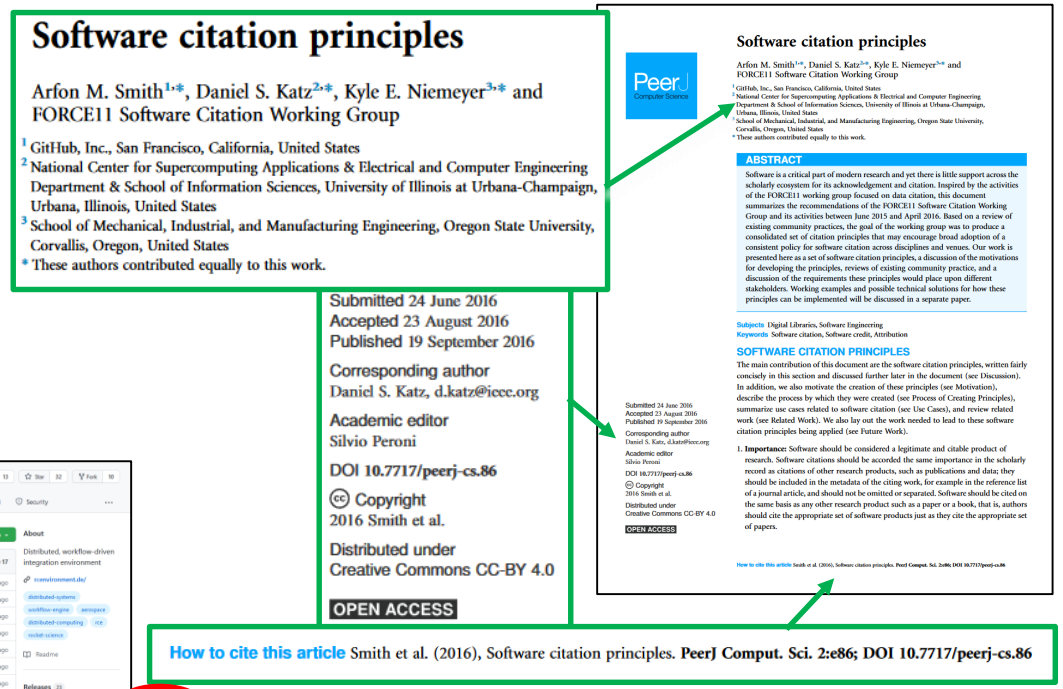
Praxis der Softwarezitierung: zitieren und zitiert werden

- Software zitieren nach den Software Citation Principles!
- Problem (für beide Seiten): Zitationsmetadaten!
  - Name der Software?
  - Autor*innen?
  - Verlag?
  - Veröffentlichungsdatum?









Praxis der Softwarezitierung: das Citation File Format (CFF)

https://citation-file-format.github.io/

- autoritativ
- kontrollierbar
- „prinzipientreu“:

1. Importance ✓
2. Credit and attribution ✓
3. Unique identification* ✓
4. Persistence* ✓
5. Accessibility** ✓
6. Specificity** ✓

```yaml
cff-version: 1.2.0
message: If you use this software, please cite it using these metadata.
title: My Research Software
abstract: This is my awesome research software. It does many things.
authors:
  - family-names: Druskat
    given-names: Stephan
    orcid: "https://orcid.org/0000-0003-4925-7248"
version: 0.11.2
date-released: "2021-07-18"
identifiers:
  - description: This is the collection of archived snapshots of all versions of My Research Software
    type: doi
    value: "10.5281/zenodo.123456"
  - description: This is the archived snapshot of version 0.11.2 of My Research Software
    type: doi
    value: "10.5281/zenodo.123457"
license: Apache-2.0
repository-code: "https://github.com/citation-file-format/my-research-software"
```

METADATEN: In einer **`CITATION.cff`**-Datei im Source Code Repository

Praxis der Softwarezitierung: das Citation File Format (CFF) - Details

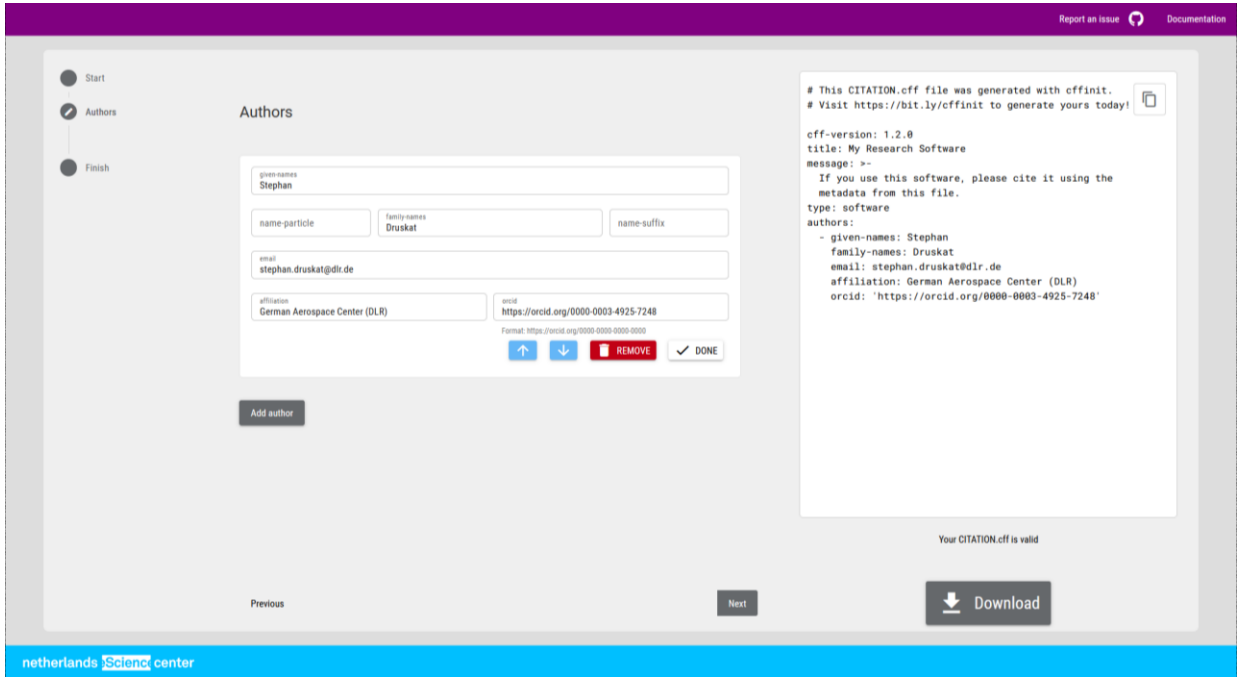
Aber im wissenschaftlichen Reporting zählen nur Papers!

```yaml
preferred-citation:
  authors:
    - family-names: Druskat
      given-names: Stephan
  title: "Software paper about My Research Software"
  type: article
title: "My Research Software"
```

Software wird genauso zitiert wie ein Paper … … auch von Software?

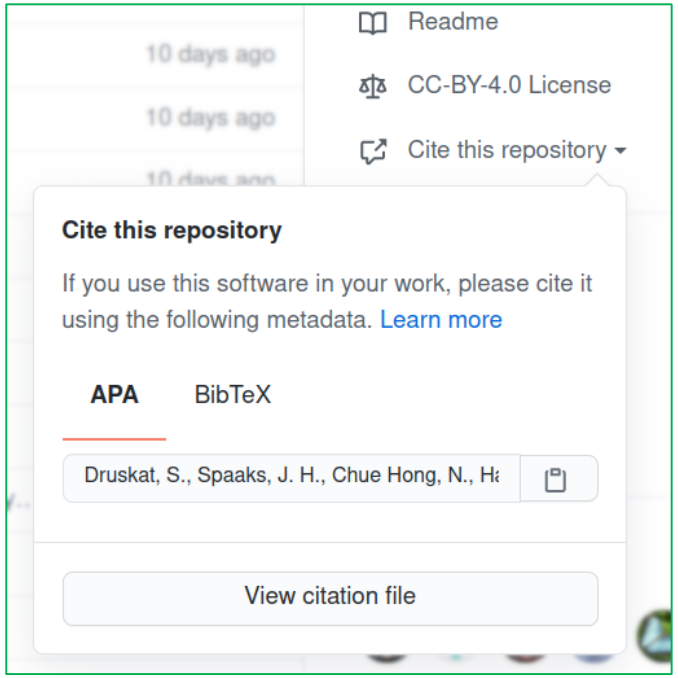
```yaml
references:
  - authors:
      - family-names: Spaaks
        given-names: "Jurriaan H."
    title: "The foundation of Research Software"
    type: software
  - authors:
      - family-names: Haines
        given-names: Robert
    title: "Ruby CFF Library"
    type: software
    version: 1.0
```

Praxis der Softwarezitierung: das Citation File Format (CFF) - Tooling



cffinit: Webtool für die Erstellung von CFF-Dateien
https://bit.ly/cffinit

- Bearbeiten & aktualisieren
- Validieren
- Konvertieren
- GitHub-Integration



Publikation von Forschungssoftware

Publikation? Von Forschungssoftware?

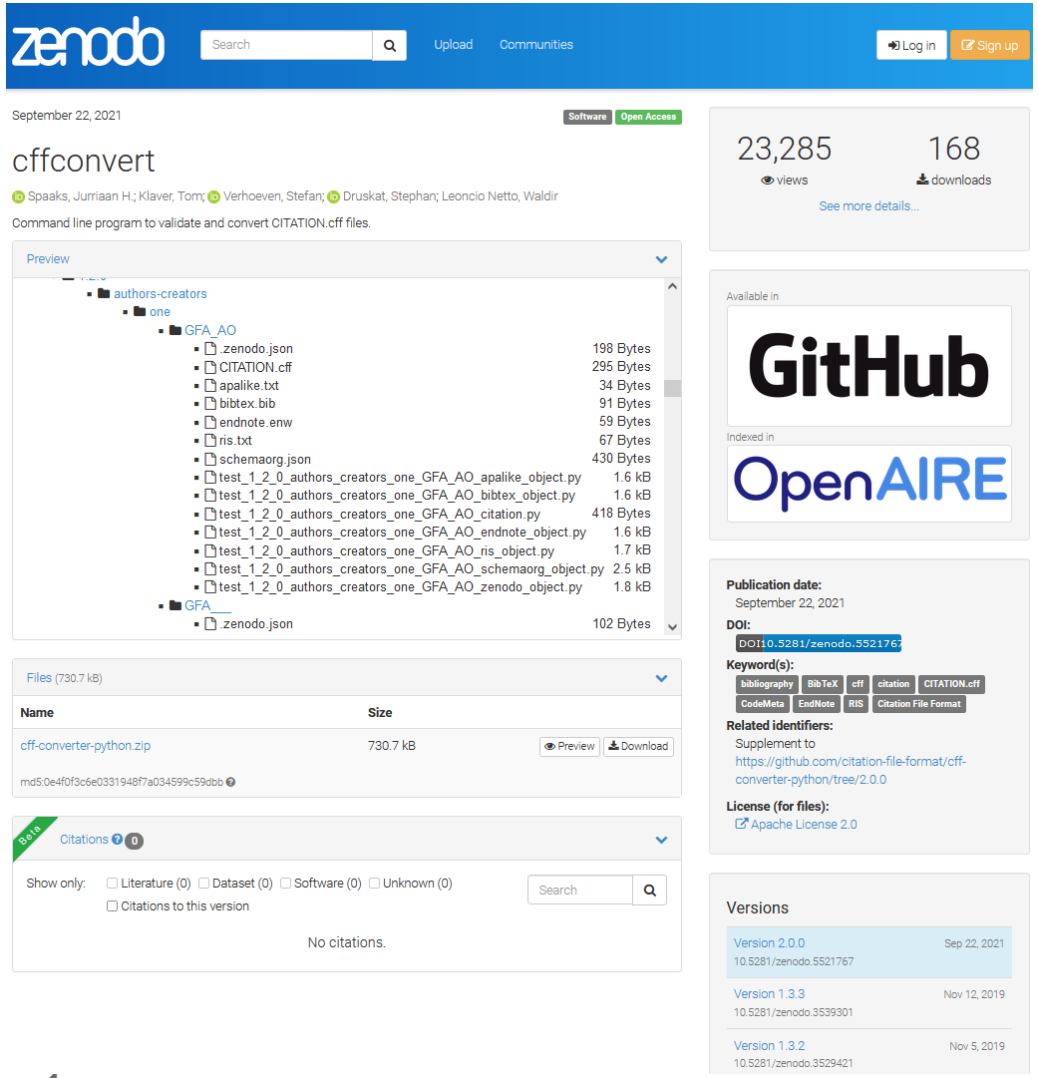
- ~~Code oder Executable per E-Mail verschicken~~
- ~~Software auf eigener Website zum Download anbieten~~
- ~~Source Code auf GitHub/GitLab/… hosten~~
- **FAIRe Softwarepublikation in einem Open Access Repositorium (mit DOI)!**
- (Publikation in einem Software Journal, z.B. JOSS)

- Offene Frage: Qualitätssicherung in der Softwarepublikation

Publikation von Forschungssoftware

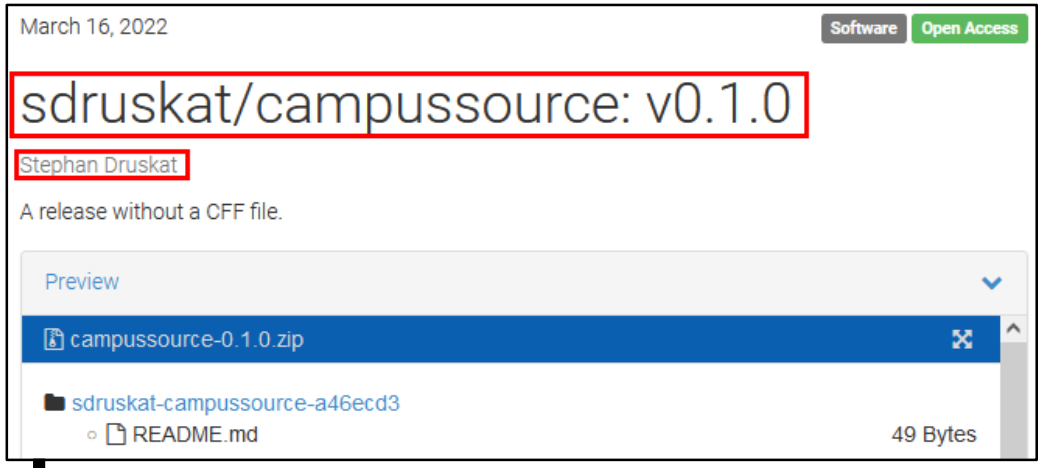
- Wann publiziere ich eigentlich?
  - Verhältnis Publikation-Release
  - Verhältnis Publikation-Commit
  - Publikationsgründe

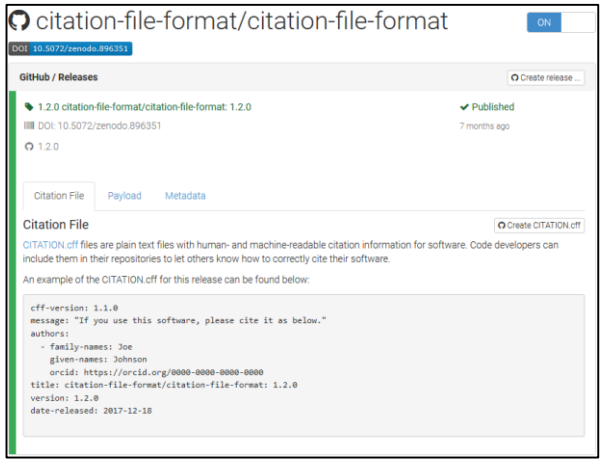
Praxis der Softwarepublikation: Beispiel Zenodo



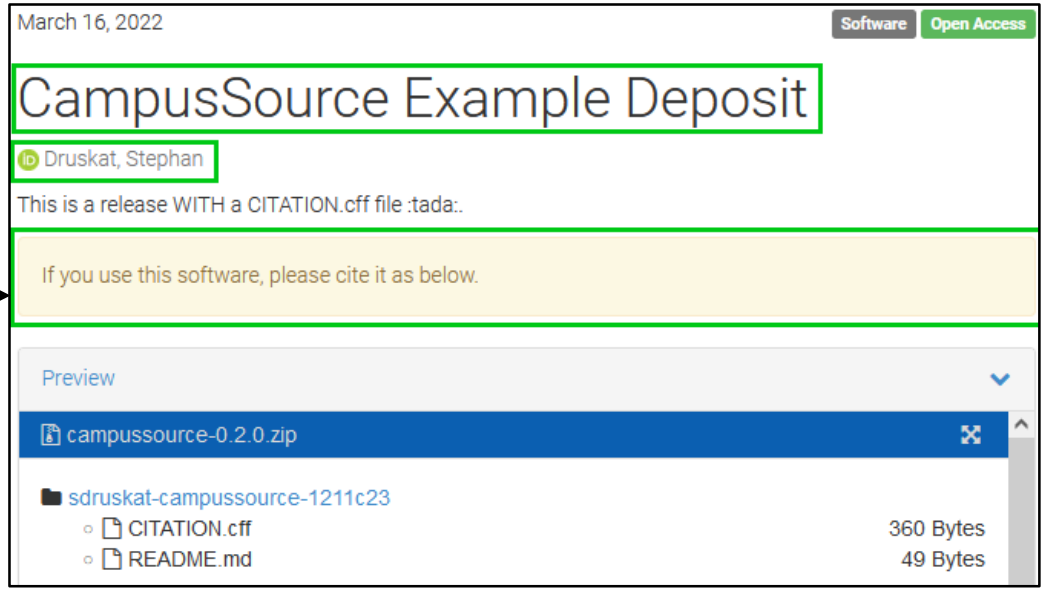
- Zenodo (https://zenodo.org) ist ein Open Access Repositorium für digitale Forschungsartefakte (inkl. Software)
- DOI und „Titelseite“
- Parent DOI
- Metadaten
- Direkter Download von Software, Dokumentation, etc.

Praxis der Softwarepublikation: Citation File Format + Zenodo



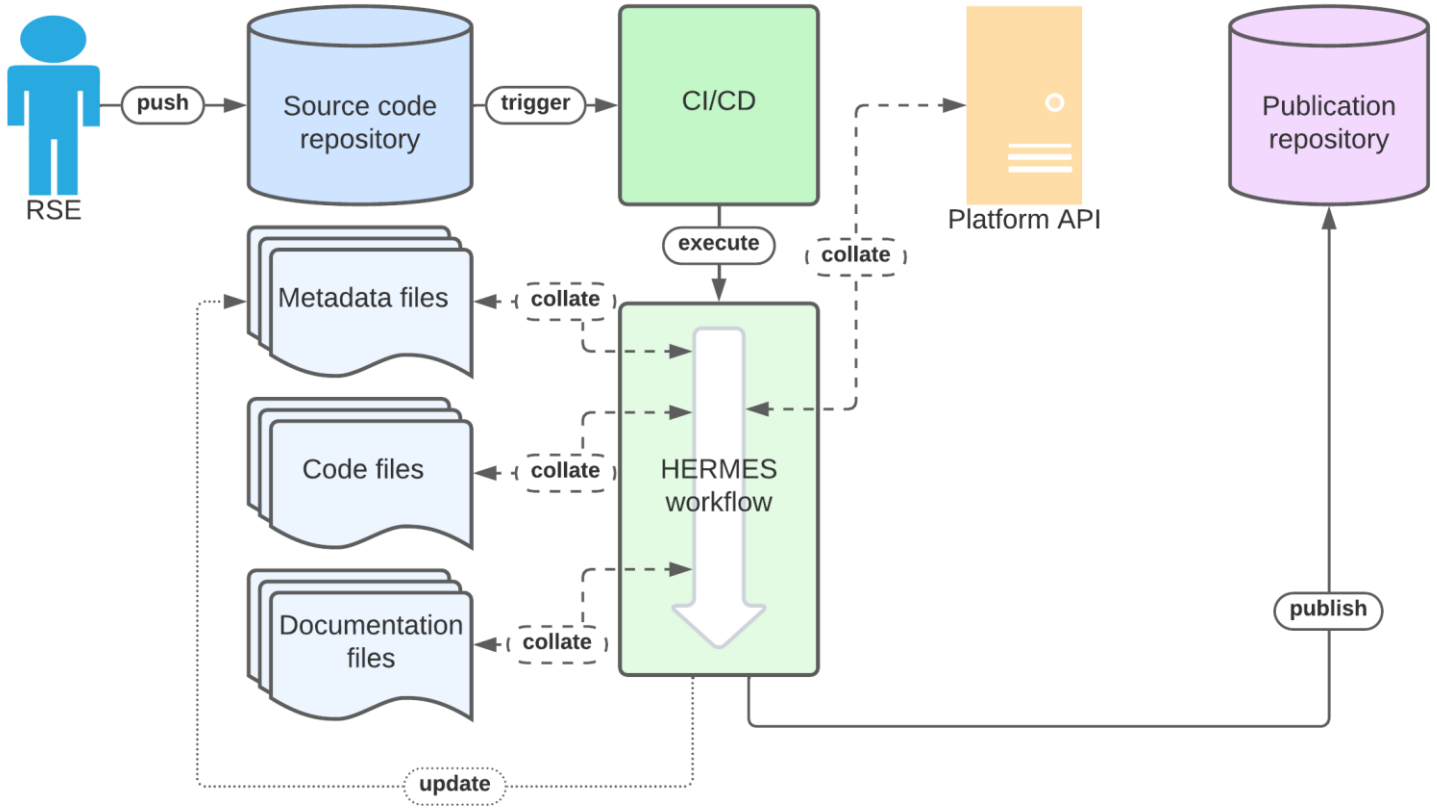



```
1  cff-version: 1.2.0
2  message: "If you use this software, please cite it as below."
3  authors:
4  - family-names: "Druskat"
5    given-names: "Stephan"
6    orcid: "https://orcid.org/0000-0003-4925-7248"
7  title: "CampusSource Example Deposit"
8  version: 0.2.0
9  doi: 10.5281/zenodo.1035710
10 date-released: 2022-03-16
11 url: "https://www.campussource.de/events/e2203hagen/#Programm"
```



Zukunft der FAIRen Softwarepublikation: HERMES

http://software-metadata.pub/ | Konzeptpapier [10]



Zusammenfassung, Ausblick, offene Frage

- Software ist wichtig, deshalb ist Nachhaltigkeit von Software wichtig
- Nachhaltigkeit wird gefördert u.a. durch Nachweis der Nützlichkeit (Zitierung) und Auffindbarkeit und Nachnutzung (FAIR, Publikation)
- Die Grundproblematik ist klar, einige Probleme sind gelöst, andere noch nicht, einige sind schwierig
- Vielleicht brauchen wir tatsächlich ein komplett anderes System (s. Metriken)?
- Praktisch für die Gegenwart von Softwarezitierung und -publikation:
  - Citation File Format
  - Zenodo
- In naher Zukunft: größtmögliche Automatisierung (HERMES)
- Zukunft von Softwarezitierung und -publikation?

- **Offene Fragen:**
  - Software und KPIs
  - Metriken: wie tappen wir nicht in die h-Index-Falle?

Vielen Dank!

- stephan.druskat@dlr.de
- Twitter: @stdruskat
- ORCID: 0000-0003-4925-7248

- citation-file-format.github.io
- software-metadata.pub

Förderung vorgestellter Arbeit durch:

- Software Sustainability Institute (EPSRC through grant EP/H043160/1; EPSRC, BBSRC and ESRC Grant EP/N006410/1; EPSRC, BBSRC, ESRC, NERC, AHRC, STFC and MRC grant EP/S021779/1)
- Helmholtz Metadata Collaboration (ZT-I-PF-3-006)

Referenzen

- [1] C. Becker *et al.*, “The Karlskrona manifesto for sustainability design,” *CoRR*, vol. abs/1410.6968, 2014 [Online]. Available: http://arxiv.org/abs/1410.6968
- [2] C. Jay, R. Haines, and D. S. Katz, “Software Must be Recognised as an Important Output of Scholarly Research,” *International Journal of Digital Curation*, vol. 16, no. 1, p. 6, Apr. 2021, doi: 10.2218/ijdc.v16i1.745.
- [3] S. Hettrick, *Software In Research Survey 2014*. Zenodo, 2018 [Online]. Available: https://zenodo.org/record/1183562.
- [4] K. Hinsen, “Dealing With Software Collapse,” *Computing in Science Engineering*, vol. 21, no. 3, pp. 104–108, May 2019, doi: 10.1109/MCSE.2019.2900945.
- [5] J. Howison and J. Bullard, “Software in the scientific literature: Problems with seeing, finding, and using software mentioned in the biology literature,” *Journal of the Association for Information Science and Technology*, vol. 67, no. 9, pp. 2137–2155, May 2015, doi: 10.1002/asi.23538.
- [6] S. Druskat, N. P. Chue Hong, P. Kornek, S. Buzzard, and A. Konovalov, “Don’t mention it: challenges to using software mentions to investigate citation and discoverability,” *PeerJ Computer Science*, forthcoming.

Referenzen

- [7] D. S. Katz, M. Gruenpeter, and T. Honeyman, “Taking a fresh look at FAIR for research software,” *Patterns*, vol. 2, no. 3, p. 100222, Mar. 2021, doi: 10.1016/j.patter.2021.100222.
- [8] S. Druskat, D. S. Katz, and I. T. Todorov, “Research Software Sustainability and Citation,” in *2021 IEEE/ACM International Workshop on Body of Knowledge for Software Sustainability (BoKSS)*, Jun. 2021, pp. 1–2, doi: 10.1109/BoKSS52540.2021.00008.
- [9] A. M. Smith, D. S. Katz, K. E. Niemeyer, and FORCE11 Software Citation Working Group, “Software citation principles,” *PeerJ Comput. Sci.*, vol. 2, no. e86, 2016, doi: 10.7717/peerj-cs.86.
- [10] S. Druskat, O. Bertuch, G. Juckeland, O. Knodel, and T. Schlauch, “Software publications with rich metadata: state of the art, automated workflows and HERMES concept,” arXiv, arXiv:2201.09015, Jan. 2022 [Online]. Available: http://arxiv.org/abs/2201.09015.