

Identifikation und Vergleich von Autorenangaben zu Software zwischen verschiedenen Datenquellen

Wismar, 30. Januar 2025

Fakultät für Ingenieurwissenschaften, Hochschule Wismar

Kevin Jahrens

E-Mail: k.jahrens@stud.hs-wismar.de



Gliederung

- 1 Einleitung
- 2 Grundlagen
- 3 Methodik
- 4 Ergebnisse
- 5 Diskussion

Einleitung

Einleitung

Motivation

- Software spielt zentrale Rolle in der Wissenschaft
- Zitation wesentlicher Bestandteil in wissenschaftlicher Publikation
- Bei wissenschaftlicher Software ist dies in diesem Umfang aktuell nicht gegeben
- Softwareautoren werden nicht immer genannt und manchmal sogar ihrer Beiträge beraubt [2]

Einleitung

Vorgehen

- Autoren aus unterschiedlichen Quellen extrahieren
- Autoren untereinander abgleichen
- Ausschließliche Betrachtung von Autoren, die Code in Git beigetragen haben
- Ergebnisse aufbereiten
- Beantwortung von Forschungsfragen

Einleitung

Forschungsfragen

- F1** Wie gut können Autoren untereinander abgeglichen werden?
- F2** Was muss ein Softwareentwickler leisten, um als Autor genannt zu werden?
- F3** Wie gut werden Autoren in den einzelnen Quellen gepflegt?

Grundlagen

Grundlagen

Prinzipien der Software-Zitation [3]

1. **Wichtigkeit:** Software sollte ein seriöses und zitierbares Produkt wissenschaftlicher Arbeit sein.
2. **Anerkennung und Zuschreibung:** Softwarezitate sollten die wissenschaftliche Anerkennung und die normative, rechtliche Würdigung aller Mitwirkenden an der Software ermöglichen.
3. **Eindeutige Identifikation:** Ein Softwarezitat sollte eine Methode zur Identifikation enthalten, die maschinell verwertbar, weltweit eindeutig und interoperabel ist.
4. **Persistenz:** Eindeutige Identifikatoren und Metadaten, die die Software und ihre Verwendung beschreiben, sollten bestehen bleiben – auch über die Lebensdauer der Software hinaus.
5. **Zugänglichkeit:** Softwarezitate sollten den Zugang zur Software selbst und weiteren Materialien erleichtern, um sie sachkundig nutzen zu können.
6. **Spezifizität:** Softwarezitate sollten die Identifikation und den Zugang zu der spezifischen Version der verwendeten Software erleichtern. Die Identifizierung der Software sollte so spezifisch wie nötig sein.

Grundlagen

Versionsverwaltung

- Verwaltet Quellcode und dessen Änderungen
- Git ist eine weit verbreitete Versionsverwaltung mit einem Marktanteil von ungefähr 75 % [1]
- Speichert Zeitpunkt und Autor, sowie die Änderungen in einem Commit
- Name und E-Mail des Autors frei wählbar
- In Git werden weitere Daten gespeichert, welche ausgelesen werden können:
 - Anzahl der eingefügten und gelöschten Zeilen
 - Anzahl der geänderten Dateien
 - Anzahl der Commits

Grundlagen

Software-Verzeichnisse und Paketverwaltung

- Eine Paketverwaltung verwaltet fertige Softwarepakete, bspw. kompilierten Code
- Softwarepakete können in einem Software-Verzeichnis abgelegt werden
- Softwarepakete enthalten Metadaten, bspw. die Autoren des Pakets
- Es werden die Verzeichnisse PyPI (Python) und CRAN (R) untersucht
- Für beide Verzeichnisse stehen APIs zur Verfügung, welche die Metadaten bereitstellen

Methodik

Ergebnisse

Diskussion

Literaturverzeichnis

- [1] Jannik Lindner. *Version Control Systems Industry Statistics*. 3. Mai 2024. URL: <https://worldmetrics.org/version-control-systems-industry-statistics/> (besucht am 21.05.2024).
- [2] Ariel Miculas. *How I got robbed of my first kernel contribution*. 27. Sep. 2023. URL: <https://ariel-miculas.github.io/How-I-got-robbed-of-my-first-kernel-contribution/> (besucht am 03.06.2024).
- [3] Arfon M. Smith, Daniel S. Katz und Kyle E. Niemeyer. „Software citation principles“. In: *PeerJ Computer Science* 2 (19. Sep. 2016), e86. DOI: [10.7717/peerj-cs.86](https://doi.org/10.7717/peerj-cs.86).