



seit 1558

# Infrastrukturstrategien für nachhaltige Forschungssoftware in befristeten Projekten

Stephan Druskat<sup>1</sup>, Thomas Krause<sup>2</sup>, Anke Lüdeling<sup>2</sup>, Volker Gast<sup>1</sup><sup>1</sup>Friedrich-Schiller-Universität Jena, <sup>2</sup>Humboldt-Universität zu Berlin

## Problem

**Forschungssoftware:** häufig entwickelt von Einzelpersonen oder Kleinstteams, in befristeten Projekten, ohne Software-Engineering-Expertise, optimiert für Projektabschluss und Features, nicht für Nachhaltigkeit.

### Nachhaltigkeit von Forschungssoftware

- Verwendbarkeit durch Dritte (oder Zweite)
- Wartbarkeit
- Reproduzierbarkeit von Forschungsergebnissen
- etc.

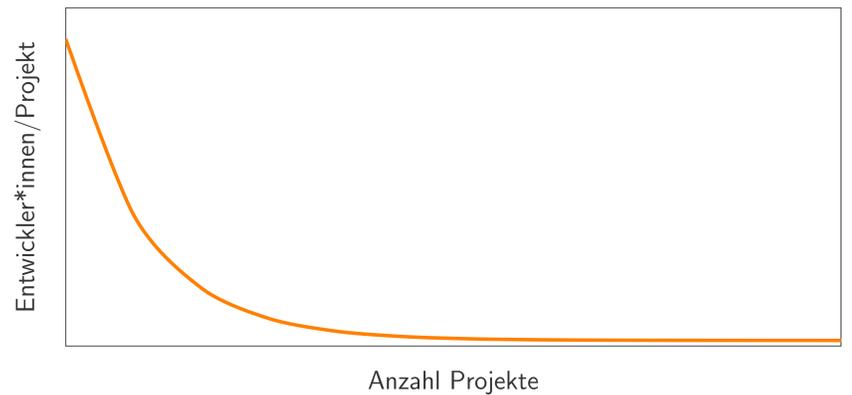
### Warum?

Tote Software → keine Überprüfbarkeit von Forschungsergebnissen → Invalidierung von Forschungsergebnissen → Zunichtemachen von Forschungsaktivitäten.

**Wo liegt Wissen über Software und Projektorganisation?**

**Wie kann man Nachhaltigkeit kostenoptimiert erreichen?**

## “The long tail of science”



Unsere Domäne: Projekte, deren Busfaktor gegen 1 tendiert.

**Lösungsansatz:** technisches Nachhaltigkeitspotential + Dokumentation + offene Infrastrukturen + Maintenance-Strategie

## technische Nachhaltigkeit

- Prinzipien des Software Engineering
  - Verständnis, Planung, Ausführung, Prüfung
- Architekturdesign
  - Erweiterbarkeit, Generizität
- Codequalität
- “Data-driven” Technologieentscheidungen

## Dokumentation

- “Document all the things!”
- Metadokumentation
- Dokumentation von Entscheidungen
- Dokumentation von Infrastruktur
- Nachhaltiges Tooling
- Code und Dokumentation zusammen vorhalten

## offene Infrastrukturen

- Vorbild: erfolgreiche Open Source-Projekte
- kostenlose “Communitystandards” nutzen
  - Entwicklungsplattformen (VCS, CI, etc.)
  - Repositorien für Buildartefakte
  - Repositorien für Dependency-Artefakte
- generische Austausch-/Konfigurationsformate

## Schlüsselrolle: Maintainer

**Ziel:** Maintenance ermöglichen = Nachhaltigkeit ermöglichen

Gewährleistet Umsetzung und funktionierendes Zusammenspiel von technischer Nachhaltigkeit, Dokumentation und offenen Infrastrukturen. Sichert und stellt Wissen über Software und Projektorganisation bereit. Sichert Funktionsfähigkeit des Softwareprojekts & der Software. Ermöglicht jederzeitige Weiterführung/Wiederaufnahme des Projekts für, z.B., funktionale Erweiterungen (Voraussetzung: *Finanzierung*).

## Fallstudie: Hexatomic

### Projekt

*A minimal infrastructure for the sustainable provision of extensible multi-layer annotation software for linguistic corpora*, <https://hexatomic.github.io>  
Gefördert durch die Deutsche Forschungsgemeinschaft (LU 856/11-1 + GA 1288/11-1) unter “Nachhaltigkeit von Forschungssoftware”.

**Ziel:** Zusammenführung zweier Prototypen für Annotationssoftware für linguistische Mehrebenenkorpora – Entwicklung, Bereitstellung, Pflege – unter Erprobung minimaler Infrastruktur.

### initiale, austauschbare Infrastrukturkomponenten:

- **Entwicklungsplattform:** GitHub
  - Host für: Sourcecode, Dokumentation(en), Landing Page; Anbindung an CI; Anbindung an Zenodo; Quelle für Software Heritage
- **Repositorium für Buildartefakte:** Zenodo & Maven Central
  - Langfristige Bereitstellung, eindeutige und versionierte Identifikation, *Software Citation*, Nachnutzbarkeit (Maven)
- **Repositorium für Dependency-Artefakte:** Maven Central, Eclipse P2
  - Langlebig, eingebunden in Build-Infrastruktur, unabhängig finanziert
- **Maintainer:** dreistufige Erprobung von Maintainerwechseln
  1. Wechsel von Projektentwickler zu Projektentwickler
  2. Wechsel von Projektentwickler zu wissenschaftlicher Hilfskraft (team-intern)
  3. Wechsel von wissenschaftlicher Hilfskraft (team-intern) zu externer wissenschaftlicher Hilfskraft (Michael Stifel Zentrum Jena)