

DigEdTnT-Workshop, Universität Graz, 23.02.2023

Tools für digitale Editionen als Helfer und Hürden: Wie nehmen wir die Werkzeuge in die Hand?

Ulrike Henny-Krahmer, Universität Rostock

Überblick

- Tools für digitale Editionen als Helfer
- Tools für digitale Editionen als Hürden
- Wo können wir ansetzen, um die Übergänge zwischen Tools effektiv zu gestalten?
 - Sache der Developer (u.a.): Nachhaltige Softwareentwicklung
 - Data und Coding Literacy: vom User zum Mit-Developer werden
 - Tool-Diskurs und Tools in anderen Diskursen: ausbauen und einbinden
 - Tool-Entwicklung: organisieren, finanzieren, fördern

Tools für digitale Editionen als Helfer

- Tools für digitale Editionen machen ihre Erstellung überhaupt erst möglich oder effizient. Einige Beispiele:

mei-friend



ediarum

Transkribus

<TEI:TOK>



LAKomp



DIGITAL MAPPA

an open-source digital humanities platform
for open-access workspaces, projects and publications

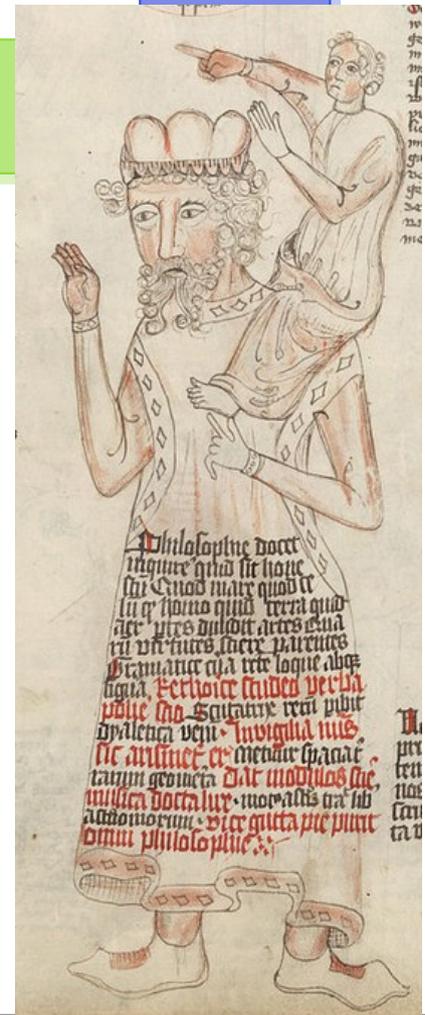


Critical
Apparatus
Toolbox

reledmac

Tools für digitale Editionen als Helfer

- Jannidis (2017, 88) zu Programmbibliotheken:
„Sie sind kondensierte, gespeicherte Arbeitszeit und Intelligenz und ermöglichen jedem, der auf sie zugreifen kann, diese Zeit und Problemlösungskompetenz in den Dienst der eigenen Sache zu stellen.“
- Das gilt auch für Tools für digitale Editionen!



Aus einem anonymen Manuskript, Rosenwald 4, Bl. 5r,
<http://lccn.loc.gov/50041709>.

Tools für digitale Editionen als Hürden

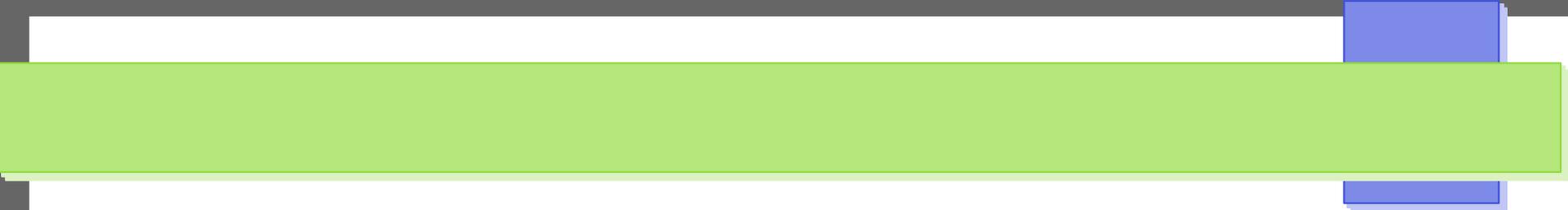
- Was, wenn das Tool mich etwas nicht machen lässt, was ich gern tun würde?
- Was, wenn ich meine Daten nicht aus dem Tool herausbekomme?
- Was, wenn ich meine Daten nicht in ein anderes Tool hineinbekomme?
- Was, wenn mein Helfer-Tool irgendwann nicht mehr funktioniert oder gar nicht mehr verfügbar ist?
- Was, wenn es gar kein Tool für das gibt, was ich tun möchte?



https://www.flaticon.com/de/kostenloses-icon/erschrockene_n_742848

Ansatzpunkte für effektive Toolübergänge

- 1) Sache der Developer (u.a.): Nachhaltige Softwareentwicklung
- 2) Data und Coding Literacy: vom User zum Mit-Developer werden
- 3) Tool-Diskurs und Tools in anderen Diskursen: ausbauen und einbinden
- 4) Tool-Entwicklung: organisieren, finanzieren, fördern



Sache der Developer (u.a.):
Nachhaltige Softwareentwicklung

Sache der Developer (u.a.): Nachhaltige Softwareentwicklung

- Was meint **Nachhaltigkeit** in der Softwareentwicklung?
 - **Transparenz:** Software sollte so entwickelt werden, dass transparent und nachvollziehbar bleibt, wie Ergebnisse entstanden sind.
 - **Effektivität:** Ressourcen für die Entwicklung von Software sollten effektiv und nachhaltig eingesetzt werden (Doppelentwicklungen vermeiden, Interoperabilität und Nachnutzbarkeit sicherstellen)
 - **Langfristigkeit:** Transparenz und Effektivität sollten möglichst langfristig sichergestellt werden (langfristige Verfügbarkeit).
- allgemein: positive ökologische, soziale oder ökonomische Wirkung (Wilkinson et al. 2016, Stürmer 2017)

Sache der Developer (u.a.): Nachhaltige Softwareentwicklung

- Ein zentraler Ansatz: **FAIR Principles for research software** (FAIR4RS, Chue Hong et al. 2022)
 - Findable
 - Accessible
 - Interoperable
 - Reusable

„**Research Software** includes source code files, algorithms, scripts, computational workflows and executables that were created during the research process or for a research purpose. Software components (e.g., operating systems, libraries, dependencies, packages, scripts, etc.) that are used for research but were not created during or with a clear research intent should be considered **software in research** and not Research Software. This differentiation may vary between disciplines.“ (Chue Hong et al. 2022, 5)

„Unter **Forschungssoftware** verstehen wir Software, die in der Forschung eingesetzt wird, um Forschungsdaten zu erzeugen, zu verarbeiten, zu analysieren, zu verknüpfen oder zu präsentieren.“ (Jettka und Henny-Krahmer 2022)

Sache der Developer (u.a.): Nachhaltige Softwareentwicklung

- Für Toolübergänge besonders relevant: ***Interoperable***

I: Software interoperates with other software by exchanging data and/or metadata, and/or through interaction via application programming interfaces (APIs), described through standards.

- I1. Software reads, writes and exchanges data in a way that meets domain-relevant community standards.
- I2. Software includes qualified references to other objects.

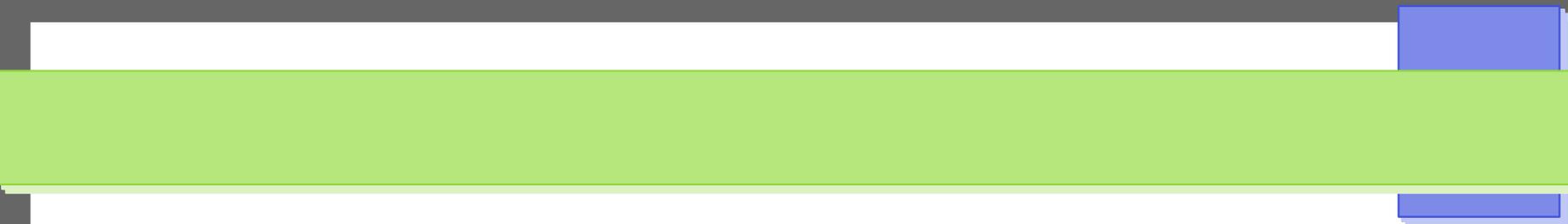
Chue Hong et al. 2022, 5.

Sache der Developer (u.a.): Nachhaltige Softwareentwicklung

- Nachhaltige Softwareentwicklung ist am Ende doch nicht nur Sache der Entwickler*innen!

Handlungsbereiche
für nachhaltige
Softwareentwicklung
und -nutzung
aus
Jettka und Henny-
Krahmer (2022)





Data und Coding Literacy:
vom User zum Mit-Developer werden

Data und Coding Literacy: vom User zum Mit-Developer werden

- **Mögliche Situation:**

- Ich kann meine Daten aus Tool 1 exportieren
- Ich kann auch Daten in Tool 2 importieren
- Aber die Datenformate sind unterschiedlich!
- Was tun?

- **Mögliche Lösungen:**

- Ich suche ein Tool für die Konvertierung
- Ich konvertiere die Daten selbst
- Ich erweitere Tool 1 oder Tool 2 um Export- und Importformate

Data und Coding Literacy: vom User zum Mit-Developer werden

- Ich suche ein Tool für die Konvertierung.
- Zum Beispiel:

TEIGarage

<https://teigarage.tei-c.org/>



Convert from: ?



Documents

- Cocoa tagging
- Compiled TEI ODD
- DocBook Document
- Markdown tagging
- Microsoft Word (.doc)
- Microsoft Word (.docx)
- ODD Document
- OpenOffice 1.0 Text (.sxd)
- OpenOffice Text (.odt)
- Plain Text (.txt)
- Rich Text Format (.rtf)
- TCP XML Document
- TEI P4 XML Document
- TEI P5 XML Document

Convert to: ?

- Comma-Separated Values (.csv)
- DocBook Document
- ePub3
- LaTeX
- Markdown tagging
- Microsoft Word (.doc)
- Microsoft Word (.docx)
- National Library of Medicine (NLM) DTD 3.0
- OpenOffice 1.0 Text (.sxd)
- OpenOffice Text (.odt)
- PDF
- Plain text
- RDF XML
- Rich Text Format (.rtf)
- TEI Simple XML Document
- TeX

Data und Coding Literacy: vom User zum Mit-Developer werden

- **Ich konvertiere die Daten selbst.**
 - z.B. händisch
 - durch Suchen-Ersetzen
 - Reguläre Ausdrücke
 - Skriptentwicklung
- **Dafür brauche ich:
Data und Code Literacy!**

Data und Coding Literacy: vom User zum Mit-Developer werden

- Bleeker et al. (2022): Umfrage und Survey zu Code Literacy in den Digital Humanities

“Should humanists learn to code?’ Less than a decade ago this question would have ignited quite a controversy in the field of digital humanities (DH). Today, the consensus is that a certain level of code literacy is preferred. Instead of arguing whether code literacy deserves to be part of DH’s skill set, the debate has moved on to discussing what it means, exactly, to be code literate.” (1-2)

Data und Coding Literacy: vom User zum Mit-Developer werden

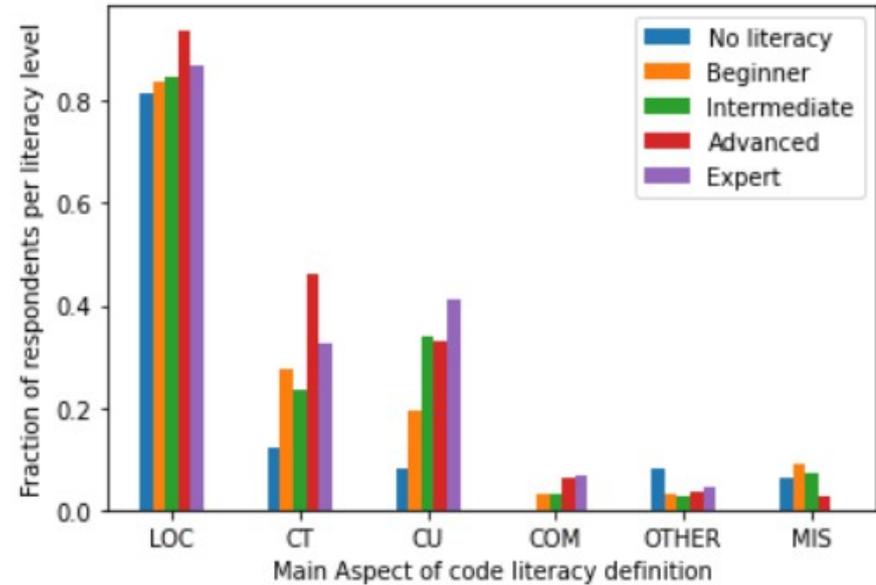
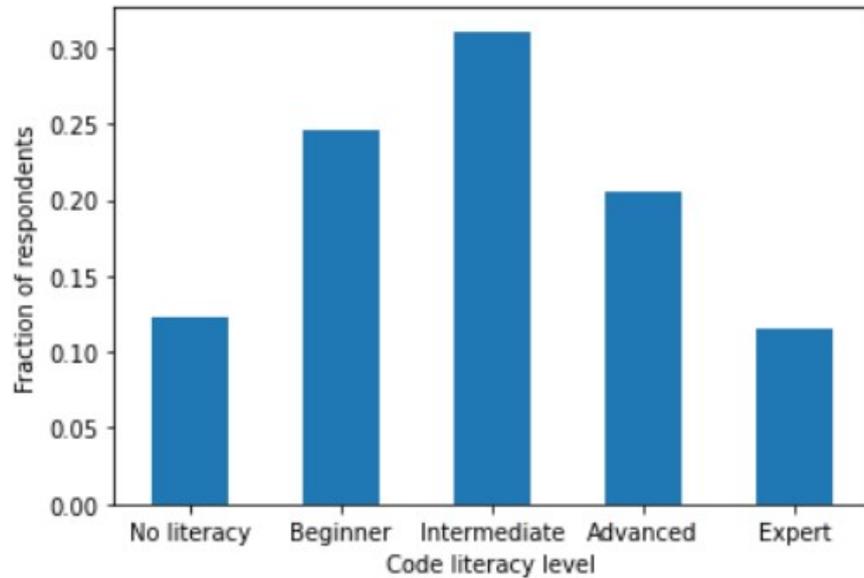
- Was ist **Code Literacy** (Bleeker et al. 2022, 14)?
 - “The ability to understand and write code and to use it to achieve some research goal.”
 - “The ability to read, write and use code.”
 - “knowledge and experience of solving problems through the use of programming skills”
 - “I’d say there’s informal and formal code literacy. Formal literacy is what you find with colleagues in the sciences and engineering. There’s a strong emphasis on style, standards, and efficiency. Tests and quality control are required. Then there are the programming historians, the self-taught, and highly pragmatic types. It’s just amazing that it works at all. The code is an odd pastiche of cut and pastes from Stack Overflow. It’s a game of persistence, self-doubt, and curiosity. I mostly work with informal code literacy to give people the practical skills of reading error messages and documentation, finding helpful solutions to problems, and making something that works but is by no means pretty or professional.”

Data und Coding Literacy: vom User zum Mit-Developer werden

Aspekte und Ebenen von Code Literacy (Bleeker et al. 2022, 15)

<i>Code</i>	<i>Description</i>	<i>Scope note</i>
COM	Communication	Knowing how to communicate about code with others, either as a coder yourself, or with someone who codes for you, including communication about purpose, workings, role and implications, as well as teaching code literacy.
CT	Code Type	Whether the code explicitly refers to a specific type of code, either 1) code as encoding of documents or 2) performative code for e.g. processing of data.
CU	Contextual Understanding	Understanding that code sits in a context and how it relates to 1) research and operationalising research questions, 2) it's possibilities, limits and biases, 3) the culture and attitudes toward code, 4) the ecosystem around code of ethics, privacy, security, maintenance, documentation, versioning, licensing, practices, platforms and software, and 5) other aspects of context.
LOC	Level of Competence	Literacy is divided into 7 different levels of competence, from 1) understanding the basics of what code is and does, to being able to 2) read code, 3) write basic code, 4) modify existing code, 5) review code, 6) create and package code from scratch, and 7) understanding of the theory of computation and coding paradigms. Separate from those levels there is a sub-category for definitions that mention there are multiple levels of competence.
OTHER	Other aspects	Any code literacy aspects not covered by the above four aspects.
MIS	Misinterpretations	Responses that are not definitions but answers to a different question.

Data und Coding Literacy: vom User zum Mit-Developer werden



Code Literacy Levels von 399 Umfrageteilnehmer*innen (Bleeker et al. 2022, 18)

Data und Coding Literacy: vom User zum Mit-Developer werden

Wenn man nun selbst Transformationen entwickelt hat:

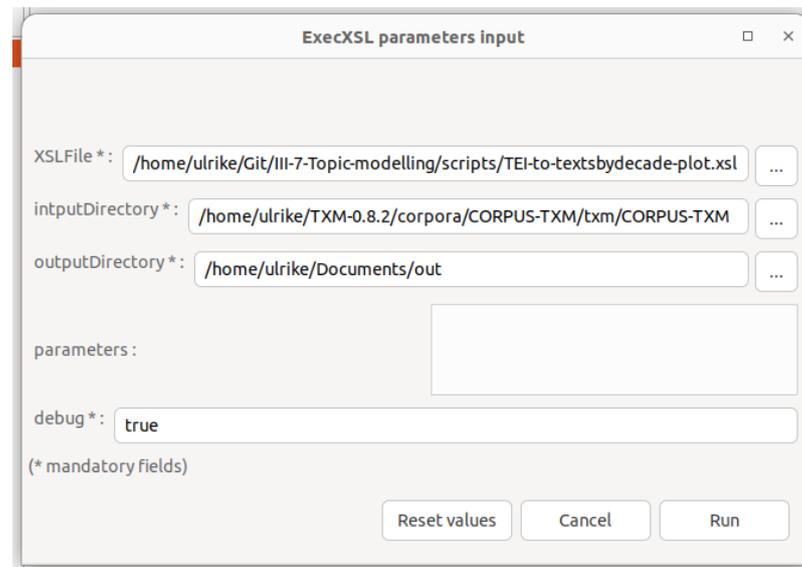
- Warum die nicht auch publizieren?
- „minions“

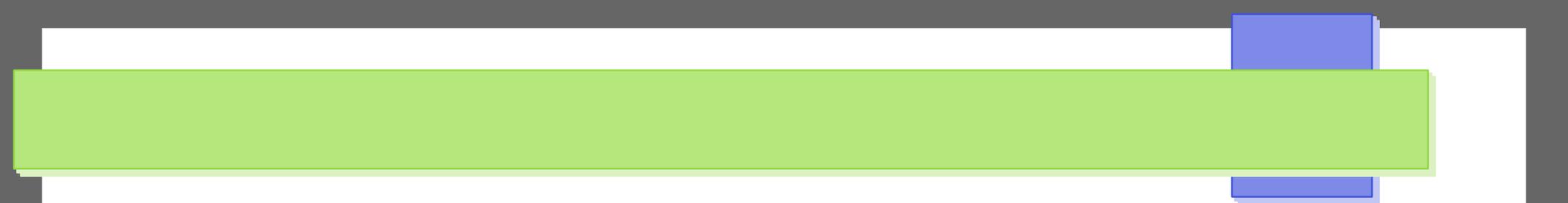
Data und Coding Literacy: vom User zum Mit-Developer werden

- **Ich erweitere Tool 1 oder Tool 2 um Export- und Importformate.**
 - erfordert ggf. mehr Code Literacy
 - wichtig: dass Tool 1 oder 2 „**accessible**“ und „**reusable**“ sind!
 - kommt direkt auch anderen Nutzer*innen der Tools zugute

Data und Coding Literacy: vom User zum Mit-Developer werden

- Ein Beispiel: das Tool **TXM** (Textométrie)
- <https://txm.gitpages.huma-num.fr/textometrie/>
- Heiden (2010)
- Open Source
- Textanalyse mit TEI-Unterstützung
- Zahlreiche Importformate, textbasiert, XML-Varianten, auch projektspezifische
- Export in TEI+CSV oder binär
- Anbindung an R
- Erweiterung durch Makros und Ausführen eigener Skripte möglich





Tool-Diskurs und Tools in anderen Diskursen:
ausbauen und einbinden

Tool-Diskurs und Tools in anderen Diskursen: ausbauen und einbinden

- Interoperabilität von Tools kann gefördert werden durch:
 - **Tool-Diskurs**
 - Veranstaltungen zu Tools (z.B. Workshops wie „DigEdTnT: Workshop - Meet the Tools and Developers“)
 - Kontakt zwischen Developer*innen und User*innen (Mailinglisten, Chats, Wikis...)
 - Veröffentlichen von Software-Papern
 - Schreiben von Software-Rezensionen
 - Tools zitieren
 - **Tools in anderen Diskursen**
 - Bedeutung und Einsatz von Werkzeugen in der Fachforschung kommunizieren und dokumentieren

Wirkung:
erhöht die
findability & trägt
zu Nachhaltigkeit
bei!

Tool-Diskurs und Tools in anderen Diskursen: ausbauen und einbinden

- Beispiel Softwarerezensionen:
 - RIDE – A review journal for digital editions and resources (<https://ride.i-d-e.de/>)
 - CKIT – Construction KIT. A review journal for research tools and data services (<https://journals.ub.uni-heidelberg.de/index.php/ckit/>)



Tool-Diskurs und Tools in anderen Diskursen: ausbauen und einbinden

- RIDE: bisher 2 Bände - Issue 11 (2020) und Issue 15 (2022) - mit 11 Rezensionen für Tools zu digitalen Editionen
- Rezensierte Tools: Reledmac, TUSTEP, Omeka Classic, Ediarum, Juxta Web Service, LERA, Variance Viewer, Digital Mappa, Mei-friend, Stylo, TEI Critical Apparatus Toolbox, TEITOK, Transkribus



Tool-Diskurs und Tools in anderen Diskursen: ausbauen und einbinden

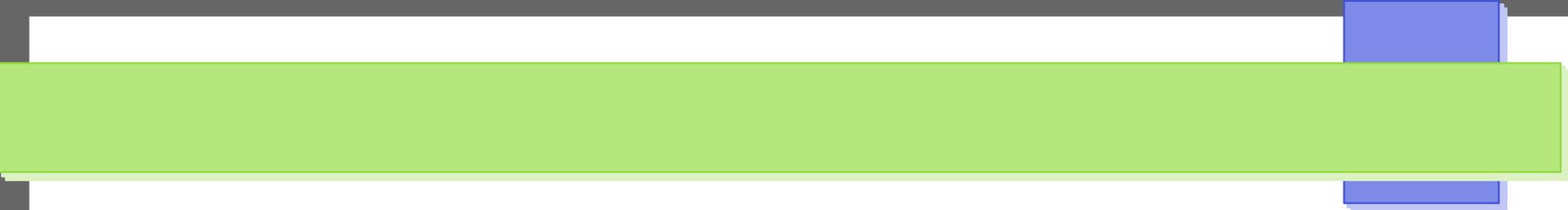
- Kriterien für die Rezension von Tools
 - „Criteria for Reviewing Tools and Environments for Digital Scholarly Editing, version 1.0“
 - Anna-Maria Sichani and Elena Spadini, in collaboration with the members of the IDE; Version 1.0, December 2018
 - <https://www.i-d-e.de/publikationen/weitereschriften/criteria-tools-version-1/>
 - Richtlinien für Beiträge in CKIT
 - Lisa Dieckmann, Maria Effinger, Anne Klammt, Daniel Röwenstrunk, Fabian Offert
 - <https://journals.ub.uni-heidelberg.de/index.php/ckit/about/submissions>

Tool-Diskurs und Tools in anderen Diskursen: ausbauen und einbinden

- Was sagen die Kriterien über Interoperabilität?
 - RIDE:
 - „2. Methods and Implementation
 - 2.4. Input, output, and data-modelling. Specify the input and output formats (e.g. .xml, .txt, .pdf, etc.) and encodings (e.g. latin-1, utf-8) supported. Point out if the internal operations require a form of data modelling or conversion, e.g. if schema control is present or pre-processing format conversion is included. With regards to multilingual audience, mention whether the tool is compatible with textual resources using non-Latin alphabet/scripts and/or multiple languages.
 - 3.8. Portability and interoperability.
 - [...] Interoperability indicates the ability to work in conjunction with other software, typically by support for multiple open-standard formats as input and output.“

Tool-Diskurs und Tools in anderen Diskursen: ausbauen und einbinden

- CKIT
 - „Die Sicht der Nutzer*innen“
 - „Obligatorisch: Dateneingabe und -ausgabe - welche Formate (Standards) verarbeitet das Tool/der Datenservice? Massen-Uploads? APIs? Wie sind die Daten mit anderen Ressourcen oder Informationen verbunden? Ist die Verwendung von oder die Verknüpfung mit Normdateien (GND, VIAF, etc.) möglich? Werden kontrollierte Vokabularien verwendet und welche?“



Tool-Entwicklung:
organisieren, finanzieren, fördern

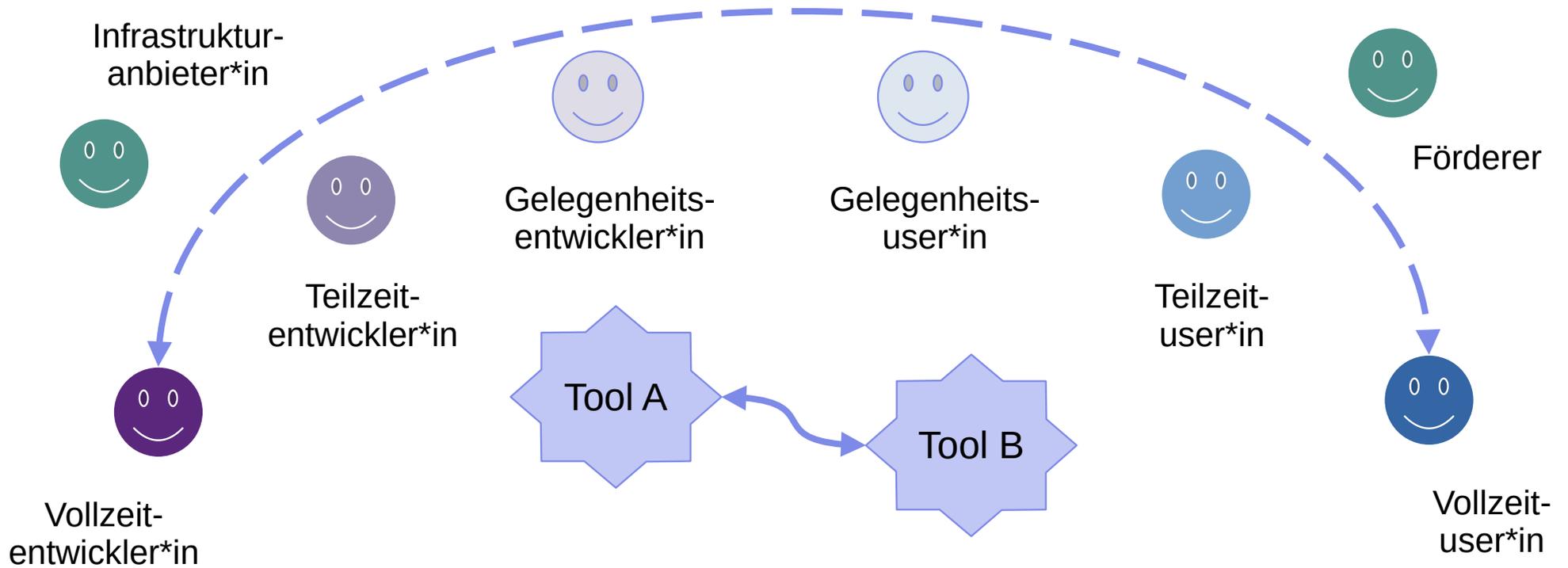
Tool-Entwicklung: organisieren, finanzieren, fördern

- Damit sich Tools und Toolübergänge etablieren können, ist es essentiell, Rahmenbedingungen zu schaffen, die Tool-Entwicklung fördern.
- Das heißt auch: die Menschen unterstützen und zusammenbringen, die Tools entwickeln.

Tool-Entwicklung: organisieren, finanzieren, fördern

- Z.B. über Arbeitsgruppen, Vereine, Projekte:
 - AG DH-RSE, <https://dh-rse.github.io/>
 - DHTech, <https://dh-tech.github.io/#/>
 - NFDI4Culture, Task Area „Research tools and data services“, <https://nfdi4culture.de>
 - De-RSE, <https://de-rse.org/de/index.html>
 - Software Sustainability Institute (SSI), <http://www.software.ac.uk/>
- Z.B. über Förderlinien:
 - DFG-Call: „Research Software – Quality Assured and Re-Usable“
 - EU-Call: „Development of community-based approaches for ensuring and improving the quality of scientific software and code“
- z.B. Integration von Tool-Entwicklung in universitäre Lehre

Fazit: Sind wir nicht alle Research Software Engineers?



Literaturhinweise (1)

- Anzt, Hartwig, Felix Bach, Stephan Druskat, Frank Löffler, Axel Loewe et al. (2021): “An environment for sustainable research software in Germany and beyond: current state, open challenges, and call for action [version 2; peer review: 2 approved].” *F1000Research* 9:295.
<https://doi.org/10.12688/f1000research.23224.2>.
- Bleeker, Elli, Marijn Koolen, Kaspar Beelen, Liliana Melgar, Joris van Zundert, and Sally Chambers (2022): „A Game of Persistence, Self-doubt, and Curiosity: Surveying Code Literacy in Digital Humanities.“ *DH Benelux Journal* 4. <https://doi.org/10.17613/03ds-9973>.
- Chue Hong, Neil P., Daniel S. Katz, Michelle Barker, Anna-Lena Lamprecht, Carlos Martínez et al. (2022): „FAIR Principles for Research Software (FAIR4RS Principles)“. RDA Recommendation.
<https://doi.org/10.15497/RDA00068>.
- Dieckmann, Lisa, Maria Effinger, Anne Klammt, Daniel Röwenstrunk und Fabian Offert (Hrsg.) (2022): „Construction Kit. A review journal for research tools and data services.“
<https://journals.ub.uni-heidelberg.de/index.php/ckit>.
- Institut für Dokumentologie und Editorik (Hrsg.) (2014-2023): „Ride. A review journal for digital editions and resources.“ <https://ride.i-d-e.de/>.

Literaturhinweise (2)

- Jannidis, Fotis (2017): „Grundbegriffe des Programmierens.“, in: *Digital Humanities. Eine Einführung*. Hrsg. Von Fotis Jannidis, Hubertus Kohle und Malte Rehbein. Stuttgart: J.B. Metzler, S. 68-95.
- Jettka, Daniel und Ulrike Henny-Krahmer (2022): “Leitfaden für die nachhaltige Entwicklung und Nutzung von Forschungssoftware.” NFDI4Culture Handreichung (Version 1.0.0).
<https://docs.nfdi4culture.de/ta3-sustainable-research-software>.
- Lamprecht, Anna-Lena, Leyla Garcia, Mateusz Kuzak, Carlos Martinez, Ricardo Del Arcila et al. (2020): “Towards FAIR principles for research software.” *Data Science* 3 (1): 37–59.
<https://doi.org/10.3233/DS-190026>.
- Stürmer, Matthias (2017): “Digitale Nachhaltigkeit: Digitale Gemeingüter für die Wissensgesellschaft und Zukunft.” *IT business* 2: 9-11.
https://www.parldigi.ch/wp-content/uploads/2017/07/DigitaleNachhaltigkeit_ITbusiness2017.pdf.
- Wilkinson, M. D. et al (2016): “The FAIR Guiding Principles for scientific data management and stewardship.” *Scientific Data* 3, <https://doi.org/10.1038/sdata.2016.18>.



Vielen Dank!

Kontakt:

ulrike.henny-krahmer@uni-rostock.de