

Nachhaltige Softwareentwicklung

Von der Inhouse-Lösung zur Open Source-Community am Beispiel von MerMEId

Henny-Krahmer, Ulrike

ulrike.henny-krahmer@uni-rostock.de
Universität Rostock, Germany

Stadler, Peter

pstadler@mail.uni-paderborn.de
Universität Paderborn, Germany

Einleitung

Die nachhaltige Entwicklung und Verfügbarmachung von Forschungssoftware ist eine der zentralen Herausforderungen in den Digital Humanities. Während Praktiken der Überlieferung und Sicherung von Forschungsdaten schon einen gewissen Reifegrad erreicht haben, steckt die Kultur der Entwicklung eines digitalen Gedächtnisses in Bezug auf Software noch in den Anfängen (Katerbow 2018: 5-6). Nachhaltige Software soll ermöglichen, dass Forschung transparent und nachvollziehbar bleibt, indem Tools und Dienste langfristig auffindbar, einsehbar und möglichst ausführbar bleiben und gut dokumentiert sind. Auch geht es darum, Software nachnutzbar zu machen, damit Mittel für ihre Entwicklung effizient eingesetzt werden. Die Zielstellung nachhaltiger Software ist damit eng mit den Ideen verknüpft, die auch hinter den FAIR-Prinzipien stehen (Lamprecht et al. 2020, Hasselbring et al. 2020). Wie nachhaltig Forschungssoftware ist und sein kann, wird durch viele Faktoren bedingt, u.a. durch technische, organisatorische sowie politisch-soziale Aspekte. Es stellt sich die Frage, welche Kriterien dafür genau erfüllt sein müssen, wofür es bereits verschiedene Vorschläge gibt, u.a. aus einer allgemein-theoretischen Sicht (Stürmer et al. 2017), aus technischer Sicht (Druskat 2017) oder aus einer förderpolitisch motivierten Sicht (Anzt et al. 2021).

Klar ist, dass nicht jede jemals entwickelte Forschungssoftware dauerhaft lauffähig gehalten werden kann. Für etablierte und verbreitete geisteswissenschaftliche Tools wie z.B. Stylo (Eder et al. 2016), CATMA (Gius et al. 2021), ediarum (Dumont et al. 2021) oder auch MerMEId (MerMEId Community 2021) ist dies jedoch anzustreben.¹ Die Frage der Nachhaltigkeit solcher Forschungssoftware wird immer dringender, je länger die Software existiert und auch, je häufiger sie eingesetzt wird, um Forschungsergebnisse zu produzieren (zur Problematik zahlreiche softwarebasierte DH-Projekte langfristig zu erhalten siehe z.B. Smithies et al. 2019).

Dieser Beitrag zielt darauf, bestehende Kriterien für nachhaltige Softwareentwicklung am Beispiel des musikwissenschaftlichen Metadaten-Editors MerMEId zu diskutieren. Dabei wird insbesondere die Entwicklungsgeschichte der Software in den Blick genommen, da sie zunächst als Inhouse-Lösung entwickelt wurde und kürzlich in ein Community-Projekt umgewandelt wor-

den ist. Auf der Grundlage der Erkenntnisse zu MerMEId wird die Anwendung der bisher hauptsächlich fachübergreifend formulierten Nachhaltigkeitskriterien auf geisteswissenschaftliche Forschungssoftware kritisch reflektiert.

Kriterien für nachhaltige Forschungssoftware

Die Darstellung bestehender Kriterien für nachhaltige Softwareentwicklung beschränkt sich auf zwei Beispiele: auf der einen Seite praktische Auswahlkriterien für Forschungssoftware, die langfristig gefördert werden sollte (Anzt et al. 2021) und auf der anderen Seite abstraktere, allgemeine Bedingungen für die (auch soziale und ökologische) Nachhaltigkeit digitaler Artefakte und ihrer Ökosysteme (Stürmer et al. 2017). Es gibt weitere Vorschläge für Kriterien für gute oder nachhaltige Software, auf die hier nicht umfassend eingegangen werden kann. Die beiden ausgewählten Kriteriengruppen ergänzen sich durch die unterschiedliche Schwerpunktlegung gut und eröffnen je eigene Perspektiven auf die Nachhaltigkeit von Software. Sie wurden jedoch nicht speziell für geisteswissenschaftliche Forschungssoftware entwickelt.² Im Folgenden werden die von Anzt et al. formulierten Kriterien sinngemäß wiedergegeben, da für die MerMEId im Einzelnen überprüft werden soll, ob sie erfüllt sind oder nicht:



Abb. 1: Nachhaltigkeitskriterien nach Anzt et al. 2021 (eigene Darstellung).

Den Hauptteil der Kriterien bei Anzt et al. machen die Transparenz und der Qualität der Software aus. Daneben zeigen die Bereiche "Nutzung und Impact" und "Reife", dass eine dauerhafte Förderung für Anzt et al. auch davon abhängt, wie stark die Software tatsächlich genutzt wird und etabliert ist. Den recht detaillierten, direkt anwendbaren Kriterien von Anzt et al. stehen die theoretisch hergeleiteten Kriterien von Stürmer et al. (2017) gegenüber, denen das Konzept der digitalen Artefakte (Daten oder

Code) zugrunde liegt. Diese benötigen eine technische und soziale Umgebung, um verarbeitet zu werden, und sind von einem sich veränderndem Ökosystem abhängig und davon, dass sie erstellt, verändert und genutzt werden. Aus dieser Eigenschaft leiten sich die in Abb. 2 gezeigten Grundbedingungen für Nachhaltigkeit ab.

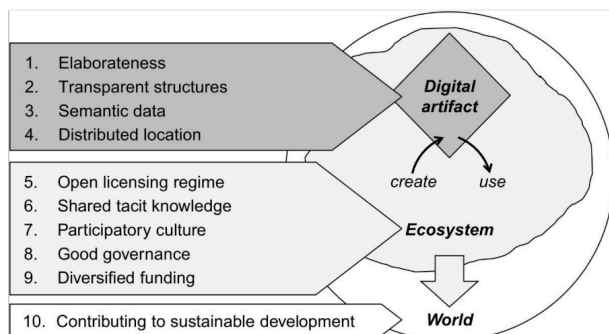


Abb. 2: Grundbedingungen für Nachhaltigkeit nach Stürmer et al. 2017.

Die Kriterien von Stürmer et al. sollen hier nicht im Einzelnen auf MerMEId angewandt werden, da sie sich in Teilen mit den Kriterien von Anzt et al. decken. Vor allem die Kriterien zum Ökosystem sind jedoch für MerMEId von Interesse, da sie den Blick verstärkt auf Aspekte der sozialen Umgebung lenken, was im Hinblick auf den Übergang der MerMEId von einer Inhouse-Lösung zu einer Community-Software besonders relevant ist. Sie werden daher in der Diskussion ergänzend berücksichtigt. Der Punkt "Shared tacit knowledge" z.B. bezieht sich darauf, dass individuelle Erfahrung und Kompetenz im Umgang mit digitalen Artefakten laufend sozial geteilt und externalisiert werden müssen, damit die Artefakte an die sich ständig ändernden Bedingungen angepasst und genutzt werden können. Dafür braucht es eine stimulierende Umgebung, was mit "Participatory culture" beschrieben wird, z.B. eine Open Source-Community, zu der gerne beigetragen wird, die aber auch durch Regeln, Normen und leitende Instanzen ("Good governance") gesteuert werden muss.

Fallbeispiel MerMEId

Die MerMEId ist ein "Metadata Editor and Repository for MEI Data", der zur Erstellung musikwissenschaftlicher (thematischer) Werkkataloge an der Königlichen Bibliothek zu Kopenhagen ab ca. 2009 entwickelt wurde. Obwohl MerMEId an erster Stelle für die eigenen Arbeiten des "Danish Centre for Music Editing" an dem "Catalogue of Carl Nielsen's Works" – dem später noch das Scheibe-Werkverzeichnis und das Hartmann-Werkverzeichnis folgen sollten – entwickelt wurde, waren doch die Entwickler Axel Teich Geertinger und Sigfrid Lundberg von Anfang an maßgeblich auch an der Gestaltung des noch jungen Datenstandards MEI beteiligt. MerMEId war dadurch frühzeitig in der MEI-Community bekannt und wurde daraufhin aktiv auf zahlreichen Workshops (z.B. bei der Edirom-Summer-School ab 2012) oder durch Konferenzbeiträge (z.B. bei der ersten MEC 2013 in Mainz) vorgestellt und verbreitet. Auch durch Axel Teich Geertingers Mitgliedschaft im MEI-Board (2015–16) und in der "Metadata and Cataloging Interest Group" gab es einen regen Austausch und wechselseitigen Einfluss zwischen dem MerMEId-Editor und dem MEI-Metadatenschema. Mehrere externe Projekte setzten daraufhin den MerMEId-Editor für die Erstellung von Werkkatalogen ein, darunter "Bruckner Online"³, die Kataloge zu Johan

Svendsen und Geirr Tveitt⁴, der "Catalogue of the Works of Frederick Delius"⁵, oder das "Bach Repertorium"⁶.

Die "Goldene Ära" von MerMEId endete dann aber 2019, als die Schließung des *Danish Centre for Music Editing* beschlossen wurde und die Entwickler keine Kapazitäten mehr für Weiterentwicklung und Pflege von MerMEId aufwenden durften.⁷ Zu diesem Zeitpunkt war der Quellcode von MerMEId bereits unter einer Apache-2.0 OpenSource-Lizenz auf GitHub veröffentlicht, es gab eine MerMEId-Sandbox (Demo-Seite), sowie eine unorganisierte (i.e. es fehlten Instrumente wie Mailinglisten, Messenger o.ä.) Community von MerMEId-Nutzer*innen.

Der *Virtuelle Forschungsverbund Edirom* 'adoptierte' daraufhin das Projekt und ziemlich schnell stießen Kolleg*innen aus der ÖAW dazu. Als weitere informelle Anfragen eingingen und Interessenten auf den Plan traten wurde klar, dass diesem breiten Interesse nur durch eine echte Community-Struktur Rechnung getragen werden konnte. Als Community-Instrumente wurden neben dem vorhandenen Code-Repository inkl. Ticketsystem und Wiki bei GitHub auch monatliche (virtuelle) Community-Meetings installiert sowie ein eigener Kanal im MEI-Slack eingerichtet. Aktuell sind die Community-Aktivitäten noch sehr auf das Refactoring des Codes konzentriert, da als erster Meilenstein ein Release einer "Community-Edition" geplant ist. Diese soll noch keine wesentlichen Feature-Neuerungen enthalten, sondern zunächst eine Umgestaltung der Softwarearchitektur zum Zwecke der besseren Wartbarkeit und des einfacheren Deployments.

Anwendung der Nachhaltigkeitskriterien auf MerMEId

Im Folgenden werden die Kriterien von Anzt et al. auf MerMEId angewandt und für jedes erfüllte Kriterium ein Punkt vergeben.

Nutzung und Impact: 3,5/5

Die meisten Kriterien zu Nutzung und Impact können für MerMEId positiv verbucht werden. Die Software wird in mehr als einer Forschungsgruppe eingesetzt (Kriterium 1) und sie ist auch die einzige Software, die das Problem eines anwenderfreundlichen Metadateneditors für MEI löst (Kriterium 4). Teilweise erfüllt ist das Kriterium 2 durch einen Aufsatz und ein Poster (Geertinger und Pugin 2011, Geertinger und Lundberg 2015). Auch das Kriterium 3 kann als teilweise erfüllt gelten, da es eine kurze Rezension von MerMEId gibt (Crandell 2015) und das Tool auf diversen Workshops vorgestellt wurde. Auch Kriterium 5 (Informations- und Lehrmaterialien) bewerten wir nur mit der halben Punktzahl, da sich im Git Repository von MerMEId zwar einige kleine Tutorials zu Spezialfällen finden, diese aber nicht als genuine Beispiele oder Tutorien für das Selbststudium gelten können.

Transparenz und Qualität der Software: 6,5/11

Für die Kriterien 6, 7 und 9 kann das Code-Repo bei GitHub als Beleg dienen, in dem u.a. auch die geforderte FLOSS-Lizenz (Apache 2.0) angegeben ist. Daneben finden sich dort auch Beispieldaten, die zusammen mit der Sandbox-Umgebung das Kriterium 11 erfüllen. Teilweise (jeweils mit 0,5 gewertet) erfüllt sind die Forderungen nach Dokumentation (zu Kriterium 8 gibt es ein User-Manual sowie ein Wiki für Entwickler*innen, in dem auch die Software-Abhängigkeiten zu eXist und Orbeon Forms beschrieben sind), sowie die Releases (Kriterium 15). Es finden

sich zwar tags in der Git-Versionsverwaltung und auch Container-Images werden bereitgestellt, allerdings gibt es keine "richtigen" Releases mit archivierbaren Binaries. Erweiterbarkeit (Kriterium 12), Interoperabilität (Kriterium 13) und Testing (Kriterium 14) müssen negativ beschieden werden, allein Kriterium 16 darf wieder positiv gewertet werden, da die Funktionalitäten von MerMEId ein Alleinstellungsmerkmal sind.

Reife: 3/5

Für MerMEId gibt es keinen Software-Management-Plan (Kriterium 17) und es kann leider auch nicht behauptet werden, dass die Software einfach zu warten wäre (Kriterium 18). Die Kriterien 19–21 wiederum werden durch das GitHub-Repositorium dokumentiert bzw. erfüllt und auch im MerMEId Slack Channel und bei den monatlichen Community-Meetings treffen sich nicht nur Entwickler*innen, sondern auch Nutzer*innen zum Austausch.

Diskussion und Fazit

Die Anwendung der Kriterien von Anzt et al. auf MerMEId funktioniert zunächst sehr gut, d.h. die Kriterien sind klar definiert und lassen sich checklistenartig beantworten. An manchen Stellen haben wir uns mit halben Punkten beholfen, da die Kriterien nur teilweise erfüllt werden konnten, uns eine komplette Negierung aber zu stark erschien. Insgesamt gibt dieser Score (13/21 -> 62%) unsere intuitive Verortung von MerMEId gut wieder: "es ist schon vieles gut, aber es gibt auch noch wesentliche Baustellen". Diese Baustellen lassen sich dank der Kriterien auch klar benennen und finden sich sowohl im Bereich der Qualität der Software als auch bei der Dokumentation.

Auffallend ist, dass sich der deutliche Bruch in der Organisation der Entwicklung (von zwei Hauptentwicklern aus derselben Institution hin zu einer internationalen Community) kaum in der Bewertung auswirkt. Obschon nicht explizit ausgeführt, würden sich die Kennzahlen kaum ändern, wenn man diese beiden Zeiträume getrennt auswerten würde. Dies mag zum einen daran liegen, dass die Community-Edition noch relativ jung ist und daher zeitlich noch keine signifikanten Änderungen bewirken konnte. Es fällt aber auf, dass die Art, wie die Entwicklung eines Softwareprojekts organisiert ist, insgesamt eine eher untergeordnete Rolle bei den Kriterien von Anzt et al. spielt. Betrachtet man die Umstellung von einer Inhouse-Lösung zur Open Source-Community mit Hilfe der Ökosystem-Kriterien von Stürmer et al., so schlägt der Systemwechsel stärker zu Buche. Eine offene Lizenzierung gab es in beiden Fällen, die übrigen vier Punkte "Shared tacit knowledge", "Participatory culture", "Good governance" und "Diversified funding" sind jedoch bei der Open Source-Community in wesentlich stärkerem Maße als bei der Inhouse-Lösung oder überhaupt erst erfüllt. Dies lässt hoffen, dass MerMEId als Community-Projekt nun gut aufgestellt ist, um nachhaltig weiterentwickelt zu werden. Allerdings werden bei Stürmer et al. als Open Source-Beispiele die Entwicklung des Linux-Kernels und Bitcoin diskutiert. Die Communities sind in diesen Fällen wesentlich größer und daher voraussichtlich stabiler als bei MerMEId, wo im jetzigen anfänglichen Stadium der Community-Entwicklung eine gute "Governance", also eine gewisse Leitung und Steuerung der Community-Prozesse, noch sehr wichtig ist.

Zur Frage der Anwendbarkeit der allgemeinen Nachhaltigkeitskriterien auf DH-Projekte wie MerMEId kann festgehalten werden, dass solche Projekte tendenziell kleiner sind, in Bezug auf Mittel, Personal und auch ihre Wirkung. Quantitative Aspekte,

wie sie bei Anzt et al. z.B. hinsichtlich Impact und Nutzung abgefragt werden, sind hier nicht unbedingt angemessen. In gleicher Weise stellt sich bei den Aspekten, die in Stürmer et al. zum Ökosystem genannt werden, die Frage, ob es bei kleineren DH-Softwareprojekten genug "kritische Masse" gibt, damit die Kriterien ihre positive Wirkung auf die Nachhaltigkeit entfalten können. Wir schließen daraus, dass es gerade bei DH-Projekten wie MerMEId essentiell ist, die weitere Entwicklung im Sinne eines Managements laufend im Blick zu behalten. So wie Anzt et al. (2021) fünfjährige Förderzyklen und Smithies et al. (2019) Managementpläne von gleicher Dauer vorschlagen, wird es auch für MerMEId erforderlich sein, "auf Sicht" zu fahren, die weitere Entwicklung der Software zu beobachten und regelmäßig zu bewerten, inwieweit Nachhaltigkeitskriterien erfüllt sind, um auf dieser Basis über die weitere aktive Entwicklung und Bewahrung der Software zu entscheiden. Neben technischen Aspekten kommt damit dem organisatorischen und sozialen Rahmen eine sehr wichtige Rolle für die Nachhaltigkeit der Softwareentwicklung zu.

Fußnoten

1. Als geisteswissenschaftliche Forschungssoftware wird hier Software verstanden, die für die Forschungsgegenstände, -daten und -methoden der Geisteswissenschaften wesentliche Funktionalitäten bereitstellt, für diese Zwecke entwickelt wurde oder entsprechend eingesetzt wird, z.B. Tools zur Textanalyse oder -annotation oder Werkzeuge zur Erfassung und Präsentation von edierten Texten und Metadaten, um nur einige Beispiele zu nennen.
2. Es gibt allerdings auch bereits spezifischere Vorschläge aus den DH, wie z.B. die "Criteria for Reviewing Tools and Environments for Digital Scholarly Editing" (Sichani und Spadini 2018) und die "Handreichung zur Rezension von Forschungssoftware in den Altertumswissenschaften" (Homburg et al. 2020), die allerdings beide die Qualität von Software im Allgemeinen adressieren und nicht primär ihre Nachhaltigkeit, auch wenn beides zusammenhängt.
3. <http://www.bruckner-online.at/> [letzter Zugriff 3. Juli 2021].
4. <https://www.musikkarven.no/english/work-catalogues/index.html> [letzter Zugriff 3. Juli 2021].
5. <https://delius.music.ox.ac.uk/catalogue/welcome.html> [letzter Zugriff 3. Juli 2021].
6. <https://www.bach-leipzig.de/de/bach-archiv/bach-repertoire> [letzter Zugriff 3. Juli 2021].
7. Die offizielle Schließung datiert vom 1. Mai 2020, die Git-Aktivitäten enden aber bereits am 13. August 2019.

Bibliographie

Anzt, Hartwig / Bach, Felix / Druskat, Stephan / Löffler, Frank / Loewe, Axel / Renard, Bernhard Y. / Seemann, Gunnar / Struck, Alexander 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]“, in: *F1000Research* 9:295 10.12688/f1000research.23224.2.

Crandell, Adam (2015): „Review of MerMEId: Metadata Editor and Repository for MEI Data, by The National Library, Danish Centre for Music Publication“, in: *Notes* 71 (3): 543-544 10.1353/not.2015.0037.

Druskat, Stephan (2017): “Kriterienbasierte Evaluation und Dokumentation technischer Nachhaltigkeit von Forschungssoftware in einem Metadatenrepositorium”, in: *DHd 2017. Digitale Nachhaltigkeit. Konferenzabstracts. Universität Bern, 13.-18. Februar 2017* 253-255 10.5281/zenodo.4622669.

Dumont, Stefan / Fehner, Martin / Grabsch, Sascha (2021): *ediarum*. <https://www.ediarum.org/>. GitHub: <https://github.com/ediarum> [letzter Zugriff 15. Juli 2021].

Eder, Maciej / Rybicki, Jan / Kestemont, Mike (2016): „Stylometry with R: A Package for Computational Text Analysis“, in: *The R Journal*. 8 (1): 107–121.

Geertinger, Axel Teich / Pugin, Laurent (2011): „MEI for bridging the gap between music cataloguing and digital critical edition“, in: *Die Tonkunst* 5 (3): 289-294.

Geertinger, Axel Teich / Lundberg, Siegfried (2015): „MerMEId: Creating Thematic Catalogues Using MEI Metadata“, in: Roland, Perry / and Kepper, Johannes (eds.): *Music Encoding Conference Proceedings 2013 and 2014*. Bavarian State Library (BSB) 122–126. URN: <http://nbn-resolving.de/urn:nbn:de:bsb:12-babs2-0000007812>.

Gius, Evelyn / Meister, Jan Christoph / Meister, Malte / Petris, Marco / Bruck, Christian / Jacke, Janina / Schumacher, Mareike / Gerstorfer, Dominik / Flüh, Marie / Horstmann, Jan (2021): *CATMA 6 (Version 6.3)*. Zenodo 10.5281/zenodo.1470118.

Hasselbring, Wilhelm / Carr, Leslie / Hettrick, Simon / Parker, Heather / Tiropanis, Thanassis (2020): „From FAIR research data toward FAIR and open research software“, in: *it - Information Technology* 62 (1): 39–47 10.1515/itit-2019-0040.

Homburg, Timo / Klammt, Anne / Mara, Hubert / Schmid, Clemens / Schmidt, Sophie Charlotte / Thier, Florian / Trognitz, Martina (2020): „Diskussionsbeitrag - Handreichung zur Rezension von Forschungssoftware in den Altertumswissenschaften / Impulse - Recommendations for the review of archaeological research software.“ *GitHub*. https://research-squirrel-engineers.github.io/Impuls_SoftwareRezensionen_DGUF/Draft.htm [letzter Zugriff 15. Juli 2021].

Katerbow, Matthias / Feulner, Georg (2018): *Handreichung zum Umgang mit Forschungssoftware*. Hrsg. von der Schwerpunktinitiative Digital Information der Allianz der deutschen Wissenschaftsorganisationen 10.5281/zenodo.1172970.

Lamprecht, Anna-Lena / Garcia, Leyla / Kuzak, Mateusz / Martinez, Carlos / Arcila, Ricardo / Martin Del Pico, Eva / Dominguez Del Angel, Victoria et al. (2020): „Towards FAIR principles for research software“, in: *Data Science* 3 (1): 37–59 10.3233/DS-190026.

MerMEId Community (2021). *MerMEId*. *GitHub.com*. <https://github.com/Edirom/MerMEId>.

Sichani, Anna-Maria / Spadini, Elena and the members of the IDE (2018): *Criteria for Reviewing Tools and Environments for Digital Scholarly Editing, version 1.0*. <https://www.i-d-e.de/publikationen/weitereschriften/criteria-tools-version-1/> [letzter Zugriff 15. Juli 2021].

Smithies, James / Westling, Carina / Sichani, Anna-Maria / Mellen, Parn / Ciula, Arianna (2019): „Managing 100 Digital Humanities Projects: Digital Scholarship & Archiving in King’s Digital Lab“, in: *Digital Humanities Quarterly* 13 (1). <http://www.digitalhumanities.org/dhq/vol/13/1/000411/000411.html> [letzter Zugriff 15. Juli 2021].

Stürmer, Matthias / Abu-Tayeh, Gabriel / Myrach, Thomas (2017): „Digital sustainability: basic conditions for sustainable digital artifacts and their ecosystems“, in: *Sustainability Science* 12 (2): 247-262 10.1007/s11625-016-0412-2.