

Bausteine Forschungsdatenmanagement
Empfehlungen und Erfahrungsberichte für die Praxis von
Forschungsdatenmanagerinnen und -managern

Die Verfügbar- und Nachnutzbarhaltung von Forschungssoftware durch Quellcodearchivierung

Herausforderungen und Möglichkeiten der Langzeitarchivierung von in
wissenschaftlicher Forschung erstellter oder verwendeter Software

Michael Höfelⁱ

2026

Zitiervorschlag

Höfel, Michael. 2026. Die Verfügbar- und Nachnutzbarhaltung von Forschungssoftware durch Quellcodearchivierung. Herausforderungen und Möglichkeiten der Langzeitarchivierung von in wissenschaftlicher Forschung erstellter oder verwendeter Software. *Bausteine Forschungsdatenmanagement. Empfehlungen und Erfahrungsberichte für die Praxis von Forschungsdatenmanagerinnen und -managern* Nr. 1/2026: S. 2-20. DOI: [10.17192/bfdm.2026.1.8948](https://doi.org/10.17192/bfdm.2026.1.8948).

Dieser Beitrag steht unter einer
[Creative Commons Namensnennung 4.0 International Lizenz \(CC BY 4.0\)](https://creativecommons.org/licenses/by/4.0/).

ⁱHochschule Darmstadt – University of Applied Sciences, S.6 Bibliothek, Bereich Hochschularchiv. ORCID: [0009-0009-2056-0914](https://orcid.org/0009-0009-2056-0914)

1 Abstract

Die Rufe innerhalb der wissenschaftlichen Community, die Replizierbarkeit und z.T. auch Rezipierbarkeit von Forschungsdaten durch die Bewahrung des Zugriffs auf sowie der Lauffähigkeit von Forschungssoftware sicherzustellen, sind in den vergangenen Jahren spätestens mit der vielseitig beklagten sogenannten ‚replicability crisis‘ der Wissenschaft lauter und zahlreicher geworden. Allerdings weisen diese Appelle mehrheitlich einen blinden Fleck auf: Sie formulieren – nachvollziehbare wie unterstützenswerte – Anforderungen an den Erhalt von Zugang und Lauffähigkeit der Softwares, blenden jedoch allzu oft die technischen, organisatorischen, personellen und finanziellen Herausforderungen aus, die mit einer Nachnutzbarhaltung der Programme über einen langen, mitunter unbefristeten Zeithorizont verbunden sind. Der vorliegende Beitrag will zu einer archivfachlich fundierten, zugleich kritischen Sensibilisierung für diese Thematik beitragen, wobei der Fokus auf der Langzeitarchivierung, d.h. der dauerhaften vertrauenswürdigen Sicherung von Quellcode von Forschungs- und Vermittlungssoftware liegen soll. Zunächst werden Vorzüge der Archivierung von Quellcode beschrieben, die aus dessen ‚Klartext‘-Charakter resultieren, beschrieben (geringer Speicherplatzbedarf, fehlender Migrationsaufwand, Maschinen- und Menschenlesbarkeit), aber auch die intellektuellen Herausforderungen für das Archivpersonal (Stichwort ‚Coding Literacy‘). In einem weiteren Schritt wird der Frage nachgegangen, welche potenziellen Probleme sich für die Sicherstellung der Lauffähigkeit langzeitarchivierter Software aus dem Textcharakter von Quellcode ergeben und inwiefern durch Quellcodearchivierung Emulationsbedarf lediglich in die Zukunft verlagert wird. Zuletzt wird auf den erhaltungsstrategischen Stellenwert von Quellcodeerhalt für die Langzeitarchivierung von Forschungs- und Vermittlungssoftware hingewiesen – sowohl als Baustein in einem ganzheitlichen Erhaltungskonzept wie auch als ‚Standalone‘-Strategie.

Schlagwörter: Forschungssoftware, Langzeitarchivierung, Softwarenachhaltigkeit, Open Research Software, Reproduzierbarkeit, Replizierbarkeit, Validierbarkeit

Keywords: research software, long-term preservation, software sustainability, open research software, reproducibility, replicability, validity

2 Einleitung¹

Es ist unbestrittener Konsens – und im Grunde ein Gemeinplatz – in der wissenschaftlichen Community und über alle Grenzen der unterschiedlichsten Fachdisziplinen hinweg, dass die dauerhafte Verfügbarhaltung von Forschungsdaten das Fundament für eine nachhaltige Wissenschaft wie auch für das Vertrauen in diese bildet. Zu dieser ungetrübten Einigkeit haben sich spätestens ab den frühen 2010er Jahren, ma-

¹Alle Hyperlinks im vorliegenden Text zuletzt geprüft am 30.09.2025.

nifestiert etwa in der Diskussion um die sogenannte „Reproduktionskrise“ in der Psychologie, zunehmend Stimmen gesellt, die die Ermöglichung einer Nachnutzung von Forschungssoftware einforderten, etwa durch die Offenlegung von Quellcode (auch: Quelltext).² Vor diesem Hintergrund initiierten engagierte Wissenschaftler*innen eine Adaption der 2016 veröffentlichten „FAIR Guiding Principles for scientific data management and stewardship“ (Wilkinson et al., 2016) für Forschungssoftware.³ Die Essenz der Apelle und Initiativen: Entfallen der Zugang zu und die Lauffähigkeit von Software, die zur Erhebung, Verarbeitung und Aufbereitung von im Rahmen von Forschung generierten Daten eingesetzt worden ist, gestaltet sich die zuverlässige Überprüfung der Daten zu einem späteren Zeitpunkt unmöglich – und damit auch die Möglichkeit, Defizite in der Datenqualität, etwa aufgrund technischer Mängel in der verwendeten Software oder wissenschaftlichen Fehlverhaltens, nachträglich belegen zu können. Diese Forderungen sind nachvollziehbar und begrüßenswert, stellen aber die die Nachnutzung verantwortenden Einrichtungen vor nicht zu unterschätzende technische, organisatorische, rechtliche, personelle und finanzielle Herausforderungen – insbesondere, wenn die Zugriffs- und Lauffähigkeitserhaltung dauerhaft gesichert, d.h. die betreffende Software **langzeitarchiviert** werden sollen. Diese Herausforderungen scheinen durch eine Archivierung der Quellcodes der genutzten Softwares anstelle ihrer ausführbaren Repräsentationen in Maschinensprache minimiert; allerdings sieht sich auch diese Erhaltungsstrategie mit langzeitarchivischen Herausforderungen konfrontiert, die aber in der Praxis für gewöhnlich nicht oder nur unzureichend kommuniziert werden.

Der vorliegende Beitrag möchte aktiv zur Füllung dieses weißen Flecks auf der fachdiskursiven Landkarte beisteuern, indem die wesentlichsten Möglichkeiten und Herausforderungen einer Langzeitarchivierung von Quellcode von Forschungs- und Vermittlungssoftware⁴ als Voraussetzung ihrer Nachnutzung kritisch und archivfachlich fundiert geprüft werden. Der Fokus soll hierbei auf technisch-organisatorischen sowie personell-kompetenzbezogenen Kriterien liegen, die sich ausschließlich aus der Quellcodearchivierung ergeben; auf Erörterungen der den dauerhaften Erhalt von Programmen allgemein betreffenden Herausforderungen wird verzichtet. Zugleich soll

²Für die frühen 2010er Jahre siehe z.B. Barnes (2010), Peng (2011), Ince et al. (2012), Morin et al. (2012); für die letzten Jahre siehe u.a. Arbeitskreis Open Science der Helmholtz-Gemeinschaft (2017), Jiménez et al. (2017), Scheliga et al. (2017), Röwenstrunk (2018), Helmholtz Open Science Office (2019), Hörnschemeyer (2019), Scheliga et al. (2019), Anzt et al. (2021), Appel & Loewe (2021), Brecher et al. (2022), Suchodoletz et al. (2023). Vgl. hierzu ferner Deutsche Forschungsgemeinschaft (10.10.2023): Ein sorgfältiger Umgang und „eine fachlich adäquate Nachnutzungsmöglichkeit“ müsse beim Umgang mit Forschungsdaten mitgedacht werden, „wenn Software für die Entstehung oder Verarbeitung von Forschungsdaten erforderlich ist“. Spannend ist in diesem Zusammenhang ein 2021 vorgestellter „Workflow for Open Reproducible Code in Science“ (WORCS), der sich durch ein in der Programmiersprache R realisiertes Package technisch implementieren lässt (Van Lissa et al., 2021; Van Lissa et al., 2025). – Für die Debatte um die sogenannte „Reproduktionskrise“ in der Psychologie vgl. u.a. Pashler & Wagenmakers (2012), Pashler & Harris (2012), Stroebe & Strack (2014).

³Z.B. Lamprecht et al. (2020), Barker et al. (2022), Chue Hong et al. (2022), Martinez-Ortiz et al. (2022), Barker et al. (2024). Zu den FAIR-Prinzipien siehe v.a. *FAIR Principles* (o. J.).

⁴Zur Definition von Forschungs- und Vermittlungssoftware siehe Abschnitt 2.1.

mit dem vorliegenden Beitrag die bereits stattfindende gegenseitige Annäherung von Forschung, wissenschaftsunterstützenden Infrastrukturen und Archiven⁵ gestärkt und die damit verbundene Diskussion im deutschsprachigen Raum um die zusätzliche Dimension der Langzeitarchivierung von Forschungssoftware(Quellcode) ergänzt werden.

3 Definitionen

3.1 Forschungsdaten – Forschungssoftware – Vermittlungssoftware

Der Terminus „Forschungsdaten“ wird mittlerweile derart selbstverständlich in verschiedenen Kontexten gebraucht, dass die Herausarbeitung einer einheitlichen und abgestimmten Definition für die Praxis sich schwierig gestaltet.⁶ Im vorliegenden Beitrag werden unter dem Begriff all jene Daten subsumiert, „die während wissenschaftlicher Tätigkeit (z.B. durch Messungen, Befragungen, Quellenarbeit) entstehen“ und als solche „eine Grundlage wissenschaftlicher Arbeit“ bilden sowie deren Ergebnisse dokumentieren (forschungsdaten.info, o. J.).⁷

Der im fachlichen Diskurs und darüber hinaus nicht weniger unscharf verwendete Begriff **Forschungssoftware** meint „explizit für die Zwecke von Forschung erstellte Software bzw. Software, die Funktionen für wissenschaftliche Analysen, Berechnungen oder andere Formen der wissenschaftlichen Datenverarbeitung bereitstellt“ (Suchodoletz et al., 2023, S. 6). Hierzu zählen z.B. Softwares von Rasterelektronenmikroskopen oder Elektronische Laborbücher (engl. *Electronic Laboratory Notebooks*, kurz: ELNs). Davon zu unterscheiden sind „alle diejenigen digitalen (Hilfs-)Werkzeuge, die zusätzlich im Kontext eines Forschungsvorhabens genutzt werden“, etwa Editoren, Bildbearbeitungsprogramme oder Softwares zur Verarbeitung von Texten oder Tabellen (Suchodoletz et al., 2023, S. 6). Zusätzlich lässt sich Forschungssoftware auch hinsichtlich ihres Stellenwertes innerhalb eines konkreten Forschungsvorhabens ausdifferenzieren: als Ergebnis wissenschaftlicher Forschung in Form statischer Daten

⁵Für den deutschen Raum vgl. v.a. Maier et al. (2020), Naumann & Neuburger (2022), Simon (2022), Naumann et al. (2023), Donig et al. (2024), Naumann et al. (2024). Bereits vor dreizehn Jahren hat Nippert (2013) auf die Forschungsdatenarchivierung als Arbeitsgebiet von Archiven von Hochschulen und wissenschaftlicher Institutionen hingewiesen; knapp zehn Jahre später bezeichnet Gastl-Hartmann (2022) die Bewertung und Archivierung von Forschungsdaten unverändert als „neues Arbeitsgebiet der Hochschularchive“.

⁶Vgl. z.B. Kaden & Kleineberg (2018, S. 67): „Gerade in geistes- und kulturwissenschaftlichen Forschungsfeldern bestehen Unsicherheiten, ob hierbei auch Quellen- bzw. Archivmaterialien mit eingeschlossen werden können“.

⁷Vgl. ergänzend auch die Definition der Deutschen Forschungsgemeinschaft (2015, S. 1): „Forschungsdaten sind eine wesentliche Grundlage für das wissenschaftliche Arbeiten. Die Vielfalt solcher Daten entspricht der Vielfalt unterschiedlicher wissenschaftlicher Disziplinen, Erkenntnisinteressen und Forschungsverfahren. Zu Forschungsdaten zählen u.a. Messdaten, Laborwerte, audiovisuelle Informationen, Texte, Surveydaten, Objekte aus Sammlungen oder Proben, die in der wissenschaftlichen Arbeit entstehen, entwickelt oder ausgewertet werden.“

oder als digitales Werkzeug zur Generierung und/oder Verarbeitung von Forschungsdaten (Röwenstrunk, 2018, S. 303).⁸

Von Forschungssoftware zu unterscheiden ist **Vermittlungssoftware**, „die für die Publikation, Darstellung, Aufbereitung oder das Verständlichmachen von Forschungserkenntnissen eingesetzt wird oder gar notwendig ist“ (Röwenstrunk, 2018, S. 303), etwa digitale historisch-kritische Editionen literarischer oder musikalischer Werke (Abb. 1). Wie jede andere Form von EDV-Programmen kann auch Forschungs- und Vermittlungssoftware **quelloffene** oder **nichtquelloffene, i.d.R. herstellergebundene** Software sein (Suchodoletz et al., 2023, S. 6f).

Forschungssoftware	(Hilfs-)Werkzeuge	Vermittlungssoftware
• Elektronische Laborbücher • Software für Rasterelektronenmikroskope • Messsoftware für Wettersatelliten • ...	• Integrierte Entwicklungsumgebungen (IDE) • Bildbearbeitungsprogramme • Tabellenkalkulationsprogramme • ...	• digitale Editionen • Datenviewer • Style- und Transformations-sheets • ...

Abbildung 1: Forschungssoftware, (Hilfs-)Werkzeuge, Vermittlungssoftware (eigene Darstellung)

3.2 Digitale Langzeitarchivierung und Forschungssoftwarearchivierung – archivfachlich erläutert

Der Begriff „(Langzeit-)Archivierung“ wird außerhalb der archivarisches Community nicht selten synonym zu einer mittel- bis längerfristigen, letztlich aber doch zeitlich limitierten Speicherung von Daten aller Art verstanden und verwendet. Auch im Bereich des Forschungsdatenmanagements fällt der Begriff mitunter dort, wo von einer definitiv oder potenziell befristeten Datenvorhaltung die Rede ist.⁹ Im archivfachlichen wie im legistischen Sinne meint „Archivierung“ die Selektion von Informationen von bleibendem Wert (Archivwürdigkeit) anhand transparenter Auswahlkriterien (Bewertung) sowie deren Übernahme, Erhaltung und dauerhafte Aufbewahrung mit dem Ziel ihrer bedarfsweisen Bereitstellung zum Zwecke der Nachnutzung, unabhängig von ihrem

⁸Vgl. auch Deutsche Forschungsgemeinschaft (2015, S. 1): „Methodische Testverfahren, wie Fragebögen, Software und Simulationen können ebenfalls zentrale Ergebnisse wissenschaftlicher Forschung darstellen und sollten daher ebenfalls unter den Begriff Forschungsdaten gefasst werden“ (Hervorhebung durch M.H.).

⁹Vgl. z.B. *Policy für Forschungsdatenmanagement an der Universität Wien* (2021, S. 1): „Insbesondere sollen Dienstleistungen für Forscher*innen angeboten werden, die es ermöglichen, Forschungsdaten, die Publikationen zugrunde liegen, zu *archivieren*. Dies entspricht einer *zeitlich definierten oder langfristigen Speicherung* von Forschungsdaten, so dass auf diese zugegriffen werden kann und diese genutzt werden können“ (Hervorhebungen durch M.H.).

Trägermaterial und ihrer Speicherungsform¹⁰; im Kontext archivwürdiger digitaler Daten ist gemeinhin von **digitaler Langzeitarchivierung** (dLZA) die Rede.¹¹ Diese folgt, wie Archivierung im Allgemeinen, bestimmten Standards (z.B. OAIS-Referenzmodell¹², PREMIS¹³) und Normen (z.B. DIN 31644¹⁴), die die Auffindbarkeit, Verfügbarkeit, Authentizität, Integrität und Nachnutzbarkeit digital archivierter Informationen gewährleisten sollen. Davon zwingend abzugrenzen ist die zeitlich begrenzte Vorhaltung von Informationen, die nach Wegfall eines Speichergrunds oder einer Speichernotwendigkeit für gewöhnlich gelöscht werden – etwa nach Ablauf der für die Datenvorhaltung vorgesehenen Frist, die im Sinne einer Sicherung der guten wissenschaftlichen Praxis empfohlen wird (Deutsche Forschungsgemeinschaft, 2025). Archivgut hat damit kein ‚Ablaufdatum‘, sondern ist eine auf rechtlicher Grundlage und durch fachlich ausgebildetes Personal – Archivar*innen – nach transparenten Kriterien gebildete Auswahl von dauerhaft erhaltenswürdigen Informationen. Folglich meint **Forschungssoftware-rearchivierung** die unbefristete Aufbewahrung von Forschungssoftware (und in einem weiteren Sinne auch von Vermittlungssoftware), deren Erhalt aus insbesondere rechtlichen und/oder wissenschaftlichen Gründen anzustreben ist.

4 Langzeitarchivierung von Forschungssoftware-Quellcode: Herausforderungen und Möglichkeiten

4.1 Quellcodearchivierung ist Textarchivierung

Die Langzeitarchivierung von Quellcode erhaltungsintendierter Software – nicht nur von Forschungs- und Vermittlungssoftware – hat gegenüber einer Vorhaltung in

¹⁰Siehe hierzu z.B.: Menne-Haritz (2001); Schenk (2014), S. 82–87; Keitel (2018), S. 99–121; Tiemann (2018); Schwarz (2023), S. 95–97; Shankar & Cushing (2024); VdA Arbeitskreis archivische Bewertung & Wendenburg (2025). Für eine legistische Definition von „Archivierung“ und „Archivwürdigkeit“ siehe, durchaus stellvertretend für die Archivgesetzgebung in Deutschland, § 2 Abs. 1 u. 3 Hessisches Archivgesetz (HArchivG), online einsehbar unter Hessisches Ministerium für Wissenschaft und Kunst (2022).

¹¹Der Begriff „(digitale) Langzeitverfügbarkeit“ hingegen ist im Archivwesen kaum gebräuchlich – nicht zuletzt, da Verfügbar- und Nachnutzbarhaltung zentrale Zielsetzungen von Archivierung darstellen (vgl. z.B. § 2 Abs. 1 HArchivG). Töwe (2025) verweist darauf, dass der Begriff „Langzeitverfügbarkeit“ gegenüber „Langzeitarchivierung“ den Vorzug aufweist, archivfachfremden Personen (z.B. Forschenden) deutlicher die Intention der Erhaltungsbemühungen näherzubringen – ein Argument, das nicht von der Hand zu weisen ist.

¹²Aktuelle Version publiziert als The Consultative Committee for Space Data Systems (2024), im März 2025 standardisiert als ISO 14721:2025-03. Die deutsche Übersetzung der inzwischen vorletzten Version von 2012 (ISO 14721:2012-09) bieten nestor-Arbeitsgruppe OAIS-Übersetzung / Terminologie (2013) und Schrimpf (2014). Zum OAIS-Referenzmodell siehe auch u.a. Schrimpf (2012) sowie Weber & Piesche (2021), S. 342–346.

¹³PREMIS Editorial Committee (2015). Zu PREMIS siehe u.a. Dappert et al. (2018).

¹⁴Enthalten und kommentiert in Keitel & Schoger (2013). Zur DIN 31644 siehe auch Keitel (2012).

Maschinen- oder Bytecode kompilierter bzw. assemblierter Programme mehrere Vorteile. Zunächst ist Quellcode **unformatierter Text** – und damit geradezu prädestiniert für eine digitale Langzeitarchivierung. Im Vergleich zu Maschinencode benötigt Quelltext grundsätzlich weniger Speicherplatz¹⁵ und bedarf – so zumindest der aktuelle fachliche Wissenstand – keiner weiterer Formatmigrationen, die zusätzliche Repräsentationen generieren, deren Langzeitarchivierung ihrerseits wiederum Speicherplatz einfordert.¹⁶

Ein weiterer Vorzug von Quellcode ist seine **Maschinenlesbarkeit**, die unterschiedliche Möglichkeiten der (teil- oder voll)automatisierten Weiterverarbeitung erlaubt, z.B. die Analyse der Funktionalitäten des betreffenden Programms oder seine Übersetzung in eine andere Programmiersprache (Codemigration). In diesem Zusammenhang nicht außer Acht gelassen werden darf das gegenwärtige wie auch perspektivische Potential, das Künstlicher Intelligenz (KI) im Hinblick auf Analyse, Interpretation und Weiterverarbeitung von Quellcodes unterschiedlicher Programmiersprachen innewohnt und das sich Software langzeitarchivierende Einrichtungen zunutze machen können.¹⁷

Nicht unterschlagen werden darf die Eignung von Quellcode zur **Menschenlesbarkeit**. Aufgrund seines ‚Klartext‘-Charakters werden keine definierten Softwarekomponenten – eine bestimmte Familie grafischer Betriebssysteme, ausgewählte proprietäre Visualisierungsprogramme usw. – benötigt. Stattdessen genügen ein schlichtes Programm zur Textwiedergabe, z.B. handelsübliche Editoren, die z.T. auch als Open-Source-Software zur Verfügung stehen, oder basale Kenntnisse einer Programmiersprache zum Auslesen von Textdateien. Hinzu kommt das Abstraktionsniveau vieler sogenannter höherer Programmiersprachen, deren Syntaxen englischsprachige Schlüsselwörter enthalten (z.B. if, else, while, print, import) oder sich der (englischen) Alltagssprache annähern und somit das ‚Lesen‘ von Quellcode grundsätzlich erleichtern.¹⁸

¹⁵Nicht unterschlagen werden dürfen an dieser Stelle die Herausforderungen, die der Import von Standard- und Drittmodulen in Quellcodes langzeitarchivierter Software für deren Nachnutzbarkeit bedeuten (vgl. auch Anm. 22); dies schließt einen erhöhten Speicherplatzbedarf ein.

¹⁶Auch Van Lissa et al. (2021, S. 35) plädieren für die Bereitstellung textbasierter anstelle binärer Dateien. Zwar wird diese Empfehlung vor dem Hintergrund einer Speicherung in Git-basierten Repositorien (z.B. GitHub) ausgesprochen, lässt sich aber für eine Langzeitverfügbarkeit nach archivfachlichen Kriterien grundsätzlich abstrichfrei adaptieren – sofern nicht andere Gründe für eine Archivierung in nichttextbasierter Form sprechen.

¹⁷Für Large Language Model-gestützte Codemigration am Beispiel des Chatbots ChatGPT siehe Banzer (2023). Auch existieren bereits mehrere KI-basierte Code-Review-Tools zur automatischen Analyse von Quellcodes auf Bugs, Sicherheitsdefizite, Performance u.ä. (z.B. Malik, 2025).

¹⁸Ein anschauliches Beispiel für Letzteres stellt LiveCode Script dar, die objektorientierte Programmiersprache der Entwicklungsumgebung LiveCode (*LiveCode script language*, o. J.). So folgt etwa die Zuweisung eines Wertes an eine Variable in LiveCode Script dem Notationsmuster "put [value] into [variable]", im Unterschied zur in vielen verbreiteten Programmiersprachen (z.B. Java, C, C++, Python, JavaScript, PHP, Swift) gängigen, einer mathematischen Gleichung ähnelnden Notationsform "[variable] = [value]". Die Folgen des zunehmenden Einsatzes von unter dem Schlagwort „fourth-generation programming language“ bzw. „4GL“ subsumierten Programmiersprachen und -umgebungen (z.B. „Fourth-generation programming language“, 2025) auf die Langzeitarchivierung

4.2 Coding Literacy – eine künftige Historische Hilfswissenschaft für Archivar*innen?¹⁹

Die Menschenlesbarkeit von Quellcode bezieht sich zunächst auf die Art und Weise, wie der Text gespeichert wird und wiedergegeben werden kann, nämlich als endliche Folge rezipierbarer Zeichen aus einem definierten Zeichensatz. Für seine intellektuelle Durchdringung bedarf es jedoch mehr als technischer Rezeptionsmöglichkeiten (z.B. Wiedergabegeräte und -softwares): Quellcode unterliegt als eine in einer formalen Sprache abgefasste Sammlung von durch den Computer auszuführenden Anweisungen bestimmten Regeln, deren Gesamtheit die Syntax einer Programmiersprache bildet. Kenntnisse dieser Syntax sowie grundlegendes Wissen über Softwareerstellung sind folglich essenziell für das Verständnis der zu langzeitarchivierenden Programme. Eine solche ‚Coding-Alphabetisierung‘ des kuratierenden Archivpersonals bildet damit eine wesentliche Voraussetzung für die erfolgreiche Langzeitarchivierung von Forschungs- und Vermittlungssoftware, durchaus vergleichbar mit paläografischen Kenntnissen älterer Druck- und Schreibschriften, die in ‚klassischen‘ Archiven aufgrund des in diesen aufbewahrten älteren Schriftgutes zum notwendigen Kompetenzarsenal zählen. Zwar können technologische Fortschritte wie KI-gestützte Tools – zur Transkription älterer Schreib- und Druckschriften wie auch zur Analyse von Quellcodes unterschiedlicher historischer sowie rezent genutzter Programmiersprachen – Archivar*innen unterstützen und entlasten, eine grundlegende „Coding Literacy“ (Vee, 2017) aber nicht ersetzen.²⁰

Die menschenmögliche (bzw. -zumutbare) Rezipierbarkeit von Forschungs- und Vermittlungssoftware kann ferner von dem Umfang des Quellcodes, der Masse und Heterogenität der die Software bildenden Einzeldateien²¹ sowie einer etwaigen Kombination mehrerer Programmiersprachen zur Softwarerealisation abhängig sein. So können sogar Softwares mit überschaubarem Funktionsumfang aus zahlreichen, über eine mehrstufige Verzeichnishierarchie verteilten Einzeldateien bestehen, die über Importanweisungen in einem komplexen Beziehungsgeflecht zueinander stehen.²² Selbst

von (nicht nur Forschungs- und Vermittlungs-)Software ist eine eigene Untersuchung wert. – Die Beispiele für englischsprachige Schlüsselwörter sind Python 3.x entlehnt.

¹⁹Unter „Historische Hilfswissenschaften“ oder „Historische Grundwissenschaften“ kann grundsätzlich jede wissenschaftliche (Teil-)Disziplin subsumiert werden, deren Ergebnisse sich für historische Forschungen nutzen lassen. Im engeren Sinne wird darunter eine Gruppe historischer Disziplinen verstanden, zu deren Kanon u.a. Paläografie (Schriftenkunde), Diplomatik (Urkundenlehre), Aktenkunde, Chronologie (Lehre der Zeitrechnung), Heraldik (Wappenkunde), Sphragistik (Siegelkunde), Numismatik (Münz- und Geldkunde) und Genealogie (Familien- und Abstammungskunde) zählen (z.B. Brandt, 2012; Rohr, 2015; in kompakter Repetitoriumsform auch Henning, 2011).

²⁰Die Notwendigkeit einer grundlegenden Coding Literacy ist bereits seit mehreren Jahren im Archivwesen, wo etwa XML-basierte Formate (z.B. METS, EAD, EAC) aus dem Bereich der digitalen Langzeitarchivierung nicht mehr wegzudenken sind, signifikant.

²¹Hier ist v.a. an Quellcode-, Konfigurations- und proprietäre Programmbibliotheksddateien zu denken. Darüber hinaus sind aber, abhängig von der betreffenden Software, auch Dokument-, Medien- und weitere Dateien denkbar.

²²Durch den Import von Standard- und Drittmodulen der verwendeten Programmiersprache(n) kann

unter der Prämisse, der Dateiumfang der zu archivierenden Software bestünde ausschließlich aus grundsätzlich menschenles- und -zumutbaren Datenkonvoluten, kann die Langzeitarchivierung des betreffenden Programms eine mitunter enorme intellektuelle Rezeptionsherausforderung darstellen.

4.3 Quellcodearchivierung als Emulationsprokrastination?

Emulation als „software-driven strategy“ (Rechert, 2022) stellt, neben der Migration von Daten in ein langzeitarchivfähiges Format, ein zentrales Instrument im Werkzeugkasten der digitalen Langzeitarchivierung dar²³ – wenngleich auch mehr in theoretisch-fachdiskursiver Hinsicht als mit Blick auf ihre Implementierungshäufigkeit. Dies hat mehrere Gründe: Zum einen stehen bei Archiven für gewöhnlich der Erhalt selektierter Inhalte sowie die Gewährleistung ihrer Authentizität und Integrität im Zentrum langzeitarchivischer Erhaltungsbemühungen; schrankenlose Totalarchivierung findet nicht statt. Die Authentizität und Integrität der archivierten Informationen haben Archive durch geeignete Übernahmestrategien und -techniken sowie eine lückenlose Dokumentation des tatsächlichen Akzessionsprozederes sicherzustellen.²⁴ Auch sind die Softwares, die der Verwaltung und Bearbeitung der Inhalte dienen (z.B. Dokumentenmanagementsysteme, datenbankgestützte Fachverfahren), selbst in Aufbau, Funktionsumfang und Erscheinungsbild i.d.R. nicht archivwürdig. Zum anderen stehen die technischen, organisatorischen, (software)rechtlichen, personellen und finanziellen Herausforderungen von Emulation als Erhaltungs- und Nutzungsstrategie in den beschriebenen Fällen in keinem verfechtungsfähigen Verhältnis zum Nutzen: Wozu ein ganzes Dokumentenmanagementsystem (DMS) mitsamt aller für den Betrieb notwendigen Software- und Hardwarekomponenten konservieren, wenn die im Zentrum der Erhaltungsbemühungen stehenden Inhalte – in diesem Falle digitale Unterlagen samt Metadaten – sich ohne Archivierung des DMS signifikant ressourcenschonender und vergleichsweise aufwandsärmer erhalten und auch wieder bereitstellen lassen?²⁵

sich die Beziehungskomplexität zusätzlich potenzieren. Greift etwa eine Software auf Module aus der Standardbibliothek der verwendeten Programmiersprache(n) zurück, sind diese Module für das Verständnis wie auch den Erhalt von Lauffähigkeit und Nachnutzbarkeit der zu erhaltenden Software gleichfalls essenziell.

²³Vgl. z.B. Borghoff et al. (2006), S. 57–78; Suchodoletz (2009); Funk (2010); Sandford (2012); Kramski (2016); Espenschied (2024). Gelungene kompakte Gegenüberstellungen von Migrations- und Emulationsstrategie im Hinblick auf Prozesse und Performance bietet Keitel (2018), S. 115 u. 142–143.

²⁴Abhängig von der Art der Daten und der Software sowie einer etwaigen Akzessionsperiodizität (elektronische Akten z.B. werden für gewöhnlich regelmäßig übernommen, bei manchen datenbankgestützten Fachanwendungen wiederum kann eine einmalige Datenübernahme vor der irreversiblen Löschung der Software erfolgen) kann eine solche Dokumentation z.B. das Schema der betreffenden Datenbank, die der Datenaussonderung zugrundeliegende Datenbankabfrage (bspw. SQL-Statement) und/oder der Quellcode für Skripte zur (teil- oder voll)automatisierten Aufbereitung der ausgesonderten Daten für deren Langzeitarchivierung umfassen.

²⁵Einen bestandserhalterischen wie überlieferungsbildnerischen Kontrapunkt hierzu bildet die Erhaltung z.B. von Computer- und Videospielen sowie deren Hardware-Voraussetzungen (vgl. hierzu z.B. Viert, 2023): Hier liegt der Fokus nicht auf der dauerhaften Sicherung von durch diese Programme

Demgegenüber wird das im vorliegenden Beitrag eingangs erwähnte Plädoyer für die Erhaltung der Zugriffs- und Lauffähigkeit von Forschungs- und Vermittlungssoftware mit einer Überlieferungsnotwendigkeit begründet: Ohne die Software, mit der Forschungsdaten generiert worden sind oder in verständlicher Form aufbereitet werden, sei eine Replizierbarkeit und ggf. auch Rezipierbarkeit nicht möglich. Vor dem Hintergrund der technisch-organisatorischen Herausforderungen von Emulation sowie des langzeitarchivkonvergenten Textcharakters von Quelltext erscheint die Archivierung von Forschungs- und Vermittlungssoftware durch Quellcodeerhalt als reizvoller Zugang, da sie die Schwierigkeiten von Emulation zu umschiffen scheint. Dies trifft jedoch nur auf den ersten Blick zu: Da Quellcode nicht von sich aus lauffähig ist – es bedarf je nach verwendeter Programmiersprache eines Interpreters, Compilers oder Assemblers sowie weiterer damit kompatibler Software- und Hardwareressourcen (z.B. Bibliotheken, Betriebssystem), um das betreffende Programm ausführen zu können bzw. ausführbar zu machen –, scheint die Notwendigkeit zur Emulation lediglich in die Zukunft prokrastiniert. Dies gestaltet sich erhaltungsstrategisch als zweischneidiges Schwert: Einerseits kann der rasche informationstechnologische Fortschritt Möglichkeiten einer nachträglichen Lauffähigkeitsgewährleistung schaffen, die zum gegenwärtigen Zeitpunkt nicht oder nur unzureichend absehbar sind. Zugleich erhöht aber eben dieser Fortschritt, in Kombination mit der zunehmenden Komplexität digitaler Systeme und mit zunehmender zeitlicher Distanz zur ‚aktiven Lebensphase‘ des betreffenden Programms, die technischen Anforderungen an eine spätere Rekonstruktion und Emulation der notwendigen Hardware- und Softwarekomponenten – und potenziert folglich das Risiko einer faktisch verunmöglichten ‚Lebenserhaltung‘ oder ‚Reanimation‘ der erhaltungsintendierten Software in der Zukunft.

Von einer grundsätzlichen bestandserhalterischen ‚Aufschieberitis‘ kann jedoch nur bedingt die Rede sein: Während in Maschinencode vorliegende Software an bestimmte technische Rahmenbedingungen (z.B. Betriebssystem, Laufzeitumgebungen, Prozessoren) gebunden ist, die in weiterer Folge emuliert werden müssen, erlaubt eine Quellcodearchivierung – zumindest theoretisch und potenziell perspektivisch – eine Anpassung der Software an künftige technische Gegebenheiten sowie – eine entsprechende lückenlose Dokumentation sowie der archivische Zugriff auf notwendige IT-Kompetenz vorausgesetzt – ihre nachhaltige Rezipierbarhaltung.

generierten Inhalten, sondern auf dem Erhalt des intrinsischen Wertes elektronischer Spiele (und ggf. ihrer Peripherie, d.h. der zur Rezeption der Spiele notwendigen Hardwarekomponenten), der sich in der individuellen spielerischen Auseinandersetzung mit diesen artikuliert. Eine zusätzliche Sicherung der von diesen Spielen erzeugten Informationen (z.B. Highscore-Listen, Speicherstände, Spielerprofile, mit Karten- bzw. Leveleditoren kreierte zusätzliche Umgebungen und Szenarien) ist damit nicht a priori von einer Langzeitarchivierung ausgenommen, setzt aber für deren Nachnutzbarkeit zwingend die Erhaltung der Lauffähigkeit der betreffenden Spiele voraus.

4.4 Quellcodearchivierung: Baustein eines Ganzen und ‚Standalone‘-Strategie

Peng (2011, S. 1226) beschreibt den digitalen Erhalt von „[l]inked and executable code [= Forschungssoftware] and data [= Forschungsdaten]“ als „gold standard“ für die Replizierbarkeit von Forschungsdaten. Die Vorhaltung von nicht ausführbarem „[c]ode and data“ hingegen garantiert zwar keinen vergleichbaren Replizierbarkeitsgrad, erlaubt aber unter gewissen Umständen, nämlich die Möglichkeit zur fehlerfreien Ausführung der im Quellcode enthaltenen Anweisungen, gleichfalls die Validierung und Reproduktion von Forschungsdaten. Zugleich bildet der Quelltext gewissermaßen den Bauplan für die zu erhaltende Software, trägt damit zu ihrem tiefergehenden Verständnis bei und stellt ferner den Ausgangspunkt für mögliche künftige Weiterverarbeitungen (z.B. Codemigrationen und -analysen) dar. Folglich ist die kombinierte Langzeitarchivierung von Quellcode, ausführbaren Repräsentationen, Kontextinformationen und mit dem Programm generierten Forschungsdaten das Ideal der Erhaltung von Forschungs- und Vermittlungssoftware (Abb. 2).

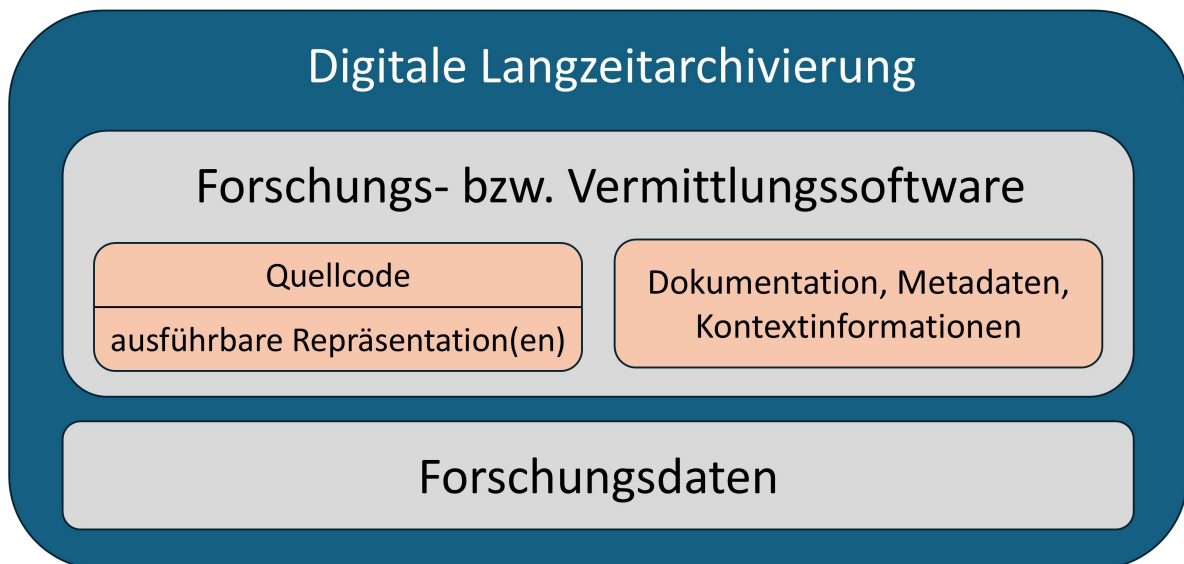


Abbildung 2: Bausteine der Langzeitarchivierung von Forschungs- und Vermittlungssoftware (eigene Darstellung)

Dieser erhaltungsstrategische Zugang kommt freilich nicht um eine aus dem Emulationszugang bekannte Achillesferse herum: Eine verlässliche Langzeitarchivierung von Software in ausführbarer Form erfordert technische, organisatorische, personelle und finanzielle Ressourcen, die eine Realisierung signifikant erschweren, im schlimmsten Falle gar scheitern lassen können; so kann die Unterstützung der digitalen Umgebungen, die für den Erhalt der Lauffähigkeit der archivierten Softwares notwendig sind, aufgrund gekürzter bzw. gestrichener finanzieller Mittel oder wegen begründeter IT-Sicherheitsbedenken auslaufen. Zugleich aber fungiert der mitarchivierte Quellcode gleichermaßen als Dokumentation, Weiterverarbeitungsgrundlage und Backup – in all

diesen Funktionen technisch langzeitarchivfähig. Damit zeigt sich die zweifache Einsatzmöglichkeit von Quellcodearchivierung: als ein Baustein in einem ganzheitlichen Erhaltungskonzept wie auch als ‚Standalone‘-Strategie.

5 Resümee und Ausblick

Der vorliegende Beitrag verfolgte das Ziel, insbesondere technisch-organisatorische und personell-kompetenzbezogene Herausforderungen der Langzeitarchivierung von Forschungs- und Vermittlungssoftware am konkreten Beispiel der Quellcodearchivierung archivfachlich zu erläutern. Dabei konnte gezeigt werden, dass die Übernahme von Quelltext den dauerhaften Erhalt von Lauffähigkeit und Nachnutzbarkeit archivwürdiger Software ermöglicht – wenngleich auch mit Einschränkungen. So ist Quellcode als *plain text* langzeitstabil, kann automatisiert oder durch menschliches Handeln weiterverarbeitet werden und benötigt keine proprietären Rezeptionswerkzeuge – und ist damit geradezu prädestiniert für die digitale Langzeitarchivierung. Ferner verkörpert er als ‚Bauplan‘ der Software deren Funktionalitäten und trägt als solcher zum tiefergehenden Verständnis des Programms bei. Zugleich aber ist Quellcode selbst nicht lauffähig bzw. ausführbar, sondern bedarf zusätzlicher Software- und Hardwarekomponenten, um die in ihm kodierten Anweisungen ausführen zu können; ferner benötigt er für seine Nachvollziehbarkeit – und damit für eine letztlich erfolgreiche Langzeitarchivierung – ein ‚coding-alphabetisiertes‘ und IT-historisch gebildetes Archivpersonal.²⁶

So lassen die Ausführungen folgendes Fazit zu: Quellcodearchivierung ist weder eine Panazee für die dauerhafte Erhaltung von Zugriffs- und Lauffähigkeit von Forschungs- und Vermittlungssoftware noch der archivfachlichen Weisheit letzter Schluss – ebenso wenig aber eine erhaltungsstrategische Sackgasse. Softwarearchivierung durch Quellcodeerhalt mag nicht den ‚Goldstandard‘ für die Nachnutzung darstellen, legt jedoch ein solides Fundament für diese – und damit auch für die Replizierbarkeit von Forschungsdaten. Das überlieferungsbildnerische Ideal stellt zweifelsfrei die gemeinsame Langzeitarchivierung von Quellcode und lauffähiger (d.h. ausführbarer) Repräsentation der Software – gewissermaßen Genotyp und Phänotyp des erhaltungsintendierten Programms – in Kombination mit Kontextinformationen und Forschungsdaten dar – allerdings können die damit verbundenen technischen, organisatorischen, personellen und finanziellen Herausforderungen ein eminentes Umsetzungshindernis darstellen. Insofern verdient Quellcodearchivierung Aufmerksamkeit durch Forscher*innen und Archivar*innen – sei es als Baustein einer ganzheitlichen Softwarearchivierung oder als ‚long-term preservation light‘.²⁷

²⁶Hinzu kommen an Softwarearchivierung allgemein zu richtende basale Kriterien, z.B. eine umfassende Dokumentation des betreffenden Programms.

²⁷Aufgrund des Fokus des Beitrags auf Quellcode als Archivierungsobjekt mussten mehrere, durchaus brennende Fragen zur Softwarearchivierung unbehandelt bleiben – so etwa jene nach dem (Nicht-?)Potential von Bytecode, der im Unterschied zu Maschinencode in der Regel unabhängig von realer

6 Referenzen

Anzt, H., & Bach, F., & Druskat, S., & Löffler, F., & Loewe, A., & Renard, B. Y., & Seemann, G., & Struck, A., & Achhammer, E., & Aggarwal, P., & Appel, F., & Bader, M., & Brusch, L., & Busse, C., & Chourdakis, G., & Dabrowski, P. W., & Ebert, P., & Flemisch, B., & Friedl, S., & Fritzs, B., & Funk, M. D., & Gast, V., & Goth, F., & Grad, J.-N., & Hegewald, J., & Hermann, S., & Hohmann, F., & Janosch, S., & Kutra, D., & Linxweiler, J., & Muth, T., & Peters-Kottig, W., & Rack, F., & Raters, F. H. C., & Rave, S., & Reina, G., & Reißig, M., & Ropinski, T., & Schaarschmidt, J., & Seibold, H., & Thiele, J. P., & Uekermann, B., & Unger, S., & Weeber, R. (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>.

Appel, F., & Loewe, A. (2021). Forschungssoftware – Nachhaltige Entwicklung und Unterstützung. *IAMO Policy Brief*, (42), 491–501. <https://doi.org/10.22004/ag.econ.314673>.

Arbeitskreis Open Science der Helmholtz-Gemeinschaft. (2017). *Zugang zu und Nachnutzung von wissenschaftlicher Software. Positionspapier des Arbeitskreises Open Science der Helmholtz-Gemeinschaft*. Helmholtz-Gemeinschaft. <https://doi.org/10.48440/os.helmholtz.042>.

Banzer, S. (2023). *Codemigration mit ChatGPT – Evaluation von ChatGPT als Tool zur teilautomatisierten Codeübersetzung von COBOL Code zu Python Code*. Fachhochschule Graubünden. https://www.fhgr.ch/fileadmin/fhgr/angewandte_zukunftstechnologien/SII/churer_schriften/sii-churer_schriften_169_banzer.pdf.

Barker, M., & Castro, L. J., & Fritzs, B., & Katz, D. S., & Martinez-Ortiz, C., & Niehues, A., & Struck, A., & Zhang, Q. (2024). *The FAIR for Research Software Principles after two years: An adoption update*. Zenodo. <https://doi.org/10.5281/zenodo.10816031>.

Barker, M., & Chue Hong, N. P., & Katz, D. S., & Lamprecht, A.-L., & Martinez-Ortiz, C., & Psomopoulos, F., & Harrow, J., & Castro, L. J., & Gruenpeter, M., & Martinez, P. A., & Honeyman, T. (2022). Introducing the FAIR Principles for research software. *Scientific Data*, 9(622). <https://doi.org/10.1038/s41597-022-01710-x>.

Hardware ist, für eine Langzeitarchivierung von (nicht nur Forschungs- und Vermittlungs-)Software. Hier ist aus archivischer Perspektive u.a. die Möglichkeit von Interesse, mittels Dekompilierung (nicht zu verwechseln oder synonym zu verwenden mit „Disassemblierung“) aus Bytecode zwar nicht den ursprünglichen, jedoch einen äquivalenten menschenlesbaren Quellcode der zugrundeliegenden höheren Programmiersprache zu erzeugen (z.B. „Decompiler“, 2025). Ebenfalls unerörtert bleiben musste die Frage nach allgemeinen Prozessen und Kriterien zur Bewertung von Forschungs- und Vermittlungssoftware. So büßt etwa das im vorliegenden Beitrag mehrfach erwähnte Argument des Softwareerhalts als Vorbedingung für eine Replizierbarkeit von Forschungsdaten unvermittelt an Zugkraft ein, wenn das Szenario der Datengenerierung selbst nicht reproduzierbar ist (z.B. bei Daten aus glaziologischen Langzeitbeobachtungen, astrofotografischen Himmelskartierungen oder wildtiertelemetrischen Aufzeichnungen).

Barnes, N. (2010). Publish your computer code: it is good enough. *Nature*, 467(7317), 753. <https://doi.org/10.1038/467753a>.

Borghoff, U. M., & Rödiger, P., & Scheffczyk, J., & Schmitz, L. [2006]. *Long-Term Preservation of Digital Documents. Principles and Practices*. Springer. <https://doi.org/10.1007/978-3-540-33640-2>.

Brandt, A. von. (2012). *Werkzeug des Historikers. Eine Einführung in die Historischen Hilfswissenschaften. Mit aktualisierten Literaturnachträgen und einem Nachwort von Franz Fuchs* (18. Aufl.). Kohlhammer.

Brecher, C., & Buchmeiser, M. R., & Burkert, A., & Busemeyer, M. R., & Conermann, S., & Ertl, S., & Friedrich, M., & Helmig, R., & Hohmann, V., & Johnston, A. J., & Kollmeier, B., & Larkum, M., & Louis, J., & Menges, A., & Morgner, U., & Müller, J., & Niessen, C., & Ohlberger, M., & Schäffner, W. ... & Flemisch, B. (2022). Commitment zu aktivem Daten- und Softwaremanagement in großen Forschungsverbünden. Commitment to active data and software management in large research alliances. *Bausteine Forschungsdatenmanagement. Empfehlungen und Erfahrungsberichte für die Praxis von Forschungsdatenmanagerinnen und -managern*, (1), 121–123. <https://doi.org/10.17192/bfdm.2022.1.8412>.

Chue Hong, N. P., & Katz, D. S., & Barker, M., & Lamprecht, A.-L., & Martinez, C., & Psomopoulos, F. E., & Harrow, J., & Castro, L. J., & Gruenpeter, M., & Martinez, P. A., & Honeyman, T., & Struck, A., & Lee, A., & Loewe, A., & van Werkhoven, B., & Jones, C., & Garijo, D., & Plomp, E., & Genova, F., & Shanahan, H., ... & RDA FAIR4RS WG. (2022). *FAIR Principles for Research Software (FAIR4RS Principles)*. Version 1.0. Zenodo. <https://doi.org/10.15497/RDA00068>.

Dappert, A., & Guenther, R. S., & Peyrard, S. (Hrsg.). (2018). *Digital Preservation Metadata for Practitioners. Implementing PREMIS*. Springer International Publishing.

"Decompiler". (2025). In *Wikipedia*. <https://en.wikipedia.org/w/index.php?title=Decompiler&oldid=1296224133>.

Deutsche Forschungsgemeinschaft. (2015). *Leitlinien zum Umgang mit Forschungsdaten*. <https://www.dfg.de/resource/blob/172112/leitlinien-forschungsdaten.pdf>.

Deutsche Forschungsgemeinschaft. (10.10.2023). *Umgang mit Forschungsdaten*. <https://www.dfg.de/de/grundlagen-themen/grundlagen-und-prinzipien-der-foerderung/forschungsdaten>.

Deutsche Forschungsgemeinschaft. (2025). *Leitlinien zur Sicherung guter wissenschaftlicher Praxis. Kodex*. Korrigierte Version 1.2. Zenodo. <https://doi.org/10.5281/zenodo.14281892>.

Donig, S., & Körfer, A.-L., & Meiners, O. (2024). Zugänge zu Datenqualität. Forschungsdatenmanagement als Schnittstelle zwischen Archiven als Gedächtnisinstitutionen und historischer Forschung. In Becker, I. C., & Haffer, D., & Lehrmann, F., & Meier, R.

(Hrsg.), *Archivists meet Historians – Transferring source criticism to the digital age. Beiträge zum 27. Archivwissenschaftlichen Kolloquium der Archivschule Marburg* (265–293). Archivschule Marburg.

Espenschied, D. (2024). Emulation. *Digital Preservation: A Critical Vocabulary*. <https://digital-preservation-a-critical-vocabulary.pubpub.org/pub/cqo43abe>.

FAIR Principles. (o. J.). GO FAIR. <https://www.go-fair.org/fair-principles/>.

forschungsdaten.info. (o. J.). *Glossar. Kernbegriffe im Forschungsdatenmanagement*. <https://forschungsdaten.info/praxis-kompakt/glossar/>.

"Fourth-generation programming language". (2025). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Fourth-generation_programming_language&oldid=1303219586.

Funk, S. E. (2010). Emulation. In Neuroth, H., & Oßwald, A., & Scheffel, R., & Strathmann, S., & Huth, K. (Hrsg.), *nestor Handbuch. Eine kleine Enzyklopädie der digitalen Langzeitarchivierung. Version 2.3* (8:16–8:23). Verlag Werner Hülsbusch.

Gastl-Hartmann, A. (2022). Forschen für die Ewigkeit. Bewertung und Archivierung von Forschungsdaten als neues Arbeitsgebiet der Hochschularchive. *Informationswissenschaft. Theorie, Methode und Praxis*, 7(1), 396–429. <https://doi.org/10.18755/iw.2022.22>.

Helmholtz Open Science Office. (2019). *Muster-Richtlinie Nachhaltige Forschungssoftware an den Helmholtz-Zentren*. Stand 21.11.2019. Helmholtz-Gemeinschaft. <https://os.helmholtz.de/open-research-software/muster-richtlinie>.

Henning, E. (2011). *Hennings HiWi-Test. 175 Fragen & Antworten rund um die Historischen Hilfswissenschaften* (2., verbesserte Auflage). BibSpider.

Hessisches Ministerium für Wissenschaft und Kunst. (2022). *Hessisches Archivgesetz (HArchivG) vom 13. Oktober 2022*. Bürgerservice Hessenrecht. <https://www.rv.hessenrecht.hessen.de/bshe/document/jlr-ArchivGHE2022pG1>.

Hörnschemeyer, J. (2019). „Möchten Sie das Programm wirklich löschen?“ Warum sich die Geisteswissenschaften mit der Nachhaltigkeit von Forschungssoftware auseinanderzusetzen sollten. *Quellen und Forschungen aus italienischen Archiven und Bibliotheken*, 99(1), 491–501. <https://doi.org/10.1515/qufiab-2019-0019>.

Ince, D., Hatton, L., & Graham-Cumming, J. (2012). The case for open computer programs. *Nature*, 482(7386), 485–488. <https://doi.org/10.1038/nature10836>.

Jiménez, R. C., & Kuzak, M., & Alhamdoosh, M., & Barker, M., & Batut, B., & Borg, M., & Capella-Gutierrez, S., & Chue Hong, N., & Cook, M., & Corpas, M., & Flannery, M., & Garcia, L., & Gelpí, J. Ll., & Gladman, S., & Goble, C., & González Ferreiro, M., & Gonzalez-Beltran, A., & Griffin, P. C., & Grüning, B., & Hagberg, J., & Holub, P., & Hooft, R., & Ison, J., & Katz, D. S., & Leskošek, B., & López Gómez, F., & Oliveira, L. J., & Mellor, D., & Mosbergen, R., & Mulder, N., & Perez-Riverol, Y., & Pergl, R., & Pichler, H., & Pope, B., &

Sanz, F., & Schneider, M. V., & Stodden, V., & Suchecki, R., & Svobodová Vařeková, R., & Talvik, H.-A., & Todorov, I., & Treloar, A., & Tyagi, S., van Gompel, M., & Vaughan, D., Via, A., & Wang, X., Watson-Haigh, N. S., & Crouch, S. (2017). Four simple recommendations to encourage best practices in research software [version 1; peer review: 3 approved]. *F1000Research*, 6(876). <https://doi.org/10.12688/f1000research.11407.1>.

Kaden, B., & Kleineberg, M. (2018). Zur Veröffentlichung dissertationsbezogener Forschungsdaten: Perspektiven und Kompetenzen von Promovierenden an Berliner Universitäten. *Bausteine Forschungsdatenmanagement. Empfehlungen und Erfahrungsberichte für die Praxis von Forschungsdatenmanagerinnen und -managern*, (1), 64–69. <https://doi.org/10.17192/bfdm.2018.1.7938>.

Keitel, C. (2012). DIN Norm 31644 „Kriterien für vertrauenswürdige digitale Langzeitarchive“: Zielsetzung, Genese und Perspektiven. In Altenhöner, R., & Oellers, C. (Hrsg.), *Langzeitarchivierung von Forschungsdaten. Standards und disziplinspezifische Lösungen* (69–85). Scivero.

Keitel, C. (2018). *Zwölf Wege ins Archiv. Umriss einer offenen und praktischen Archivwissenschaft*. Franz Steiner Verlag.

Keitel, C., & Schoger, A. (Hrsg.) (2013). *Vertrauenswürdige digitale Langzeitarchivierung nach DIN 31644*. Beuth.

Kramski, H. W. (2016). Digitale Dokumente im Archiv. In Lepper, M., & Raulff, U. (Hrsg.), *Handbuch Archiv. Geschichte, Aufgaben, Perspektiven* (178–197). J.B. Metzler.

Lamprecht, A.-L., & Garcia, L., & Kuzak, M., & Martinez, C., & Arcila, R., & Del Pico, E. M., & Dominguez Del Angel, V., & van de Sandt, S., & Ison, J., & Martinez, P. A., & McQuilton, P., & Valencia, A., & Harrow, J., & Psomopoulos, F., & Gelpí, J. Ll., & Chue Hong, N., & Goble, C., & Capella-Gutierrez, S. (2020). Towards FAIR principles for research software. Position Paper. *Data Science*, 3(1), 37–59. <https://doi.org/10.3233/DS-190026>.

LiveCode script language. (o. J.). Livecode Wiki. https://livecode.fandom.com/wiki/LiveCode_script_language.

Maier, G., & Föhle, D., & Neuburger, A. (2020). Bereitstellung, Aufbereitung, Langzeitsicherung. Funktionen der Archive in der Forschungsdateninfrastruktur. *Archivar. Zeitschrift für Archivwesen*, 73(1), 13–18.

[Malik], S. (2025). *10 AI Code Review Tools That Find Bugs & Flaws in 2025*. DigitalOcean. <https://www.digitalocean.com/resources/articles/ai-code-review-tools>.

Martinez-Ortiz, C., & Katz, D. S., & Lamprecht, A.-L., & Barker, M., & Loewe, A., & Fouilloux, A., & Wyngaard, J., & Garijo, D., & Moldon, J., & Castro, L. J., & Wheeler, D., & Albers, J. R. D., & Lee, A. (2022). *FAIR4RS: Adoption support*. Zenodo. <https://doi.org/10.5281/zenodo.6258365>.

Menne-Haritz, A. (2001). Archivische Bewertung. Der Prozess der Umwidmung von geschlossenem Schriftgut zu auswertungsbereitem Archivgut. *Schweizerische Zeitschrift für Geschichte*, 51(4), 448–460. <https://doi.org/10.5169/seals-107873>.

Morin, A., & Urban, J., & Adams, P. D., & Foster, I., & Sali, A., & Baker, D., & Sliz, P. (2012). Shining Light into Black Boxes. *Science*, 336(6078), 159–160. <https://doi.org/10.1126/science.1218263>.

Naumann, K., & Neuburger, A. (2022). Nutzungsperspektiven durch Querverbindung. Die Rolle der Archive als Teil der digitalen beziehungsweise Nationalen Forschungsdateninfrastruktur. In Becker, I. C., & Haffer, D., & Meier, R., & Uhde, K. (Hrsg.), *Nutzung 3.0 – Zwischen Hermeneutik und Technologie? Beiträge zum 25. Archivwissenschaftlichen Kolloquium der Archivschule Marburg* (75–102). Archivschule Marburg.

Naumann, K., & Raphael, L., & Siegers, P. (2024). Abstimmungsbedarf zur Überlieferungsbildung und Archivierung von Forschungsdaten. In Becker, I. C., & Haffer, D., & Lehrmann, F., & Meier, R. (Hrsg.), *Archivists meet Historians – Transferring source criticism to the digital age. Beiträge zum 27. Archivwissenschaftlichen Kolloquium der Archivschule Marburg* (217–239). Archivschule Marburg.

Naumann, K., & Hodenberg, C. von, & Siegers, P. (2023). Wie Archive und Forschungsdaten zueinanderfinden – Ein gegenseitiger Lernprozess. *ARCHIV. theorie & praxis*, 76(3), S. 187–194.

nestor-Arbeitsgruppe OAIS-Übersetzung / Terminologie. (2013). *Referenzmodell für ein Offenes Archiv-Informationssystem. Deutsche Übersetzung 2.0*. nestor – Kompetenznetzwerk Langzeitarchivierung. [urn:nbn:de:0008-2013082706](https://nbn-resolving.org/urn:nbn:de:0008-2013082706).

Nippert, K. (2013). Forschungsdaten. Ein Arbeitsgebiet für Archive von Hochschulen und wissenschaftlichen Institutionen. *Archivar. Zeitschrift für Archivwesen*, 66(2), 154–159.

Pashler, H., & Harris, C. R. (2012): Is the replicability crisis overblown? Three arguments examined. *Perspectives on Psychological Science*, 7(6), 531–536. <https://doi.org/10.1177/1745691612463401>.

Pashler, H., & Wagenmakers, E.-J. (2012): Editors' Introduction to the Special Section on Replicability in Psychological Science: A Crisis of Confidence? *Perspectives on Psychological Science*, 7(6), 528–530. <https://doi.org/10.1177/1745691612465253>.

Peng, R. D. (2011). Reproducible Research in Computational Science. *Science*, 334(6060), 1226–1227. <https://doi.org/10.1126/science.1213847>.

Policy für Forschungsdatenmanagement an der Universität Wien. (2021). https://rdm.univie.ac.at/fileadmin/user_upload/p_forschungsdatenmanagement/Dokumente/RDM_Policy_UNIVIE_v1_de.pdf.

PREMIS Editorial Committee (2015). *PREMIS Data Dictionary for Preservation Metadata. Version 3.0*. <https://www.loc.gov/standards/premis/v3/premis-3-0-final.pdf>.

Rechert, K. (2022). Emulation options for legacy databases. In Naumann, K. (Hrsg.), *Databases for 2080. Workshop Proceedings* (16–20). Landesarchiv Baden-Württemberg. URN: [urn:nbn:de:101:1-2022071903](https://nbn-resolving.org/urn:nbn:de:101:1-2022071903).

Rohr, C. (2015). *Historische Hilfswissenschaften. Eine Einführung*. Böhlau.

Röwenstrunk, D. (2018). Langzeitverfügbarkeit von wissenschaftlicher Software im Bereich historisch-kritischer Musikedition. *Bibliothek – Forschung und Praxis*, 42(2), 302–308. <https://doi.org/10.1515/bfp-2018-0029>.

Sandford, J. (2012). Emulation and the Pursuit of Preservation. *Archivi*, VII(1), 85–102.

Scheliga, K., & Pampel, H., & Bernstein, E., & Bruch, C., & Zu Castell, W., & Diesmann, M., & Fritzsche, B., & Fuhrmann, J., & Haas, H., & Hammitzsch, M., & Laehnemann, D., & McHardy, A., & Konrad, U., & Scharnberg, G., & Schreiber, A., & Steglich, D. (2017). *Helmholtz Open Science Workshop „Zugang zu und Nachnutzung von wissenschaftlicher Software“ #hgfos16. Report; November 2016*. Deutsches GeoForschungsZentrum GFZ. <https://doi.org/10.2312/LIS.17.01>.

Scheliga, K., & Pampel, H., & Konrad, U., & Fritzsche, B., & Schlauch, T., & Nolden, M., & Zu Castell, W., & Finke, A., & Hammitzsch, M., & Bertuch, O., & Denker, M. (2019). *Dealing with research software: Recommendations for best practices*. Helmholtz Open Science Coordination Office. <https://doi.org/10.2312/OS.HELMHOLTZ.003>.

Schenk, D. (2014). *Kleine Theorie des Archivs* (2., überarbeitete Auflage). Franz Steiner Verlag.

Schrimpf, S. (2012): Überblick über das OAIS-Referenzmodell. In Altenhöner, R., & Oellers, C. (Hrsg.), *Langzeitarchivierung von Forschungsdaten. Standards und disziplinspezifische Lösungen* (51–64). Scivero.

Schrimpf, S. (2014). *Das OAIS-Modell für die Langzeitarchivierung. Anwendung der ISO 14721 in Bibliotheken und Archiven*. Beuth.

Schwarz, K. (2023). Archive. In Kuhlen, R., & Lewandowski, D., & Semar, W., & Womser-Hacker, C. (Hrsg.), *Grundlagen der Informationswissenschaft* (7., völlig neu gefasste Ausgabe, 93–101). De Gruyter Saur. <https://doi.org/10.1515/9783110769043-007>.

Shankar, K., & Cushing, A. (2024). Appraisal. *Digital Preservation: A Critical Vocabulary*. <https://digital-preservation-a-critical-vocabulary.pubpub.org/pub/6557jxbg>.

Simon, H. (2022). Daten für die Forschung – Der Aufbau der Nationalen Forschungsdateninfrastruktur und eine Verortung des Bundesarchivs. In Becker, I. C., & Haffer, D., & Meier, R., & Uhde, K. (Hrsg.), *Nutzung 3.0 – Zwischen Hermeneutik und Technologie? Beiträge zum 25. Archivwissenschaftlichen Kolloquium der Archivschule Marburg* (125–140). Archivschule Marburg.

Stroebe, W., & Strack, F. (2014): The Alleged Crisis and the Illusion of Exact Replication. *Perspectives on Psychological Science*, 9(1), 59–71. <https://doi.org/10.1177/1745691613514450>.

Suchodoletz, D. von [2009]. *Funktionale Langzeitarchivierung digitaler Objekte – Erfolgsbedingungen des Einsatzes von Emulationsstrategien*. nestor – Kompetenznetzwerk Langzeitarchivierung. [urn:nbn:de:0008-2008070219](https://nbn-resolving.org/urn:nbn:de:0008-2008070219).

Suchodoletz, D. von, & Brettschneider, P., & Axtmann, A., & Heber, M., & Oberländer, L., & Leendertse, J., & Schumm, I., & Brandt, O., & Schmidt, K., & Gertis, L., & Selzer, M., & Ulrich, R., & Iglezakis, D., & Boda, V. (2023). Sicherstellung der Reproduzierbarkeit von Forschungsergebnissen durch Bewahrung des Zugriffs auf Forschungssoftware. *Bausteine Forschungsdatenmanagement. Empfehlungen und Erfahrungsberichte für die Praxis von Forschungsdatenmanagerinnen und -managern*, (5), 3–13. <https://doi.org/10.17192/bfdm.2023.5.8555>.

The Consultative Committee for Space Data Systems. (2024). *Reference Model for an Open Archival Information System (OAIS)* (Magenta Book). NASA. https://ccsds.org/wp-content/uploads/gravity_forms/5-448e85c647331d9cbaf66c096458bdd5/2025/01/650x0m3.pdf.

Tiemann, K. (2018). Bewertung und Übernahme von amtlichem Registraturgut. In Stumpf, M., & Höötmann, H.-J. (Hrsg.), *Praktische Archivkunde. Ein Leitfaden für Fachangestellte für Medien- und Informationsdienste - Fachrichtung Archiv* (4. aktualisierte Auflage, 87–106). Ardey-Verlag.

Töwe, M. (25.09.2025). *Langzeitarchivierung von Forschungsdaten: Erfahrungen aus der Schweiz*. LaVaH Workshop, Frankfurt am Main. https://hebis.rz.uni-frankfurt.de/LaVaH/lavah-workshop-25-9-25/5_Toewe_LaVaH_Abschluss.pdf.

Van Lissa, C. J., & Brandmaier, A. M., & Brinkman, L., & Lamprecht, A.-L., & Peikert, A., & Struiksmä, M. E., & Vreede, B. M. I. (2021). WORCS: A workflow for open reproducible code in science. Resource Paper. *Data Science*, 4(1), 29–49. <https://doi.org/10.3233/DS-210031>.

Van Lissa, C. J., & Peikert, A., & Brandmaier, A. M. [2025]. *Workflow for Open Reproducible Code in Science*. 0.1.20.1. <https://cjvanlissa.github.io/worcs/>.

VdA Arbeitskreis archivische Bewertung, & Wendenburg, A. (2025). *Überlieferung und archivische Bewertung*. VdA – Verband deutscher Archivarinnen und Archivare e.V.

Vee, A. (2017). *Coding Literacy. How Computer Programming Is Changing Writing*. The MIT Press.

Viert, A. (2023). *Herausforderungen in der Aufbewahrung von Videospielen und ihrer Peripherie. Fragen und Antworten insbesondere zur Peripherie und zur Emulation als Lösungsansatz*. Fachhochschule Graubünden.

URN: [urn:nbn:de:101:1-2505191556551.677128368851](https://nbn-resolving.org/urn:nbn:de:101:1-2505191556551.677128368851).

Weber, A., & Piesche, C. (2021). Datenspeicherung, -kuration und Langzeitverfügbarkeit. In Putnings, M., & Neuroth, H., & Neumann, J. (Hrsg.), *Praxishandbuch Forschungsdatenmanagement* (327–355). De Gruyter Saur. <https://doi.org/10.1515/9783110657807-019>.

Wilkinson, M. D., & Dumontier, M., & Aalbersberg, I. J., & Appleton, G., & Axton, M., & Baak, A., & Blomberg, N., & Boiten, J.-W., & Bonino da Silva Santos, L., & Bourne, P. E., & Bouwman, J., & Brookes, A. J., & Clark, T., & Crosas, M., & Dillo, I., & Dumon, O., & Edmunds, S., & Evelo, C. T., & Finkers, R., ... & Mons, B. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, (3), 1–9. <https://doi.org/10.1038/sdata.2016.18>.