

Muster-Richtlinie Nachhaltige Forschungssoftware an den Helmholtz-Zentren

Stand 21.11.2019

Hinweis: Die Muster-Richtlinie wurde von der Task Group Forschungssoftware - unter Einbeziehung weiterer Expert*innen aus der Helmholtz-Gemeinschaft - erarbeitet und mit dem Arbeitskreis Technologietransfer und Gewerblicher Rechtsschutz sowie dem Arbeitskreis Recht abgestimmt. Die vorliegende Version wurde am 21.11.2019 vom Arbeitskreis Open Science verabschiedet.

Die Muster-Richtlinie soll als richtungsweisende und nachnutzbare Vorlage zur Erstellung von Regelungen für einen nachhaltigen Umgang mit Forschungssoftware dienen. Die Regelungen können an die individuellen Bedingungen und Bedürfnisse der Helmholtz-Zentren angepasst werden.

Minimalanforderung für die Erstellung einer auf ein Helmholtz-Zentrum bezogenen Richtlinie, auf Basis der vorliegenden Muster-Richtlinie, ist die entsprechende Anpassung der wie folgt markierten Textfragmente: <hier zum Beispiel den Namen des Helmholtz-Zentrums einfügen>.

Die „Muster-Richtlinie Nachhaltige Forschungssoftware an den Helmholtz-Zentren“ ist in den Open Science Bestrebungen der Helmholtz-Gemeinschaft verankert. Das Papier „Empfehlungen zur Implementierung von Leit- und Richtlinien zum Umgang mit wissenschaftlicher Software an den Helmholtz-Zentren“¹ ist als ergänzende und richtungsweisende Informationsgrundlage zu der Muster-Richtlinie zu beachten. Des Weiteren soll eine ergänzende und kontinuierlich gepflegte Informations- und Referenzsammlung für die praktischen Aspekte des Umgangs mit Forschungssoftware dienen². Zu berücksichtigen sind auch „[Empfehlungen für Richtlinien der Helmholtz-Zentren zum Umgang mit Forschungsdaten](#)“. Auch die Digitalisierungsstrategie der Helmholtz-Gemeinschaft ist zu beachten. Im weiteren Kontext ist der Kodex „[Leitlinien zur Sicherung guter wissenschaftlicher Praxis](#)“ der Deutschen Forschungsgemeinschaft (DFG) grundlegend.

¹ Papier „Empfehlungen zur Implementierung von Leit- und Richtlinien zum Umgang mit wissenschaftlicher Software an den Helmholtz-Zentren“: <https://os.helmholtz.de/index.php?id=3197>

² www.helmholtz.de/software (work in progress)

Richtlinie für nachhaltige Forschungssoftware am <Helmholtz-Zentrum>

Präambel

Mit zunehmender Unverzichtbarkeit von Softwarelösungen im Prozess der wissenschaftlichen Erkenntnisgewinnung muss der nachhaltige Umgang mit Forschungssoftware Teil der guten wissenschaftlichen Praxis sein.

Mit Forschungssoftware sind in dieser Richtlinie alle Formen von Programmcode (z. B. Quellcode nebst zugehörigen Dokumentationen, Parametern und Workflows) und daraus generierten ausführbaren Programmen gemeint, die im Rahmen einer wissenschaftsbezogenen Tätigkeit an Helmholtz-Zentren entwickelt und / oder (nach)genutzt werden.

Die Entwicklung von Forschungssoftware ist Teil eines kreativen Prozesses und ist in diesem Sinne ausführbares Wissen. Die Entwicklung von Forschungssoftware ist eine intellektuelle und urheberrechtlich geschützte Leistung. Forschungssoftware ist als zentrales und eigenständiges Produkt der wissenschaftlichen Arbeit zu betrachten und wertzuschätzen.

Für einen nachhaltigen Umgang mit Forschungssoftware sind die FAIR-Prinzipien³ (**F**indable, **A**ccessible, **I**nteroperable, **R**eusable) zu berücksichtigen. Darüber hinaus sollte, zur Gewährleistung der Qualität der Forschungssoftware, eine Orientierung an gängigen Standards stattfinden.

Im Sinne von Open Science sind der Zugang zu und die Nachnutzung von Forschungssoftware – im Zusammenspiel mit Textveröffentlichungen und dazugehörigen Forschungsdaten – die Basis für Nachvollziehbarkeit, Verifizierbarkeit und Reproduzierbarkeit von wissenschaftlichen Ergebnissen.

Im Streben, die wissenschaftliche Erkenntnisgewinnung transparent zu gestalten, verfolgt das <Helmholtz-Zentrum> das Ziel, die im Rahmen der Tätigkeit seiner Mitarbeiter*innen allein oder gemeinsam mit Dritten entwickelte Software nach den oben genannten Prinzipien zu behandeln. Verwertungsoptionen sind hierbei zu berücksichtigen.

Ziel der Richtlinie ist es, am <Helmholtz-Zentrum> einen nachhaltigen Umgang mit Forschungssoftware zu etablieren und die Wertschätzung qualitativ hochwertiger Forschungssoftware zu erhöhen. Die Richtlinie ist auf den Kontext und die Sichtweise der Helmholtz-Gemeinschaft auf Open Science abgestimmt.

³ <https://www.force11.org/group/fairgroup/fairprinciples>; <https://www.nature.com/articles/sdata201618>

Nachhaltigkeit

Im Kontext von Forschungssoftware zielt Nachhaltigkeit im Wesentlichen darauf ab, die Nachvollziehbarkeit, Verifizierbarkeit und Reproduzierbarkeit wissenschaftlicher Erkenntnisse, sowie auch die Nachnutzbarkeit der Software sicherzustellen. Dazu trägt die oben erwähnte Anwendung der FAIR-Prinzipien auf Forschungssoftware bei.

Das <Helmholtz-Zentrum> verpflichtet sich, Aspekte der Nachhaltigkeit in allen Etappen des Lebenszyklus einer Forschungssoftware zu berücksichtigen.

Nachhaltigkeit ist als Faktor bei Entscheidungsprozessen insbesondere bei der Softwareentwicklungs- und Dokumentationspraxis, bei der Bereitstellung, Publikation und Archivierung von Forschungssoftware, der mittel- und langfristigen Wartung sowie der Bereitstellung entsprechender Infrastrukturen einzubeziehen.

Entwicklung, Verwendung und Nachnutzung von Forschungssoftware

Entwicklung und Dokumentation

Die Umsetzung und Ausgestaltung einer guten Softwareentwicklungs- und Dokumentationspraxis hängt von einer Vielzahl von Faktoren, wie beispielsweise der wissenschaftlichen Domäne, den eingesetzten Programmiertechnologien oder der Komplexität ab. Daher empfiehlt das <Helmholtz-Zentrum> den Entwicklergruppen schrittweise erforderliche Anforderungen durch die Einführung passender Anwendungsklassen abzustimmen und festzulegen. Anwendungsklassen spiegeln typischerweise die Komplexität eines Softwareprojektes wider, so können z. B. für Skripte je nach Komplexitätsgrad unterschiedliche Qualitätsanforderungen gelten.

Für die unterschiedlichen Anwendungsklassen sind am <Helmholtz-Zentrum> folgende Standards zu etablieren:

1. Die Entwickler*innen setzen auf übliche Standards und Werkzeuge auf aktuellem Stand der Technik während der Entwicklung, Validierung, Verifizierung und dem Deployment.
2. Die Beachtung rechtlicher Aspekte (z. B. Rechte Dritter) [*→ Rechtliche Aspekte*] ist integraler Bestandteil des Softwareentwicklungsprozesses und erfordert Dokumentation (z. B. Urheberrechtsvermerke).
3. Versionskontrollsysteme sind bei jeder Anwendungsklasse einzusetzen.
 - a. Das zugehörige Repository enthält bzw. referenziert sämtliche zur Nutzung notwendigen Bestandteile der Forschungssoftware.
 - b. Eine kurze, aussagekräftige Beschreibung des Zwecks der Forschungssoftware ist beizulegen (z. B. in Form einer README-Datei).

- c. Metadaten sind maschinenlesbar als Zitationshinweis⁴ beizulegen. Darüber hinaus dienen sie nach Indizierung der Auffindbarkeit von Forschungssoftware.
4. Stabile Versionen (Release) sind für Nutzer*innen eindeutig, beispielsweise über Release-Nummern⁵, zu kennzeichnen.
5. Zum Zweck der Zitierbarkeit der Forschungssoftware ist dieser ein persistenter Identifikator (PID) zuzuordnen [*→ Bereitstellung, Publikation und Zitation*]. Dazu notwendige Metadaten sind im Zitationshinweis (s. o.) zu finden.
6. Wird die Forschungssoftware von einer Gruppe entwickelt, so legt diese die Prozesse und Regeln für Entwicklung und Zusammenarbeit fest, die regelmäßig überprüft und dokumentiert werden. Dies betrifft u. a. Themen wie:
 - a. Dokumentation von Quelltext, Schnittstellen und Konzepten
 - b. Teststrategien mit Testarten, Automatisierung und Zeitpunkte
 - c. Entwicklungsmethoden (agil, iterativ, ...)
 - d. Change Management und Code Reviews
 - e. Kollaborative Nutzung von Werkzeugen und Infrastruktur
 - f. Definition der Entwicklungszyklen
 - g. Kommunikationskanäle
7. Zur Verwendung der Forschungssoftware stellen die Entwickler*innen eine angemessene Installations- und Nutzerdokumentation bereit und benennen eine*n Ansprechpartner*in.

Zur Unterstützung der Entwickler*innen stellt das <Helmholtz-Zentrum> entsprechende Infrastrukturen und Beratungsangebote zur Verfügung [*→ Angebote zur Unterstützung und Beratung*].

Angebote zur Unterstützung und Beratung

Um die Entwicklungsarbeit an Forschungssoftware zu vereinfachen, stellt das <Helmholtz-Zentrum> verschiedene Beratungs- und Unterstützungsangebote sowie geeignete Werkzeuge auf dem aktuellen Stand von Bedarf und Technik bereit:

1. Versionskontrollsystem/e, bevorzugt verknüpft mit kollaborativen Funktionen für Communities [*→ Entwicklung und Dokumentation*].
2. Automatisierungswerkzeuge und Umgebungen für Validierung und Verifizierung von Forschungssoftware im Sinne der Qualitätssicherung.
3. Archivierung von Forschungssoftware:
 - a. Primär für Versionen im Kontext von wissenschaftlichen Publikationen.

⁴ <https://research-software.org/citation/developers>

⁵ Sollten einem Schema unterliegen, siehe Praxisdokument und <http://semver.org>

- b. Speicherung der Forschungssoftware inklusive begleitender Daten (z. B. Metadaten, Laufzeitumgebung, Testdaten).
 - c. Einhaltung von Speicherfristen entsprechend der Regeln guter wissenschaftlichen Praxis in den Fachdisziplinen.
4. Lizenzrechtliche Beratung und Freigabeprozesse [-> *Rechtliche Aspekte*].

Qualitätssicherung und Archivierung

Das <Helmholtz-Zentrum> möchte langfristig die Qualität der entwickelten Software steigern, um einen wissenschaftlichen Mehrwert und Sichtbarkeit in den Communities zu generieren. Es entwickelt Compliance-Level und Qualitätsstandards für die Anwendungsklassen sowie Checklisten zu deren Überprüfung.

Entwickler*innen, Software-Verantwortliche, Projektleiter*innen und Vorgesetzte achten gemeinsam auf die Umsetzung der Maßnahmen zur Qualitätssicherung unter Berücksichtigung der wissenschaftlichen und wirtschaftlichen Perspektiven:

1. Gelebte Praxis der Empfehlungen zur Entwicklung und Dokumentation von Forschungssoftware.
2. Eingruppierung der Forschungssoftware in die Anwendungsklassen und deren regelmäßigen Evaluation.
3. Kontinuierliche Verbesserung als strategisches Ziel.
4. Aufbau einer Community für Nutzer*innen und Entwickler*innen als strategisches Ziel.
5. Einhaltung der Regeln guter wissenschaftlicher Praxis durch langfristig verfügbare Repositorien und ggf. (Langzeit)-Archivierung.
6. Einführung von Prozessen zur Qualitätskontrolle.

Nachhaltige Forschungssoftware setzt voraus, dass auch inaktive oder archivierte Software nachnutzbar ist. Die Schnelllebigkeit der Technologien im Software-Umfeld stellt im Hinblick auf die Archivierung von Forschungssoftware eine Herausforderung dar. Entwickler*innen, Projektleiter*innen und Vorgesetzte müssen bei der Planung, Umsetzung und Verwertung von Forschungssoftware gemeinsam anhand der Anwendungsklasse entscheiden, wie im konkreten Einzelfall zu handeln ist. Es sind entsprechende (langfristige) Ressourcen einzuplanen.

Um die vorhandene Expertise bei der Erstellung qualitativ hochwertiger Forschungssoftware langfristig am Zentrum zu halten, setzt sich das <Helmholtz-Zentrum> darüber hinaus dafür ein, nachhaltige und möglichst langfristige Karriereperspektiven für mit Forschungssoftware verbundene Rollen (z. B. RSE, Software-Verantwortliche) zu schaffen.

Weiterbildung, Karriereperspektiven und Vernetzung

Die Qualität der Forschungssoftware hängt in entscheidendem Maße von den vorhandenen Kenntnissen und Fähigkeiten der Entwickler*innen ab, daher hat die kontinuierliche Weiterbildung große Bedeutung für das <Helmholtz-Zentrum> sowie die Helmholtz-Gemeinschaft. Insbesondere soll damit bei Forschenden mit Schwerpunkten außerhalb der Informatik die Motivation zur aktiven und nachhaltigen Softwareentwicklung langfristig ansteigen. Die möglichst vielfältigen Weiterbildungs- und Vernetzungsangebote am <Helmholtz-Zentrum> werden komplementär zu den zentrenübergreifenden Angeboten der Helmholtz-Gemeinschaft⁶ gestaltet.

Das <Helmholtz-Zentrum> unterstützt seine Mitarbeiter*innen bei folgenden Maßnahmen:

1. Zur Honorierung der Leistung bei der Erstellung, Wartung und Pflege der *Forschungssoftware* sind die Qualitätssicherung und Software-Veröffentlichungen in das wissenschaftliche Reporting einzubeziehen.
2. Bereitstellung von Informationsangeboten und fachlicher Unterstützung (mit unterschiedlichem Aufwand):
 - a. Zentrales, regelmäßig aktualisiertes und moderiertes Informationsangebot
Beispiele: Best Practice Guides, Tutorialsammlung, Wikis zum Wissensaustausch
 - b. Austausch am Arbeitsplatz
Beispiele: Community-Chats, FAQs, informelle Treffen (Hacky Hours), Sprechstunden (Code Clinic), Kommunikation mit Software-Verantwortlichen
 - c. Weiterbildungsangebote
Beispiele: Software Carpentries, Schulungen, Workshops, Hackathons, Consulting
 - d. Consulting durch interne und/oder externe Research Software Engineers.
3. Ermöglichung der Teilnahme an regelmäßigen Weiterbildungsangeboten zu Programmiersprachen und -techniken, Methoden rund um Softwareentwicklung sowie zu rechtlichen Aspekten und Lizenzierung für alle Mitarbeiter*innen. Soweit möglich, sind Grundlagen für einen nachhaltigen Umgang mit Forschungssoftware bereits in den <Graduiertenschulen> zu verankern.
4. Nutzung existierender Schulungsangebote wie Software Carpentries und Online-Angebote als ergänzendes niederschwelliges Angebot.
5. Förderung der Etablierung eines Netzwerks für Entwickler*innen von Forschungssoftware innerhalb des <Helmholtz-Zentrum>, um ein Forum für Community-Building und Erfahrungsaustausch zu schaffen.
6. Einführung von Anlaufstellen für diverse Angelegenheiten zu Forschungssoftware.

⁶ z. B. Helmholtz Information & Data Science Academy (HIDA): <https://www.helmholtz.de/forschung/information-data-science/helmholtz-information-data-science-academy-hida-research-schools/>

7. Unterstützung von Aktivitäten für eine Vernetzung von Entwickler*innen auch außerhalb des Zentrums.
8. Unterstützung der Mitarbeit von Entwickler*innen in Helmholtz-weiten Aktivitäten und an Open-Source-Software.
9. Unterstützung der Vernetzung von Entwickler*innen von Forschungssoftware über diverse on- und offline Kommunikationskanäle.
Beispiele: Mailinglisten, Forum, Netzwerkveranstaltungen
10. Ausbau von Kooperationen mit anderen universitären oder außeruniversitären Einrichtungen im Bereich Softwareentwicklung.

Bereitstellung, Publikation und Zitation

Zur Erfüllung des Kriteriums der Nachhaltigkeit im Wissenschaftsbetrieb erwartet das <Helmholtz-Zentrum> die Zugänglichmachung von Forschungssoftware einerseits und die Kenntlichmachung der Verwendung durch Zitation andererseits durch die Wissenschaft, analog zu Forschungsdaten und wissenschaftlichen Publikationen.

Es gelten folgende Rahmenbedingungen für Bereitstellung und Publikation:

1. Entwickler*innen, Vorgesetzte und Projektleiter*innen entscheiden gemeinsam und möglichst frühzeitig über eine Strategie zur Zugänglichkeit der Forschungssoftware.
 - a. Zugänglichkeit zur Forschungssoftware ist für einen möglichst großen, idealerweise öffentlichen, Kreis an Nutzer*innen anzustreben.
 - b. Von Beginn an ist die Möglichkeit der transparenten Entwicklung zur Sicherung der Qualität zu prüfen.
 - c. Frühzeitige Beachtung der rechtlichen Aspekte und Freigabeprozesse ist unabdingbar für die Weitergabe und Publikationen.
 - d. Unabhängig von der Zugänglichkeit der Forschungssoftware an sich, sind Metadaten zur Forschungssoftware im Sinne einer Softwarepublikation zu veröffentlichen.
2. Die Realisierung der langfristigen Zugänglichkeit durch Nutzung der internen und externen Unterstützungs- und Beratungsangebote.
3. Für Softwarepublikationen gilt:
 - a. Jede Publikation ist durch einen persistenten Identifikator (PID) zitierbar zu machen.
 - b. Publikationen enthalten mindestens Metadaten⁷.
 - c. Die Publikation kann mit Embargofristen (im Sinne von Rückhalte- und Sperrfristen) verknüpft werden.

⁷ <https://research-software.org/citation/researchers>

HELMHOLTZ

Open Science

Zur Erhöhung der Sichtbarkeit des Beitrags von Forschungssoftware an wissenschaftlichen Ergebnissen ist diese durchgängig in wissenschaftlichen Publikationen zu zitieren.

Forschungssoftware ist unabhängig von ihrer öffentlichen Zugänglichkeit mit folgenden Angaben in einer wissenschaftlichen Publikation zu zitieren⁸ 9: Autor*innen, Name der Software, PID, das Veröffentlichungsdatum, eventuell vorhandene Release-Nummer. Falls es keine PID gibt, kann alternativ eine Revision und die URL des Quelltext-Repositorys angegeben werden.

Rechtliche Aspekte

Die Entwicklung, wissenschaftliche Verwendung und Nachnutzung einschließlich wirtschaftlicher Verwertungsaspekte einer Forschungssoftware muss den einschlägigen rechtlichen Kontext mit in den Blick nehmen. Sowohl eine Zugänglichmachung der Forschungssoftware im Sinne von Open Science als auch deren wirtschaftliche Nutzung erfordern zwingend das Vorliegen der notwendigen Verfügungsberechtigung. Die Prüfung erfolgt durch die zuständigen Fachabteilungen.

Grundsätzlich unterfällt eine Forschungssoftware dem Urheberrecht. § 69a Urhebergesetz (UrhG) schützt Software in jeder Gestaltung, einschließlich Entwurfsmaterial, in allen Ausdrucksformen (QC, C, EXE, Module). Nicht geschützt sind dagegen Ideen und Grundsätze, die dem Werk zugrunde liegen.

Das Urheberpersönlichkeitsrecht ist zwar nicht übertragbar. Zulässig sind jedoch die Einräumung von Nutzungsrechten sowie Vereinbarungen zu Verwertungsrechten. Sobald die Entwickler*innen sich in einem Arbeits- und Dienstverhältnis mit dem <Helmholtz-Zentrum> befinden, liegen die Nutzungs- und Verwertungsrechte automatisch beim Zentrum (Einzelheiten sind im § 69 b UrhG-Gesetz geregelt). Spezielle Regelungen sind daher bei der Einbeziehung Dritter, die keinen Arbeitsvertrag o. Ä. mit dem Zentrum haben (z. B. Abschluss eines Contribution Agreement), beispielsweise bei Ausgründungen oder wirtschaftlicher Nutzung zu treffen.

Auch beim Einsatz fremder Software müssen Rechte durch den Urheber eingeräumt werden. Dabei sind zwingend die Lizenzbedingungen der eingesetzten (Open-Source-) Software zu beachten, da eine Nichtbeachtung zu Urheberrechtsverletzungen führen kann. Dies gilt auch, wenn lediglich Programmbestandteile oder Teilsequenzen in eine eigene Forschungssoftware durch Dritte integriert werden. Dann bedarf es einer intensiven rechtlichen Prüfung, inwieweit die verschiedenen Lizenzen der integrierten Drittsoftware sich widersprechende Regelungen enthalten. Das kann dazu führen, dass die eigene Software nicht unter der beabsichtigten Open-Source-Lizenz veröffentlicht werden kann.

⁸ Siehe auch Empfehlungen der FORCE 11 WG, siehe: Smith AM, Katz DS, Niemeyer KE, FORCE11 Software Citation Working Group. (2016) Software citation principles. PeerJ Computer Science 2:e86 <https://doi.org/10.7717/peerj-cs.86>

⁹ <https://research-software.org/citation/researchers>

HELMHOLTZ

Open Science

Zudem kann die Open-Source-Lizenz der eingesetzten Software wegfallen, wenn die Lizenzbedingungen nicht eingehalten werden.

Die Vorgesetzten von Entwickler*innen müssen deshalb zwingend vor Beginn der Arbeit Anleitung geben, welche programmtechnische Maßnahmen diese sogenannte Infektion vermeiden und die mit Softwareentwicklung befassten Mitarbeiter*innen auf die möglichen zivil- und strafrechtlichen Konsequenzen hinweisen sowie entsprechende Verhaltensanweisungen herausgeben. Verstöße gegen das Urheberrecht können schwerwiegende Folgen für das Zentrum nach sich ziehen und zur Strafbarkeit führen (siehe §106 UrhG).

Gleiches gilt für die Nutzung von externen Förderungen bei der Entwicklung von Forschungssoftware. Hier müssen die jeweiligen Förderbestimmungen und Kooperationsverträge beachtet werden, wenn z. B. Ausschließlichkeit zugunsten des Fördergebers eines Dritten vereinbart wurde.

Zusammenfassung der wichtigsten Punkte:

- Nur Personen an der Entwicklung von Software arbeiten lassen, die einen Arbeitsvertrag oder vergleichbaren Vertrag mit dem Zentrum haben.
- An der Entwicklung beteiligte Dritte (z. B. Praktikant*innen, Student*innen), die keinen Arbeitsvertrag o. Ä. Vertrag haben, über ein Contribution Agreement ihre Rechte an das <Helmholtz-Zentrum> abtreten lassen.
- Vor Lizenzierung sorgfältig prüfen, wo die Software herkommt, wer daran mitgearbeitet hat und welche Komponenten Dritter dafür verwendet worden sind.
- Bei Konflikten mit Rechten Dritter die Forschungssoftware bzw. deren Komponenten so ersetzen oder nachprogrammieren, dass Infektion vermieden wird.
- Durch programmtechnische Maßnahmen dafür sorgen, dass eigene Software nicht durch Copyleft-Software infiziert wird.
- Bei verschiedenen Softwarekomponenten die Lizenzkompatibilität prüfen.
- Dokumentation verwenden: Versionskontrollsystem, Dokumentation der Open-Source-Software, der Lizenztexte und Urheber.
- Prüfung und Berücksichtigung von Förderbestimmungen.

Wissenschaftliche und wirtschaftliche Verwertungsaspekte

Diese Richtlinie schließt ausdrücklich nicht eine nachgelagerte wirtschaftliche Verwertung aus. Vielmehr können die wissenschaftliche Verwendung und wirtschaftliche Verwertungsaspekte durchaus gemeinsame Bausteine einer abgestimmten Verwertungsstrategie für eine Forschungssoftware sein. Es ergeben sich nicht sofort im Verlauf der Entwicklung oder Verwendung einer Forschungssoftware wirtschaftliche Verwertungsaspekte. Die Verwertung erfolgt meist nachgelagert im Rahmen der auftragsbezogen bzw. kooperativen Drittmittelakquise im technologiegetriebenen Industriekontext.

Daher müssen insbesondere mit Blick auf eine wirtschaftliche Verwertung frühzeitig alle notwendigen wirtschaftlichen Nutzungs- und Verwertungsrechte gesichert werden (z. B. bei der Einbindung Dritter wie Freelancer in die Entwicklungsarbeiten oder Forschungsarbeiten).

Unabhängig von der späteren Verwendung bzw. Nachnutzung ist es daher wichtig, die bereitgestellten Angebote zur Unterstützung und Beratung wahrzunehmen und die dargestellten Maßnahmen zur Entwicklung, Verwendung und Nachnutzung von Forschungssoftware mit Beginn der Entwicklungsarbeit umzusetzen. Nur so ist es hinreichend möglich, im jeweiligen Einzelfall zu prüfen, ob Rechte Dritter verletzt werden und damit eine wissenschaftliche oder wirtschaftliche Verwendung bzw. Nachnutzung eingeschränkt oder sogar verhindert wird.

Die Entwickelnden und ihre Vorgesetzten werden hiermit aufgefordert, so früh wie möglich mit den jeweiligen Fachabteilungen (z. B. Rechtsabteilung, Technologietransfer, IT) Kontakt aufzunehmen und ihre jeweilige Situation zu besprechen.

Anhang

Glossar

Anwendungsklassen

Anwendungsklassen¹⁰ dienen zur Kategorisierung von Forschungssoftware und legen dazugehörige Regeln und Empfehlungen in Bezug auf eine angemessene Softwareentwicklungspraxis und Dokumentation fest. Sie erleichtern die Prüfung der Regeln und die Kommunikation zu den dazugehörigen Themen. Weitere Informationen zum Begriff können dem Dokument „Software-Engineering-Empfehlungen des DLR“ (ab Seite 7)¹¹ entnommen werden. Darin befindet sich eine exemplarische Definition von aufeinander aufbauenden Anwendungsklassen. Die beschriebene „Anwendungsklasse 1“ legt beispielsweise minimale Empfehlungen für kleine, unkritische Forschungssoftware (z. B. Datenauswertungsskripte) fest und ist kompatibel zur der in diesem Dokument festgelegten Minimalpraxis in Bezug auf Entwicklung und Dokumentation.

Copyleft

Weiterentwicklungen der Software müssen unter den gleichen Lizenzbedingungen weitergegeben werden wie die Software selbst. Strenges Copyleft: keine Ausnahmen! (GPL, AGPL, CPL) alle Bearbeitungen müssen der Ursprungslizenz unterstellt werden. Beschränktes Copyleft: Ausnahmen sind möglich (MPL, LGPL)-Ausnahme vom strengen Copyleft sind zugelassen.

Non-Copyleft

Weiterentwicklungen der Software können unter anderen Lizenzbedingungen freigegeben werden als die Software selbst (z. B. BSD, Apache) keine Pflichten für die Lizenzierung von neuem oder geänderten Code. Lizenzen mit Sonderrechten – (z. B. NPL, QPL). Der Bearbeiter muss dem Inhaber der Ursprungssoftware Sonderrechte einräumen. Lizenzen mit Wahlmöglichkeit (Artistic / Perl). Der Bearbeiter einer Software darf zwischen verschiedenen Möglichkeiten zur Lizenzierung seiner Bearbeitung wählen.

Entwicklergruppe

Die Entwicklergruppe besteht aus den natürlichen Personen, die an der Entwicklung einer Forschungssoftware beteiligt sind.

¹⁰ https://rse.dlr.de/guidelines/00_dlr-se-guidelines_de.html#anwendungsklassen

¹¹ <https://doi.org/10.5281/zenodo.1344608>

HELMHOLTZ

Open Science

Entwickler*innen

Entwickler*innen sind Personen, die im Rahmen ihrer Tätigkeit Software für wissenschaftliche Zwecke entwickeln (das umfasst sowohl Wissenschaftler*innen, die Software entwickeln als auch Software-entwickler*innen, die Software in wissenschaftlichen Projekten entwickeln).

Forschungssoftware

Mit Forschungssoftware sind alle Formen von Programmcode (z. B. Quellcode nebst zugehörigen Dokumentationen, Parametern und Workflows) und daraus generierten ausführbaren Programmen gemeint, die im Rahmen einer wissenschaftsbezogenen Tätigkeit an Helmholtz-Zentren entwickelt und / oder (nach)genutzt werden.

Lebenszyklus einer Forschungssoftware

Der Lebenszyklus einer Forschungssoftware beschreibt alle wesentlichen Entwicklungsstadien ausgehend von der Idee und Konzeption, über die Entwicklung, Nutzung und Pflege bis hin zur Archivierung und Außerbetriebnahme.

PID

Persistent Identifier (https://en.wikipedia.org/wiki/Persistent_identifier)

Projektleiter*in

Im Zusammenhang mit wissenschaftlichen Projekten ist in der Regel eine Führungskraft mit der Leitung des Projekts betraut. Diese kann in Personalunion auch Software-Verantwortliche/r sein, je nach Größe und Anteil der Forschungssoftware am Projekt.

Release

Bei einem Release handelt es sich um eine stabile Version einer Forschungssoftware, die Nutzern zur Verfügung gestellt wird oder einen Beitrag zu einer wissenschaftlichen Publikation leistet. Einem Schema folgend¹², stellt eine Release-Nummer sicher, dass Release und damit verbundener Inhalt eindeutig gekennzeichnet sind.

RSE

Research Software Engineer(s) (<https://rse.ac.uk/about/>; <https://www.de-rse.org/de/index.html>; <https://www.de-rse.org/de/index.html>)

Softwarepublikation

Analog zu Forschungsdaten und wissenschaftlichen Texten ist auch Forschungssoftware ein Ergebnis wissenschaftlicher Arbeit, sodass sie als wissenschaftliche Publikation einem breiten Publikum bekannt gemacht werden kann. Nichtsdestotrotz kann der Zugang zur Forschungssoftware aus der Publikation

¹² siehe z. B. <https://semver.org/>

HELMHOLTZ

Open Science

analog zu Forschungsdaten eingeschränkt sein. Wie Forschungsdaten und Textpublikationen beinhaltet eine Softwarepublikation Metadaten, mit denen sie auffindbar und zitierbar ist.

Software-Verantwortliche

Der*die Software-Verantwortliche ist Sprecher*in der Entwicklergruppe und fungiert als direkte/r Ansprechpartner*in zu einer Forschungssoftware. Im Fall von Forschungssoftware mit geringem Umfang ist dies i. d. R. der/die Hauptentwickler*in. Der/die Software-Verantwortliche plant die Weiterentwicklung der Forschungssoftware unter Einbeziehung der Nutzerwünsche und der Anforderungen aus unterschiedlichen Forschungsprojekten, stellt Information zum Entwicklungsstatus bereit und gestaltet den Entwicklungsprozess aus. Der*die Software-Verantwortliche wirkt bei der Koordination und Definition von Zugriffs- und Nutzungsrechten mit.

Versionskontrollsystem

Das Versionskontrollsystem dient zur Erfassung von Änderungen an Daten (z. B. Dokumente, Programmcode) durch eine definierte Personengruppe, die in einem gemeinsamen Speicherort („Repository“) verwaltet werden. Dabei wird jede Änderung mit einem Zeitstempel und dem Urheber der Änderung gespeichert. Die sich daraus ergebende Historie erlaubt es Änderungen nachzuvollziehen und zu früheren Ständen zurückzukehren.