

Empfehlungen zur Implementierung von Leit- und Richtlinien zum Umgang mit Forschungssoftware an den Helmholtz-Zentren

Dieses Positionspapier wurde von der Task Group Forschungssoftware¹ des Arbeitskreises Open Science der Helmholtz-Gemeinschaft erarbeitet und vom Arbeitskreis Open Science am 21.11.2019 verabschiedet.

Einführung

Mit der voranschreitenden Digitalisierung von Forschung und Lehre steigt an Forschungseinrichtungen die Zahl an wissenschaftlichen Software-Lösungen, die zur Verarbeitung von Forschungsdaten genutzt werden. Diese Softwarelösungen sind heute unverzichtbar im Prozess der Erkenntnisgewinnung und für die Nachvollziehbarkeit der Ergebnisse.

Mit Forschungssoftware ist in diesem Kontext insbesondere Programmcode (Quellcode nebst zugehörigen Dokumentationen, Parametern und Workflows) gemeint, der im Rahmen einer wissenschaftsbezogenen Tätigkeit entwickelt und / oder genutzt wird. Im Folgenden wird der Quellcode nicht isoliert betrachtet, sondern der gesamte Lebenszyklus von Software-Projekten in den Blick genommen, da auch die Nutzung von kommerzieller Software und vorhandener Open Source Software von Bedeutung für das wissenschaftliche Arbeiten sein kann.

Die unter dem Stichwort Open Science geforderte Verifizierbarkeit und Reproduzierbarkeit von wissenschaftlichen Ergebnissen kann in vielen Fachgebieten nur sichergestellt werden, wenn neben den Forschungsdaten auch der Programmcode nach definierten Kriterien offen zugänglich gemacht wird. Der Programmcode ist ein zentrales und eigenständiges Produkt der wissenschaftlichen Arbeit und muss analog zu anderen wissenschaftlichen Methoden dokumentiert, publiziert und anerkannt werden. Der Zugang zu und die Nachnutzung von Forschungssoftware (Open Research Software) sind somit, neben offenem Zugang zu Publikationen (Open Access) und Forschungsdaten (Open Research Data), ein wesentliches Element von Open Science.²

Forschungssoftware an Forschungseinrichtungen gewinnt deshalb zunehmend an Bedeutung und Aufmerksamkeit. Gleichzeitig mangelt es an Standards, Leit- und Richtlinien, Best Practices sowie Fördermechanismen in Bezug auf die Entwicklung, Publikation und Pflege Forschungssoftware. Wissenschaftlerinnen und Wissenschaftler benötigen Unterstützung beim Management von Forschungssoftware über den gesamten Softwarelebenszyklus. Ein wichtiger Aspekt ist hierbei die Nachvollziehbarkeit von Entwicklungen, Änderungen und Anpassungen. Vor diesem Hintergrund sollte der Umgang mit Forschungssoftware offen und transparent gestaltet werden.

Der Arbeitskreis Open Science der Helmholtz-Gemeinschaft hat im März 2017 das Positionspapier „Zugang zu und Nachnutzung von wissenschaftlicher Software“ verabschiedet.³ Im Folgenden werden diese Positionierungen aufgegriffen, ergänzt und durch praktische Handlungsempfehlungen für die Zentren der Helmholtz-Gemeinschaft konkretisiert.⁴

Die Schaffung von definierten Prozessen und das Aufzeigen von Handlungsoptionen sind Ziele dieser Schrift. Festgehalten wird jedoch auch, dass die Ausgangssituationen je nach Zentrum und nach Forschungsbereich variieren. So muss jedes Zentrum prüfen, welche Bedeutung eine Empfehlung hat, ob diese umgesetzt werden soll und wie sie umgesetzt werden kann. Bei einigen der im Folgenden genannten Handlungsfelder bietet sich eine Kooperation der Zentren an.

Anreize und Metriken

Die bei der Entwicklung von Programmcode erbrachte intellektuelle Leistung und der damit geleistete Beitrag an einem Forschungsergebnis sollte als eigenständiges Produkt des Forschungsprozesses anerkannt und gewürdigt werden. Ein nachhaltiger und offener Umgang mit Forschungssoftware, der effiziente Einsatz von Ressourcen und die Verifizierbarkeit von Forschungsergebnissen kann nur gewährleistet werden, wenn Anreize für die entsprechenden Praktiken zur Realisierung dieser Ziele existieren.

Den Helmholtz-Zentren wird empfohlen, Anreize zu setzen und adäquate und transparente Metriken in Bezug auf Forschungssoftware einzuführen, um die geleistete Arbeit zu erfassen und anzuerkennen.

Zur Anerkennung der wissenschaftlichen Leistung und zur Verankerung der Wertschätzung von Forschungssoftware im Wissenschaftssystem sollte die Veröffentlichung von Programmcode in Form einer Publikation gefördert werden, wie es auch für Fachtexte und Forschungsdaten üblich ist. Im Dialog mit wissenschaftlichen Verlagen sollte darauf hingewirkt werden, die Referenzierungen genutzter Software in den Artikeln zu fordern. Diese Praxis kann sicherstellen, dass die erstellte oder genutzte Software für Dritte nachvollziehbar wird und in scientometrischen Analysen berücksichtigt werden kann. Um die Forschungsleistung eines Zentrums und seiner Mitarbeiterinnen und Mitarbeiter umfassender als bisher zu dokumentieren, sollten die Zentren in Publikationsdatenbanken neben Text- und Datenpublikationen auch Softwarepublikationen als eigenständigen Publikationstyp erfassen, damit dieser bei Evaluierungen analog zu anderen Veröffentlichungsarten berücksichtigt werden kann.⁵

Die Entwicklung und Nutzung von Programmcode und dessen Einfluss auf die Forschungsarbeit Dritter sollte unter Berücksichtigung der Publikationskultur des jeweiligen Forschungsbereichs aktiv über adäquate Mechanismen nachvollzogen werden. Beispiele für Metriken, die Rückschlüsse auf die Anzahl und die Art der Nutzungen zulassen, sind Zitierungen von Software in Publikationen, Downloadstatistiken, Zugriffe auf webbasierte Softwareangebote, Quantität des Feedbacks von Nutzern, Nutzung von Software-Repositories durch Forks und Pull-Requests, oder die Einbindung des Programmcodes in Software Dritter. Weitere Metriken hierfür könnten auch die Anzahl der Erwähnungen in Anträgen, erfolgreich eingeworbene Forschungsgelder und Einnahmen aus Kommerzialisierungsaktivitäten beinhalten. Auch Aspekte wie Alleinstellungsmerkmale, Marktführerschaft und Auszeichnungen können einbezogen werden.

Zusätzlich zu Evaluierungen können die Helmholtz-Zentren Maßnahmen zur Sichtbarmachung und Wertschätzung Forschungssoftwareentwicklung einführen, z. B. in Form von Preisen oder Auszeichnungen. Außerdem bieten Informations- und Beratungsangebote, Aus- und Weiterbildungsangebote sowie entsprechende Karrierewege Anreize für Mitarbeiterinnen und Mitarbeiter, um die Entwicklung und Pflege von Forschungssoftware ihrem Stellenwert entsprechend zu betreiben.⁶

Softwareentwicklungs- und Dokumentationspraxis

Nachhaltige Softwareentwicklung geht Hand in Hand mit guter Dokumentationspraxis. Beide Maßnahmen sollten über entsprechende Infrastrukturen ermöglicht, über entsprechende Fortbildungsangebote vermittelt und durch Richtlinien eingefordert werden.

Allgemeine Grundanforderungen können hierfür, in Analogie zum Laborbuch für die experimentelle Arbeit, in den Richtlinien zur guten wissenschaftlichen Praxis festgehalten werden.⁷

Mit Einführungsveranstaltungen für neue Mitarbeiterinnen und Mitarbeiter können solche Anforderungen unmittelbar in den Zentren verankert werden. Dabei sollte festgehalten werden, dass Änderungen am Programmcode (durch Versionsverwaltung) ebenso wie die Durchführung einzelner Analysen („Laborbuch“) klar dokumentiert werden müssen, um die wissenschaftliche Arbeit nachvollziehbar zu machen. Die Vermittlung dieser Mindeststandards kann außerdem genutzt werden, um auf entsprechende Weiterbildungsmaßnahmen⁸ und weitergehende Empfehlungen zu verweisen. Solche weitergehenden Empfehlungen zur Entwicklungs- und Dokumentationspraxis sollten konkrete Verweise auf Methoden, Software und Plattformen bieten, die nutzbar sind für:

- Software-Planung (Use Cases, Klassendiagramme, UML Modelle, Anforderungs-, Designdokumente etc.),
- Software-Entwicklung (Versionierung, Issue Tracking, Code Review, Style Guides, etc.),
- Software-Testing (Unit Tests, Integration Tests, Debugging, etc.),
- Software-Distribution (klare Versionierung, einfach Installation, etc.),
- Software-Dokumentation (Code-Dokumentation, API-Dokumentation, Nutzer-Dokumentation, etc.).

Empfehlungen dieser Art sind nicht Helmholtz-spezifisch und können daher auch als online verfügbare Empfehlungen öffentlich und kollaborativ zusammengeführt und ergänzt werden.⁹

Auch ein öffentliches Software-Repository unter Betreuung eines Akteurs auf Ebene der Helmholtz-Gemeinschaft ist denkbar. In einem solchen Kontext könnte auch eine Bedarfsanalyse zu den Software- und Plattformlösungen realisiert werden.¹⁰

Zugänglichmachung, Publikations- und Transferstrategien

Der offene Zugang zu Forschungssoftware ist neben dem Zugang zu Publikationen und Daten ein zentraler Baustein von Open Science. Offener Zugang trägt maßgeblich zur Nachhaltigkeit von Forschungsprozessen bei und fördert die verbesserte Wahrnehmung von Softwareentwicklung als fundamentaler

Bestandteil der Forschungstätigkeit. Erst durch das Zusammenspiel der zugänglich gemachten Forschungssoftware mit Textveröffentlichungen und dazugehörigen Forschungsdaten kann die Nachvollziehbarkeit von wissenschaftlichen Resultaten gesichert werden.

Den Helmholtz-Zentren wird daher die Etablierung eines definierten Prozesses empfohlen, der die Zugänglichmachung, die Öffnung und den Transfer von Forschungssoftware begleitet. Hierbei muss zwischen der Veröffentlichung und Bereitstellung von Software im wissenschaftlichen Kontext und einem möglichen Transfer der Software für die kommerzielle Nutzung ebenso unterschieden werden, wie zwischen kleinen Anwendungen für einzelne Analysen, umfangreichen Softwarebibliotheken sowie komplexen Softwaresystemen.¹¹

Wenn keine kommerziellen Verwertungsoptionen oder rechtliche Belange entgegenstehen, ist es Zielsetzung der Zentren der Helmholtz-Gemeinschaft Forschungssoftware ganz oder in Teilen zur freien Nutzung auf einer vertrauenswürdigen Infrastruktur zu veröffentlichen. Dabei sollten, soweit möglich, bereits von Beginn an mit der Erstellung des Codes alle Verwertungsoptionen (Open Source, kommerzielle Nutzung, ...) mitgedacht werden, um später auch alle Optionen offen zu haben. Angemessene Embargofristen (z. B. verbindliche Regelungen für Publikationen oder Abschlussarbeiten) sowie die Wahrung von Wettbewerbsvorteilen sind unter bestimmten Bedingungen zu beachten.

Innerhalb des Zentrums sollte ein kollaborativer Zugang zur Förderung von Synergien¹² und des interdisziplinären Austauschs ermöglicht werden. Der Zugang für Dritte muss unter Berücksichtigung rechtlicher Rahmenbedingungen und der mit der Öffnung und des Transfers einhergehenden Zielstellungen ausgestaltet werden.¹³ Wichtig ist dabei, dass die Software langfristig verfügbar und versionsgenau zitierbar ist.¹⁴

Zur offenen Zugänglichmachung von Software sollte die Bereitstellung zentreneigener sowie zentrenübergreifender Plattformen geprüft werden.¹⁵ Mitarbeitern und Mitarbeiterinnen sollte es ermöglicht werden, Software unter Nutzung von Digital Object Identifiers (DOIs) und anderen Persistent Identifiers (PIDs) zu veröffentlichen.¹⁶

In den Prozess müssen die unterschiedlichen Akteure an den Zentren eingebunden werden: Leitungsebene, Rechtsabteilung, Technologietransfer, Bibliothek, Rechenzentrum / CIO sowie die Öffentlichkeitsarbeit. Der Prozess sollte nachvollziehbar, transparent, offen und proaktiv kommuniziert werden, um softwareentwickelnden Mitarbeiterinnen und Mitarbeitern eine Orientierung zu geben sowie über Möglichkeiten der Zugänglichmachung von Forschungssoftware zu informieren. So kann bei der Planung von Forschungsvorhaben bereits frühzeitig die Zugänglichmachung, Publikation und ein möglicher Transfer von Forschungssoftware berücksichtigt werden, z. B. im Rahmen von Softwaremanagementplänen.¹⁷ Diese Praxis ermöglicht auch eine bessere Berücksichtigung und Planung der benötigten Ressourcen.

Wird Software zur Nachnutzung bereitgestellt, müssen vorab rechtliche Rahmenbedingungen geklärt werden.¹⁸ Es ist zu definieren, wie mit offiziellen Releases und Versionsständen umzugehen ist und wie deren Bereitstellung erfolgen kann. Außerdem sind Aussagen zum Umfang von entgeltfreien und entgeltlichen Services¹⁹ sowie zum Betrieb von Infrastrukturen zu treffen und die dazu notwendigen Ressourcen festzulegen.

Infrastrukturen

Geeignete Infrastrukturen können Mitarbeiterinnen und Mitarbeitern dabei unterstützen Software zu entwickeln, verfügbar zu machen und langfristig zu archivieren. Den Helmholtz-Zentren wird daher empfohlen, hierfür Plattformen zur kollaborativen Softwareentwicklung sowie zur dauerhaften Zugänglichkeit der Software bereit zu stellen. So kann sichergestellt werden, dass die Zentren die Autonomie über Softwareentwicklungen behalten und sich nicht in Abhängigkeit von kommerziellen Dienstleistern begeben. Es muss dabei beachtet werden, dass die gewählten Werkzeuge, die gerade im wissenschaftlichen Umfeld praktizierte agile Softwareentwicklung unterstützen.

Verwaltet werden können solche Infrastrukturen je nach Bedarf und Möglichkeiten von einzelnen Arbeitsgruppen, zentralen Einrichtungen der Helmholtz-Zentren oder kooperativ für die gesamte Helmholtz-Gemeinschaft.

Wenn solche Plattformen offen und kostenlos nutzbar sind, sollte ein Partner- oder Sponsoren-Status erwogen werden, um offene Infrastrukturen nachhaltig zu stützen und den Einsatz der Helmholtz-Gemeinschaft für Open Science zu verdeutlichen.

Darüber hinaus sollte Forschungssoftware, die komplex und essentiell für eine wissenschaftliche Gemeinschaft ist, als Infrastruktur verstanden werden, die über Jahre entwickelt, gewartet und betrieben wird. Es bestehen hohe Anforderungen an Zuverlässigkeit, Qualität und Dokumentation sowie an Training und Community Building. Die Helmholtz-Gemeinschaft und ihre Zentren müssen Forschungssoftware entsprechend langfristig sichern.

Empfohlene Maßnahmen und Rahmenbedingungen für die Schaffung und Einführung geeigneter Infrastrukturen und Prozesse sind:

- Ergänzung der Publikationsrichtlinie und Publikationsdatenbank für die Erfassung und Freigabe von Softwarepublikationen mit Verweis auf geeignete Infrastrukturen zur Speicherung.²⁰
- Aufbau von fachspezifischen Kompetenzzentren (Core Facilities) zur Unterstützung der Wissenschaftlerinnen und Wissenschaftlern bei Design, Implementierung und Optimierung von Software.
- Bereitstellung von Software-Entwicklungsplattformen, die für das Training, den Test, sowie die Startphase neuer Software-Projekte genutzt werden können.²¹
- Bereitstellung einer möglichst diversen und leistungsfähigen Hardware-Testplattform zur kontinuierlichen Integration der entwickelten Versionen.²²
- Archivierung der veröffentlichten Releases in Repositorien mit Vergabe eines Digital Object Identifier (DOI) oder eines anderen Persistent Identifier (PID).

Für einen Teil dieser Maßnahmen bietet es sich an, mittelfristig eine zentrenübergreifende Lösung für die Helmholtz-Gemeinschaft zu entwickeln.

Bei der Bereitstellung eigener Infrastrukturen sollte auf eine nachhaltige Personalpolitik geachtet werden, die sicherstellt, dass das Know-how der Mitarbeiterinnen und Mitarbeiter zum Betrieb der Infrastrukturen und zur Unterstützung der Forschenden längerfristig an den Zentren gehalten wird.

Qualitätssicherung

Der Erkenntnisgewinn in der Wissenschaft hängt zunehmend von Forschungssoftware ab. Damit hat Software einen maßgeblichen Einfluss auf die Qualität der erzielten Forschungsergebnisse sowie deren Reproduzierbarkeit und Verifizierbarkeit. Den Helmholtz-Zentren wird daher empfohlen, angemessene Qualitätsstandards für die Entwicklung und Zugänglichmachung von Forschungssoftware zu definieren und einzuführen.

Bei der Einhaltung von Qualitätsstandards spielen Best Practices des Software Engineerings eine entscheidende Rolle. Die Anwendung bewährter Methoden nach definierten Regeln dient den Mitarbeiterinnen und Mitarbeitern bei der Softwareentwicklung als Orientierung, um Software mit einem hohen Qualitätsniveau zu erstellen, welches Nachvollziehbarkeit, Nachnutzung und Weiterentwicklung ermöglicht. Mindeststandards bei der Entwicklung von Forschungssoftware sollten deshalb erarbeitet, eingesetzt und überprüft werden.

Durch die Nutzung von Werkzeugen und Infrastrukturen, die Prozesse vorgeben und Personal bei der Softwareentwicklung unterstützen, können Arbeitsprozesse in der Softwareentwicklung standardisiert und somit verbessert werden. Hierfür sollten die Helmholtz-Zentren ein entsprechendes Angebot an Werkzeugen und Infrastrukturen bereitstellen und sicherstellen, dass dieses genutzt wird.²³

Mit Hilfe von Checklisten können Software entwickelnde Mitarbeiterinnen und Mitarbeiter selbständig prüfen, ob der entwickelte Programmcode den gestellten Qualitätsanforderungen entspricht. Dabei kann je nach Art der Forschungssoftware zwischen verschiedenen Compliance Levels unterschieden werden. Die Qualitätssicherung kann somit mittels definierter Qualitätskriterien für unterschiedliche Qualitätsstufen erfolgen.²⁴

Daneben sollten die Helmholtz-Zentren geeignete Strukturen schaffen, um Prozesse zur gegenseitigen Unterstützung und wechselseitigen Review als Teil der wissenschaftlichen Arbeit in interdisziplinären Teams zu etablieren. Die Umsetzung geeigneter Reviewverfahren sollte gefördert werden, um Forschungssoftware sowohl aus wissenschaftlicher als auch aus technologischer Perspektive zu begutachten. So kann sichergestellt werden, dass die durch die Software gewonnenen Forschungsergebnisse wissenschaftlich korrekt sind und Dritte die Software einsetzen und Ergebnisse nachvollziehen sowie reproduzieren können. Reviews sollten die Entwicklung von Software kontinuierlich begleiten.

Von besonderer Bedeutung sind Reviewverfahren bei der Veröffentlichung von Forschungssoftware.²⁵ Wünschenswert wären Verfahren, die zu einem zentrenübergreifenden „Software Seal of Approval“ für veröffentlichte Forschungssoftware führen.

Die Veröffentlichung von Forschungssoftware kann auch der Qualitätssicherung und Nachhaltigkeit dienen.²⁶ Dabei sollten Ansprüche an Referenzierung und Analysen zur Relevanz einer bestimmten Software beachtet und die entsprechenden Metriken bedient werden.²⁷ Dies erfordert auch eine entsprechende Qualität der Metadaten. Neben diesen technologischen und organisatorischen Maßnahmen ist ein besonderes Augenmerk auf die kontinuierliche Aus- und Weiterbildung der Mitarbeiterinnen und Mitarbeiter zu richten.²⁸

Lizenzierung und weitere rechtliche Themen

Da Software mehrheitlich urheberrechtlich geschützt ist und ggf. auch ein kommerziell nutzbares Potential aufweist, ist es wichtig, dass bei einer Zugänglichmachung der Software eine bewusste Entscheidung über die Art der Nutzung und damit der Lizenzierung getroffen wird. Unterschieden werden kann zwischen der Lizenzierung zur Erzielung von Einnahmen durch Lizenzgebühren durch Nutzung herkömmlicher Lizenzen oder der weitergehenden Zugänglichmachung der Software durch Nutzung von Open-Source-Lizenzen. Durch deren Nutzung wird die Software lizenzgebührenfrei für Dritte zur Nachnutzung bereitgestellt (wobei auch unter solchen Lizenzen Einnahmen erzielt werden können, bspw. durch Supportleistungen oder Zusatzfunktionen). Die Nutzung offener, etablierter und z. B. durch die Open Source Initiative (OSI) anerkannter Lizenzen wird empfohlen. Dabei sind insbesondere die Vor- und Nachteile von Copyleft-Lizenzen gegenüber Nicht-Copyleft-Lizenzen im Einzelfall abzuwägen.

Den Helmholtz-Zentren wird empfohlen, Prozesse zu etablieren, die sicherstellen, dass eine Klärung der Urheberrechte, der Verwertungsmöglichkeiten und die Wahl der geeigneten Lizenzierungsform für die jeweilige Software garantiert sind.

Ist eine öffentliche Zugänglichmachung der Software vorgesehen so sollte diese Veröffentlichung durch standardisierte und etablierte Open-Source-Lizenzen erfolgen, damit Dritten die Nachnutzung der Software rechtlich abgesichert ermöglicht wird.

Informationsinfrastrukturen und Administration eines Helmholtz-Zentrums sollten Forschende bei der Lizenzierung von Software durch Empfehlungen, Best Practices und klar definierte Prozesse unterstützen. Eine Verankerung des Themas in den Publikationsrichtlinien eines Zentrums kann hier Verbindlichkeit schaffen.²⁹ Die Zentren sollten ein auf Software im Wissenschaftsbetrieb spezialisiertes Informations- und Beratungsangebot schaffen und bei Bedarf die Konsultation von Expertinnen und Experten ermöglichen.³⁰

Aus- und Weiterbildung

Die Qualität von Forschungssoftware hängt entscheidend von den Kenntnissen und Fertigkeiten der Software entwickelnden Mitarbeiterinnen und Mitarbeiter ab. Den Helmholtz-Zentren wird daher empfohlen, Aus- und Weiterbildungsverfahren im Dialog mit Hochschulpartnern zu thematisieren und in den Zentren bewusst Ausbildungs- und Karrierewege im Bereich der Entwicklung von Forschungssoftware zu etablieren.³¹

Neben einer möglichst frühen Verankerung von Programmierkenntnissen in der Ausbildung von Fachwissenschaftlerinnen und -wissenschaftlern sollten auch Möglichkeiten zur Ausbildung von Fachinformatikerinnen und -informatikern für Anwendungsentwicklung oder auch duale / berufsbegleitende Studiengänge der Informatik in Kooperation mit Fachhochschulen evaluiert werden. Auch können Qualifikationsarbeiten (Bachelor, Master und Dissertationen), in denen die Entwicklung von Forschungssoftware eine Rolle spielt, in Kooperation zwischen Fachwissenschaft und Informatik erstellt werden.

Die Stärkung von internen und externen Aus- und Weiterbildungsmaßnahmen sollte im Interesse nachhaltiger Personalentwicklung durch gezielte Wege zur Übernahme exzellenter Kandidatinnen und Kandidaten nach ihrem Abschluss ergänzt werden.

Insbesondere den Helmholtz-Graduiertenschulen oder auch der Helmholtz Data Science Academy kann eine zentrale Bedeutung zukommen, indem z. B. Kurse zu Softwareentwicklung³² angeboten werden.³³ Ergänzend zu zentreninternen Lehrveranstaltungen, die über die Ausbildung von Multiplikatorinnen und Multiplikatoren nachhaltig etabliert werden können, sollte auch eine flexible Anerkennung von externen Angeboten für Doktorandinnen und Doktoranden ermöglicht werden.

Bei der Zusammenarbeit zwischen Mitarbeiterinnen und Mitarbeitern aus Fachwissenschaft und Informatik sollten deren Kommunikationsfähigkeit sowie das Verständnis für die Denkweise der jeweils anderen Seite durch geeignete Maßnahmen gestärkt werden.

In der Praxis führen wechselseitige Kooperationen zwischen Fachwissenschaft und Informatik auch häufig zu flexiblen Verschiebungen zwischen Arbeitsanteilen in der Softwareentwicklung und der jeweiligen Fachwissenschaft – ein Prozess den die Helmholtz-Zentren unterstützend durch gezielte Angebote zur Weiterbildung begleiten sollten. Hierzu können beispielsweise an den oben genannten Helmholtz-Graduiertenschulen Kurse zur Softwareentwicklung auch für Postdocs und weitere Mitarbeiterinnen und Mitarbeiter geöffnet werden. Im Umkehrschluss sollten aber auch Möglichkeiten angedacht werden, damit Informatikerinnen und Informatiker sowie weiteres technisches Personal fachwissenschaftliche Kurse besuchen kann.

Die Helmholtz-Zentren sollten zudem die Vernetzung der oft verteilt arbeitenden Mitarbeiterinnen und Mitarbeiter, die Software entwickeln, innerhalb der Zentren und über Zentren hinweg unterstützen und für koordinierte Fortbildungen sorgen. Hier können neue Lernformate und -methoden gefördert werden.³⁴ Auch Netzwerke auf Ebene der Helmholtz-Forschungsbereiche oder der Helmholtz-Gemeinschaft sind denkbar, z. B. im Rahmen der Helmholtz-Akademie.

Nachhaltigkeit in der Softwareentwicklung sollte von den Zentren darüber hinaus durch klare und längerfristige Karriereperspektiven für Entwicklerinnen und Entwickler gefördert werden. Hierbei ist die Finanzierung von Dauerstellen in verschiedenen Arbeitsgruppen ebenso denkbar wie dedizierte bzw. zentrale Teams mit Entwicklern, die flexibel, zentrenübergreifend und bedarfsabhängig eingesetzt werden. Diese können als eine Art „zentrale Serviceplattform“ über Arbeitsgruppen oder sogar Zentren hinweg arbeiten.³⁵ In jedem Fall sollten dabei die Karrierewege der einzelnen Entwicklerinnen und Entwickler auch über die Arbeit am Zentrum hinaus mitgedacht werden (u. a. Mitnahme von Software-Projekten zu neuen Arbeitsstellen³⁶ sowie die Anerkennung qualitativ hochwertiger und nachhaltig nutzbarer Forschungssoftware als wissenschaftliche Leistung).³⁷

Leit- und Richtlinien

Leit- und Richtlinien sind die Grundlage für einen abgestimmten und organisierten Umgang mit Forschungssoftware. Sie unterstützen und entlasten Personen, die mit Forschungssoftware befasst sind, indem sie grundlegende Verfahren im Umgang mit der Software berücksichtigen, Orientierung geben sowie Ansprechpartnerinnen und Ansprechpartner benennen. Entsprechende Leit- und Richtlinien sollten den gesamten Lebenszyklus von Forschungssoftware erfassen und insbesondere Aussagen zu folgenden Bereichen enthalten: Anreize und Metriken; Softwareentwicklungs- und Dokumentationspraxis; Qualitätssicherung; Lizenzierung und weitere rechtliche Aspekte; Zugänglichmachung, Publikations- und Transferstrategien; Infrastrukturen und Archivierung; Aus- und Weiterbildung.

Die Verankerung von Leit- und Richtlinien bietet eine verlässliche Grundlage für die Weiterentwicklung gemeinsamer Prozesse und Infrastrukturen sowie für einen organisierten Umgang mit Forschungssoftware im digitalen Zeitalter. Die Implementierung sollte durch konkrete Vorlagen und klare Ansprechpartnerinnen und Ansprechpartner unterstützt werden und proaktiv in alle relevanten Bereiche der Zentren getragen werden.

Über die Formulierung von Leit- und Richtlinien hinaus empfiehlt es sich konkrete Muster für Softwaremanagementpläne zu erstellen. Diese Pläne können Mitarbeiterinnen und Mitarbeiter bei der Planung und Durchführung von Aufgaben in der Softwareentwicklung unterstützen, indem sie Handlungsoptionen für den Umgang mit Forschungssoftware benennen und eine Vorlage für den verantwortungsvollen Umgang mit Software schaffen. Insbesondere bei größeren Kollaborationen helfen solche Pläne auch, einen verlässlichen Konsens über den Umgang mit der Software zu schaffen.

Den Helmholtz-Zentren wird daher empfohlen, Expertengremien zum Thema Forschungssoftware mit allen relevanten Akteuren aus Wissenschaft, Informationsinfrastruktur (Bibliotheken, Daten- und Rechenzentren), Wissens- und Technologietransfer und Rechtsabteilungen zu bilden. Dabei sollten alle vorhandenen Kompetenzen für die Formulierung, Anwendung und Einhaltung von Leit- und Richtlinien gebündelt werden. Insbesondere bestehende Best Practices und Policies – z. B. Regeln zur guten wissenschaftlichen Praxis, Publikationsrichtlinien oder Vorgaben zum Technologietransfer – müssen dabei berücksichtigt und mögliche Spannungsfelder offen thematisiert werden. Nur eine abgestimmte und aufeinander referenzierte Policy-Landschaft ermöglicht es, den vielfältigen Herausforderungen der digitalen Transformation in der Wissenschaft zu begegnen und eine stetige Anpassung der Policies zu gewährleisten. Zur erfolgreichen Umsetzung der Leit- und Richtlinien sollten – über die Bereitstellung von Infrastrukturen und Vorlagen hinaus – klare Ansprechpartner benannt werden, die bei Fragen und Anliegen rund um die Leit- und Richtlinien Unterstützung bieten und Know-how vermitteln.

¹ <https://os.helmholtz.de/open-science-in-der-helmholtz-gemeinschaft/akteure-und-ihre-rollen/arbeitskreis-open-science/task-group-forschungssoftware/>

² Siehe hierzu das Selbstverständnis des Arbeitskreis Open Science der Helmholtz-Gemeinschaft unter: <https://os.helmholtz.de/open-science-in-der-helmholtz-gemeinschaft/akteure-und-ihre-rollen/arbeitskreis-open-science/selbstverstaendnis-des-arbeitskreises-open-science-der-helmholtz-gemeinschaft/>

³ Siehe: <https://os.helmholtz.de/open-science-in-der-helmholtz-gemeinschaft/akteure-und-ihre-rollen/arbeitskreis-open-science/zugang-zu-und-nachnutzung-von-wissenschaftlicher-software/>
Darüber hinaus gibt ein Workshop-Report der Task Group „Wissenschaftliche Software“ des Arbeitskreises Open Science einen umfassenden Einblick in das Thema: Scheliga, K. S.; Pampel, H.; Bernstein, E.; Bruch, C.; zu Castell, W.; Diesmann, M.; Fritsch, B.; Fuhrmann, J.; Haas, H.; Hammitzsch, M.; Lähnemann, 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. Potsdam: Deutsches GeoForschungsZentrum GFZ.
<https://doi.org/10.2312/lis.17.01>

⁴ Diese Empfehlungen folgen den vom Arbeitskreis Open Science formulierten „Empfehlungen für Richtlinien der Helmholtz-Zentren zum Umgang mit Forschungsdaten“
<https://doi.org/10.2312/os.helmholtz.002>

⁵ Siehe hierzu auch den Abschnitt „Zugänglichkeit, Publikations- und Transferstrategien“.

⁶ Siehe hierzu auch den Abschnitt „Aus- und Weiterbildung“.

⁷ Siehe hierzu auch den Abschnitt „Abschnitt Leit- und Richtlinien“.

⁸ Siehe hierzu auch den Abschnitt „Aus- und Weiterbildung“.

⁹ Denkbar wäre z. B. auch die Behandlung dieses Thema im Rahmen der Schwerpunktinitiative „Digitale Information“ der Allianz der deutschen Wissenschaftsorganisationen.

¹⁰ Siehe hierzu auch den Abschnitt „Infrastruktur“.

¹¹ Siehe hierzu auch den Abschnitt „Lizenzierung und weitere rechtliche Themen“.

¹² Z. B. durch einen effizienter Ressourceneinsatz und die Vermeidung redundanter Entwicklungen.

¹³ Siehe hierzu auch den Abschnitt „Lizenzierung und weitere rechtliche Themen“.

¹⁴ Siehe hierzu auch den Abschnitt „Anreize und Metriken“.

¹⁵ Siehe hierzu auch den Abschnitt „Infrastruktur“.

¹⁶ Eine Cross-Referenzierung der Software auf Entwicklungsplattformen, Repositorien und Journalen unter Einsatz von PIDs erleichtert die Nachvollziehbarkeit. Dabei kann das gesamte Spektrum an Software-Artefakten mit Programmcode (von Quelltext bis Binaries), Dokumentationen, Anleitungen und Metadaten für die Katalogisierung und Suche berücksichtigt werden, wobei jeweils spezifische Anforderungen an die Qualitätssicherung zu beachten sind. Siehe hierzu auch den Abschnitt „Qualitätssicherung“.

¹⁷ Siehe hierzu z. B. <https://www.software.ac.uk/software-management-plans>

¹⁸ Siehe hierzu auch den Abschnitt „Lizenzierung und weitere rechtliche Themen“.

¹⁹ Z. B. User-Support, Community Building sowie Unterstützung von Entwicklerinnen und Entwicklern.

²⁰ Siehe hierzu auch den Abschnitt „Lizenzierung und weitere rechtliche Themen“.

²¹ Die Plattform sollte Entwicklungswerkzeuge und Projektmanagement-Tools bereitstellen sowie mit externen Plattformen gekoppelt werden können (z. B. GitHub, SourceForge oder GNU Savannah). Mögliche Entwicklungsplattformen sind z. B. GitLab oder Redmine.

²² Hier ist wichtig, dass es den Software-Projekten ermöglicht wird, die Qualität der Software durch automatisierte Tests (Unit-Tests aber auch Tests zur Ergebnisverifikation) zu sichern. Entwicklungs- und Testplattformen können als Software-Container passend für jedes Projekt bereitgestellt und über längere Zeit bewahrt werden.

²³ Siehe hierzu auch den Abschnitt „Infrastruktur“.

²⁴ Hierbei umfasst der Qualitätsbegriff bei Forschungssoftware viele Facetten wie Funktionalität, Benutzerfreundlichkeit, Dokumentation, Zuverlässigkeit, Wartbarkeit, Sicherheit, Portabilität, Kompatibilität und Performance.

²⁵ Hier könnten auch die bereits erwähnten Checklisten zum Einsatz kommen. Wünschenswert wären zentrenübergreifende Checklisten, die gemeinsamen Ansprüchen der Zentren gerecht werden. Siehe hierzu auch den Abschnitt „Zugänglichkeit, Publikations- und Transferstrategien“.

²⁶ Siehe hierzu auch den Abschnitt „Zugänglichkeit, Publikations- und Transferstrategien“.

²⁷ Siehe hierzu auch den Abschnitt „Anreize und Metriken“.

²⁸ Siehe hierzu auch den Abschnitt „Aus- und Weiterbildung“.

²⁹ Siehe hierzu auch den Abschnitt „Leit- und Richtlinien“.

³⁰ Siehe hierzu z. B. das Beratungsangebot am DLR.

³¹ Z. B. unter Berücksichtigung aktueller Entwicklungen zum Thema Research Software Engineers (RSE).

³² Z. B. Best Practices in der Softwareentwicklung als Teil der guten wissenschaftlichen Praxis, Programmiersprachen usw.

³³ Z. B. in Kooperation mit Hochschulpartnern oder anderen wissenschaftsnahen Anbietern und unter Rückgriff auf Open Educational Resources - OER.

³⁴ Z. B.: Massive Open Online Course (MOOCs) und Hackathons.

³⁵ Beispielsweise für die Betreuung größerer Softwareprojekte oder komplizierterer Analyse-Pipelines.

³⁶ Siehe hierzu auch den Abschnitt „Lizenzierung und weitere rechtliche Themen“.

³⁷ Siehe hierzu auch den Abschnitt „Anreize und Metriken“.