

LEITLINIE FÜR DIE ENTWICKLUNG UND WEITERGABE VON SOFTWARE AM FORSCHUNGSZENTRUM JÜLICH

Version 12.1 | Datum: 03.11.2022

Inhalt

1	Executive Summary	2
2	Einleitung	3
3	Grundsätze zur Entwicklung Dokumentation und Weitergabe von Software am Forschungszentrum Jülich	4
3.1	Klassifizierung nach Anwendungsklassen	4
3.2	Mindeststandards und Maßnahmen für die Anwendungsklassen	5
3.3	Bereitstellung, Weitergabe und Zitation	8
4	Unterstützung und Angebote am Forschungszentrum Jülich	9
4.1	Beratungsangebote und Entscheidungshilfen	9
4.2	Unterstützungs- und Weiterbildungsangebote	10
4.3	Wissenschaftliche & wirtschaftliche Verwertung und Lizenzwahl	10
4.4	Open Source-Lizenzen	11
4.5	Proprietäre Lizenzen	11
5	Fazit	12
6	Begriffserläuterungen	12

Anhang

- Checkliste
- Entscheidungsbaum

1 Executive Summary

Software ist ein zentraler Bestandteil der akademischen Forschung und wird in den verschiedenen Instituten am Forschungszentrum Jülich entwickelt und genutzt. Mit Software sind alle Formen von Programmcode und daraus generierten ausführbaren Programmen gemeint, die im Rahmen der Tätigkeit am Forschungszentrum Jülich entwickelt, bereitgestellt und weitergegeben werden.

Bisher existierte am Forschungszentrum Jülich kein praktischer und handlungssicherer Rahmen, in dem sich Mitarbeitende bei der Entwicklung von Software bewegen konnten. Mit dieser Leitlinie legt das Forschungszentrum Jülich nun ein Dokument vor, das unterstützend dazu beitragen will, Software mit hohen Qualitätsstandards zu entwickeln, zu managen und Wirkung zu erzielen. Diese Leitlinie richtet sich vor allem an alle Software-Entwickler:innen und Führungskräfte, die am Forschungszentrum Jülich an der Entwicklung, Verwaltung und Weitergabe von Software direkt oder indirekt beteiligt sind. Ziel der Leitlinie ist es, am Forschungszentrum Jülich einen handlungssicheren und nachhaltigen Umgang mit programmierter Software zu etablieren, die Softwarequalität zu verbessern und die Wertschätzung für qualitativ hochwertige Software zu erhöhen. Mit der dadurch generierten Professionalisierung im Bereich der Software-Entwicklung soll ein nachhaltiger Umgang mit Software erreicht und gute wissenschaftliche Praxis im Sinne von Verifizierbarkeit und Reproduzierbarkeit von Forschungsergebnissen gefördert werden.

Bei der Entwicklung der Leitlinie wurden sowohl interne Rahmenbedingungen berücksichtigt, wie z.B. die Regeln zur Sicherung guter wissenschaftlicher Praxis, die internen Vorgaben der „Veröffentlichungsrichtlinie“, der „Leitlinie Forschungsdaten“, die Innovationsstrategie „Forschung und Innovation 2025“ als auch externe Empfehlungen, wie z.B. die „Checkliste zur Unterstützung der Helmholtz-Zentren bei der Implementierung von Richtlinien für nachhaltige Forschungssoftware“ des Arbeitskreises Open Science der Helmholtz Gemeinschaft.

Der Entwurf basiert auf der Muster-Richtlinie „Nachhaltige Forschungssoftware an den Helmholtz-Zentren“ des Arbeitskreises Open Science der Helmholtz Gemeinschaft, welche an die Gegebenheiten und Bedürfnisse des Forschungszentrum Jülich angepasst wurde.

In der Leitlinie werden Software bzw. Software-Projekte in vier verschiedene Anwendungsklassen eingeteilt. Dabei wird Software, welche nur innerhalb des eigenen Teams bzw. innerhalb des Forschungszentrums genutzt werden soll (Anwendungsklasse 0) bis hin zu Software, welche an Dritte lizenziert werden soll und einen Produktcharakter aufweist (Anwendungsklasse 3) abgebildet. Abhängig von der jeweiligen Anwendungsklasse werden gewisse Mindestanforderungen und Maßnahmen, welche sich vor allem auf die Software-Entwicklung und die Dokumentation beziehen, empfohlen, um hohe Standards bei Softwareentwicklung, Softwarequalität und Management zu ermöglichen.

Wenn eine Bereitstellung bzw. Weitergabe der Software gewünscht oder gefordert ist, stimmen sich die Software-Entwickler:innen, die Projektverantwortlichen und/oder Institutsleitungen über den geeigneten Transferweg ab. Dabei besteht die Möglichkeit, die Software entweder über eine Open Source Lizenz offen zu stellen, oder über eine proprietäre Lizenz an Dritte weiterzugeben. Falls die Weitergabe über eine Open Source Lizenz gewünscht ist, obliegt dem jeweiligen Institut die Auswahl der entsprechenden Open Source Lizenz. Sind eine wirtschaftliche Verwertung bzw. eine Einschränkung der Nutzungsrechte und somit eine proprietäre Lizenzierung der Software an Dritte gewünscht, so stimmt sich das Institut mit dem Fachbereich Innovation und Strategie der Unternehmensentwicklung (UE-I) ab, welcher die Erstellung und Verhandlung jeglicher proprietärer Lizenzverträge verantwortet.

Das Forschungszentrum Jülich unterstützt die Bereitstellung und Weitergabe von am Forschungszentrum Jülich entwickelter Software unabhängig davon, ob der Code mit einer proprietären Lizenz, also zur eingeschränkten Nutzung, oder Open Source geteilt werden soll. Bei einer Veröffentlichung als Open Source Software ist die Reichweite und der Transfereffekt enorm, da Millionen von Entwickler:innen auf Plattformen wie GitHub aktiv sind. Aus diesem Grund und zur Erfüllung des Kriteriums der Nachhaltigkeit im Wissenschaftsbetrieb ermutigt das Forschungszentrum Jülich die Forschenden dazu, Software als „Open Source“ offen zugänglich zu machen.

Für die Weitergabe bzw. Bereitstellung müssen u.a. die rechtlichen Aspekte, mögliche Förderbedingungen, geeignete Transferwege und Freigabeprozesse berücksichtigt werden. Als Entscheidungshilfe stehen hierfür die Checkliste (Anlage A) und der Entscheidungsbaum (Anlage B) zur Verfügung.

Darüber hinaus können bei Bedarf die Beratungsangebote der Unternehmensentwicklung (UE), dem Geschäftsbereich Recht (R) und der Zentralbibliothek (ZB) in Anspruch genommen werden. Die Entscheidungshilfen und Beratungsangebote des Forschungszentrums Jülich sollen zukünftig zentral über das neu aufgesetzte RSE-Portal zur Verfügung gestellt werden. Dieses Portal wird gemeinsam von UE, R und ZB moderiert und inhaltlich mit den Software-Entwickler:innen gestaltet. Dort werden ergänzende Informationen wie Best Practice Beispiele, FAQ, Ansprechpersonen und Ankündigungen zu interne/externe Trainings und Weiterbildungen bereitgestellt und regelmäßig aktualisiert.

Schulungen zu Softwareentwicklung sind von entscheidender Bedeutung, um wissenschaftlichen Nachwuchs zu hohen Standards in der Softwareentwicklung und Dokumentation zu befähigen. Daher möchte das Forschungszentrum Jülich das bestehende interne Schulungsangebot zu Softwareentwicklungs-Methodiken und Best Practices zukünftig weiter ausbauen.

2 Einleitung

Software wird in praktisch allen Instituten des Forschungszentrums Jülich entwickelt und genutzt. Mit Software sind in dieser Leitlinie alle Formen von Programmcode (z. B. Quellcode nebst zugehöriger *Dokumentation* [1]) und daraus generierten ausführbaren Programmen gemeint, die im Rahmen der Tätigkeit am Forschungszentrum Jülich entwickelt, bereitgestellt und weitergegeben werden. Die Entwicklung von Software ist Teil eines kreativen Prozesses und generiert in diesem Sinne ausführbares Wissen. Sie ist integraler Bestandteil moderner Publikationszusammenhänge aus schriftlicher Publikation, Datensätzen und Software. Darüber hinaus ist die Software-Entwicklung eine intellektuelle und urheberrechtlich geschützte Leistung und im Kontext der Forschung ein eigenständiges Produkt der wissenschaftlichen Arbeit.

Die meisten Wissenschaftler:innen, die Codes schreiben, sind keine ausgebildeten Software-Entwickler:innen und orientieren sich beim Software Engineering häufig an ihrem unmittelbaren Arbeitsumfeld und den Erfahrungen der Kolleg:innen, ohne dabei zwangsläufig die Beratungsangebote, Instrumente und Best Practice Erfahrungen der Einrichtung zu kennen.

Um diese Wissenschaftler:innen in ihrer Eigenständigkeit und Handlungsfähigkeit zu unterstützen, bildet diese Leitlinie einen Rahmen für die Entwicklung, Verwaltung und Weitergabe von Software am Forschungszentrum Jülich. Die Leitlinie legt die grundsätzlichen Anforderungen an die Entwicklung, Verwaltung und Weitergabe von Software fest, beschreibt praktische Abläufe sowie entsprechende Beispiele und Beratungsangebote.

Die Leitlinie behandelt den *Software-Lebenszyklus*, von der Software-Entwicklung über die *Dokumentation* bis zu Weitergabe und Pflege der Software. Die Leitlinie soll unterstützend dazu beitragen, moderne Software Engineering Methoden am Forschungszentrum Jülich zu etablieren, welche hohe Standards bei Softwareentwicklung, Softwarequalität und Management ermöglichen. Mit dieser Professionalisierung wird mehr Nachhaltigkeit erreicht und gute wissenschaftliche Praxis im Sinne von Verifizierbarkeit und Reproduzierbarkeit von Forschungsergebnissen gefördert.

Diese Leitlinie richtet sich vor allem an alle Entwickler:innen und Führungskräfte, die am Forschungszentrum Jülich an der Entwicklung, Verwaltung und Weitergabe von Software direkt oder indirekt beteiligt sind. Ziel der Leitlinie ist es, am Forschungszentrum Jülich einen handlungssicheren und nachhaltigen Umgang mit programmierter Software zu etablieren, die Softwarequalität zu verbessern und die Wertschätzung für qualitativ hochwertige Software zu erhöhen.

Die Leitlinie berücksichtigt die Sichtweise der Helmholtz-Gemeinschaft auf Open Science und basiert auf der Muster-Richtlinie „Nachhaltige Forschungssoftware an den Helmholtz-Zentren“ [2] des

1 Die in dieser Leitlinie kursiv und blau markierten Begriffe sind am Ende des Dokuments unter Kapitel 6 „Begriffserläuterungen“ zum besseren Verständnis weiter ausgeführt.

2 doi.org/10.2312/os.helmholtz.007

Arbeitskreises Open Science der Helmholtz Gemeinschaft. Sie reflektiert und implementiert die Regeln zur Sicherung guter wissenschaftlicher Praxis [3], die internen Vorgaben der „Veröffentlichungsrichtlinie“ [4], der „Leitlinie Forschungsdaten“ [5], der „Richtlinie *Informationssicherheit*“ (in Erstellung) und berücksichtigt darüber hinaus die Herausforderungen und Handlungsfelder der Innovationsstrategie des Forschungszentrums [6] in diesem Zusammenhang. Zusätzlich wurden bei der Erstellung der Leitlinie die vom Helmholtz Forum Forschungssoftware und dem Helmholtz Open Science Büro formulierten „Checkliste zur Unterstützung der Helmholtz-Zentren bei der Implementierung von Richtlinien für nachhaltige Forschungssoftware“ [7] berücksichtigt.

Im Sinne von Open Science sind der Zugang zu und die Nachnutzung von Software die Basis für Nachvollziehbarkeit, Verifizierbarkeit und Reproduzierbarkeit von wissenschaftlichen Ergebnissen. In Parallele zu den FAIR-Prinzipien [8] (**F**indable, **A**ccessible, **I**nteroperable, **R**eusable), die für Forschungsdaten gelten, wendet das Forschungszentrum Jülich diese Grundsätze zur Nachhaltigkeit auch für Software an. Zur Erfüllung des Kriteriums der Nachhaltigkeit im Wissenschaftsbetrieb ermutigt das Forschungszentrum Jülich die Forschenden dazu, Software als „*Open Source*“ offen zugänglich zu machen. Am Beginn eines Software-Projektes ist jedoch von dem entwickelnden Institut abzuwägen, auf welchem Transferweg die beste Wirkung für Gesellschaft und Wirtschaft erzielt werden kann. Gibt es kommerzielle Anwendungen für die Software, kann eine wirtschaftliche Verwertung über *proprietäre Lizenzen* sinnvoller sein als eine *Open Source Veröffentlichung*. Auch *proprietäre Lizenzen* können Innovationen z.B. in einem Unternehmen ermöglichen und einen Technologievorsprung z.B. einer Ausgründung sichern. Open Science einerseits und kommerzielle Verwertung andererseits stehen nicht im Widerspruch zueinander, sondern sind sich ergänzende Elemente der Innovationsstrategie des Forschungszentrums Jülich.

3 Grundsätze zur Entwicklung Dokumentation und Weitergabe von Software am Forschungszentrum Jülich

Software und *Software-Projekte* unterscheiden sich untereinander sowohl in ihrer Komplexität als auch in ihrem Reifegrad. Die Komplexität eines *Software-Projektes* kann aus technischen Aspekten resultieren aber auch aus Management-Aspekten, wie der gemeinsamen Softwareentwicklung in einem großen Konsortium. Darüber hinaus spielen das wissenschaftliche Fachgebiet, die eingesetzten Programmiertechnologien, die Zielgruppe des Projektes und das mögliche Risiko (z.B. Haftung bei Fehlfunktionen/Ausfall), das mit einem späteren Software-Produkt verbunden sein kann, eine Rolle. In der Praxis ist die Entwicklung von Software eine iterative und fortlaufende Aktivität, die sich mit dem *Software-Lebenszyklus* verändert. Die Maßnahmen und Qualitätsstandards, die das Forschungszentrum Jülich für die Software-Qualität festlegt, richten sich nach der *Anwendungsklasse* (AK) der entwickelten Software. Zu Beginn eines Projektes wird von den Verantwortlichen für jede Software-Entwicklung eine *Anwendungsklasse* festgelegt, für die die beschriebenen Maßnahmen zu Qualität und *Dokumentation* einzuhalten sind. Da erfahrungsgemäß viele Projekte klein starten, sich dann aber weiterentwickeln, empfiehlt es sich, bereits zu Beginn Standards höherer Anwendungsklassen zu verwenden. Diese *Anwendungsklassen* orientieren sich an den *Anwendungsklassen*, die seitens des Deutschen Zentrums für Luft- und Raumfahrt (DLR) [9] vorgeschlagen und implementiert wurden. Die Qualitätsanforderungen für das Software-Engineering steigen dabei mit jeder *Anwendungsklasse*.

3.1 Klassifizierung nach Anwendungsklassen

Die *Anwendungsklassen* richten sich v.a. danach, wie die Software genutzt und ob bzw. in welchem Umfang sie weitergegeben werden soll. Eine geplante oder geforderte Weitergabe der Software ist

3 IR 118-2 Regeln zur Sicherung guter wissenschaftlicher Praxis in der Forschungszentrum Jülich GmbH

4 Richtlinie Nr. 2/2020 Veröffentlichungen des Forschungszentrums Jülich GmbH

5 Rundschreiben Nr. 06/2019 Umgang mit Forschungsdaten

6 Forschung und Innovation 2025: Innovationsstrategie des Forschungszentrums Jülich; Version 1, 02.04.2019

7 <https://doi.org/10.48440/os.helmholtz.031>

8 www.force11.org/fairprinciples

9 go.fzj.de/dlr-rse

ein entscheidendes Kriterium für die Auswahl der *Anwendungsklasse*. Die Maßnahmen und Standards für die entsprechende *Anwendungsklasse* beziehen sich jedoch v.a. auf Software-Entwicklung und *Dokumentation*. Die Komplexität eines Softwareprojektes und die Reife der Software spiegeln sich deshalb nicht zwingend in den *Anwendungsklassen* wider. Ein Wechsel zwischen den *Anwendungsklassen* ist möglich, i. d. R. jedoch von niedrigeren in höhere Klassen. Es kann deshalb hilfreich sein, sich schon an den Standards der höheren *Anwendungsklasse* zu orientieren, wenn eine Weitergabe der Software nicht ausgeschlossen ist. Zu beachten ist der notwendige Ressourceneinsatz, der über den gesamten *Software-Lebenszyklus* im Institut sichergestellt werden muss.

Tabelle 1: Kurzübersicht zu den Anwendungsklassen für Software

Anwendungs- klasse	0	1	2	3
Entwicklung	im FZJ	im FZJ	im FZJ oder mit Dritten	im FZJ oder mit Dritten
Nutzung	Persönlich und intern im Projekt-Team.	Im Institut	Als längerfristig verwertbares Ergebnis aus Drittmittelprojekten oder als Produkt geplant.	Produktcharakter
Anforderung	Beachtung rechtlicher Aspekte	Versionskontrollsystem wird verwendet. Für Unbeteiligte ist Weiterentwicklung möglich.	Wartbarkeit und Verwertbarkeit sind gegeben. Nutzungs- und Verwertungsrechte liegen bei FZJ.	Testautomatisierung, Release- und Wartungsmanagement sind gegeben.
Weitergabe	Außerhalb des Entwickler:innen-Teams nicht vorgesehen.	Nur in eng begrenztem Rahmen z.B. im Institut oder an die Einrichtung der Betreuer:innen. Bei notwendiger Weitergabe an Personen außerhalb des FZJ, wenn möglich mit Lizenz.	Ja, mit Lizenz (OSS oder proprietär).	Ja, mit Lizenz (OSS oder proprietär).
Beispiele	Code im geringsten Umfang, einzelne Funktionen, einfache Skripte	Software aus Dissertationen mit Demonstrationscharakter.	Softwarepublikationen, Software, die in Kooperationen mit Partnern entwickelt und verwendet wird.	Software, die kommerziell verwertet werden soll, z.B. in einer Ausgründung. Software, die in einem großen Open Source Projekt entsteht.

3.2 Mindeststandards und Maßnahmen für die Anwendungsklassen

Ab Anwendungsklasse 0:

1. Die Beachtung rechtlicher Aspekte (z. B. Rechte Dritter) ist integraler Bestandteil des Softwareentwicklungsprozesses und erfordert eine entsprechende *Dokumentation* (z. B. Urheberrechtsvermerke). Hilfestellung zu diesem Thema bietet die Checkliste (Anlage A).

Beispiel: Wird bei der Programmierung auf vorbestehender Software aufgebaut oder werden Bibliotheken verwendet, sind diese Lizenzbedingungen zu beachten.

Die Verwendung eines Versionskontrollsystems wird empfohlen.

Ab Anwendungsklasse 1 zusätzlich:

2. Versionskontrollsysteme und *Code Repositories* werden verwendet, die sämtliche zur Nutzung notwendigen Bestandteile der Software enthalten bzw. referenzieren und für das Änderungsmanagement eingesetzt werden.

Beispiel: Jegliche Änderungen am Quellcode werden mit Hilfe von Git dokumentiert.

3. Eine kurze, aussagekräftige Beschreibung des Zwecks der Software ist beizulegen (z. B. in Form einer README-Datei im *Repository*). Neue Beitragende sollen sich an der Entwicklung der Software beteiligen können.

Beispiel: Erläuterung des wissenschaftlichen Kontexts der Software, Verweis auf Publikationen und Beteiligte sowie technische Hinweise zur Nutzung und Weiterentwicklung. Insbesondere neuen Beitragenden erleichtert dies den Zugang.

4. Zur Sicherstellung der Zitierbarkeit sind *Metadaten* maschinenlesbar als *Zitationshinweis* [10] beizulegen.

Beispiel: Das „Citation File Format“ und „CodeMeta“ erlauben Indizierung und Einblenden von Zitationshinweisen auf Plattformen wie Github.

5. Die Werkzeuge und Standards für die Software-Entwicklung, -Validierung, und -Verifizierung sind mit dem Entwicklungsteam (und den Vorgesetzten) abzustimmen und ihre Nutzung ist zu dokumentieren. Softwareentwicklung erfolgt nach Möglichkeit nach anerkannten Standards und mit Werkzeugen auf aktuellem *Stand der Technik* während der Entwicklung, Validierung, Verifizierung und der Bereitstellung.

Beispiel: Tools wie SoftWipe <https://github.com/adrianzap/softwipe> und Publikationen wie „Best Practice for Scientific Computing“ (Wilson et al. 2014; PLOS Vol.12 Issue 1). Mehr Best Practice Beispiele, Tools und Publikationen sind auf dem RSE-Portal (siehe Kapitel 4) aufgeführt.

6. Versionen (*Releases*) sind für Nutzer:innen eindeutig, beispielsweise über *Release*-Nummern, zu kennzeichnen. Für große Software-Projekte empfiehlt sich die Festlegung einer gemeinsamen Release-Policy.

Beispiel: Für Reproduzierbarkeit von Ergebnissen und Kommunikation sind Git „tags“ enorm hilfreich, insbesondere in Kombination mit einem persistenten Identifikator.

7. Zum Zweck der Referenzier- und Zitierbarkeit der Software sind Versionen (und ggf. der Software als solche) ein *persistenter Identifikator (PID)* zuzuordnen. Dazu notwendige *Metadaten* sind im *Zitationshinweis* (s. o.) zu finden.

Beispiel: Auch nicht öffentlich zugängliche Software kann FAIR sein und z. B. einen DOI erhalten, die zur Verwendung in Textpublikationen einfacher und etablierter ist. Zusätzlich ermöglicht ein DOI nachträgliches Verschieben ohne die Referenz zu brechen. Reproduzierbarkeit wird erheblich durch referenzierbare Versionen von Daten, Software und Umgebung erleichtert.

8. Jede *Software-Publikation* des Forschungszentrums ist analog zu Daten-Publikationen im institutionellen *Repository* mindestens mit *Metadaten* nachzuweisen. Sofern keine *Publikation* erstellt wird, sollte Software in einem zentralen internen Software-Katalog auffindbar gemacht werden.

Ab Anwendungsklasse 2 zusätzlich:

9. Zur Weitergabe und Verwendung der Software wird eine angemessene Installations-, Entwicklungs- und Anwendungsdokumentation bereitgestellt. Die *Dokumentation* sollte für andere leicht verständlich sein, um Beiträge anderer Beitragender zu erleichtern.

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

10. Funktionstests der Software werden systematisch ausgeführt. Dazu wird eine adäquate Teststrategie festgelegt und dokumentiert.
11. Wird die Software im Rahmen einer Kooperation entwickelt, sind die Zuwendungsbedingungen und die Regelungen der Verträge, bzw. des Grant Agreements und/oder die Vorgaben des Konsortiums zu beachten.

Tragen einzelne Entwickler:innen des Forschungszentrums Jülich Code zu Software-Projekten von Dritten bei, sollte ein Contributor License Agreement (CLA) geschlossen werden, falls kein Kooperationsvertrag geschlossen wurde. Ein Muster CLA ist im RSE-Portal abrufbar.

12. Vor der Weitergabe an Dritte außerhalb des Forschungszentrums wird eine [Lizenz](#) für die Software festgelegt, die die Nutzung der Software durch Dritte regelt. Lizenzbestimmungen verwendeter Fremdsoftware werden eingehalten. Die Software wird nicht ohne [Lizenz](#) weitergegeben. Ein Urheberrechtsvermerk ist für die Weitergabe notwendig.

Beispiel 1: Die Software wird Open Source gestellt. Eine geeignete Open Source Lizenz wird vom Institut mit Hilfe der Checkliste ausgewählt und zur Weitergabe verwendet (siehe Kapitel 3.3).

Beispiel 2: Die Software wird eingeschränkt weitergegeben. UE-I erstellt gemeinsam mit dem Institut einen individuellen Software-Lizenzvertrag, der die Nutzung der Software regelt.

13. Für die Weitergabe von Software werden die Freigabeprozesse des Forschungszentrums/des Instituts beachtet.

Beispiel 1: Die Software soll Open Source gestellt werden. Die jeweilige Institutsleitung oder eine von der Institutsleitung bevollmächtigte bzw. beauftragte Person genehmigt die Veröffentlichung.

Beispiel 2: Die Software soll eingeschränkt an Dritte weitergegeben werden. Das Institut hält Rücksprache mit UE-I. UE-I erstellt gemeinsam mit dem Institut einen individuellen Software-Lizenzvertrag, der die Nutzung der Software regelt.

14. Vor der Weitergabe an Dritte außerhalb des Forschungszentrums ist sicher zu stellen, dass die Regelungen zur Exportkontrolle (Außenwirtschaftsrecht) eingehalten wurden.

Beispiel: Vor der Veröffentlichung wird durch das Institut anhand der EG-Dual-Use Liste geprüft, ob die Software gelistet ist. Bestehen Anhaltspunkte dafür, ist der Geschäftsbereich Recht einzubeziehen. So ist beispielsweise eine Software nach 3D001 gelistet, wenn diese für die Entwicklung oder Herstellung von Ausrüstung für die Fertigung von Halbleiterbauelementen oder -materialien besonders entwickelt worden ist. Hinweise zur Prüfung anhand der Liste sind im RSE-Portal zu finden.

Ab Anwendungsklasse 3 zusätzlich:

15. Ein hoher Grad an Abdeckung mit Testautomatisierung ist gegeben.

Beispiel: Continuous Integration oder sogar Continuous Delivery können für die Testautomatisierung genutzt werden. So sind GitLab CI, GitHub Actions, etc. niederschwellig nutzbare und in die Weboberflächen gut integrierte Angebote.

16. Das Institut hat die Software-Wartung und Fehlerbehebung organisiert.

Beispiel: Die Software kann entweder mit eigenen Ressourcen gewartet werden oder aber die Community übernimmt Wartung und Fehlerbehebung. Möglich ist auch, eine Firma mit dieser Aufgabe zu beauftragen.

Ausnahmen:

Sogenannte [Snippets](#), kurze Ausschnitte aus dem Quelltext, unterfallen keiner Anwendungsklasse und können ohne Lizenz weitergegeben werden.

3.3 Bereitstellung, Weitergabe und Zitation

Das Forschungszentrum Jülich unterstützt die Bereitstellung und Weitergabe von am Forschungszentrum Jülich entwickelter Software unabhängig davon, ob die Software im *Source Code* oder im *Object Code* weitergegeben wird und unabhängig davon, ob der Code mit einer proprietären *Lizenz*, also zur eingeschränkten Nutzung, oder Open Source geteilt werden soll. Bei einer Veröffentlichung als Open Source Software ist die Reichweite und der Transfereffekt enorm, da Millionen von Entwickler:innen auf Plattformen wie GitHub aktiv sind. Aus diesem Grund und zur Erfüllung des Kriteriums der Nachhaltigkeit im Wissenschaftsbetrieb ermutigt das Forschungszentrum Jülich die Forschenden dazu, Software als „Open Source“ offen zugänglich zu machen (siehe Kapitel 4.3).

Nach der oben genannten Einordnung der Software in *Anwendungsklassen* ist Software der *Anwendungsklassen* 2 und 3 zur Weitergabe geeignet. Dabei wird unterschieden, ob die Weitergabe als frei zugängliche *Publikation* erfolgt, also z.B. über Daten-Repositorien, Journale und *Archive*, oder im Rahmen einer *Lizenz*, mit der dem User *Nutzungsrechte* an der Software eingeräumt werden,

Für jede Bereitstellung von Software durch Mitarbeitende des Forschungszentrums Jülich ist der Freigabeprozess des Instituts zu beachten, dem der Mitarbeitende angehört. Der Freigabeprozess wird von dem jeweiligen Institut eigenverantwortlich festgelegt. Die OE-Leitung kann sich dabei an der Richtlinie Nr. „2/2020 Veröffentlichungen des Forschungszentrums Jülich GmbH“ orientieren. Die Verantwortung kann dabei von der Institutsleitung z.B. an die Projektverantwortlichen oder die Software-Entwickler:innen delegiert werden.

Für die Zitation von Software gilt:

Im Sinne der *Leitlinien guter wissenschaftlicher Praxis der DFG* fordert das Forschungszentrum Jülich seine Mitarbeitenden zur Kenntlichmachung der Verwendung eigener oder fremder wissenschaftlicher Software durch Zitation bzw. Referenzierung in ihren Publikationen auf.

Zitation und Referenzierung sind die Kernelemente für Reproduzierbarkeit von Ergebnissen und Sichtbarkeit von nichtwissenschaftlichen Mitarbeitenden. Idealerweise stehen dazu *FAIR Software-Publikationen* mit *persistenten Identifikatoren* zur Verfügung. Diesen können i. d. R. die notwendigen Daten entnommen werden. Sind diese nicht vorhanden, sind *Zitationshinweise* ggf. in den *Code-Repository* vorhanden (siehe Kapitel 3.2).

Software ist unabhängig von ihrer öffentlichen Zugänglichkeit mit folgenden Angaben in einer wissenschaftlichen *Publikation* zu zitieren [11], [12]:

Autor:innen, Name der Software, PID, Veröffentlichungsdatum, [eventuell vorhandene Release-Nummer]

Falls es keine *PID* gibt, kann alternativ eine Revision und die URL des Quelltext-*Repositoriums* (oder eines qualifizierten Software Archives wie das „Software Heritage Archive“ [13]) angegeben werden.

Für die Bereitstellung und Weitergabe von Software gilt:

1. Die Erstellung der Software erfolgte nach den Standards der jeweiligen *Anwendungsklasse*, ist gut dokumentiert und versioniert.
2. Die (Urheber-)rechtlichen Aspekte, potentielle Rechte Dritter, Fördervorgaben und Freigabeprozesse müssen für die Weitergabe und *Publikationen* der Software beachtet werden.
3. Vor der Bereitstellung und Weitergabe von Software an Dritte sollte der gewünschte Transferweg (Open Source/proprietäre *Lizenz*) definiert werden.

11 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>

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

13 <https://www.softwareheritage.org>

Wichtige zu berücksichtigende Punkte sind hierbei u.a. der Grad der Offenheit und der gewährte Zugang, mögliche kommerzielle Verwertungsvorhaben und etwaige (rechtliche) Einschränkungen. Dies bildet die Grundlage für die Entscheidung über den Transferweg. Bei Bedarf können hierzu die in Kapitel 4.1 genannten Beratungsangebote und Entscheidungshilfen hinzugezogen werden.

4. Wann immer möglich, hinterlegen Entwickler:innen die der *Publikation* zugrunde liegende Software zugänglich in anerkannten *Archiven* und *Repositorien* [14].
5. Unabhängig von der Zugänglichkeit der Software selbst, sind *Metadaten* zur Software im Sinne einer *Software-Publikation* zu veröffentlichen.
6. Unabhängig von der Art der Bereitstellung und Weitergabe, sei es frei zugänglich oder proprietär, müssen dem User die entsprechenden *Nutzungsrechte* durch eine *Lizenz* eingeräumt werden (siehe Kapitel 4.3).
7. Bei der Einräumung von *Nutzungsrechten* an einer am Forschungszentrum Jülich entwickelten Software wird, unabhängig davon, ob die Software mit einer Open Source oder einer proprietären *Lizenz* weitergegeben wird, das *Urheberrecht* i. d. R. folgendermaßen angegeben:

© (20JJ) Forschungszentrum Jülich GmbH, [<beteiligte Institute>].

Im *Zitationshinweis* oder in einem gesonderten Dokument (contributor list) sind alle Urheber:innen zu nennen.

8. Die Realisierung der langfristigen Zugänglichkeit sollte unter Einhaltung der Qualitätsstandards sichergestellt sein.
9. Die *Publikation* kann mit Embargofristen verknüpft werden (Analog zu Rückhalte- und Sperrfristen bei Daten- und Textpublikationen).

Für die Umsetzung der Maßnahmen zur Bereitstellung und Weitergabe von Software können die Checkliste (Anlage A) und der Entscheidungsbaum (Anlage B) zu Hilfe genommen werden.

4 Unterstützung und Angebote am Forschungszentrum Jülich

Um die Entwicklung von Software, die hohen Qualitätsstandards genügt, in Jülich zu unterstützen, stellt das Forschungszentrum Jülich Beratungs-, Unterstützungs- und Weiterbildungsangebote bereit. Dabei beziehen sich Beratung und Informationen auf alle Phasen des *Software-Lebenszyklus* und die Weitergabe der Software. Die verschiedenen Angebote zum Thema Software werden in ausführlicher Form zentral auf dem RSE-Portal dokumentiert und regelmäßig aktualisiert. Das Portal wird gemeinsam von der Unternehmensentwicklung (UE), dem Geschäftsbereich Recht (R) und der Zentralbibliothek (ZB) moderiert und inhaltlich gemeinsam mit den Software-Entwickler:innen gestaltet.

4.1 Beratungsangebote und Entscheidungshilfen

Bei Fragen und Unklarheiten im Umgang mit selbst programmierter Software kann ein persönliches Beratungsgespräch mit den Ansprechpersonen von UE, R und/oder ZB vereinbart werden, deren aktuelle Kontaktdaten auf dem RSE-Portal zu finden sind. In den Beratungsgesprächen werden dabei Fragen geklärt wie z.B. welche Wirkung die Software entfalten soll, welcher Transfer- oder Publikationsweg geeignet ist (z.B. Open Source Software oder proprietäre Software) und welche (Urheber-)rechtlichen Rahmenbedingungen gegeben sind bzw. berücksichtigt werden müssen. Bei einer gewünschten proprietären Lizenzierung der Software (siehe Kapitel 4.5) werden in einem diesbezüglichen Beratungsgespräch die Rahmenbedingungen geklärt und anschließend ein geeigneter Lizenzvertrag erarbeitet.

14 Damit sind z. B. Daten-Repositorien gemeint

Neben der persönlichen Beratung bieten die auf dem RSE-Portal bereitgestellten Informationen und Werkzeuge praktische Hilfe an. Die Checkliste (Anlage A) und der Entscheidungsbaum (Anlage B) helfen, offene Punkte und wichtige Aspekte auf dem Entscheidungsweg vor der Weitergabe von Software zu klären.

Ergänzende Informationen, wie Tutorialsammlungen, Best Practice Beispiele und Antworten zu häufig gestellten Fragen werden zusätzlich über das RSE-Portal zur Verfügung gestellt. Diese können für die Überprüfung bzw. Einschätzung des eigenen Software-Projekts und als Vorbereitung für persönliche Beratungsgespräche verwendet werden. Des Weiteren können zusätzlich über die genannte Best Practice Beispiele am Forschungszentrum Jülich übliche Vorgehensweisen zu Software und gängige Open Source [Lizenzen](#) und deren rechtliche Einordnung als Entscheidungshilfe herangezogen werden.

4.2 Unterstützungs- und Weiterbildungsangebote

Schulungen zu Software-Engineering spielen eine wichtige Rolle, um wissenschaftlichen Nachwuchs zu guter Software-Entwicklung zu befähigen und die Standards für gute Software im Forschungszentrum umzusetzen. Das Forschungszentrum Jülich möchte daher einen einfachen Zugang zu internen und externen Trainingsangeboten schaffen. Des Weiteren soll das am Forschungszentrum bestehende Schulungs-Angebot zu Softwareentwicklungs-Methodiken und Best Practices zukünftig weiter ausgebaut werden.

Auf dem RSE-Portal sind interne und externe Angebote und Informationen aufgelistet, welche regelmäßig aktualisiert und laufend ergänzt werden. Dort sind unter anderem die Aktivitäten und Kurse der Helmholtz-Gemeinschaft wie der HIFIS Software Helpdesk, Schulungen von [HIFIS](#) und [HIDA](#), das [Helmholtz Open Science Forum „Forschungssoftware“](#) und die Angebote der RWTH Aachen z.B. im Rahmen von JARA-CSD zu finden. In Ergänzung und enger Vernetzung zum RSE-Portal wird auf der Jülicher Lernplattform JuLearn ein Arbeitsbereich für Softwareentwickler:innen mit entsprechenden Angeboten eingerichtet.

Das Forschungszentrum Jülich ermutigt alle Software-Entwickler:innen die genannten Angebote aktiv zu nutzen, sowie sich selbst dort einzubringen. Die Führungskräfte sind dazu aufgerufen, geeignete Mitarbeitende zu ermutigen, sich als Trainer:innen im Bereich der Softwareentwicklung zu engagieren (z.B. im Rahmen von carpentries Workshops oder im internen Trainingsangebot) und gewährt ihnen entsprechende Freiräume.

4.3 Wissenschaftliche & wirtschaftliche Verwertung und Lizenzwahl

Das Forschungszentrum Jülich unterstützt und befürwortet die Veröffentlichung von Software als Open Source Software, um so zu einer Stärkung von „Open Science“ beizutragen und dadurch zu einem effektiveren und offeneren Informationsaustausch innerhalb der Wissenschaft zu kommen und den Transfer der Ergebnisse in die Gesellschaft zu fördern.

In den Fällen, in denen eine wirtschaftliche Verwertung der Software möglich und sinnvoll ist und im Interesse der Software-Entwickler und des Instituts liegt, empfiehlt das Forschungszentrum Jülich eine solche kommerzielle Verwertung. Dabei können die wissenschaftliche Verwendung und wirtschaftliche Verwertungsaspekte durchaus gemeinsame Bausteine einer abgestimmten Verwertungsstrategie für eine Software sein und damit die Interessen des Instituts und des Forschungszentrums Jülich für die Erstellung, Weitergabe und Bereitstellung von Software bestmöglich wahren.

Zur Entscheidung für eine bestimmte Lizenzart ist frühzeitig in der Softwareentwicklung und in Abstimmung zwischen Software-Entwickler:innen, Führungskräften und Projektverantwortlichen

- die [Anwendungsklasse](#) festzulegen,
- die Qualität der Software zu sichern,
- die Nutzer-Zielgruppe zu bestimmen,
- die Art und Weise der Weitergabe (Lizenzart) zu überlegen und
- der rechtliche Rahmen zu prüfen.

Eine Checkliste und ein Entscheidungsbaum stehen über das RSE-Portal zur Verfügung und helfen bei diesen Entscheidungen.

Nur wenn das Forschungszentrum Jülich alle notwendigen *Nutzungs-* und *Verwertungsrechte* an der Software hat, kann diese unter Beachtung ggf. geltender (Open Source)-Lizenzbedingungen weitergegeben werden. Dies gilt insbesondere für *abgeleitete Werke*. Die Wahl der *Lizenz* obliegt dabei dem jeweiligen Institut und kann bei Bedarf und je nach Prozess an die Projektverantwortlichen, Führungskräfte und Software-Entwickler:innen delegiert werden. Bei Fragen und Unklarheiten zur Weitergabe von Software oder zur Lizenzwahl können sich Software-Entwickler:innen und Projektbeteiligte an die Ansprechpersonen auf dem RSE-Portal aus der Unternehmensentwicklung und der Rechtsabteilung wenden, um gemeinsam die jeweilige Situation zu analysieren und die best-mögliche Lösung zu finden.

Abhängig von den vertraglichen Rahmenbedingungen (Kooperationsverträge, Konsortialverträge, Förderbestimmungen etc.), der möglicherweise enthaltenen und verbundenen Fremdsoftware, dem gewünschten Transferweg, der zutreffenden *Anwendungsklasse* etc. können unterschiedliche *Lizenzen* für die Weitergabe der Software gewählt werden. Dabei besteht die Wahl zwischen verschiedenen *Open Source Lizenzen* [15] oder einer proprietären Lizenzierung.

4.4 Open Source-Lizenzen

Das Forschungszentrum Jülich empfiehlt bei der Veröffentlichung von Software als Open Source [13] eine von der *Open Source Initiative* anerkannte *Lizenz* zu verwenden. Anerkannte und populäre Open Source *Lizenzen* sind beispielsweise EUPL, AGPL, GPL, LGPL, MIT, Apache oder Mozilla Public License.

Auf dem RSE-Portal sind Lizenztexte und aktuelle Informationen zu den empfohlenen Open Source *Lizenzen* sowie Tools für Kompatibilitätscheck verlinkt. Als Entscheidungshilfe bei der Wahl einer geeigneten *Lizenz* können die Best Practices Beispiele auf dem RSE-Portal und der Entscheidungsbaum (Anlage B) verwendet werden. Falls eine Abweichung von den oben genannten *Lizenzen* oder eine Beratung zu der Auswahl der passenden *Lizenz* gewünscht ist, empfiehlt das Forschungszentrum Rücksprache mit UE-I oder R-R zu halten (siehe Kapitel 4.1).

4.5 Proprietäre Lizenzen

Wird eine wirtschaftliche Verwertung der Software oder eine stärker individuell ausgestaltete *Lizenz* angestrebt, weil z.B. die Weitergabe des Codes bzw. die Art der Verwendung der Software (z.B. kommerzielle Nutzung) eingeschränkt werden soll, empfiehlt das Forschungszentrum Jülich eine proprietäre *Lizenz* zu vergeben.

Möglich sind dabei auch Vertragsmodelle, mit denen die Software proprietär aber kostenlos für akademische Zwecke lizenziert wird, eine kommerzielle Nutzung aber kostenpflichtig ist [16]. Soll eine Software in Kombination mit Know-how, wie einem Bauplan, lizenziert werden, sollte immer eine proprietäre Lizenzierung gewählt werden.

Besteht der Bedarf darin, nur innerhalb einer geschlossenen Community Software gemeinsam frei zu nutzen und zu modifizieren, bieten sich eigene *Lizenzen* an, da eine bestehende Open Source *Lizenz* einer geschlossenen Community entgegensteht. Bei der gemeinsamen Software-Entwicklung in großen Communities, deren Parteien auch Background Software in die Zusammenarbeit einbringen, ist dringend auf Lizenzkompatibilitäten zu achten.

Auch Software, die mit einer Open Source *Lizenz* versehen ist, kann kommerziell verwertet werden, wenn das Forschungszentrum Jülich die *Urheberrechte* hält. Zu der Verwertung von Open Source Software können verschiedene Modelle verfolgt werden. Wird Software als Background-IP in Ko-

15 <https://opensource.org/licenses>

16 Beispiel: <https://www.13cflux.net/13cflux2/>

operationsprojekte eingebracht, ist diese vor der Weitergabe an Partner mit einer *Lizenz* zu versehen, da Software auch in Kooperationen nicht ohne Einräumung expliziter *Nutzungsrechte* (Open Source oder proprietär) weitergegeben werden darf.

Bei Unsicherheiten, aufkommenden Fragen oder als Entscheidungshilfe zu den unterschiedlichen Transferwegen können die oben genannten und über das RSE-Portal verfügbaren Beratungsangebote zur Unterstützung herangezogen werden. So kann bei Bedarf im jeweiligen Einzelfall gemeinsam geprüft werden, ob z.B. Rechte Dritter verletzt werden und damit eine wissenschaftliche oder wirtschaftliche Verwendung bzw. Nachnutzung eingeschränkt oder sogar verhindert wird.

Die Erstellung und Verhandlung jeglicher Lizenzverträge, die eine proprietäre Nutzung einräumen, werden am Forschungszentrum Jülich federführend von UE-I verantwortet. Ist eine proprietäre Lizenzierung gewünscht, wird gemeinsam mit dem Institut über mögliche Lizenzkonditionen, vertragliche Rahmenbedingungen, die IP-Situation etc. beraten und ein entsprechender Lizenzvertrag aufgesetzt. Dabei stimmt sich UE-I bezüglich der vertragsrechtlichen Rahmenbedingungen mit dem Fachbereich Recht (R-R) ab und verhandelt den Lizenzvertrag mit dem potentiellen Lizenznehmer.

5 Fazit

Die Implementierung dieser Software-Leitlinie am Forschungszentrum Jülich ermöglicht den Mitarbeitenden in einem praktischen und handlungssicheren Rahmen, Software mit hohen Qualitätsstandards zu entwickeln und weiterzugeben.

Durch die Einteilung in Anwendungsklassen, das Aufzeigen von Qualitätsstandards und entsprechend empfohlenen Maßnahmen und der Berücksichtigung von möglichen Transferwegen bzw. Lizenzarten und die damit einhergehenden rechtlichen Aspekte wird eine Professionalisierung in der Breite im Zentrum realisiert.

Ergänzt wird dies durch die vom Forschungszentrum Jülich bereitgestellten und zukünftig über das RSE-Portal kanalisierten Entscheidungshilfen und Beratungsangebote von UE, R und ZB.

Darüber hinaus möchte das Forschungszentrum Jülich bestehende interne Schulungsangebote zu Softwareentwicklungs-Methodiken und Best Practices zukünftig weiter ausbauen, um so den wissenschaftlichen Nachwuchs zu hohen Standards in der Softwareentwicklung und Dokumentation zu befähigen.

Mit der durch diese Leitlinie generierten Wirkung soll eine Nachhaltigkeit und eine gute wissenschaftliche Praxis bei der Entwicklung und Weitergabe von Software und somit ein Mehrwert in der Gesellschaft, Wirtschaft, Wissenschaft und Politik geschaffen werden.

6 Begriffserläuterungen

Abgeleitete Werke:

Auch beim Einsatz bzw. der Einbindung von Code, dessen Nutzungsrechte nicht (ausschließlich) beim Forschungszentrum Jülich, sondern bei Dritten, z.B. Kooperationspartnern, liegen (Fremdsoftware) müssen Rechte durch den Urheber bzw. dessen Arbeitgeber eingeräumt werden. Dabei sind auch zwingend die Lizenzbedingungen der eingesetzten (Open-Source) Software zu beachten, da eine Nichtbeachtung eine Vertragsverletzung darstellt und zu Schadensersatzforderungen führen kann. Dies gilt auch, wenn lediglich Programmbestandteile oder Teilsequenzen in eine eigene Software integriert werden. Dann bedarf es einer intensiven Prüfung, inwieweit die verschiedenen Lizenzen der zu integrierenden Drittsoftware inkompatible Regelungen enthalten. Das kann dazu führen, dass die eigene Software nicht unter der beabsichtigten Open-Source-Lizenz veröffentlicht werden kann. Zudem kann die Open-Source-Lizenz der eingesetzten Software erlöschen, wenn die Lizenzbedingungen nicht eingehalten werden.

Anwendungsklasse:

Anwendungsklassen¹⁷ dienen zur Kategorisierung von Software 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)¹⁸ entnommen werden. Darin befindet sich eine exemplarische Definition von aufeinander aufbauenden Anwendungsklassen. Die beschriebene „Anwendungsklasse 1“ legt beispielsweise minimale Empfehlungen für kleine, unkritische Software (z. B. Datenauswertungsskripte) fest und ist kompatibel zur der in diesem Dokument festgelegten Minimalpraxis in Bezug auf Entwicklung und Dokumentation.

Archivierung:

Eine Archivierung der am Forschungszentrum Jülich programmierten Software trägt zur Nachhaltigkeit der Softwareentwicklung bei. Dabei dient die Archivierung primär der Sicherung von Versionen zu Nachweis-, Provenienz- und Reproduzierbarkeits-Zwecken im Kontext wissenschaftlicher Ergebnisse.

Die Archivierung erfolgt inklusive begleitender Daten (z. B. Metadaten, Dokumentation, ggf. Laufzeitumgebung und Testdaten). Zusätzlich muss eine Einhaltung von Speicherfristen entsprechend der Regeln guter wissenschaftlicher Praxis in den Fachdisziplinen eingehalten werden.

Möglichkeiten zur Archivierung können je nach Anforderungen und Anwendungsklasse unter anderem sein: Repositorien ([Zenodo](#), [Jülich DATA](#), etc....), Software-Journale und das Software Heritage Archive. Nähere Informationen dazu finden sich in den Best Practices und auf dem RSE-Portal.

Nachhaltige Software setzt voraus, dass auch inaktive oder archivierte Software nachnutzbar bleibt. Die Schnelllebigkeit der Technologien im Software-Umfeld stellt im Hinblick auf die Archivierung von Software eine Herausforderung dar. Software-Entwickler:innen, Projektleiter:innen und Führungskräfte müssen bei der Planung, Umsetzung und Verwertung von selbstprogrammierter Software gemeinsam anhand der Anwendungsklasse entscheiden, wie im konkreten Einzelfall zu handeln ist. Die entsprechenden (langfristigen) Ressourcen dafür sind durch die Projektverantwortlichen von Beginn an einzuplanen.

Code-Repository:

Ein Code-Repository ist ein Archiv für den Code, an dem gearbeitet wird. Neben dem Code selbst können Sie in Ihrem Repository auch Dinge wie Dokumentation, Notizen, Webseiten und andere Dinge aufbewahren. Ein Code-Repository ist für jedes erfolgreiche Softwareentwicklungsprojekt erforderlich.

Contributor Agreement:

Ein Contributor Agreement oder Contributor License Agreement (CLA), ist ein Dokument, in dem die Bedingungen beschrieben sind, unter denen [geistiges Eigentum](#) zu einem Projekt oder Vorhaben beigetragen werden kann; meist handelt es sich um ein [Software](#)-Projekt unter einer [Open-Source-Lizenz](#). Das Forschungszentrum Jülich stellt eine geprüfte Vorlage über das RSE-Portal zur Verfügung, welche den externen Beitragenden vorgelegt werden kann

Exportkontrolle:

Sobald Person aus Staaten außerhalb der EU oder gleichgestellten Staaten (UK, USA, JP,...) am Projekt beteiligt sein sollen, kann eine außenwirtschaftsrechtliche Prüfung des Projekts erforderlich sein. Ob dies für das eigene Projekt notwendig ist, kann anhand einer Checkliste im RSE-Portal ermittelt werden.

Git:

Git ist ein Versionskontrollsystem, mit dem man verfolgen kann, wer wann welche Änderungen am Code vorgenommen hat. Git bietet Optionen zur einfachen Aktualisierung einer freigegebenen oder öffentlichen Version eines Codes auf github (<https://github.com/>).

Informationssicherheit:

Die Stabsstelle Datenschutz und Informationssicherheit stellt in der „Richtlinie Informationssicherheit“ (in Erstellung) ein Klassifikationssystem für das Schutzniveau von Daten bereit. Bei der Erstellung von Software wird daher empfohlen, sich bereits frühzeitig Gedanken über die Einstufung der eingeflossenen Daten und der Software an sich anhand der Richtlinie zu machen. Spätestens bei Vorliegen von substantiellen Code-Teilen muss eine Klassifikation vorliegen.

¹⁷https://rse.dlr.de/guidelines/00_dlr-se-guidelines_de.html#anwendungsklassen

¹⁸<https://doi.org/10.5281/zenodo.1344608>

Lizenzen:

OSS- Lizenzen: Nach allgemeiner Ansicht gibt die Open Source Definition (<http://www.opensource.org/docs/osd>) korrekt wieder, was in der Open Source Community unter dem Begriff "Open Source Software" verstanden wird. Entscheidendes Kriterium der Definition ist, dass die Lizenz einer Software die umfassende "freie" Nutzung des Programms gestattet und keine Beschränkung der Lizenznehmer oder Verwendungsgebiete vorsieht, also von jedermann zu jedem Zweck genutzt werden darf. Die Lizenzgebührenfreiheit und der offen zugängliche Sourcecode sind dabei wesentliche Voraussetzungen für die freie Nutzbarkeit. Entscheidend ist also stets der Lizenzvertrag, unter dem eine Software angeboten wird.

(<https://www.ifross.org>)

Proprietäre Lizenzen:

Der wesentliche Unterschied zwischen proprietären und Open Source Lizenzen ist, dass die proprietäre Lizenz völlig frei gestaltet werden kann und insbesondere die Möglichkeit eröffnet, die Weitergabe von Code vollständig einzuschränken. Zwar verpflichtet die Verwendung einer Open Source Lizenz nicht pauschal zur Weitergabe, allerdings kann einem Dritten gegenüber dieses Recht nicht eingeschränkt werden. Ferner greift die Open Source Lizenz nur dann, wenn der Dritte die Software wieder weitergibt. So wäre z.B. das Anbieten einer Dienstleistung mit der Software ohne Weitergabe dieser ohne weiteres möglich. Eine proprietäre Lizenz wird also dann interessant, wenn man die Weitergabe des Codes an sich oder eine bestimmte Art der Verwendung (kommerzielle Nutzung) einschränken möchte.

Nutzungs- und Verwertungsrecht:

Sobald Software-Entwickler:innen sich in einem Arbeits- und Dienstverhältnis mit dem Forschungszentrum Jülich befinden, liegen die Nutzungs- und Verwertungsrechte gemäß Urheberrecht automatisch beim Forschungszentrum. Spezielle Regelungen sind daher bei der geplanten Einbeziehung Dritter, die keinen Arbeitsvertrag o. Ä. mit dem Forschungszentrum haben, vor Projektstart zu treffen. Externe Beiträge können bei Bedarf über ein Contributor License Agreement eingebunden werden. Das Forschungszentrum Jülich stellt eine geprüfte Vorlage über die RSE-Portal zur Verfügung, welche den externen Beitragenden vorgelegt werden kann.

PID:

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

Software-Projekt:

Ein Softwareprojekt ist ein komplexes Unterfangen von zwei oder mehr Personen, das im Rahmen von Zeit, Budget und Personalressourcen einen neuen oder verbesserten Computercode hervorbringt.

Software-Publikation:

Analog zu Forschungsdaten und wissenschaftlichen Texten ist auch Software ein Ergebnis wissenschaftlicher Arbeit, sodass sie als wissenschaftliche Publikation einem breiten Publikum bekannt gemacht werden kann. Nichtsdestotrotz kann der Zugang zur Software aus der Publikation analog zu Forschungsdaten eingeschränkt sein. Wie Forschungsdaten und Textpublikationen beinhaltet eine Software-Publikation Metadaten, mit denen sie auffindbar und zitierbar ist.

Release

Bei einem Release handelt es sich um eine Version einer Software, die Nutzern zur Verfügung gestellt wird oder einen Beitrag zu einer wissenschaftlichen Publikation leistet. Einem Schema folgend,¹⁹ 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>)

Software:

¹⁹ siehe z. B. :<https://semver.org/>

Mit Software sind im Kontext dieser Leitlinie 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.

Software Lebenszyklus:

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

Standards/aktueller Stand der Technik:

Bei dem Begriff „Stand der Technik“ handelt es sich um einen rechtlich-technischen Begriff, welcher für das Produkt gewisse Mindestanforderungen impliziert. Konkret heißt es: Stand der Technik ist der Entwicklungsstand fortschrittlicher Verfahren, Einrichtungen und Betriebsweisen, der nach herrschender Auffassung führender Fachleute das Erreichen des gesetzlich vorgegebenen Zieles gesichert erscheinen lässt. Verfahren, Einrichtungen und Betriebsweisen oder vergleichbare Verfahren, Einrichtungen und Betriebsweisen müssen sich in der Praxis bewährt haben oder sollten – wenn dies noch nicht der Fall ist – möglichst im Betrieb mit Erfolg erprobt worden sein.

Deutlicher wird es, wenn man die Position des Begriffs im Vergleich zu „allgemein anerkannte Regeln der Technik“ (darunter) und Stand von Wissenschaft und Technik (darüber) sieht. (https://www.bmju.de/DE/The-men/RechtssetzungBuerokratieabbau/HDR/HDR_node.html; siehe 4.5.1.)

Grundsätzlich gibt es also zwei Möglichkeiten. Entweder man definiert eigene möglichst konkrete Kriterien, oder man entscheidet sich für einen der drei Begriffe und achtet auf die korrekte Verwendung innerhalb des Projekts.

Urheberrecht:

Software unterliegt 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 nach deutschem Recht zwar nicht übertragbar, zulässig sind jedoch die Einräumung von umfangreichen Nutzungsrechten sowie Vereinbarungen zu Verwertungsrechten. Legt man z.B. amerikanisches Recht zugrunde, dann ist auch eine Abtretung des gesamten Urheberrechts möglich.

Sobald die Software-Entwickler:innen sich in einem Arbeits- und Dienstverhältnis mit dem Forschungszentrum Jülich befinden, liegen die Nutzungs- und Verwertungsrechte automatisch beim Forschungszentrum (Einzelheiten sind im § 69 b UrhG-Gesetz geregelt). Spezielle Regelungen sind daher bei der geplanten Einbeziehung Dritter, die keinen Arbeitsvertrag o. Ä. mit dem Forschungszentrum haben, vor Projektstart beispielsweise bei Ausgründungen oder wirtschaftlicher Nutzung zu treffen.

Versionskontrollsystem:

Wichtig bei der Entwicklung von qualitativ hochwertiger Software ist die Verwendung von Versionskontrollsystemen, um so u.a. eine Übersicht über sämtliche Änderungen am Code und den daran beteiligten Software-Entwickler:innen zu erhalten. Diese sollen mit kollaborativen Funktionen für Projekte und Communities verknüpft sein. 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.

Das Forschungszentrum Jülich bietet aktuell eine solche kollaborative Plattform für seine Mitarbeiter:innen (z.B. aktuell JuGIT, Stand 2021). Darüber hinaus ist wünschenswert, dass diese Plattform für die nationale und internationale Zusammenarbeit genutzt werden kann, indem etablierte Identitätsföderationen wie derzeit (2021) eduGain, DFN AAI und/oder die Helmholtz AAI angebunden und über diese auch für Nichtbetriebsangehörige in diesem Sinne nutzbar ist.

Ebenso können weitere solcher Plattformen für geschlossene Gruppen oder aus anderen wichtigen Gründen angeboten werden. Die von der Helmholtz Gemeinschaft im Rahmen von HIFIS bereitstehenden oder etablierte globale Plattformen wie GitHub.com oder GitLab.com stehen ebenfalls zur Auswahl. Diese können nach gemeinsamen Ermessen der Entwickler:innen, Führungskräfte und Projektverantwortlichen genutzt werden, um beispielsweise die Reichweite und Sichtbarkeit zu erhöhen.

CHECKLISTE FÜR DIE WEITERGABE VON SOFTWARE

Datum:

Name der Software:

Institut:

Kontakt:

Urheber und Rechte Dritter <i>„Alle Urheberrechte müssen bei Forschungszentrum Jülich liegen“ und „Alle Urheber:innen müssen bekannt sein“</i>	
☐	Alle Urheber:innen der Software sind bekannt und benannt.
☐	Alle Urheber:innen haben als Mitarbeiter:innen des Forschungszentrums Jülich programmiert und die Nutzungsrechte liegen bei Forschungszentrum. Falls Nein:
☐	- Dritt-Institutionen sind bekannt.
☐	- Forschungszentrum liegen Nutzungsrechte dieser Institutionen in schriftlicher Form vor.

Vertragliche Bindungen	
☐	Förder- oder Zuwendungsvorgaben, Kooperationsverträge, Grant Agreements und Arbeitsverträge wurden auf mögliche Vorgaben oder Einschränkungen zur Publikation und Weitergabe von Software geprüft.
☐	Es ist bekannt, wenn/ob die Software als Background in Projekten eingebracht ist.
☐	Im Falle von medizinischer Software sind die Limitationen bekannt und werden beachtet.
☐	Die Regelungen zur Exportkontrolle wurden geprüft und werden, wenn notwendig, eingehalten.

Kompatibilitäten	
☐	Die Software wurde ohne Einbindung von vorbestehenden Softwareteilen oder Bibliotheken geschrieben. Falls Nein:
☐	- Die Lizenzbedingungen der vorbestehenden/veränderten Software bzw. der verknüpften Bibliotheken sind bekannt und Kompatibilitäten werden beachtet.
☐	- Falls für die Lizenzierung/Weitergabe der eigenen Software eine kostenpflichtige Entwicklerlizenz für die vorbestehende Software/Bibliothek benötigt wird, liegt diese vor.

	Transferweg, Verwertung
☐	Es wurde geprüft, ob die Software ein kommerzielles Potential hat.
☐	Die Zielgruppe ist bekannt.
☐	Das Interesse und die Zielsetzung der Entwickler:innen ist formuliert.
☐	Die zukünftige Nutzung/Behandlung und Zugänglichkeit der Software im Institut ist geklärt.
☐	Die Zustimmung des Institutsverantwortlichen liegt vor und der entsprechende Freigabeprozess wurde eingehalten

	Lizenzwahl
☐	Es ist geklärt, welches Maß an Zugriff die Urheber auf die Software zulassen wollen (source code oder object code).
☐	Es ist entschieden, ob die Software proprietär oder als OSS weitergegeben werden soll.
☐	Für Open Source Software Lizenzen: Die ausgewählte Lizenz entspricht den „approved licenses“ (siehe Seite der Open Source Initiative https://opensource.org/) Falls Nein:
☐	- Es wurde Rücksprache mit den Ansprechpersonen von UE/R gehalten.
☐	Für Open Source Software Lizenzen: Der Text der ausgewählten Lizenz wurde gelesen und verstanden und passt zur Zielgruppe und zur Zielsetzung der Weitergabe. Falls Nein:
☐	- Es wurde Rücksprache mit den Ansprechpersonen von UE/R gehalten.

Erläuterungen zur Checkliste

Grundsätzlich stehen zwei Aspekte in rechtlicher Hinsicht im Vordergrund. Bestehende Rechte des Forschungszentrums am Code vor Lizenzierung und die Rechte, die ein Dritter durch die Lizenzierung erhalten soll.

Vor der Lizenzierung

Bevor eine am Forschungszentrum entwickelte Software weitergegeben und dabei mit einer Lizenz versehen werden kann, muss sichergestellt werden, dass das Forschungszentrum die Urheberrechte daran hält.

§ 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.

Sobald die Softwareentwickler:innen sich in einem Arbeits- und Dienstverhältnis mit dem Forschungszentrum Jülich befinden, liegen die Nutzungs- und Verwertungsrechte automatisch beim Forschungszentrum. Spezielle Regelungen sind daher bei der geplanten Einbeziehung

Dritter, die keinen Arbeitsvertrag o. Ä. mit dem Forschungszentrum haben, vor Projektstart zu treffen.

Externe Beiträge können bei Bedarf über ein Contributor License Agreement eingebunden werden. Das Forschungszentrum Jülich stellt eine geprüfte Vorlage über die [Entwickler:innen-Portal](#) zur Verfügung, welche den externen Beitragenden vorgelegt werden kann.

Auch beim Einsatz bzw. der Einbindung von Code, dessen Nutzungsrechte nicht (ausschließlich) beim Forschungszentrum Jülich, sondern bei Dritten, z.B. Kooperationspartnern, liegen (Fremdsoftware) müssen Rechte durch den Urheber bzw. dessen Arbeitgeber eingeräumt werden. Dabei sind auch zwingend die Lizenzbedingungen der eingesetzten (Open-Source) Software zu beachten, da eine Nichtbeachtung eine Vertragsverletzung darstellt und zu Schadensersatzforderungen führen kann. Dies gilt auch, wenn lediglich Programmbestandteile oder Teilsequenzen in eine eigene Software integriert werden. Dann bedarf es einer intensiven Prüfung, inwieweit die verschiedenen Lizenzen der zu integrierenden Drittsoftware inkompatible Regelungen enthalten. Das kann dazu führen, dass die eigene Software nicht unter der beabsichtigten Open-Source-Lizenz veröffentlicht werden kann. Zudem kann die Open-Source-Lizenz der eingesetzten Software erlöschen, wenn die Lizenzbedingungen nicht eingehalten werden.

Weitergabe an Dritte

Bei der Wahl einer Lizenz kommt es in erster Linie darauf an, welche Rechte und Pflichten den Lizenznehmer treffen sollen. Das [Spektrum](#) reicht von z.B. der MIT (alle Rechte, kaum Pflichten) bis zu GPL v3 (beschränkte Rechte, Pflichten bei Weitergabe).

Ist es im Projekt wichtig, dass möglichst viele Partner unkompliziert und schnell Zugang zum Code bekommen und mitarbeiten können, dann spricht vieles für eine Lizenz mit allen Rechten aber ohne Pflichten.

Soll aber sichergestellt werden, dass der Code nicht von Dritten übernommen und unter Einschränkungen weiterverbreitet wird, dann müssen den Lizenznehmern auch Pflichten auferlegt werden (z.B. die Weitergabe des Quellcodes).

Einen wirksamen Schutz vor einer z.B. kommerziellen Verwendung bietet aber nur eine proprietäre Lizenz (mehr dazu im Kapitel Lizenzwahl der Software-Leitlinie).

Weitergabe von Software

Entscheidungsbaum



Beratungsangebote

Entscheidungshilfen

Weitere Tools im [RSE Portal](#)

Beratung durch
UE-I & R-R

Checkliste

Beratung durch
UE-I & R-R

Checkliste

Beratung durch
UE-I / R-R / ZB

Tool: FOSSA

Beratung durch
UE-I

Checkliste

Beratung durch
UE-I

Software-Leitlinie

Beratung durch
UE-I

*Software darf nur mit Lizenz (Proprietäre oder Open Source) an Dritte weitergegeben werden!