

# Software ist kein Beiprodukt!

## Nachhaltige Forschungssoftware durch Software-Management-Pläne

Yves Vincent Grossmann und Michael Franke

Software ist kein Nebenprodukt, sondern integraler Bestandteil der Forschung. Das stellt nicht nur die Volkswagen-Stiftung als Forschungsförderer in ihrer Open Science Policy 2022 fest.<sup>1</sup> Auch aus verschiedenen Wissenschaftsdisziplinen ist Software als eigenständiges Forschungsergebnis nicht mehr wegzudenken. Häufig benötigt man den geschriebenen Code und die Forschungsdaten, um die Ergebnisse einer Textpublikation nachzuvollziehen. Forschungssoftware entwickelt sich damit zu einem eigenen Publikationsgenre und ist kein Nischenprodukt mehr. Dies geht einher mit gesteigerten Anforderungen an beispielsweise die Struktur des Codes, die Auswahl der externen Komponenten oder den Ort der Veröffentlichung. Ähnlich wie bei Forschungsdaten kann ein expliziter Umgang mit Software im wissenschaftlichen Kontext frühzeitig helfen, Probleme zu vermeiden, Qualitätsstandards zu erreichen und Nachhaltigkeit zu gewährleisten. Dieser Beitrag diskutiert exemplarisch aktuelle Entwicklungen rund um Forschungssoftware. Normative Rahmenbedingungen wie die FAIR-Prinzipien, institutionelle Richtlinien und Peer Review sind hier relevante Aspekte, die näher betrachtet werden. Dies leitet über zum Publikationsverhalten bei Forschungssoftware. Nach diesen Betrachtungen wird präsentiert, wie ein Software-Management-Plan Wissenschaftlerinnen und Wissenschaftler, aber auch Institutionen bei diesen Themen unterstützen kann.



Abbildung 1: The Turing Way Community, & Scriberia. (2022). Illustrations from The Turing Way: Shared under CC-BY 4.0 for reuse. Zenodo. <https://doi.org/10.5281/zenodo.6821117>.

### Abstract

Forschungssoftware ist ein integraler Bestandteil der Wissenschaft und kein Nebenprodukt mehr. Je nach Disziplin wird dem Code immer mehr Aufmerksamkeit geschenkt. Die fachinterne Anerkennung von Software wird für Wissenschaftlerinnen und Wissenschaftler immer wichtiger. Gleichzeitig entwickelt sich Forschungssoftware zu einem eigenständigen Publikationsgenre neben Text- und Datenpublikationen. Allerdings ist der professionelle Umgang mit Softwareprojekten in der Wissenschaft noch nicht überall gelebte Praxis. Dabei bietet ein expliziter Umgang mit Forschungssoftware erhöhten Nutzen im wissenschaftlichen Alltag und darüber hinaus. Ein Software-Management-Plan als Organisations-Werkzeug besitzt hierfür großes Potenzial. Der Beitrag diskutiert diese aktuellen Entwicklungen und zeigt zugleich mögliche Lösungswege auf.

Research software is an integral part of science and no longer a by-product. Depending on the discipline, more and more attention is paid to code. Interdisciplinary recognition of software is becoming more important for scientists. At the same time, research software is becoming a publication genre in its own right, alongside text and data publications. However, the professional management of software projects in science is not yet common practice everywhere. Explicit use of research software offers increased benefits in everyday scientific life and beyond. A software management plan as a management tool has great potential for this. The article discusses these current developments and shows possible future solutions.

### Forschungssoftware in der Diskussion

Am Anfang waren die FAIR Data Principles<sup>2</sup>. Seit ihrer Veröffentlichung im Jahr 2016 haben sie eine weitreichende normative Kraft entwickelt. Diese geht weit über den Bereich der Forschungsdaten hinaus, vielleicht gerade weil diese Prinzipien aus der Wissenschaft für die Wissenschaft entwickelt wurden. In der Folge wurden 2021 die FAIR4RS-Prinzipien veröffentlicht.<sup>3</sup>

Ziel war es, mit diesen „FAIR for Research Software Principles“ die FAIR-Prinzipien aus der Datendomäne in Richtung

1 Volkswagen-Stiftung, Open Science Policy und Umsetzung, 2022, S. 5, [https://www.volkswagenstiftung.de/sites/default/files/documents/Open\\_Science\\_Policy\\_und\\_Umsetzung\\_VolkswagenStiftung.pdf](https://www.volkswagenstiftung.de/sites/default/files/documents/Open_Science_Policy_und_Umsetzung_VolkswagenStiftung.pdf).

2 Mark D. Wilkinson u. a., The FAIR Guiding Principles for Scientific Data Management and Stewardship, in: Scientific Data 3. 2016, Heft 1, S. 160018, <https://doi.org/10.1038/sdata.2016.18>.

3 Neil P Chue Hong u. a., FAIR Principles for Research Software (FAIR4RS Principles), 2021, S. 32.



Abbildung 2: The Turing Way Community, & Scriberia. (2022). Illustrations from The Turing Way: Shared under CC-BY 4.0 for reuse. Zenodo. <https://doi.org/10.5281/zenodo.6821117>.

Software zu transformieren.<sup>4</sup> Eine besondere Herausforderung stellte hierbei die Definition von Forschungssoftware dar. Man einigte sich auf ein vergleichsweise weites Verständnis, sodass nach FAIR4RS unter „research software“ jeglicher Code, Algorithmen, Skripte und ähnliches verstanden werden kann. Die Einschränkung besteht vor allem darin, dass diese mit einer klaren wissenschaftlichen Intention geschrieben sein müssen.<sup>5</sup> Nach diesem Verständnis sind beispielsweise handelsübliche Programme für die Tabellenkalkulation keine Forschungssoftware, sondern ausschließlich Software, welche in der Forschung verwendet wird. Warum sollten Wissenschaftler und Wissenschaftlerinnen sich überhaupt mit solchen Fragen bezüglich ihrer Software beschäftigen?

### Nachhaltigkeit bei Forschungssoftware

Ein wesentlicher Vorteil des bewussten Umgangs mit Forschungssoftware ist die erhöhte Nachhaltigkeit über ihren gesamten Lebenszyklus. So „zeitgeistig“ das klingt, so relevant ist der Aspekt der Nachhaltigkeit für Software. Konkret bedeutet dies beispielsweise, dass Code so geschrieben und dokumentiert wird, dass er auch von anderen installiert und genutzt werden kann. Dies zielt insbesondere auf den in der Wissenschaft zentralen Aspekt der Reproduzierbarkeit durch Dritte ab. Gleichzeitig ist das Prinzip der guten Dokumentation auch für ein zukünftiges „Ich“ relevant, da die eigene Software nach einigen Jahren Pause nicht unbedingt ohne Hindernisse wieder verwendet werden kann. Auch hier helfen z.B. eine klare Code-Struktur und ein ausformuliertes Handbuch, um

Forschungssoftware über einen längeren Zeitraum reproduzierbar zu halten.<sup>6</sup>

Nachhaltigkeit von Software kann zudem über diese Perspektive hinausgedacht werden. Gerade die Ressourcenschonung ist bei Software im wissenschaftlichen Kontext von großer Relevanz. Der Einsatz rechenintensiver Software kann im Bereich des High Performance Computing (HPC) schnell viele Ressourcen – insbesondere Zeit und Energie – in Anspruch nehmen. Die Optimierung von Forschungssoftware in solchen Größenordnungen erfordert einen vergleichsweise hohen Aufwand. Einsparungen bei Zeit und Energie wirkt sich positiv und sichtbar in verfügbaren finanziellen Ressourcen aus. Eine solche ökonomische Ressourcenschonung geht Hand in Hand mit einer ökologischen. „Grüne Informationstechnik“ (Green-IT) hilft, den Energie- und Rohstoffbedarf zu reduzieren. Wenn bei Forschungssoftware bereits in der Konzeptionsphase die CO<sub>2</sub>-Äquivalente berechnet werden, kann der ökologische Fußabdruck reduziert werden.

Diese verschiedenen Aspekte der Nachhaltigkeit zeigen, dass es eine Vielzahl strategischer Überlegungen gibt, die einen bewussten Umgang und ein explizites Projektmanagement mit Forschungssoftware ratsam erscheinen lassen. Dies reicht bis hin zu der grundsätzlichen Frage während der Planungsphase, ob es nicht bereits eine Anwendung mit ähnlichen Ergebnissen oder Mechanismen gibt. Ein Softwareprojekt kann daher besonders ressourcenschonend sein, wenn die Anwendung nicht eigenständig entwickelt werden muss. Vielmehr kann es auch interessant sein, bereits existierende Forschungssoftware aus der eigenen Fachkultur zu nutzen und an die eigenen Fragestellungen anzupassen. Gerade im Umfeld von freier und offener Software bzw. „Free and Open Source Software“ (FOSS) ist dies gelebte Praxis. Je nach Qualität kann eine so entwickelte Forschungssoftware innerhalb der eigenen Fachdisziplin große Aufmerksamkeit und Anerkennung finden.

### Forschungssoftware anerkennen und Anerkennung einfordern

Software ist kein Nebenprodukt. Gerade die fachinterne Anerkennung der erbrachten Leistung bei der Entwicklung und Pflege von Forschungssoftware gewinnt für Wissenschaftlerinnen und Wissenschaftler zunehmend an Bedeutung. Dies sehen auch die großen Forschungsförderer und honorieren den wissenschaftlichen Beitrag, den Forschungssoftware erbringen kann. Die eingangs erwähnte Volkswagen-Stiftung ist hier nur ein besonders promi-

4 Anna-Lena Lamprecht u. a., Towards FAIR Principles for Research Software, in: Data Science 3. 2020, Heft 1, S. 37–59, hier S. 42, <https://doi.org/10.3233/DS-190026>.

5 Defining Research Software: a controversial discussion, v. 1, 2021, S. 16, <https://doi.org/10.5281/ZENODO.5504016>.

6 Für eine Analyse der aktuellen Entwicklung in Deutschland siehe besonders Hartwig Anzt u. a., An Environment for Sustainable Research Software in Germany and Beyond: Current State, Open Challenges, and Call for Action, in: F1000Research 9. 2020, S. 295, <https://doi.org/10.12688/f1000research.23224.1>.

ntes Beispiel. Auch im Kodex der DFG wird klar betont, dass „[s]elbst programmierte Software [...] unter Angabe des Quellcodes öffentlich zugänglich gemacht“ wird.<sup>7</sup>

Im Sinne guter wissenschaftlicher Praxis sollte Forschungssoftware für andere verfügbar sein. Dies muss nicht zwangsläufig in Form einer offenen Publikation erfolgen, sondern kann beispielsweise auch über einen restriktiven Zugang oder über eine kommerzielle Verwertung geschehen. Die FAIR4RS-Prinzipien sind in der Tat nicht deckungsgleich mit der Notwendigkeit eines offenen Zugangs. Letztlich ist aber die eingesetzte Forschungssoftware für die Reproduzierbarkeit von Ergebnissen oft notwendig. Im zweiten Plan für Open Science in Frankreich von 2021 spiegelt sich dies in der expliziten Nennung von Forschungssoftware als dritte eigenständige Säule als Publikationsgattung – neben Text- und Datenpublikationen – wider. Damit verbunden ist die ausdrückliche Aufforderung, die entwickelte Forschungssoftware nach Möglichkeit öffentlich zugänglich zu machen.<sup>8</sup>

Auch in Deutschland zeichnet sich eine solche Entwicklung ab. Der Aspekt der Anerkennung von Forschungssoftware macht schließlich auch vor wissenschaftlichen Einrichtungen nicht halt. Innerhalb der Nationalen Forschungsdateninfrastruktur (NFDI) arbeitet bereits eine eigene Arbeitsgruppe zu diesem Thema.<sup>9</sup> Im bundesdeutschen Kontext nimmt die Helmholtz-Gemeinschaft hier eine Vorreiterrolle ein. Forschungssoftware wird in den verschiedenen Helmholtz-Zentren schon seit einiger Zeit verstärkt beachtet. So existieren diesbezüglich bereits seit 2019 eine Muster-Richtlinie<sup>10</sup> und seit 2021 eine eigene Checkliste für die Implementierung von Software-Leitlinien bei Helmholtz-Instituten.<sup>11</sup> Dies zeigt, dass Forschungssoftware einen hohen Stellenwert für den wissenschaftlichen Erkenntnisgewinn genießt. Forschungssoftware wird daher auch in deren Open-Science-Monitoring berücksichtigt.<sup>12</sup> Neben der internen Anerkennung ist es ein paralleles Ziel, die Veröffentlichung von Software so einfach wie möglich zu gestalten. Dies setzt die Helmholtz-Gemeinschaft mit ihrem HERMES-Konzept um.<sup>13</sup>

Ähnliche Wege beschreiten auch die TU Delft mit ihrer „Research Software Policy“<sup>14</sup> und das Max-Planck-Institut für Meteorologie mit einer eigenen „Software License Policy“<sup>15</sup>. Beide schaffen einen normativen Rahmen, in dem Wissenschaftlerinnen und Wissenschaftler Software explizit ohne Genehmigung durch Vorgesetzte veröffentlichen können. Es besteht dadurch nicht mehr die Notwendigkeit jede Code-Publikation im Vorhinein abzuklären. Neben dieser Rechtssicherheit für die Forschenden ist ein weiterer Aspekt mindestens ebenso wichtig: Diese Institu-



Abbildung 3: The Turing Way Community, & Scriberia. (2022). Illustrations from The Turing Way: Shared under CC-BY 4.0 for reuse. Zenodo. <https://doi.org/10.5281/zenodo.6821117>.

tionen positionieren sich klar dahingehend, dass sie in der Veröffentlichung von Forschungssoftware einen Mehrwert sehen und ihre Mitglieder dazu ermutigen, ihre Arbeiten mit der wissenschaftlichen Gemeinschaft zu teilen.

### Publikationsverhalten

Normative Vorgaben können vieles regeln, aber Softwareprojekte scheitern auch im Kleinen an der konkreten Umsetzung. So ist gerade die Übergabe der Forschungssoftware z.B. von einer Promotionsgruppe an eine Neue eine kritische Transformationsphase. Werden solche Pro-

7 Deutsche Forschungsgemeinschaft, Kodex »Leitlinien zur Sicherung guter wissenschaftlicher Praxis«, 2019, Abschnitt 13, <https://wissenschaftliche-integritaet.de/>.

8 Ministre de l'enseignement supérieur et de la recherche, Second French Plan for Open Science – Generalising Open Science in France 2021–2024, 2021, S. 17, [https://www.ouvrirlascience.fr/wp-content/uploads/2021/10/Second\\_French\\_Plan-for-Open-Science\\_web.pdf](https://www.ouvrirlascience.fr/wp-content/uploads/2021/10/Second_French_Plan-for-Open-Science_web.pdf). Siehe hierzu auch den Bericht Daniel Le Berre u. a., Forges de l'enseignement supérieur et de la recherche – Définition, usages, limitations ren-contrées et analyse des besoins, 2023, <https://doi.org/10.52949/34>. Siehe hierzu auch Michelle Barker u. a., Amsterdam Declaration on Funding Research Software Sustainability, 2023, <https://doi.org/10.5281/zenodo.7740084>.

9 Martin Hammitzsch u. a., Concept for Setting up a Working Group Research Software Engineering in the NFDI Section Common Infrastructures, 2022, <https://doi.org/10.5281/zenodo.6483448>.

10 Felix Bach u. a., Muster-Richtlinie Nachhaltige Forschungssoftware an den Helmholtz-Zentren, 2019, S. 12, <https://doi.org/10.2312/os.helmholtz.007>.

11 Reinhard Messerschmidt u. a., Checkliste zur Unterstützung der Helmholtz-Zentren bei der Implementierung von Richtlinien für nachhaltige Forschungssoftware, 2021, <https://doi.org/10.48440/os.helmholtz.031>.

12 Helmholtz Open Science Office, Helmholtz Open Science Policy, 2022, <https://os.helmholtz.de/open-science-in-helmholtz/open-science-policy/>.

13 Stephan Druskat u. a., Software publications with rich metadata: state of the art, automated workflows and HERMES concept, 2022, <https://doi.org/10.48550/arXiv.2201.09015>. HERMES steht hierbei für HELmholtz Rich Metadata Software Publication.

14 Anton Akhmerov u. a., TU Delft Research Software Policy, 2021, <https://doi.org/10.5281/zenodo.4629662>.

15 Max-Planck-Institut für Meteorologie, Software Licensing and Copyright Policy for Research Software CODE @ MPI-M, 2022, [https://mpimet.mpg.de/fileadmin/02\\_Forschung/08\\_Gute\\_wissenschaftliche\\_Praxis/MPI-M-SW-Policy.pdf](https://mpimet.mpg.de/fileadmin/02_Forschung/08_Gute_wissenschaftliche_Praxis/MPI-M-SW-Policy.pdf).

jektübergaben von den beteiligten Akteuren nicht unterstützt, können langfristig fatale Informationslücken entstehen, die die zukünftige Nutzbarkeit einschränken oder gar verhindern.<sup>16</sup> Oder es tritt der umgekehrte Fall ein, dass niemand mehr nachvollziehen kann, warum die Software so gute Ergebnisse liefert.

In welchen Fällen ist es sinnvoll, die Forschungssoftware zu veröffentlichen? Diese einfache Frage stellt im konkreten Forschungsalltag mitunter eine große Herausforderung dar, da sie mit vielen Bedingungen und Abhängigkeiten verbunden sein kann. Das Deutsche Zentrum für Luft- und Raumfahrt empfiehlt z.B. vier Größen von Software.<sup>17</sup> Je nach Umfang, Intensität und Zielsetzung des Projektes kann die Forschungssoftware kategorisiert und daraus Handlungsempfehlungen für das Management abgeleitet werden. Dies bezieht sich keineswegs nur auf

und GitHub ist auch im wissenschaftlichen Kontext weit verbreitet.<sup>19</sup> Ebenso entwickeln sich zunehmend eigene Publikationsjournale für Forschungssoftware.<sup>20</sup> Letztendlich lässt sich beobachten, dass – je nach Fachdisziplin – sich neue Publikationsverhalten bei Forschungssoftware zu entwickeln beginnen.

Im Vergleich zu Software im Allgemeinen weist Forschungssoftware jedoch einige Besonderheiten auf, die es ebenfalls zu beachten gilt. Von besonderer Bedeutung sind hier die beschreibenden Metadaten, die die Forschungssoftware z.B. mit den Autoren und deren Affiliationen verbindet. Häufig bestehen auch z.B. Querverbindungen zu anderen Forschungsprojekten und Publikationen, da die Software in den seltensten Fällen für sich alleine steht. Um diese Verbindungen sichtbar zu machen, werden Metadaten über die Forschungssoftware benötigt. Insbesondere die Helmholtz-Gemeinschaft versucht, mit dem bereits erwähnten HERMES-Konzept die Anreicherung von Forschungssoftware mit qualitativ hochwertigen Metadaten zu erreichen.<sup>21</sup> Dies wird vor allem durch benutzerfreundliche Mechanismen erreicht, die für die Wissenschaftlerinnen und Wissenschaftler in der Regel leicht zu handhaben sind. Ein prominentes Beispiel ist das maschinen- und menschenlesbare Format .cff, das Metadaten für die Zitation von Forschungssoftware bereitstellt und auf Github.com schon über 16.000 Mal verwendet wurde.<sup>22</sup>



Abbildung 4: The Turing Way Community, & Scriberia. (2022). Illustrations from The Turing Way: Shared under CC-BY 4.0 for reuse. Zenodo. <https://doi.org/10.5281/zenodo.6821117>.

die Frage der Publikation, sondern auf den gesamten Lebenszyklus.

Gerade bei der öffentlichen Verfügbarkeit von Software werden verschiedene Fragenkomplexe deutlich. Forschungssoftware wird häufig im Sinne von Open Source veröffentlicht.<sup>18</sup> Hierunter wird meist verstanden, dass der Code veröffentlicht wird, für Außenstehende uneingeschränkt zugänglich ist und unter bestimmten Bedingungen genutzt und verändert werden darf. Die Nutzung spezieller Repositorien für Open-Source-Software wie GitLab

### Eigene Beobachtungen und Erkenntnisse

Für den Umgang mit Forschungssoftware existieren bereits Prozesse, fachspezifische Verhaltensweisen und Anwendungen. Trotz allem lässt sich beobachten, dass zwischen Theorie und Empirie eine deutliche Lücke klafft. Wie kann dies erklärt werden?

Eine explizite Kultur im Umgang mit Forschungssoftware ist trotz der obigen Ausführungen in der Regel noch selten oder steht erst am Anfang eines Prozesses. Dies geht einher mit einer geringeren Wertschätzung von Forschungssoftware und Softwarepublikationen im Vergleich zu Text- und auch Datenpublikationen. Durch den enormen Veröffentlichungsdruck im System „Wissenschaft“ steht die Ressource „Zeit“, die letztlich in Forschungssoftware investiert werden müsste, nur in geringem Maße zur Verfügung. Hinzu kommt, dass die wenigsten Wissenschaftlerinnen und Wis-

16 Jörg Fehr u. a., Sustainable Research Software Hand-Over, in: Journal of Open Research Software 9. 2021, Heft 1, S. 5, <https://doi.org/10.5334/jors.307>.

17 Tobias Schlauch u. a., DLR Software Engineering Guidelines, 2018, S. 7-8, <https://doi.org/10.5281/ZENODO.1344612>.

18 Siehe hierzu beispielsweise die statistischen Verteilungen von Forschungssoftware aufgeschlüsselt nach Forschungsgebieten in W. Hasselbring u. a., Open Source Research Software, in: Computer 53. 2020, Heft 8, S. 84-88, hier S. 86-87, <https://doi.org/10.1109/MC.2020.2998235>.

19 So wurde beispielsweise in annähernd 20% aller arXiv-Publikationen auf ein github.com-Repositorium verwiesen, siehe Emily Escamilla u. a., The Rise of GitHub in Scholarly Publications, 2022, v. 1, hier S. 1, <https://doi.org/10.48550/ARXIV.2208.04895>.

20 Hier sind beispielsweise erwähnenswert das Journal of Open Source Software (<https://joss.theoj.org>) oder etwa das Journal of Open Research Software (<https://openresearchsoftware.metajnl.com>).

21 Druskat wie Fußnote 13.

22 Stephan Druskat, Citation File Format (CFF), 2021, <https://citation-file-format.github.io/>. Siehe ebenso Stephan Druskat, The Citation File Format: Providing Citation Metadata for Research Software, 2023, S. 4, <https://doi.org/10.5281/zenodo.7655140>.

senschaftler ausgebildete Softwareentwicklerinnen und -entwickler sind. Vielmehr sind sie häufig Autodidakten auf dem Gebiet der Softwareentwicklung, die sich im Laufe ihrer wissenschaftlichen Karriere immer weiter in die Materie einarbeiten. Dadurch wird der Funktionalität die größte Aufmerksamkeit geschenkt. Dies geht letztlich zu Lasten der Software-Dokumentation und erst recht zu Lasten von Fragen der Langzeiterhaltung. Hinzu kommt der Umstand, dass häufig Forschungssoftware auch von Generation zu Generation an Promovierenden quasi „vererbt“ wird. Das Ergebnis gleicht einer Art „Stiller Post“, bei der am Ende kaum noch Verständnis für Funktionsweise, Probleme und Architektur der Software vorhanden ist. Vielmehr ist es eine angenehme Überraschung, wenn die Forschungssoftware wie erhofft funktioniert und offenbar brauchbare Ergebnisse liefert.

Dieses skizzierte Bild des Einsatzes von Forschungssoftware ist vielleicht zu pessimistisch. Es gibt bereits viele positive Ausnahmen und erfolgreiche Beispiele. Dennoch gibt es noch viel Potenzial zu heben. Ein erster wesentlicher Schritt dazu ist vor allem die Transformation von implizitem in explizites Wissen: Wenn Forschungssoftware

als eigenständige Einheit wahrgenommen wird, dann können dedizierte Prozesse wie Versionierung, Pflege, Publikation, Archivierung etc. gezielt geplant werden. Gerade mit dem Anspruch, in Deutschland und Europa exzellente Wissenschaft zu betreiben, sollte ein solcher expliziter Umgang mit Forschungssoftware geboten sein.

### Software-Management-Pläne

Es ist viel einfacher, den (mangelhaften) Einsatz von Forschungssoftware zu kritisieren, als konkrete Dienstleistungen zu entwickeln, die helfen, diese Situation zu verbessern. Ein gangbarer Weg scheint die Adaption von Datenmanagementplänen (DMPs) für Forschungssoftware zu sein. Diese DMPs sind in der Wissenschaft, z.B. bei Drittmittelanträgen, bereits etabliert. Es liegt daher nahe, dieses Konzept auch für das Management von Forschungssoftware zu nutzen.

Erstmals wurde ein Artikel konkret über Software-Management-Pläne (SMPs) durch das schottische Software Sustainability Institute in 2014 veröffentlicht.<sup>23</sup> Ungefähr seit dem Jahr 2021 lässt sich ein gesteigertes Interesse an SMPs beobachten.<sup>24</sup> Eine der ersten ausformulierten SMPs

23 Chue Hong, Neil, Writing and Using a Software Management Plan, 2014, <https://www.software.ac.uk/resources/guides/software-management-plans>. Siehe ebenso The Software Sustainability Institute, Software Management Plans, 2016, <https://www.software.ac.uk/software-management-plans>.

24 Es dürfte kaum ein Zufall sein, dass dies zeitgleich mit der Veröffentlichung der FAIR4RS-Prinzipien entstand.

## Perfekt kombiniert!

### Entdecken Sie die Kombinationsmöglichkeiten bei EasyCheck!

Intelligente Regale mit Bezahlfunktion, Vormerkschränke mit Selbstverbuchung, Laptop-Ausleihe mit iPad-Butler und vieles mehr: Kombinieren Sie unsere Komponenten ganz nach Ihren Wünschen – für den optimalen Komfort in Ihrer Bibliothek!

Sprechen Sie mit unseren Experten über Ihre individuelle Lösung.

[easycheck.org](https://easycheck.org)

Ein Unternehmen der **ekz** Gruppe

**easy** **Check**  
library technologies

zur Nachnutzung veröffentlichte die europäische, biomedizinische Forschungsinfrastruktur ELIXIR.<sup>25</sup> Dem folgte in 2022 das niederländische eScience Centre zusammen mit dem niederländischen Forschungsrat (NWO) mit einer Handreichung zum expliziten Umgang mit Forschungssoftware samt konkreten Empfehlungen.<sup>26</sup> Die Max Planck Digital Library veröffentlichte Ende 2022 einen nachnutzbaren SMP-Fragenkatalog für die Software „Research Data Management Organiser“ (RDMO).<sup>27</sup> RDMO findet mittlerweile breite Verwendung als Service für das Erstellen von DMPs an deutschen Hochschulen und Forschungseinrichtungen. In 2023 veröffentlichte die DINI/Nestor Unterunterarbeitsgruppe „SMP“ auf <https://forschungsdaten.info> eine eigene SMP-Definition.<sup>28</sup>

### Software-Management-Pläne erstellen

Das Erstellen eines SMPs hat relativ viele Ähnlichkeiten mit dem Erstellen eines DMPs. Jedoch gibt es Unterschiede, z.B. ist die langfristige Kuratierung von Software wesentlich komplexer als die von Daten. Letztlich weisen SMPs und DMPs aber viele Gemeinsamkeiten auf. Dies gilt insbesondere für die Zielgruppen. Software-Management-Pläne sind in erster Linie ein Werkzeug für Wissenschaftlerinnen und Wissenschaftler, um das Projektmanagement für ihre Forschungssoftware zu organisieren. Gleichzeitig können SMPs auch Teil eines Beratungskonzeptes sein, das z.B. der lokale IT-Support, die Scientific Computing Unit oder ähnliche anbieten. Auch für das unterstützende Personal bieten SMPs Vorteile. So können Drittmittelanträge unterstützt, die Forschungsorganisation in der eigenen Einrichtung verbessert und Fragen des Projektmanagements gezielter diskutiert werden. Über all dem schwebt zugleich der Aspekt der Qualitätssicherung. Auch die Forschenden profitieren von ihren SMPs. Die explizite Darstellung der Informationen rund um ein Forschungssoftware-Projekt ermöglicht, das eigene Vorgehen strukturiert zu reflektieren. Gleichzeitig kann die Software gemeinsam mit Kolleginnen und Kollegen, z.B. aus der IT, klarer gestaltet werden. Die erfolgreiche Integration neuer Projektmitglieder wird durch das explizite Wissen in einem SMP erleichtert, da weniger Informationen verloren gehen und die Wissensorganisation bereits vorgegeben ist. Eine solche Informationserhaltung erhöht auch die langfristige Nachhaltigkeit von Forschungssoftware.

Bei der Archivierung kann ein mitgelieferter SMP wichtige Informationen für eine spätere Nach- oder Weiternutzung liefern. Und natürlich kann ein SMP auch bei der Beantragung von Drittmitteln beim Förderer eingereicht werden, wie dies bei DMPs bereits häufig der Fall ist.

### Umsetzung in RDMO

Ähnlich wie bei Forschungssoftware ist auch der Weg hin zu einem Forschungsservice durch viele Komplexitätsstufen gekennzeichnet. Mit dem SMP-Katalog für die RDMO-Software hat ein Team der Max Planck Digital Library in München in 2022 als Open Source der RDMO-Nutzergemeinschaft frei verfügbar übergeben.<sup>29</sup> Der Fragenkatalog mit dem Titel: „Software-Management-Plan für Forschende“ ist seit dem auf Deutsch und Englisch verfügbar. Der Code besteht aus 2.200 Zeilen XML und ist offen zugänglich sowie nachnutzbar.<sup>30</sup> Inhaltlich besteht der Katalog aus 49 Fragen, strukturiert in fünf Fragesets. Der erste allgemeine Teil beschäftigt sich vor allem mit den beteiligten Personen und dem Projektmanagement. Im zweiten Abschnitt werden besonders technische Informationen wie z.B. Code, Infrastruktur und Sicherheit abgefragt. Daran anknüpfend wird im dritten Segment die Qualitätssicherung über Software Testing und Dokumentation geklärt. Unter Punkt vier folgt die öffentliche Verfügbarkeit mit relevanten Informationen zu Software Releases, Metadaten und Archivierung. Im letzten Abschnitt werden die rechtlichen und ethischen Fragen wie etwa das Urheberrecht, Lizenzen und Dual Use thematisiert. Diese Fragen stehen samt erklärenden Hilfstexten und weiterführenden Informationen den Nutzenden in RDMO zur Verfügung. Die Antworten können in Freitextfeldern, Checkboxes oder Dropdown-Menüs gespeichert und über verschiedene Exportmöglichkeiten weiterprozessiert werden, beispielweise als Anhang eines Drittmittelantrages. Neben der primären Nutzung in RDMO steht ebenso eine docx.-Datei zur Verfügung, die alle englischen Fragen und Hilfstexte beinhaltet.<sup>31</sup> Dieses Dokument ist mit einer „Creative Commons Zero“ (CC0) lizenziert, sodass eigene Adaptionen ohne Einschränkung möglich sind.

### Fazit

Dieser Beitrag zeigt den Mehrwert für Forschende auf, die eine bewusste Einbindung von Forschungssoftware in den

25 Renato Alves u. a., ELIXIR Software Management Plan for Life Sciences, 2021, <https://doi.org/10.37044/osf.io/k8znb>.

26 Martinez-Ortiz, Carlos u. a., Practical Guide to Software Management Plans, v. 1.0, 2022, <https://doi.org/10.5281/ZENODO.7248877>.

27 Michael Franke und Yves Vincent Grossmann, Nachhaltige und FAIR Software in der Forschung – Ein neuer RDMO-Katalog für das Arbeiten mit Software-Management-Plänen, 2023, <https://doi.org/10.17617/2.3496327>. Zum Research Data Management Organiser (RDMO) siehe besonders <https://rdmorganiser.github.io>.

28 <https://forschungsdaten.info/praxis-kompakt/glossar/#c822402>.

29 Yves Vincent Grossmann, Pull Request #165: Add Catalogue for Software Management Plan, 2022, <https://github.com/rdmorganiser/rdmo-catalog/pull/165>.

30 <https://github.com/rdmorganiser/rdmo-catalog/blob/master/rdmorganiser/questions/SMP-Questions.xml>.

31 Yves Vincent Grossmann und Michael Franke, Template „Software Management Plan for Researcher“, 2022, <https://doi.org/10.17617/2.3481986>. Falls Ihre Institution keine RDMO-Instanz betreibt und/oder den SMP-Katalog nicht implementiert hat, können Sie sich für einen „Live-Eindruck“ gerne an den RDMO-Support der Max Planck Digital Library via [rdmo@mpdl.mpg.de](mailto:rdmo@mpdl.mpg.de) wenden.

wissenschaftlichen Alltag mit sich bringen. Die Relevanz und Anerkennung von Forschungssoftware wird in der Wissenschaft (weiter) zunehmen. Daher empfiehlt es sich für Forschende aber auch darüber hinaus, den Blick auch auf dieses Genre wissenschaftlicher Publikationen zu richten. Der explizite Umgang mit Forschungssoftware bringt einen gesteigerten Nutzen mit sich. Das Konzept der SMPs ist eines der Werkzeuge für den bewussten und reflektierten Umgang mit Software. Durch die Ähnlichkeit zu Datenmanagementplänen ist das Grundprinzip des SMP den Forschenden oft bereits bekannt. Es bietet sich daher an bereits bestehende Formate weiter auszuarbeiten.

Darauf aufbauend hat die Max Planck Digital Library einen SMP-Fragenkatalog in RDMO verfügbar gestellt. Dies ist eine Software, die bereits an deutschen Hochschulen und Forschungseinrichtungen als Dienst angeboten wird. Der SMP-Katalog der MPDL erfordert daher keine neue Infrastruktur, sondern kann nahtlos mit der bestehenden genutzt werden. Die weitere Nutzung und Iterationen zur Überarbeitung des Fragenkatalogs werden in Zukunft zeigen, in welchen Fächergruppen das Konzept der SMPs einen deutlichen Mehrwert liefern wird. Auch hierauf weist die eingangs zitierte Open Science Policy der Volkswagen-Stiftung hin, in dem sie zu recht festhält, dass „[d]er Bereich Open Science so dynamisch [ist],

dass sich viele Entwicklungen, Themen und Fragestellungen heute erst abzeichnen“.<sup>32</sup>

### Dank

Dank für die Unterstützung gebührt für Korrektorat Catharina Hofmann und Larissa Leiminger (Max Planck Digital Library) sowie für Hinweise Jens Saak (Max-Planck-Institut für Dynamik komplexer technischer Systeme). **I**



### Michael Franke

ist Diplomsoziologe und Bereichsleiter an der Max Planck Digital Library und im Bereich der digitalen Infrastrukturen für die Max-Planck-Gesellschaft tätig.

franke@mpdl.mpg.de

<https://orcid.org/0000-0002-2661-8242>



### Yves Vincent Grossmann

ist promovierter Zeithistoriker und Referent für Forschungsdatenmanagement an der Max Planck Digital Library. Er ist Service-Leiter für mehrere Forschungsdatendienste innerhalb der Max-Planck-Gesellschaft.

grossmann@mpdl.mpg.de

<https://orcid.org/0000-0002-2880-8947>

32 Volkswagen-Stiftung, Open Science Policy und Umsetzung, S. 7.

# BIS-C PLUS<sup>®</sup>

## BibliotheksInformationsSystem

Gesellschaft für Datenbank-Informationssysteme

+ Fernleihe Workshops Präsenz- & Leihbibliothek OAI SRU MAB virtueller Verbund responsive Webdesign automatisierte Abläufe Cloud WebSpace offene- & genormte Schnittstellen MARC 21 Musiknotation & Notendarstellung GND Zeitschriftenverwaltung bibliothekarische Systemschulungen Statistikauswertung UTF-8 Server Hosting Software Wartung MultiMedia Barrierefreiheit freie Generierbarkeit Support RDA Updates Fremddaten eBooks

Ihr PLUS an Individualität!

SCHNELL

EFFIZIENT

ZUVERLÄSSIG

DABIS GmbH, Heiligenstädter Str. 213, A-1190 Wien  
Telefon: +43 (1) 318 9777, eMail: [office@dabjs.eu](mailto:office@dabjs.eu)  
[www.dabjs.eu](http://www.dabjs.eu)

ERFAHRUNG + WISSEN + KOMPETENZ

SEIT 1989