


































Code for Science: Bessere Forschung durch bessere Forschungssoftwarekompetenzen

Simon Christ ¹, Bernadette Fritzsche ², Yves Vincent Grossmann ³, Dominic Kempf ⁴, Jan Linxweiler ⁵, Frank Löffler ⁶, Lydia Riedl ⁷, Jan Philipp Thiele ⁸, Adina S. Wagner ⁹, Lorenzo Zanon ¹⁰, Cora Assmann ¹¹, Dirk Brömmel ¹², René Caspart ¹³, Mustafa Dogan ¹⁴, Sven Fillinger ¹⁵, Jasmin Frangenberg ¹⁶, Michael Franke ¹⁷, Annalena Frey ¹⁸, Jonas Hagenberg ¹⁹, Carina Haupt ²⁰, Volodymyr Kushnarenko ²¹, Sarah Neuwirth ²², Tobias Ribizel ²³, Jens Saak ²⁴, Andreas Schieberle ²⁵, Kristine Schima-Voigt ²⁶, Peter Schmidt ²⁷, Heidi Seibold ²⁸, Robert Speck ²⁹, Mathijs Vleugel ³⁰, Alexander Wellmann ³¹, Philipp Wieder ³², und Samantha Wittke ³³

¹Leibniz Universität Hannover, ²Alfred-Wegener-Institut, Helmholtzzentrum für Polar- und Meeresforschung Bremerhaven, ³Universität Frankfurt, ⁴Scientific Software Center, Universität Heidelberg, ⁵TU Braunschweig, ⁶Universität Jena, ⁷Universität Marburg, ⁸Weierstraß-Institut Berlin, ⁹Forschungszentrum Jülich, ¹⁰Hochleistungsrechenzentrum Stuttgart (HLRS), Universität Stuttgart, ¹¹Universität Jena, ¹²Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH, ¹³Karlsruher Institut für Technologie, ¹⁴SUB Göttingen, ¹⁵Zentrum für Quantitative Biologie, Universität Tübingen, ¹⁶Leibniz-Institut für Naturstoff-Forschung und Infektionsbiologie, ¹⁷Max Planck Digital Library, ¹⁸Universität Heidelberg, ¹⁹Max-Planck-Institut für Psychiatrie, ²⁰Deutsches Zentrum für Luft- und Raumfahrt, ²¹Hochleistungsrechenzentrum Stuttgart (HLRS), Universität Stuttgart, ²²Universität Mainz, ²³TU München, ²⁴Max Planck Institute for Dynamics of Complex Technical Systems, ²⁵Universität Gießen, ²⁶SUB Göttingen, ²⁷Podcast Code4Thought, ²⁸Digital Research Academy, ²⁹Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH, ³⁰Helmholtz Open Science Office, ³¹Leibniz Supercomputing Center, ³²Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen, ³³CSC-IT Center for Science, Finland, CodeRefinery

27. Februar 2025

Zusammenfassung

Die Bedeutung von Forschungssoftware im wissenschaftlichen Erkenntnisprozess nimmt zu. Die Entwicklung und der Umgang mit Forschungssoftware erfordern jedoch Kompetenzen, die derzeit im deutschen Wissenschaftssystem noch nicht in genügendem Maße und in ausreichender Breite vorhanden sind. Dieses Problem wurde in einem Workshop mit Unterstützung der Volkswagen-Stiftung adressiert. Die Veranstaltung fand im Dezember 2024 statt und entwickelte Handlungsempfehlungen für verschiedene Akteure, die der Förderung von Forschungssoftwarekompetenzen an deutschen Wissenschaftsinstitutionen dienen.

Persistenter Identifier: <https://doi.org/10.5281/zenodo.14273366>

Dieses Werk ist lizenziert unter einer [Creative Commons](#) “Namensnennung 4.0 International” Lizenz.



1 Einleitung

Der zunehmende Einsatz von Forschungssoftware eröffnet in vielen Forschungsfeldern neue Möglichkeiten, erfordert jedoch auch entsprechende Kompetenzen bei Wissenschaftlerinnen und Wissenschaftlern.¹ Trotzdem wird Forschungssoftware und insbesondere der entsprechende Kompetenzaufbau von wissenschaftlichen Einrichtungen nicht immer im notwendigen Umfang adressiert. Oft existiert keine Übersicht, wer innerhalb einer Institution Forschungssoftware entwickelt, betreibt und erhält. Synergien und Kommunikationswege entstehen dabei selten oder meist zufällig. Ähnlich ist es um Kompetenzentwicklung in diesem Bereich bestellt. Während beispielsweise lokal einige Fachbereiche durchaus erfolgreiche Anstrengungen unternehmen, wird von der breiten Masse der Studierenden – und damit der zukünftigen Forschenden – erwartet, sich diese Kompetenzen selbstständig in der Freizeit anzueignen und dies oft ohne nützliche Hinweise, welche Fähigkeiten eigentlich später gebraucht werden.

Diese Situation findet vor einer zunehmend dynamischen Entwicklung rund um die Handhabung von Forschungssoftware in Deutschland, Europa und der Welt statt. Mit der jüngsten DFG-Handreichung zu Forschungssoftware² vom Oktober 2024 wurde beispielsweise auf nationaler Ebene die Bedeutung von Forschungssoftware aber auch der notwendigen Kompetenzen prominent unterstrichen.³ Dies lässt sich in einer bereits laufenden Diskussion zu Forschungssoftware hinsichtlich Dauerhaftigkeit und Effizienz im deutschen Wissenschaftssystem verorten.

Unter dem Titel „Code for Science oder: Bessere Forschung durch bessere Forschungssoftwarekompetenzen“ fand daher im Rahmen der „Themenwoche Digitale Kompetenzen in der Wissenschaft“ der Volkswagenstiftung vom 2. bis 4. Dezember 2024 im Schloss Herrenhausen in Hannover ein Workshop mit zahlreichen Vertretern der deutschsprachigen Research Software Engineering (RSEng)⁴ Community statt.⁵ Das zentrale Ziel dieser Veranstaltung war Empfehlungen zu erstellen, auf welche Art und Weise die verschiedenen Akteure und ihre Institutionen befähigt werden können, mit Forschungssoftware und entsprechender Kompetenzgewinnung bewusster umzugehen. Der Antragstext für den Workshop ist im Vorfeld bereits offen publiziert worden.⁶ Zusätzlich entstand im Nachgang eine „Code for Thought“-Podcast-Folge von Peter Schmidt, welcher den Workshop begleitete.⁷

Die erarbeiteten Empfehlungen basieren auf einer Vielzahl an Vorarbeiten, welche in Deutschland insbesondere die deRSE-Community geleistet hat. Das deRSE-Positionspapier Nr. 1 dokumentierte im Jahr 2021 den damaligen Stand zum Research Software Engineering⁸ und verband die Auflistung von offenen Herausforderungen mit dem Appell nach Handlungen. Das darauf folgende deRSE-Positionspapier Nr. 2⁹ Ende 2023 spezifizierte dann die wesentlichen Kompetenzen und Verantwortungen von Forschungssoftwareentwickler:innen. Gedanklich daran anknüpfend verstehen sich die hier präsentierten Handlungsempfehlungen als weiterer Debattenbeitrag zur Diskussion von Forschungssoftware und deren Verankerung in der bundesdeutschen Wissenschaftslandschaft.

2 Der status quo zu Forschungssoftware und Kompetenzen in Deutschland

Für einen Überblick über konkrete Ansätze, wie Software Engineering-Praktiken im akademischen Alltag verwirklicht werden, gab es auf dem Workshop in Hannover sechs Impulse in der Form von Vorträgen, welche das Verhältnis Forschungssoftware und Kompetenzen aus unterschiedlichen Perspektiven beleuchteten. Diese Beiträge werden im folgenden Kapitel skizziert, um die eingehenden Debatten aus unterschiedlichen Blickrichtungen verständlich zu gestalten, denn auf ihnen bauten die weiteren Diskussionen Richtung Handlungsempfehlungen maßgeblich auf.

¹Für die Diskussion der verschiedenen Kategorien und Konzepte zu „Forschungssoftware“ siehe beispielsweise Hasselbring et al. [2024].

²Deutsche Forschungsgemeinschaft [2024].

³Siehe hierzu ebenfalls die skizzierten Thesen 3.2 und 3.3 in [Deutsche Forschungsgemeinschaft, Ausschuss für Wissenschaftliche Bibliotheken und Informationssysteme, 8-9].

⁴Der folgende Text verwendet die Abkürzung „RSE/RSEs“ für „Research Software Engineer(s)“ also die Menschen. Für die sprachliche Klarheit wird demgegenüber „RSEng“ verwendet, um das „Research Software Engineering“ also die Disziplin, zu verdeutlichen.

⁵Volkswagen-Stiftung [2024].

⁶Kempf et al. [2024].

⁷Schmidt [2025].

⁸Anzt et al. [2021].

⁹Goth et al. [2024].

2.1 Forschungssoftware in wissenschaftlichen Bibliotheken

Bibliotheken stehen im Wandel. Die digitale Transformation verändert die Forschung, Lehre und die Bibliothek selbst. Bücher als Datenträger verlieren dabei ihren Exklusivstatus und neue Träger und Formate für Forschungsergebnisse entstehen. In diesem Kontext präsentierte Mustafa Doğan exemplarisch die Staats- und Universitätsbibliothek Göttingen (SUB) und deren RSE-als-Service.¹⁰ Durch die Zentralisierung von RSEng-Spezialist:innen kann an der SUB eine umfassende Expertise für Forschungssoftware aufgebaut werden. Durch strategische Partnerschaften mit Einrichtungen wie der GWDG kann ein erweitertes Serviceportfolio angeboten und Innovationsförderung im akademischen Kontext betrieben werden. Damit lässt sich eine höhere Effizienz bei der Softwareentwicklung erzielen, was strategische Vorteile bietet. Problemfelder sind aber weiterhin die oft nicht anerkannte Rolle der Forschungssoftware in einer Bibliothek und die schwache Verknüpfung von Forschungssoftwarekompetenzen mit „traditionellen“ Bibliotheksaufgaben.

2.2 Perspektiven von Rechenzentren auf Forschungssoftware

Ähnlich wie Bibliotheken sind auch Rechenzentren mit neuen Herausforderungen bei der digitalen Transformation konfrontiert. Philipp Wieder von der Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen (GWDG) präsentiert hierzu seine Perspektiven auf diese Entwicklungen.¹¹ Dabei arbeitete er heraus, dass für Software, die von einem Rechenzentrum angeboten und unterstützt wird, teilweise eine Vielzahl von Regelungen, Verordnungen, Normen etc. gelten, welche für diverse Zertifizierungen zu beachten sind. Diese Anforderungen werden bei der Eigenentwicklung im Rechenzentrum bereits von Anfang an berücksichtigt. Es existieren dafür bereits etablierte Standards und Prozesse, wie etwa Qualitäts-Managementprozess für Software, Reifegradmodelle oder ähnliches. Problematisch wird es hingegen, wenn der Betrieb und die Weiterentwicklung einer Forschungssoftware aus RZ-fremden Bereichen, wie Projekten, universitären Einrichtungen, übernommen werden sollen. Denn dort existieren oft die entsprechenden Regularien nicht und auch die daraus folgenden Software Engineering-Praktiken sind wenig oder nicht bekannt. Es ist daher zu überlegen, inwiefern es von RSEs und Rechenzentrum gemeinsam erarbeitete Prinzipien, Standards und *Best Practices* für das RSEng braucht, so dass die Anforderungen für einen künftigen Betrieb der Forschungssoftware im Rechenzentrum von Anfang an mitgedacht werden können. Eventuell sind auch gemeinsame Entwicklungsplattformen oder Praktiken, z.B. DevOps, als Lösungsansatz möglich. Dabei sollte stets die Anschlussfähigkeit an die European Open Science Cloud (EOSC) und die Nationalen Forschungsdateninfrastrukturen (NFDI) mitbedacht werden, um die Kompatibilität und die Integration von Forschungssoftware in die Infrastrukturen der Rechenzentren sicherzustellen.

2.3 Open Science und Forschungssoftwarekompetenzen

In ihrem Impulsvortrag¹² betonte Heidi Seibold von der Digital Research Academy die Bedeutung von Open Science, um die Qualität und Reproduzierbarkeit von Forschungsergebnissen sicherzustellen. Dazu bedarf es einer Reihe von Werkzeugen, etwa für die digitale Organisation von Dateien, Versionskontrolle oder Programmierkenntnisse, deren Handhabung durch die Forschenden entsprechende Fähigkeiten voraussetzt. Oft mangelt es im Forschungsalltag jedoch an der Zeit, die Forschenden für die Umsetzung von Open Science eingeräumt wird. Um dies zu verbessern, sind Expert:innen erforderlich, die sich mit Open Science und Forschungssoftware hinreichend auskennen und die andere Forschende darin unterstützen können. Erfolge gibt es bereits, wie zum Beispiel die zunehmende Anerkennung des Bedarfs an RSEng-Kompetenzen. Ebenso zu nennen sind die vielen Initiativen, die sich für Open Science, Reproduzierbarkeit und die dazugehörigen digitalen Fähigkeiten einsetzen, wie beispielsweise „The Turing Way“¹³, „The Carpentries“¹⁴ und „CodeRefinery“¹⁵. Diese Projekte zeigen, dass es möglich ist, Open Science und RSEng-Kompetenzen zu fördern und erfolgreich zu implementieren.

¹⁰Doğan [2024].

¹¹Wieder [2024].

¹²Seibold [2024].

¹³<http://the-turing-way.org>.

¹⁴<https://carpentries.org>.

¹⁵<https://coderefinery.org>.

2.4 Forschungssoftwarekompetenzen an Einrichtungen des Höchstleistungsrechnens

Forschungssoftwarekompetenzen und HPC sind laut Sarah M. Neuwirth von der Universität Mainz Schlüssel zu Innovation und Effizienz in der wissenschaftlichen Forschung.¹⁶ Im HPC-Bereich dient Software als Brücke zwischen leistungsfähiger Hardware und wissenschaftlichen Fragestellungen. Durch standardisierte Softwarepraktiken können interdisziplinäre Teams unterstützt werden. Jedoch sind solche Standards noch nicht überall etabliert. Außerdem verfügen Wissenschaftler:innen oft nicht über formale Ausbildung in Software-Engineering-Praktiken. Probleme entstehen auch durch die mitunter unzureichend Dokumentation von Forschungssoftware und die fehlende Nachhaltigkeit in der Softwareentwicklung, was die Reproduzierbarkeit wiederum erschwert. Beim Entwicklungsprozess treten häufig Transferprobleme zwischen Disziplinen auf, da Softwareentwickler:innen und Fachwissenschaftler:innen oft unterschiedliche „Sprachen“ sprechen.

2.5 Forschungsdatenmanagement und Forschungssoftware

Cora Assmann von der Universität Jena verwies in ihrem Vortrag darauf, dass Forschungssoftwarekompetenzen ein zentraler Baustein im Forschungsdatenmanagement sind, da Software in allen Phasen des Datenlebenszyklus eine entscheidende Rolle spielt.¹⁷ Software begleitet Forschende bei der Erzeugung, Analyse, Bearbeitung, Qualitätssicherung und Verwaltung von Forschungsdaten. Allerdings gibt es institutionelle Herausforderungen bei der Förderung von Forschungssoftwarekompetenzen. Teilweise fehlt es am Bewusstsein für das Thema oder es wird nicht als prioritäres Handlungsfeld gesehen. Institutionen verfügen häufig nicht über die finanziellen und personellen Ressourcen, um nachhaltige Angebote oder Strukturen zu schaffen. Auch auf individueller Ebene bei den Wissenschaftler:innen gibt es vielfältige Hürden für den Erwerb von Forschungssoftwarekompetenzen. Forschende haben oft wenig Zeit, um sich weiterzubilden, und erhalten nicht immer die notwendige Unterstützung durch ihre Vorgesetzten. Bestehende Fortbildungsmöglichkeiten sind teilweise unbekannt oder es ist Wissenschaftler:innen gar nicht bewusst, dass ihnen Kompetenzen im Bereich Forschungssoftware fehlen. Hinzu kommt, dass Forschungsdaten sehr vielfältig und komplex sind und notwendige Softwarekompetenzen häufig disziplinspezifisch benötigt werden, solche Angebot aber teilweise noch nicht existieren.

2.6 Inner- und überinstitutionelle Weiterbildung bei Forschungssoftware

Die Entwicklung von Forschungssoftware ist für Lorenzo Zanon vom Höchstleistungsrechenzentrum an der Universität Stuttgart (HLRS) oft ein komplexes Unterfangen, das nicht nur Fachwissen, sondern auch spezielle Kompetenzen in der Softwareentwicklung erfordert.¹⁸ Wissenschaftler:innen, die sich mit der Entwicklung von Forschungssoftware auseinandersetzen, müssen nicht nur Kenntnisse in Code-Best-Practices haben, sondern auch in der Lage sein größere Programme zu lesen, zu pflegen und zu erweitern. Durch Weiterbildungen und Trainings kann dies schon jetzt erreicht werden, indem die jeweiligen Anbieter sich auf die Bedürfnisse der Forschenden und die Herausforderungen der Softwareentwicklung konzentrieren.

3 Handlungsempfehlungen zu besseren Forschungssoftwarekompetenzen an deutschen Wissenschaftsinstitutionen

Die Impulsvorträge des Symposiums zeigen zum einen Maßnahmen mit Vorbildcharakter, zum anderen verdeutlichen sie den Bedarf an Forschungssoftwarekompetenzen in unterschiedlichen wissenschaftlichen Bereichen. Zudem werden Hindernisse aufgezeigt, welche für eine Stärkung von Forschungssoftwarekompetenzen überwunden werden müssen. Im Rahmen des Workshops sind daher konkrete Handlungsempfehlungen formuliert worden. Sie bieten verschiedenen Akteur:innen im Wissenschaftssystem Möglichkeiten, die (Weiter-)Entwicklung von Kompetenzen in Forschungssoftware aus unterschiedlichen Blickwinkeln zu fördern. Für eine langfristige, nachhaltige Verbesserung der Forschungssoftwarekompetenzen in der Wissenschaft ist ein Zusammenspiel all dieser Akteur:innen wünschenswert.

¹⁶Neuwirth [2024].

¹⁷Assmann [2024].

¹⁸Zanon [2024].

Zielgruppen

Es existiert keine universelle Maßnahme zur Förderung von Forschungssoftwarekompetenzen, die für alle gültig ist. Stattdessen sollten je nach Kontext und Funktion zielgerichtete Vorgehen etabliert und entsprechend zielgruppenspezifische Handlungsempfehlungen für die verschiedenen Funktionsgruppen in unterschiedlichen wissenschaftlichen Institutionen entwickelt werden. Dazu haben wir eine Reihe konkreter Zielgruppen definiert: Wissenschaftspolitik, Fördergeber, Wissenschaftliche Infrastruktur, Institutsleitungen, Führungskräfte, Lehrpersonal, Interessensvertretungen und -verbände, sowie Entwickler:innen von Forschungssoftware.

Entwickler:innen von Forschungssoftware und deren Netzwerke: Obwohl sie die „kleinste Einheit“ sind, können Entwickler:innen von Forschungssoftware und Netzwerke, „RSEng Champions“ (siehe Definition in Kapitel 3.2) oder lokale RSE-Netzwerke in ihrem unmittelbaren Umfeld bedeutungsvolle Beiträge leisten, insbesondere bei Community Aspekten. Als Nutzende der Ergebnisse von Handlungsempfehlungen, etwa Lernangeboten, können sie auch maßgeblich an höhere Zielgruppen vermitteln, dass ein Bedarf im Bereich von Forschungssoftwarekompetenzen besteht.

Interessensvertretungen und -verbände: Interessensvertretungen und -verbände wie etwa die Research Software Alliance¹⁹ oder de-RSE e.V.²⁰ können Interessen institutionsübergreifend bündeln. Sie repräsentieren sowohl lokal eingebundene RSEs als auch Entwickler:innen ohne Netzwerk und können systemische Anliegen auf nationaler und internationaler Ebene verfechten. Hierzu zählen auch die Fachgesellschaften, wie etwa die Gesellschaft für Informatik (GI)²¹, welche darüber hinaus die spezifischen Kompetenzen innerhalb ihres Bereichs identifizieren und fördern können.

Wissenschaftliches Lehrpersonal: Lehrkräfte sind in der einzigartigen Position, Forschungssoftwarekompetenzen einer großen Menge zukünftiger Forschender zu demonstrieren und zu vermitteln. Diese Weitergabe kann zu einem langfristigen Kulturwandel beitragen.

Wissenschaftliche Führungskräfte: Innerhalb des wissenschaftlichen Systems haben wissenschaftliche Führungskräfte großen direkten Einfluss auf die wissenschaftliche Praxis auf den verschiedenen Ebenen, die sie verantworten – sei es als Fachbereichs-, Arbeitsgruppen- oder Projektleitung. Sie bestimmen mit Anreizen, Arbeitskultur oder Anforderungen an Mitarbeitende maßgeblich, ob sich digitale Kompetenzen entwickeln können oder nicht.

Einrichtungsleitungen: Führungspersonal von Wissenschaftsorganisationen, Universitäten, Hochschulen, außeruniversitären Forschungseinrichtungen, oder im Bereich der Forschungscoordination hat zwar weniger direkten Einfluss auf wissenschaftliche Praxis, kann aber über zentrale Vorgaben und Prioritätensetzung digitalen Kompetenzerwerb fördern.

Lokale wissenschaftliche Infrastruktur: Unter lokaler wissenschaftlicher Infrastruktur verstehen wir Services, wie Studiengangskoordination, Kommunikationsteams, Data Stewards, oder Open Science Offices. Sie können von Institution zu Institution unterschiedlich präsent sein und nicht in jeder Einrichtung zur Verfügung stehen. Wo sie vorhanden ist, kann lokale wissenschaftliche Infrastruktur aber abteilungsübergreifende unterstützende Rahmenbedingungen schaffen, um Forschungssoftwarekompetenzen zu fördern.

Wissenschaftliche Großinfrastruktur/zentrale Einrichtungen: Wissenschaftliche Großinfrastruktur- und Serviceanbieter wie Bibliotheken und Rechenzentren sind seit jeher an der Schnittstelle zwischen unterschiedlichen wissenschaftlichen Bereichen. Da ihr Angebot nicht originär fachspezifisch ausgelegt ist, wirken sie bereits fachübergreifend als Multiplikatoren und können vermittelnd und werbend im Bereich RSEng aktiv werden. Zudem beschäftigen diese Einrichtungen typischerweise auch selbst Personen mit Softwarekompetenzen, um ihr Serviceportfolio weiterzuentwickeln, so dass eine Nutzung von Synergien zwischen RSE-Aktivitäten in der Wissenschaft und in wissenschaftlichen Serviceeinrichtungen möglich wird.

Fördergeber: Im Wissenschaftssystem haben die Förderer eine zentrale Rolle inne, indem sie durch die Vergabe von Mitteln entscheidend das Profil der wissenschaftlichen Arbeit beeinflussen. Neben der direkten Vergabe von Mitteln zur Förderung von Forschungssoftware ist dabei insbesondere

¹⁹<https://www.researchsoft.org/>

²⁰<https://de-rse.org>.

²¹<https://gi.de>.

die Festlegung der Rahmenbedingungen zur Förderung von Vorhaben mit Softwareaspekten ein wichtiger Hebel der Fördergeber. In den vergangenen Jahren hat sich bereits einiges in diesem Bereich getan.²²

Wissenschaftspolitik: Die Zielgruppe mit der langfristigen Perspektive ist die der Wissenschaftspolitik. Sie kann insbesondere dafür Sorge tragen, dass sich die Arbeitsbedingungen in der Wissenschaft so verändern, dass Softwarekompetenz im Wissenschaftssystem gefördert wird. Neben klassischen Akteuren wie Parteien spielen dabei auch politische Vernetzungsinitiativen wie etwa die Allianz der Wissenschaftsorganisationen²³ eine entscheidende Rolle.

Die folgenden Handlungsempfehlungen gliedern sich in drei Hauptbereiche: „Lernangebote bereiten“, „RSE-Communities fördern“ sowie „Anreize und Rahmen entwickeln“. Innerhalb dieser Hauptbereiche formulieren wir konkrete Handlungsempfehlungen - wenn möglich zielgruppenspezifisch.

3.1 Lernangebote bereiten

Ein zentrales Element in der Förderung von Forschungssoftwarekompetenzen sind Ausbildungsangebote an Personen im wissenschaftlichen System.

3.1.1 Wissenschaftliches Lehrpersonal

Vermittelt Open Science! Open Science sollte Einzug in die Lehre erhalten. Dazu gehören sowohl theoretisches als auch praktisches Wissen. Im ersten Schritt wünschen wir uns dazu eine Reflexion der (digitalen) Kompetenzen, die in vorhandenen Lehrveranstaltungen vorkommen, sowie der Tools, die dies unterstützen. Im Sinne der offenen Wissenschaft sollte, wo möglich, in der Lehre auf offene Softwarewerkzeuge (FOSS/FLOSS²⁴) gesetzt und begründet werden, warum diese bevorzugt werden, z.B. mit „Public money, public code“.

Gibt konkrete Empfehlungen zum Selbststudium! In der praktischen Anwendung der Inhalte einer Vorlesung werden häufig digitale Tools verwendet. Diese sollten im Rahmen der Literaturempfehlung, z.B. in Vorlesungsskripten, ebenso Erwähnung finden. Das gibt Studierenden Impulse dazu, was neben der Vorlesung interessant und relevant für das Selbststudium ist.

Nutzt vorhandene Materialien! Lehrkräfte müssen neue Materialien selten allein entwickeln, da insbesondere für die Lehre von digitalen Kompetenzen viele frei nutzbare Materialien, etwa open educational resources (OER²⁵), verfügbar sind. Im Bereich der Forschungssoftwarekompetenzen gibt es hier bereits koordinierte Bestrebungen und Projekte, wie z.B. The Carpentries²⁶, CodeRefinery²⁷, Elixir²⁸, INTERSECT²⁹, UniverseHPC³⁰, oder „The Missing Semester of your CS education“³¹. Diese stellen ihre Materialien mit einer nachnutzbaren offenen Lizenz – häufig unter CC-BY³² – zur Verfügung und sind offen für Mitarbeit und Beiträge.

Bildet euch weiter! Um die Qualität der Lehrveranstaltungen zu sichern und weiter zu erhöhen, bieten manche der oben genannten Organisationen/Projekte verschiedene Formen von „Train-the-Trainer“-Workshops an.³³ Bei diesen kann man sich für die Vermittlung von digitalen Kompetenzen fortbilden.

²²Siehe beispielsweise [Deutsche Forschungsgemeinschaft \[2024\]](#).

²³<https://www.allianz-der-wissenschaftsorganisationen.de>.

²⁴https://en.wikipedia.org/wiki/Free_and_open-source_software

²⁵https://de.wikipedia.org/wiki/Open_Educational_Resources

²⁶<https://carpentries.org/>

²⁷<https://coderefinery.org/>

²⁸<https://elixir-europe.org/>

²⁹<https://intersect-training.org>

³⁰<https://www.universe-hpc.ac.uk/>

³¹<https://missing.csail.mit.edu/>

³²<https://creativecommons.org/licenses/by/4.0/>

³³Siehe hierzu beispielsweise die „Train-the-Trainer“-Workshop im Forschungsdatenmanagement, [Biernacka et al.](#) oder das „Carpentries Instructor Training“ <https://carpentries.org/instructor-training/>.

Schafft innovative Formate für Lernende ohne bestehende Community! Insbesondere für Forschungssoftwareentwickler:innen ohne lokales RSE Netzwerk sind Lernmöglichkeiten zum Selbststudium und Austausch, wie Wikis, Online-Foren oder Chatkanäle wichtig. Praktische Formate wie Hackathons, Advent of Code oder Code Katas gestalten den Lernprozess interaktiv und fördern den direkten Kontakt mit der Community. Gleichzeitig sollten Forschende ermutigt werden, sich aktiv in die Community einzubringen, etwa durch Peer-Coding, Mentoring oder die Moderation von Diskussionsforen.

3.1.2 Wissenschaftliche Führungskräfte

Seid Mentor:in! Wenn Führungskräfte selbst digital kompetent sind, sollten sie Mitarbeitenden als Mentor:in im Softwarekompetenzbereich zur Verfügung stehen. Beispiele sind Onboarding in digitale Arbeitsprozesse oder Werkzeuge im Rahmen von wissenschaftlicher Arbeit, das Identifizieren relevanter digitaler Kompetenzen bei Mentees und die Ermutigung zum Erwerb dieser Kompetenzen, oder die Einbindung in relevante Angebote und Netzwerke.

3.1.3 Wissenschaftliche Infrastruktur

Integriert digitale Kompetenzen frühzeitig in die Studienpläne! Lehrveranstaltungen zur Vermittlung von digitalen Kompetenzen sollten im Studiengang verfügbar sein, und die Teilnahme etwa in Beispielstudienplänen erwähnt werden. Eine verpflichtende Teilnahme kann auch in Betracht gezogen werden, wenn ein hinreichender Anteil der Studierenden die vermittelten digitalen Kompetenzen im Laufe ihres Studiums benötigt. Eine niederschwellige Möglichkeit der Einführung allgemeiner Kompetenzen bieten fachübergreifende Schlüsselqualifikationskurse. Es kann allerdings von Vorteil sein, wenn allgemeine Kurse mit fachspezifischer Zielgruppe angeboten werden können, um eine einladende und verbindende Atmosphäre zu schaffen und auf typische Erfahrungen und Schwierigkeiten einzugehen. Thematisch spielen je nach Fachbereich sowohl allgemeine Einführungskurse als auch Schwerpunktkurse eine große Rolle. Beispiele für erstere sind etwa Kurse zu fachübergreifenden Themen wie Versions- oder Literaturverwaltung. Letztere sind Kurse zu fachspezifischeren Themen, wie etwa die Nutzung von Jupyter Notebooks, HPC-Systemen oder Forschungsdatenmanagement.

Macht Lehrveranstaltungen zu digitalen Kompetenzen sichtbar! Zusätzlich zur Integration in die Studienpläne sollten Lehrveranstaltungen sichtbar und einfach verfügbar sein. Digitale Kompetenzen haben während des Studiums fortwährende Relevanz, aber Studierenden kann das Fehlen dieser Kompetenzen oder die Verfügbarkeit der Angebote unter Umständen nicht bewusst sein. Bestehende Angebote sollten daher bereits möglichst zu Studienbeginn beworben werden. Die größte Rolle spielen diese Kompetenzen in spezifischen Studienleistungen wie Praktika und Bachelor- und Masterarbeiten. Daher können Einführungs- und Informationsveranstaltungen verwendet werden, um erneut auf verfügbare Angebote zu verweisen.

3.1.4 Wissenschaftliche Großinfrastruktur und Serviceanbieter

Nutzt eure Netzwerke! Da Serviceanbieter wie wissenschaftliche Bibliotheken oder Rechenzentren fachbereichsübergreifend wirken, können diese Strukturen auch als idealer Einstieg gesehen werden, eine Kooperation zwischen verschiedenen Akteuren im Bereich Forschungssoftware zu vermitteln oder Querverweise zu erstellen. Unter Einbezug aller zentralen Einrichtungen und mit den Netzwerken der Infrastrukturanbieter über lokale Akteure hinaus können föderierte Angebote erarbeitet und vorhandene Kompetenzen überregional gebündelt werden.

Baut niederschwellige Angebote aus! Infrastruktur- und Serviceanbieter können Hemmschwellen zwischen Fächern abbauen, indem sie ihr Kursangebot trotz fachübergreifendem Inhalt zielgruppenspezifisch aufarbeiten beziehungsweise anbieten. Um das Kursangebot gezielter ausrichten zu können, sollten sie die Ergebnisse von Zielgruppenevaluation mit relevanten Akteuren aus dem Bereich RSEng teilen. Und sie sollten Werbung und Außenwirkung um das Thema Softwarekompetenz erweitern.

Bildet euch selbst weiter! Bei internen Weiterbildungen sollten sich Infrastruktur- und Serviceanbieter nicht nur auf klassische Personalentwicklungsthemen konzentrieren, sondern auch RSEng-Themen einbinden. Wenn es in einem Fachbereich akuten Bedarf an einer Schulung im Bereich Forschungssoftwarekompetenzen gibt, und gleichzeitig ein bestehendes Angebot in einer

anderen Abteilung, können Ressourcen gebündelt und Lehrkräfte „verliehen“ werden. So können Ressourcen innerhalb der Organisation effizient eingesetzt werden. Dies dient auch der Etablierung einer breiten Kompetenzbasis und der Werbung für die unterschiedlichen Arbeitsfelder im RSEng-Bereich.

Stärkt Kompetenzen mit fortgeschrittenen Lernangeboten und Unterstützungsangeboten nachhaltig! Forschende, die erste Erfahrungen gesammelt haben, benötigen Gelegenheiten, ihre Fähigkeiten in einem größeren Kontext anzuwenden und zu vertiefen. Dies ist besonders wichtig in der Phase nach den ersten Coding-Erfahrungen, in der Einsteigerwissen nicht mehr ausreicht, um domänenspezifische Coding-Projekte weiterzuentwickeln. Fortgeschrittene Formate wie „Bring your own code“-Hackathons ermöglichen es, konkrete Projekte mit Unterstützung von Expert:innen weiterzuentwickeln. Schulungen mit direktem Bezug zur eigenen Forschungsdomäne helfen, die Brücke zwischen Theorie und Empirie zu schlagen. Ein wichtiger Aspekt ist das Erwartungsmanagement. Der Fokus sollte darauf liegen, Forschende zu befähigen, selbstständig zu agieren und ihre Erkenntnisse in die eigene Forschungsumgebung zu transferieren. Grenzen der Begleitung sollten möglichst frühzeitig definiert und kommuniziert werden. Auf diese Weise lassen sich realistische Ziele setzen und Enttäuschungen vermeiden.

3.1.5 Fördergeber

Setzt die offene Bereitstellung von Lehrmaterial durch! Bei der Evaluation von Projekten soll darauf hingewirkt werden, dass die dort entwickelten Lehrmaterialien als Open Educational Resources (OER) bereitgestellt werden, um sie über das Projekt hinaus und später nach zu nutzen. Dies muss zum Projektabschluss geprüft werden und bei Nichtbeachtung Konsequenzen haben, beispielsweise bei weiteren Anträgen.

3.1.6 Zielgruppenübergreifende Aufgaben

Ermittelt Bedarf oder meldet Bedarf! Sämtliche Zielgruppen sollten aktiv an und in Bedarfsanalysen mitwirken. Anbietende sollten in der Interaktion mit Forschenden, Lehrenden, und Studierenden die spezifischen Bedarfe für verschiedene Fachdisziplinen ermitteln und ihr Angebot daran orientieren; Empfänger:innen dieser Angebote sollten proaktiv Bedarf kommunizieren und Angebote annehmen.

Erhältet längerfristig Angebote! Sämtliche Zielgruppen, die Lernangebote bereiten, sollten sicherstellen, dass innerhalb ihrer Abteilung oder innerhalb der Institution ausreichend Menschen mit den zu vermittelnden Softwarekompetenzen vorhanden sind. Dafür sollten sie Weiterbildungsangebote für Lehrende und daran Interessierte anbieten („Train-the-trainer“), um genügend Nachwuchs sicherzustellen.

3.2 RSE-Community fördern

Der Schlüssel zu nachhaltigem Kompetenzaufbau liegt darin, eine lebendige Community zu schaffen, die den Wissensaustausch fördert und Forschende auf allen Ebenen einbindet. Die folgenden Handlungsempfehlungen zielen darauf, solche Communities zu fördern. Je zugänglicher der Einstieg gestaltet wird, desto mehr Forschende fühlen sich ermutigt, Teil der RSE-Gemeinschaft zu werden und ihr Wissen aktiv einzubringen. Gleichzeitig ist es wichtig, diese Communities nachhaltig weiterzuentwickeln.

Innerhalb von RSE-Gemeinschaften gibt es Personen, die Sonderrollen einnehmen, die im Vorfeld der Handlungsempfehlungen definiert werden sollen. In unserem Workshop wurden auf Basis von Erfahrungen in verschiedenen Institutionen und Kontexten zwei Sonderrollen identifiziert. Wir stellen sie hier unter den Begriffen „Multiplikator:innen“ und „Champions“ vor. Zuweilen haben Institutionen bereits formale Rollen oder Instrumente etabliert, kennen sie aber unter anderem Titel; zuweilen fallen Personen ohne dedizierte Rollenbeschreibung in diese Kategorien; theoretisch kann eine Einzelperson auch beide Rollen gleichzeitig ausfüllen. Wichtiger als der konkrete Name ist was diese Rollen tun:

Multiplikator:innen sind ein zentraler Knotenpunkt im lokalen RSE-Netzwerk. Sie dienen als erste Anlaufstelle für neue und etablierte Mitglieder, und vernetzen innerhalb und über die eigene Einrichtung hinaus. Ihre Aufgaben bestehen darin, relevante Themen zu identifizieren, zu verstärken und Wissen zu verbreiten, sowie den Austausch und die Zusammenarbeit über verschiedene

Disziplinen und Einrichtungen hinweg zu fördern. Sie stärken die Sichtbarkeit von Forschungssoftwarethemen, verbreiten *Best Practices* und fördern die interdisziplinäre Zusammenarbeit. Indem Multiplikator:innen nicht nur das Wissen innerhalb der Community verbreiten, sondern auch Impulse von außen aufnehmen, tragen sie aktiv dazu bei, die Community weiterzuentwickeln und nachhaltig auszubauen. Sie erfüllen also Funktionen, die auch Community Manager:innen, Koordinator:innen oder Open-Science-Beauftragte zum Teil abdecken. Dabei müssen Multiplikator:innen nicht zwangsläufig Einzelpersonen sein. Die Funktion einer Multiplikator:in kann beispielsweise auch durch Gremien, Fachgesellschaften, Initiativen oder spezialisierte Arbeitsgruppen erfüllt werden. Multiplikator:innen müssen dabei keine technischen Expert:innen sein, sollten jedoch ein grundlegendes Verständnis für die Herausforderungen der Softwareentwicklung und Bedürfnisse der Forschenden mitbringen. Ihre Fähigkeit, klar zu kommunizieren und als Knotenpunkt eines Netzwerks zu agieren, ist entscheidend für den Erfolg ihrer Rolle.

Champions sind in Abgrenzung dazu immer Fachexpert:innen. Als eine Art Vorbild- oder Botschafter:innen-Rolle sind sie engagierte Unterstützer:innen der RSE-Community und haben entweder selbst Erfahrungen als RSEs oder arbeiten eng mit ihnen zusammen. Darüber hinaus genießen sie jedoch auch Anerkennung und Glaubwürdigkeit innerhalb der Wissenschaftsgemeinschaft, und können Forschungsergebnisse, Erfolgsbeispiele und die Bedeutung von RSEng-Arbeit überzeugend vermitteln. Sie sind also Wissenschaftler:innen, die sich intrinsisch motiviert aktiv und öffentlich für die RSE-Community einsetzen und diese vertreten. Damit machen sie die Relevanz von RSEs für den wissenschaftlichen Fortschritt sichtbar, motivieren Forschende und Institutionen, die Bedeutung von Softwareentwicklung in der Forschung ernst zu nehmen und setzen sich für bessere Arbeitsbedingungen, Anerkennung und Ressourcen für RSEs ein. Beispiele für Champions sind Professor:innen oder leitende Wissenschaftler:innen, die öffentlich die Bedeutung von RSEng-Arbeit betonen; RSEs selbst und ihre Teamleitungen, die durch ihre Projekte oder Publikationen herausstechen und eine Vorbildrolle einnehmen; oder Mitarbeitende von Infrastruktureinrichtungen und Forschende, beispielsweise Software Stewards (äquivalent zu Data Stewards), „Researchers Who Code“ (Forschende mit Softwareentwicklungsfähigkeiten), Fachreferent:innen oder „embedded developers“ (äquivalent zu sogenannten „embedded librarians“)

3.2.1 Entwickler:innen von Forschungssoftware und deren Netzwerke

Erleichtert den Einstieg in existierende Communities! Um Forschende für die Community zu gewinnen, müssen die Hürden beim Einstieg minimiert werden. Viele Forschende haben Berührungängste mit Themen wie Softwareentwicklung oder betrachten sich nicht als Teil der Forschungssoftware-Community. Mögliche Lösungen liegen in persönlichem Kontakt durch benannte Ansprechpersonen (beispielsweise Champions) oder Mentoring-Programme, aber auch in digitalen Werkzeugen wie Wikis, Online-Foren oder Chatkanäle, oder der Organisation von praktischen Formaten wie Hackathons, „Advent of Code“ oder Code Katas, um den direkten Kontakt mit der Community zu fördern.

Findet und erzählt Geschichten! Durch die Präsentation von Erfolgsbeispielen von RSEs („RSE-Geschichten“) oder Fallstudien mit klaren Herausforderungen, Lösungen und Ergebnissen („Case Studies“), können Nachwuchsforscher:innen und institutionelle Entscheider:innen motiviert werden, sich für die Entwicklung von Kompetenzen in Forschungssoftware einzusetzen. RSE-Champions können diese Geschichten organisieren und Stories von den „Stars der Geschichte“ erzählen, die ihre Erfahrungen mit der Forschungssoftware teilen.

Tu Gutes und rede darüber! Durch die Nutzung von Social Media können „RSE Geschichten“ auch einer breiten Öffentlichkeit und der Forschungscommunity zugänglich gemacht werden. Die Zielgruppe für diese Geschichten sind nicht nur Forschende und Entscheidende, sondern auch Nachwuchswissenschaftler:innen, die durch diese Geschichten inspiriert werden können, sich für RSEng-Themen zu interessieren.

3.2.2 Interessenverbände

Kreiert zentrale Ressourcen! Sowohl für gut vernetzte als auch nicht-vernetzte Akteure bietet eine Übersicht über Ressourcen, Netzwerke, oder Materialien Mehrwert. Sinnvoll wäre etwa die Erstellung einer zentralen Website mit Erfolgsbeispielen (z.B. Sammlung von Policies oder Forschungssoftware) von RSE-Netzwerken.

3.2.3 Wissenschaftliche Führungskräfte

Vernetzt euch und eure Mitarbeitenden! Führungskräfte verfügen häufig über institutionelles Wissen zu vorhandenen Angeboten oder die Existenz von Expert:innen in digitalen Kompetenzen an ihrer Einrichtung. Dies sollten sie nutzen, um als Mentor:innen ihre Mitarbeitenden in Netzwerke einzubinden, vorhandene Services zu bewerben und als Multiplikator:innen für digitale Forschungskompetenzen zu wirken. Führungskräfte können zudem gezielt Champions für digitale Methoden und Forschungssoftware innerhalb ihres Teams identifizieren und fördern. Wenn Führungskräfte nicht über die entsprechenden Netzwerke verfügen, gilt es, vorhandene Netzwerke zu identifizieren.

3.2.4 Wissenschaftliche Infrastruktur

Verhelft RSE-Stories zu mehr Reichweite! Kommunikationsteams können „RSE-Geschichten“ oder „Case Studies“ von Entwickler:innen von Forschungssoftware unterstützen, indem sie sie auf Social Media und anderen Plattformen teilen oder in Form von Pressemitteilungen verbreiten. Darüber hinaus kann ein neues RSEng-Journal gegründet werden, um Erfolgs- und Schreckensgeschichten (äquivalent zu Data Scarytales³⁴) in Form von Story-basierten Veröffentlichungen und Repositorien zu veröffentlichen.

3.2.5 Fördergeber

Finanziert RSE-PR! Durch die Finanzierung von PR-Arbeit, Social-Media-Kampagnen und Content-Produktion können „RSE-Geschichten“ oder „Case Studys“ effektiv verbreitet werden.

3.2.6 Zielgruppenübergreifende Maßnahmen

Identifiziert, fördert, und vernetzt Multiplikator:innen und Champions! Einrichtungen wie Universitäten, Forschungseinrichtungen, Institute und Arbeitsgruppen sollten gezielt Personen oder Teams benennen, die als zentrale Ansprechstelle fungieren. Hierdurch werden Multiplikator:innen etabliert und ein effektives Netzwerk geschaffen, das sowohl intern als auch extern vernetzt ist. Sie sollten Champions identifizieren und fördern, und Champions und Multiplikator:innen in bestehende Netzwerke integrieren, um die Sichtbarkeit von RSEs nachhaltig zu steigern.

Kreiert klar definierte Einstiegspunkte in die RSE-Community! Institutionen sollten Einsteiger:innen direkt zu Beginn, beispielsweise ihres ersten Codingprojektes innerhalb der Forschung, explizit ansprechen und personalisierte Unterstützung anbieten: Persönlicher Kontakt durch benannte Ansprechpersonen wie Champions oder Mentoring-Programme können den Einstieg erleichtern.

3.3 Anreize und Rahmen entwickeln

Die Entwicklung von Forschungssoftware ist in das Wissenschaftssystem eingebettet und hat damit andere Rahmenbedingungen als die Entwicklung von Software in der Industrie. Hierzu zählen die derzeit noch unzureichend entwickelten Anreize für die Qualitätssicherung von Forschungssoftware und sehr heterogene Forschungssoftwarekompetenzen. Viele Forschende haben Berührungsängste mit Themen wie Softwareentwicklung oder betrachten sich nicht als Teil der Forschungssoftware-Community. Daher müssen die Hürden zum Einstieg minimiert werden. Anreize beispielsweise in Form von neuen Indikatoren für die Einbeziehung von Forschungssoftware in die Forschungsevaluation können die Sichtbarkeit von RSEng-Kompetenzen verbessern und die Qualität von Forschungssoftware fördern.

3.3.1 Entwickler:innen von Forschungssoftware und deren Netzwerke

Macht euch mit den Richtlinien vertraut oder fordert sie ein! An einigen Einrichtungen sind bereits Richtlinien zur Entwicklung und dem Umgang mit Forschungssoftware in Kraft oder in Entwicklung³⁵. Sie bieten einen Rahmen, innerhalb dessen sich die Entwickler:innen bei ihrer Arbeit bewegen (z.B. hinsichtlich Lizenzen, Anforderungen an Entwicklungspraxis etc.). Daher

³⁴Siehe beispielsweise [Gerlach et al.](#).

³⁵Beispielsweise die Muster-Richtlinie "Nachhaltige Forschungssoftware an den Helmholtz-Zentren", siehe [Task Group Forschungssoftware des Arbeitskreises Open Science der Helmholtz-Gemeinschaft](#), oder die gemeinsame Richtlinie der Fachgruppe der Gesellschaft für Informatik (GI) und der Gesellschaft für Forschungssoftware (de-RSE e.V.) in [Czerniak et al. \[2025\]](#).

müssen die Entwickler:innen diese Richtlinie kennen und umsetzen. Wo solche Richtlinien noch nicht existieren, können sie aktiv werden und deren Erarbeitung auf der Grundlage vorhandener Muster initiieren.

3.3.2 Wissenschaftliche Führungskräfte

Wertschätzt Softwarekompetenz! Digitale Kompetenzen sind essentiell in der wissenschaftlichen Arbeit. Führungskräfte sollten digitale Kompetenzen wertschätzen – nicht nur in Lippenbekenntnissen, sondern auch in Formen, die digital kompetenten Mitarbeitenden im Arbeitsleben zugutekommen. Ähnlich wie bei der Einbeziehung von Forschungssoftwarepublikationen im DFG-Antragsverfahren sollten digitale Kompetenzen auch beispielsweise in Einstellungsverfahren, bei der Evaluation von Mitarbeitenden oder in der Kommunikation im Team hervorgehoben werden.

Fördert eine gesunde Fehlerkultur! Der Erwerb und die Erweiterung von Softwarekompetenzen sind ein stetiger Prozess, der von Kollaboration profitiert und auch von *trial-and-error* geprägt ist. Um diesen Prozess zu fördern, sollten Führungskräfte eine Fehlerkultur etablieren, in der Mitarbeitende unabhängig von Hierarchieebenen Lernerfahrungen oder Fragen teilen können und auch offen mit Misserfolgen oder Fehlern umgehen können. Dabei helfen ein wertfreier und konstruktiver Umgang mit Problemen, eine Lernkultur, die Unwissen als Chance statt Defizit sieht, oder eine Praxis des „Fehler Feierns“ in der Fehler als notwendige Lernerfahrung normalisiert und zelebriert werden. Praktiken wie etwa *Code Sharing* oder *Code Peer Review* können gefördert werden.

Seid euch des Ressourcenbedarfs von Softwarekompetenzen bewusst! Führungskräfte bestimmen maßgeblich die Prioritäten in ihrem Arbeitsumfeld. Ihnen sollte bewusst sein, dass Softwarekompetenzen relevant sind und dass dafür finanzielle und personelle Ressourcen benötigt werden. Dieses sollte in der Personalplanung, bei Projektanträgen, der Zeitplanung von wissenschaftlichen Projekten oder Abschlussarbeiten, sowie bei der Aufstellung von Budgets bedacht werden.

Fordert und fördert die Weiterentwicklung von Softwarekompetenz bei den Mitarbeitenden! Durch den schnellen technologischen Wandel ändern sich die Anforderungen stetig. Daher endet der Erwerb digitaler Kompetenzen nicht mit dem Abschluss eines Studiums, sondern ist ein beständiger Prozess, für den wegen der oft hohen Arbeitsbelastung in der Wissenschaft häufig nicht genügend Zeit bleibt. Führungskräfte sollten intrinsisch motivierte Mitarbeitende beim Ausbau ihrer digitalen Kompetenzen z.B. durch frei nutzbare Arbeitszeitkontingente fördern, um ihnen Raum für ihre Weiterbildung zu geben. Allerdings sind nicht alle Mitarbeitenden motiviert, sich selbst die notwendige Softwarekompetenz anzueignen. In diesen Fällen sollten Führungskräfte Anforderungen für den Ausbau der digitalen Kompetenzen formulieren und etwa die Teilnahme an bestimmten Schulungen, insbesondere institutionalisierte Schulungen und Fortbildungsprogramme im RSEng-Bereich fordern.

Leistet Lobbyarbeit für Softwarekompetenzen! Führungskräfte sind ein Bindeglied zwischen Organisationseinheiten im wissenschaftlichen System. Die Wichtigkeit von digitalen Kompetenzen sollten sie nicht nur „nach unten“ innerhalb ihrer eigenen Mitarbeitenden propagieren, sondern auch „nach oben“ Lobbyarbeit leisten – in der Verhandlung von Stellen oder Budgets, im Review von Projektanträgen, oder im Wirken in wissenschaftlichen Gesellschaften.

Denkt RSE im On- und Offboarding mit! Für das Onboarding und Offboarding neuer Mitarbeiter:innen sollten Checklisten genutzt oder entwickelt werden, die auch die Aspekte umfassen sollten, die im Zusammenhang mit der Entwicklung und Nutzung von Forschungssoftware stehen, wie etwa Hinweis auf Richtlinien.

3.3.3 Fördergeber

Unterstützt den Aufbau von Softwarekompetenz in den geförderten Projekten durch entsprechende Ressourcen! Um die Qualität der in den Projekten entwickelten Forschungssoftware zu sichern, muss es möglich sein, dass auch Ressourcen für die Weiterbildung der Projektmitarbeiter:innen beantragt werden können, um die notwendigen digitalen Kompetenzen im Projekt zu verankern. Dies kann von Kursen für einzelne Entwickler:innen für erforderliche spezielle Softwarekompetenzen bis hin zu allgemeineren digitalen Kompetenzen für das ganze Projekt reichen, um eine gemeinsame Basis für die Arbeit im Projekt zu haben.

Stellt sicher, dass genügend Softwarekompetenz im Review vorhanden ist! Auch im Begutachtungsprozess muss stärker verankert werden, dass die Entwicklung digitaler Kompetenzen in der Projektarbeit Raum hat. Dafür muss gesichert werden, dass entsprechende Expert:innen für das Thema im Gremium vorhanden sind.

3.3.4 Einrichtungseleitungen

Schafft Karrierewege und Perspektiven für Forschende mit RSEng-Kompetenzen!

Entwickelt inklusive Lebenslauf-Muster, die die Diversität von Research-Outputs (inklusive Daten- und Softwarepublikationen) anerkennen.

3.3.5 Zielgruppenübergreifende Aufgaben

Erhöht die subjektive Anziehungskraft! Um Forschungssoftwarekompetenzen attraktiver für Nachwuchsforscher:innen, institutionelle Entscheider:innen, und „schlafende“ RSEs zu machen, können wissenschaftliche Infrastruktur, Fördergeber, Interessenvertretungen oder Entwickler:innen von Forschungssoftware attraktive Formulierungen (z.B. „Researchers Who Code“) benutzen, Events organisieren, Personen mit Anziehungskraft einbinden, oder Erfolgsgeschichten aus der Community teilen (siehe 3.2).

Erhöht die objektive Anziehungskraft! Führt Preise und Badges ein, um (Spitzen-)Leistung zu fördern und zu honorieren. Entwickelt Mentoring-Programme und Aktivitätspläne, um RSE Champions einzubinden, und ihnen Anerkennung durch Preise, Förderungen oder öffentliche Plattformen verschaffen. Für die Finanzierung dieser Initiativen können institutionelle Mittel für Preise oder Mentoring-Programme genutzt werden. Zudem stehen individuelle Förderprogramme von Forschungsorganisationen wie der VW-Stiftung, DFG- oder EU-Projekten zur Verfügung.

Nutzt RSEng-Kompetenzen als Evaluationsbestandteil! Zur Verbesserung der Rahmenbedingungen gehört auch die Einbeziehungen von Forschungssoftware in das Research Assessment (z.B. bei Evaluationen). Aber auch bei Personaleinstellungen, Beförderungen, oder internen Anreizsystemen z.B. in Form von Boni für Teams, die RSEs effektiv einbinden, können gezielt Impulse gesetzt werden, um Forschungssoftwarekompetenzen sichtbar zu machen. Die quantitativen und qualitativen Key Performance Indicators (KPI) und Evaluationsmechanismen sollten dabei explizit gemacht und veröffentlicht werden.

Entwickelt, implementiert, und teilt Policies! Policies zur Unterstützung der Forschungssoftwarekompetenzen sind notwendig, da moderne Forschung zunehmend auf Software angewiesen ist. Sie fördern Standards, ermöglichen nachhaltige Karriereperspektiven und sichern die Qualität sowie Reproduzierbarkeit wissenschaftlicher Ergebnisse. In enger Zusammenarbeit mit RSEs und anderen relevanten Communities sollten Arbeitsgruppen, Universitäten, Forschungseinrichtungen, oder Forschungsverbünde Richt- und Leitlinien oder Checklisten für RSEng-Kompetenzen entwickeln, implementieren, und zur Nachnutzung intern und extern veröffentlichen. Dies umfasst beispielsweise die Definition von RSE-Rollen und Verantwortlichkeiten in der Code-Arbeit, die klare Festlegung von Verantwortlichkeiten für Forschungssoftware in Projekten und Arbeitsgruppen und die Nennung von Ressourcen. Dazu können entweder bereits existierende Forschungssoftware-Policies genutzt werden, oder neue institutionelle Forschungssoftware-Policies erstellt (bzw. existierende Open-Science-/Forschungsdaten-Policies ergänzt) und darauf aufbauende konkrete Handlungsempfehlungen in einer Leitlinie oder Richtlinie verfasst werden. Die resultierenden Policies sollten regelmäßig evaluiert und aktualisiert werden.

Richtet zentrale Anlaufstellen für Software-Policies ein! mit Ansprechpersonen für Fragen zu Software-Policies. Die Implementierung einer Forschungssoftwarepolicy sollte von lokalen Unterstützungsangeboten zur Umsetzung dieser Policy begleitet werden. Fehlen solche niederschweligen Angebote, besteht die Gefahr, dass Forschende die Inhalte nicht wirklich in ihre Arbeitsweise integrieren, sondern diese nur *pro forma* erfüllen.

4 Zusammenfassung und Ausblick

Digitale Kompetenzen im Allgemeinen und Forschungssoftwarekompetenzen im Speziellen werden in der modernen Wissenschaft immer relevanter. Insbesondere mit dem eigenen Anspruch an die gu-

te wissenschaftliche Praxis und damit einhergehende Reproduzierbarkeit von softwareunterstützter Forschung.

Daher haben bereits viele verschiedene Organisationen für sich festgestellt, dass in der Fortbildung und Unterstützung von Wissenschaftler:innen Handlungsbedarf besteht. Die Impulsvorträge haben zum einen die Diversität der handelnden Organisationen gezeigt und zum anderen einen Überblick über Maßnahmen und Angebote gegeben, die bereits jetzt umgesetzt werden und von weiteren Organisationen umgesetzt werden können.

Die hier formulierten Handlungsempfehlungen wurden von einer interdisziplinären und diversen Gruppe von Menschen aus unterschiedlichen Bereichen und Hierarchieebenen im deutschen wissenschaftlichen System entwickelt. Damit konnten wir die vielschichtige Fragestellung nach der Förderung von Forschungssoftwarekompetenzen aus zahlreichen Blickwinkeln betrachten. Unsere Handlungsempfehlungen bieten konkrete Ansatzpunkte für eine große Zahl von Zielgruppen - vom einzelnen RSE bis hin zum Förderer sollten Akteure darin konkrete Maßnahmen finden, die sie unmittelbar oder langfristig umsetzen können.

Zugleich erheben wir keinen Anspruch auf Vollständigkeit: Das Feld Forschungssoftware ist im beständigen Wandel, und so werden sich auch Handlungsempfehlungen über die Zeit verändern müssen. In diesem Sinne ist dieses Dokument versioniert, sodass es zukünftigen Anforderungen angepasst werden kann.

Ebenso wie die Diskussion selbst sind auch die Förderung von Forschungssoftware und Forschungssoftwarekompetenzen beständige Prozesse. Dementsprechend werden manche der Handlungsempfehlungen bereits kurzfristige Erfolge bieten, während andere auf langfristige Veränderungen hinauslaufen. Daher sollte diese Langfristigkeit und Nachhaltigkeit bei der Umsetzung der Handlungsempfehlungen beachtet werden.

Wie in der Einleitung beschrieben, empfinden wir die hier aufgeschriebenen Impulse und Handlungsempfehlungen als Debattenbeitrag innerhalb einer größeren Diskussion. Diese erste Version zeigt das Ergebnis der Diskussion zwischen den Teilnehmer:innen des Workshops in Hannover und soll eine Diskussionsgrundlage für weitere Diskussionen in den einzelnen Fachbereichen der Teilnehmer:innen sein.

Dementsprechend geben wir als Ausblick auch eine finale Handlungsempfehlung:

Nehmt aktiv an Diskussion und Austausch teil! Insbesondere haben wir im Plenum mit den anderen Workshops im Rahmen der Themenwoche festgestellt, dass sowohl eine breite Diskussion über fächerübergreifende Grundlagen und Gemeinsamkeiten als auch fachspezifische Bedarfe im Bereich digitaler Kompetenzen nötig sind.

Literatur

- Hartwig Anzt, Felix Bach, Stephan Druskat, Frank Löffler, Axel Loewe, Bernhard Y. Renard, Gunnar Seemann, Alexander Struck, Elke Achhammer, Pioush Aggarwal, Franziska Appel, Michael Bader, Lutz Brusch, Christian Busse, Gerasimos Chourdakis, Piotr wojciech Dabrowski, Peter Ebert, Bernd Flemisch, Sven Friedl, Bernadette Fritzsche, Maximilian D. Funk, Volker Gast, Florian Goth, Jean-Noel Grad, Jan Hegewald, Sibylle Hermann, Florian Hohmann, Stephan Janosch, Dominik Kutra, Jan Linxweiler, Thilo Muth, Wolfgang Peters-Kottig, Fabian Rack, Fabian H.C. Raters, Stephan Rave, Guido Reina, Malte Reißig, Timo Ropinski, Joerg Schaarschmidt, Heidi Seibold, Jan P. Thiele, Benjamin Uekermann, Stefan Unger, and Rudolf Weeber. An environment for sustainable research software in Germany and beyond: current state, open challenges, and a call for action. *F1000Research*, 9(295), 2021. doi: 10.12688/f1000research.23224.2. URL <https://doi.org/10.12688/f1000research.23224.2>.
- Cora Assmann. Bedeutung von Forschungssoftwarekompetenzen im Forschungsdatenmanagement. Code for Science oder: Bessere Forschung durch bessere Forschungssoftwarekompetenzen, Dec 2024. URL <https://doi.org/10.5281/zenodo.14287152>.
- Katarzyna Biernacka, Ron Dockhorn, Claudia Engelhardt, Kerstin Helbig, Juliane Jacob, Tereza Kalová, Adienne Karsten, Kristin Meier, Andreas Mühlichen, Janna Neumann, Britta Petersen, Benjamin Slowig, Ute Trautwein-Bruns, Jeanne Wilbrandt, and Cord Wiljes. *Train-the-Trainer-Konzept zum Thema Forschungsdatenmanagement*. Zenodo. URL <https://doi.org/10.5281/zenodo.10122152>.
- Andreas Czerniak, Adrian Ehrenhofer, Bernadette Fritzsche, Maximilian Funk, Florian Goth, Reiner Hähnle, Carina Haupt, Marco Konersmann, Jan Linxweiler, Frank Löffler, Alexander Lüpkes, Bernhard Rumpe, Ina Schieferdecker, Tobias Schlauch, Robert Speck, Alexander Struck, Jan Philipp Thiele, Matthias Tichy, and Inga Ulusoy. GI- und DE-RSE Muster-Leitlinie zur effizienten Entwicklung von Forschungssoftware, 2025. URL <https://dl.gi.de/handle/20.500.12116/45664>.
- Deutsche Forschungsgemeinschaft. Umgang mit Forschungssoftware im Förderhandeln der DFG, 2024. URL <https://doi.org/10.5281/zenodo.13919790>.
- Deutsche Forschungsgemeinschaft, Ausschuss für Wissenschaftliche Bibliotheken und Informationssysteme. Digitale Forschungspraxis und kooperative Informationsinfrastrukturen. Ein Diskussionspapier der Deutschen Forschungsgemeinschaft zu Förderung und Finanzierung wissenschaftlicher Informationsinfrastrukturen. URL <https://zenodo.org/doi/10.5281/zenodo.14621979>.
- Zeki Mustafa Doğan. Forschungssoftware in Bibliotheken. Code for Science oder: Bessere Forschung durch bessere Forschungssoftwarekompetenzen, Dec 2024. URL <https://doi.org/10.5281/zenodo.14273293>.
- Roman Gerlach, Jessica Rex, Kevin Lang, Nadine Neute, Annett Schröter, Volker Schwartz, Cora Assmann, and Anne Lehmann. Research Data ScaryTales. URL <https://doi.org/10.5281/zenodo.4066679>.
- Florian Goth, Renato Alves, Matthias Braun, Leyla Jael Castro, Gerasimos Chourdakis, Simon Christ, Jeremy Cohen, Fredo Erxleben, Jean-Noël Grad, Magnus Hagdorn, Toby Hodges, Guido Juckeland, Dominic Kempf, Anna-Lena Lamprecht, Jan Linxweiler, Frank Löffler, Michele Martone, Moritz Schwarzmeier, Heidi Seibold, Jan Philipp Thiele, Harald von Waldow, and Samantha Wittke. Foundational Competencies and Responsibilities of a Research Software Engineer. *F1000Research*, 2024. doi: 10.12688/f1000research.157778.1. URL <https://doi.org/10.12688/f1000research.157778.1>.
- Wilhelm Hasselbring, Stephan Druskat, Jan Bernoth, Philine Betker, Michael Felderer, Stephan Ferenz, Anna-Lena Lamprecht, Jan Linxweiler, and Bernhard Rumpe. Toward Research Software Categories, 2024. URL <https://arxiv.org/abs/2404.14364>.
- Dominic Kempf, Yves Vincent Grossmann, Frank Löffler, and Jan Linxweiler. Öffentliche Dokumente der 2024 Themenwoche der VW-Stiftung zum Thema "Code for Science oder: Bessere Forschung durch bessere Forschungssoftwarekompetenzen", 2024. URL <https://doi.org/10.5281/zenodo.14265405>.

- Sarah Neuwirth. Forschungssoftwarekompetenzen und HPC: Schlüssel zur Innovation und Effizienz. Code for Science oder: Bessere Forschung durch bessere Forschungssoftwarekompetenzen, Dec 2024. URL <https://doi.org/10.5281/zenodo.14273217>.
- Peter Schmidt. Software Kompetenzen in der Wissenschaft - Die Tagung in Hannover, Dezember 2024, 2025. URL <https://www.buzzsprout.com/1326658/episodes/16483665>.
- Heidi Seibold. Forschungssoftwarekompetenzen & Open Science. Code for Science oder: Bessere Forschung durch bessere Forschungssoftwarekompetenzen, Dec 2024. URL <https://doi.org/10.5281/zenodo.14266459>.
- Task Group Forschungssoftware des Arbeitskreises Open Science der Helmholtz-Gemeinschaft. Muster-Richtlinie Nachhaltige Forschungssoftware an den Helmholtz-zentren. pages 432 KB, 13 pages. doi: 10.2312/OS.HELMHOLTZ.007. URL https://gfzpublic.gfz-potsdam.de/pubman/item/item_4906899.
- Volkswagen-Stiftung. Themenwoche Digitale Kompetenzen in der Wissenschaft, 2024. URL <https://www.volkswagenstiftung.de/de/foerderung/foerderangebot/themenwoche-digitale-kompetenzen-der-wissenschaft-beendet>.
- Philipp Wieder. Impuls zur Bedeutung von Forschungssoftware für wissenschaftliche Rechenzentren. Code for Science oder: Bessere Forschung durch bessere Forschungssoftwarekompetenzen, Dec 2024. URL <https://doi.org/10.5281/zenodo.14730918>.
- Lorenzo Zanon. Impuls zu: Training in HPC. Code for Science oder: Bessere Forschung durch bessere Forschungssoftwarekompetenzen, Dec 2024. URL <https://doi.org/10.5281/zenodo.14273082>.