Umgang mit Forschungssoftware an der Universität Stuttgart

Positionspapier

Bernd Flemisch (IWS), Sibylle Hermann (UB/SimTech), Christian Holm (ICP), Miriam Mehl (IPVS), Guido Reina (VISUS), Benjamin Uekermann (IPVS), David Boehringer (TIK), Thomas Ertl (VISUS), Jean-Noël Grad (ICP), Dorothea Iglezakis (FoKUS), Alexander Jaust (IPVS), Timo Koch (IWS), Anett Seeland (FoKUS), Rudolf Weeber (ICP), Florian Weik (ICP), Kilian Weishaupt (IWS)

Dieses Positionspapier ist aus Sicht von Forschenden an der Universität Stuttgart geschrieben. Wir empfehlen die Einrichtung einer Organisationseinheit „Forschungssoftware-Entwicklung“ an der Universität Stuttgart und eines daran angegliederten Stellenpools von Research Software Engineers (RSEs). Dazu schlagen wir Maßnahmen zur Schaffung und Finanzierung entsprechender neuer RSE-Stellen, zur Integration bestehender Stellen sowie zur Gewinnung und Förderung geeigneter Personen vor. RSEs sind Personen, die sich um Konzeption, Organisation, Implementierung, Testen, Dokumentation und Wartung von Forschungssoftware kümmern. Die institutionelle Förderung von Forschungssoftware-Entwicklung ist notwendig, da die Bedeutung von Software für die Forschung und Anforderungen an die entsprechende Software, u.a. durch die DFG, stetig zunimmt, siehe Abschnitt 1. In Abschnitt 2 erläutern wir, warum der aktuelle Zustand der Forschungssoftware-Entwicklung diesen Anforderungen nicht gerecht wird. Aufbauend auf den in Abschnitt 3 geschilderten grundlegenden Anforderungen an Forschungssoftware und Research Software Engineers schlagen wir in Abschnitt 4 konkreten Maßnahmen und deren Umsetzung vor. Durch diese kann die Universität Stuttgart ihre Forschenden von zentraler Stelle aus umfassend unterstützen und eine Vorreiterrolle in diesem Bereich übernehmen.

1 Hintergrund und Motivation

Ein Großteil des wissenschaftlichen Erkenntnisgewinns basiert heutzutage auf der Anwendung von Forschungssoftware, also auf Software, die von Forschenden für wissenschaftliche Zwecke entwickelt wird. Ein entsprechend erzieltes Resultat kann nur dann verstanden, validiert und reproduziert werden, wenn die zugrunde liegende Forschungssoftware nachhaltig entwickelt und getestet wird sowie nachnutzbar ist. Die DFG hat die Wichtigkeit von Forschungssoftware und deren Entwicklung in ihrem aktuellen Kodex „Leitlinien zur Sicherung guter wissenschaftlicher Praxis“[1] erkannt, unter anderem in den folgenden drei aufgeführten Leitlinien, siehe auch [2]:

- Leitlinie 7: Phasenübergreifende Qualitätssicherung. Die DFG fordert: „Wenn wissenschaftliche Erkenntnisse öffentlich zugänglich gemacht werden [...], werden stets die angewandten Mechanismen der Qualitätssicherung dargelegt.“ Und erläutert: „Qualitätssicherung bezieht sich insbesondere auf [...] die Auswahl und Nutzung von Forschungssoftware, deren Entwicklung und Programmierung“ sowie „Der Quellcode von öffentlich zugänglicher Software muss persistent, zitierbar und dokumentiert sein.“
- Leitlinie 13: Herstellung von öffentlichem Zugang zu Forschungsergebnissen. Daraus: „Ist eine Entscheidung, Ergebnisse öffentlich zugänglich zu machen, erfolgt, beschreiben Wissenschaftlerinnen und Wissenschaftler diese vollständig und nachvollziehbar. Dazu gehört es auch, [...] die eingesetzte Software verfügbar zu machen [...]. Selbst programmierte Software wird unter Angabe des Quellcodes öffentlich zugänglich gemacht.“
- Leitlinie 17: Archivierung. Daraus: „Wissenschaftlerinnen und Wissenschaftler sichern öffentlich zugänglich gemachte [...] Forschungsergebnisse sowie [...] die eingesetzte Forschungssoftware [...].“ Weiter: „Hochschulen [...] stellen sicher, dass die erforderliche Infrastruktur vorhanden ist, die die Archivierung ermöglicht.“

[1] Deutsche Forschungsgemeinschaft, Leitlinien zur Sicherung guter wissenschaftlicher Praxis - Kodex, September 2019.
[2] Stellungnahme der FDM-AG der Universität Stuttgart zu den Leitlinien zur Sicherung guter wissenschaftlicher Praxis (GWP) der DFG vom September 2019.

Um künftig Fördermittel durch die DFG erhalten zu können, muss die Universität Stuttgart diese drei und auch die anderen 16 Leitlinien rechtsverbindlich umsetzen. Dies kann nur gelingen, wenn sie die Wichtigkeit von Forschungssoftware anerkennt und ihre Entwicklung und Anwendung fördert. Wir wollen unsererseits als Forschende und insbesondere Forschungssoftware-Entwickelnde aktiv zur Einrichtung entsprechender Strukturen und Prozesse beitragen. Durch die vorhandene Expertise sowohl seitens der Forschenden als auch der Infrastruktur-Einrichtungen sind wir davon überzeugt, dass die Universität Stuttgart eine Vorreiterrolle im wissenschaftsunterstützenden Bereich der Entwicklung nachhaltiger Forschungssoftware spielen kann und muss.

2 Status Quo

Große und weltweit genutzte Softwareprojekte haben meistens einen großen Förderer im Hintergrund. Beispiele sind der LAMMPS Molecular Dynamics Simulator[3], der hauptsächlich vom US Department of Energy (DOE) durch Projekte (OASCR, OBER, ASCI, LDRD, Genomes-to-Life) gefördert wurde. Weitere prominente Beispiele sind der NAMD Molekulardynamikcode[4] und das Visualisierungstool VMD[5], deren Entwicklung vom National Institute of Health (NIH) und der National Science Foundation (NSF) mit Geldmitteln (> 1 Mio. US-Dollar/Jahr) finanziell unterstützt wurden.

In Europa fand erst in den letzten fünf Jahren eine dedizierte Förderung von Software statt. Als Beispiele seien die Centres of Excellence[6] genannt, die im Rahmen des Horizon 2020 Programms der EU über fünf Jahre unterstützt wurden. Diese relativ großen Zentren bekamen Geldmittel in Höhe von etwa 1 Mio. Euro/Jahr, und fördern die gemeinsame Entwicklung von ausgewählten Softwarebibliotheken und Programmpaketen. Auf Grund dieser und anderer projektbasierter Förderung haben es beispielsweise große Community-Codes geschafft, ein bis drei RSEs über mehrere Jahre für ihre Open-Source-Software zu beschäftigen, siehe etwa GROMACS[7].

In Deutschland gibt es begrenzte und zum Teil sehr spezielle Fördermöglichkeiten für Softwareentwicklung, zum Beispiel den durch das BMBF finanzierten Prototype Fund. Dieser geht aktuell in die letzte Runde und ist auch auf neue Ideen oder neue Erweiterungen sowie eine Laufzeit von sechs Monaten eingeschränkt. Diese Initiative hat zudem das inhärente Problem, dass Software entsteht, deren Nachhaltigkeit nur bei schnellem, großem Erfolg und durch die geforderte Open-Source-Eigenschaft und der Bildung einer Community gewährleistet werden kann.

Mit der im November 2016 erfolgten Ausschreibung „Nachhaltigkeit von Forschungssoftware“ verschrieb sich die DFG erstmals explizit der Förderung von Forschungssoftware-Entwicklung[8]. Die Ausschreibung war ungefähr siebenfach überzeichnet. Von den schließlich gut zwanzig geförderten Projekten sind vier an der Universität Stuttgart angesiedelt:

- ICP (C. Holm): ESPResSo
  - Teilchendynamik mit Hydrodynamik und Elektrostatik Kopplung.
  - Wird zur Zeit durch den Projektbearbeiter, drei weitere Kern-Entwickler und acht Doktorand*innen/ Masterstudierende/HiWis weiterentwickelt.
  - DFG-Projekt 391126171.
- VISUS (T. Ertl, G. Reina): MegaMol
  - Visualisierungsframework mit Fokus auf Partikeldatensätzen.
  - Wird zur Zeit durch den Projektbearbeiter, einen Entwickler aus einem MWK-Projekt, sieben Doktoranden, einen PostDoc aus dem Mittelbau und einen HiWi weiterentwickelt.
  - DFG-Projekt 391302154.

[3] https://lammps.sandia.gov
[4] https://www.ks.uiuc.edu/Research/namd/
[5] https://www.ks.uiuc.edu/Research/vmd/
[6] https://www.nomad-coe.eu/externals/european-centres-of-excellence
[7] http://www.gromacs.org
[8] Nachhaltigkeit von Forschungssoftware.

- IPVS (M. Mehl, B. Uekermann), zusammen mit der TUM (H.-J. Bungartz): preCICE
  - Parallele Kopplung von Black-Box-Einphysiklösern.
  - Wird zur Zeit durch den Projektbearbeiter, einen weiteren Doktoranden an der Universität Stuttgart (auf Institutsstelle), drei Doktoranden an der TU München (finanziert aus anderen Projekten), einen Postdoc in Eindhoven und mehrere HiWis und Masterstudenten weiterentwickelt.
  - DFG-Projekt 391150578.
- IWS (B. Flemisch), TIK (D. Boehringer), UB (S. Hermann): DuMux
  - Mehrphasen-Hydrodynamik in porösen Medien.
  - Wird zur Zeit durch einen Projektbearbeiter und rund zwanzig Doktoranden, PostDocs und Professoren weiterentwickelt.
  - DFG Projekt 391049448.

Auch wenn wir uns als Begünstigte glücklich über die temporäre Förderung schätzen dürfen, so können wir mit den Fördermitteln nur thematisch und zeitlich begrenzt arbeiten und längst nicht alle Anforderungen an eine nachhaltige Entwicklung und Bereitstellung unserer Programmpakete abdecken. Die maßgeblichen Beiträge hierzu werden weiterhin als Nebeneffekt klassischer Forschungsprojekte erzielt.

Die folgende vier Punkte erschweren die nachhaltige Entwicklung und Bereitstellung von Forschungssoftware:

1. **Fehlen langfristiger Finanzierungsmöglichkeiten und zentraler Stellen**: Die Entwicklung von Forschungssoftware ist ein kontinuierlicher Prozess, der stark von der entsprechenden Verfügbarkeit und dem Know-How der Entwickelnden abhängt. Mit ausschließlich befristeten Stellen kann die erforderliche Kontinuität nicht erbracht werden. Durch fehlende zentrale Stellen und Koordination werden Synergieeffekte bisher verhindert. Das Kompetenzzentrum für Forschungsdaten der Universität Stuttgart FoKUS beschäftigt sich im Rahmen des DFG-Projektes ReSUS mit der Bereitstellung und Lizenzierung von Forschungssoftware, hat aber bisher keine Kapazitäten, um die Koordination von RSEs oder auch einzelne RSE-Aufgaben zu übernehmen.
2. **Reduzierter Forschungsoutput der Promovierenden**: Kleineren Softwarepaketen bleibt meist nichts anderes übrig, als die Softwareentwicklung von Promovierenden und Masterstudierenden im Rahmen von Förderprojekten durchführen zu lassen. Diese Personen haben jedoch alle wissenschaftlichen Output als Ziel, und nicht die Softwareentwicklung per se. Diese Struktur führt zu Software, die aufgrund fehlender Nachhaltigkeit in der Regel nur eine Promovierendengeneration überlebt. Daraus entsteht ein vervielfältigter Aufwand, da die Software für bereits gelöste Probleme in diesen Fällen erneut geschrieben werden muss. Das wiederum reduziert den Forschungsoutput der Promovierenden.
3. **Fehlende Ausbildung und Erfahrung in der Softwareentwicklung**: Wissenschaftler*innen haben in der Regel eine mangelnde Ausbildung und Erfahrung in der Softwareentwicklung. Sie sind primär Forschende und als solche ausgebildet. Programmierung wird entweder nur durch grundlegende Lehrangebote oder im Selbststudium erlernt[9] und befähigt nicht zum Entwurf und der Entwicklung von Software, welche den Kriterien der Nachhaltigkeit entsprechen kann. Die Autor*innen möchten im Speziellen darauf hinweisen, dass dies selbst für Informatik-Absolvent*innen gilt.
4. **Fehlende Infrastruktur für das Entwickeln von Forschungssoftware**: Nachhaltige Softwareentwicklung profitiert stark von Diensten zum Entwickeln, Testen und Validieren von Software. Im Speziellen fehlen Hardware und personelle Unterstützung für Continuous Integration/Continuous Delivery (CI/CD), welche eine breite Unterstützung für Betriebssysteme bietet und sowohl mit der GitHub-Enterprise-Installation am TIK als auch mit externen Lösungen (GitHub, GitLab, Self-Hosted) verwendet werden kann. Dieser Ansatz kann auch Arbeitsabläufe beschleunigen, da die zur Zeit genutzten CI/CD-Systeme mit starken Einschränkungen bezüglich Rechenlast, Rechenzeit und Parallelität einhergehen.

[9] Philippe et al. 2019, *softwaresaved/international-survey: Public release for 2018 results* (Version 2018-v.1.0.2). Zenodo. 10.5281/zenodo.2585783

Wir werden im Folgenden darlegen, wie wir diesen Hindernissen begegnen und damit die nachhaltige Entwicklung und Bereitstellung von Forschungssoftware begünstigen können.

3 Anforderungen

Ausgehend von Großbritannien bildete sich der Dachverband der „Research Software Engineers International“, der das neue Berufsbild vertritt und fördert. Einige deutsche Universitäten haben bereits zentrale RSE-Gruppen geschaffen, zum Beispiel die Universität Göttingen im Rahmen der Göttingen eResearch Alliance oder die Universität Heidelberg mit einem Scientific Software Center. Wir skizzieren zunächst einige Anforderungen an die Forschungssoftware und anschließend kurz entsprechende Ansprüche an die Research Software Engineers.

Anforderungen an die Forschungssoftware: Ein übergreifendes Kriterium, um nachhaltige Forschungssoftware zu identifizieren, ist die (Nach)Nutzbarkeit der Software. Dies beinhaltet die möglichst einfache Installation, Betrieb und Bedienung und ist unabhängig vom Einsatzzweck und Fachbereich. Die Nutzbarkeit ist wichtig, damit die Software weiterhin für die Nutzer relevant bleibt und um neue Nutzer zu erschließen. Ein Teil der Hürden kann durch Bereitstellung von Dokumentation abgemildert werden, doch kontinuierliche Veränderungen der Infrastruktur, Hardware und des Software-Ökosystems machen potentiell wiederkehrende Wartungsarbeiten an der Software notwendig, auch wenn diese nicht weiterentwickelt wird, sondern nur zum Zwecke der Reproduzierbarkeit lauffähig bleiben soll.

Anforderungen an die Research Software Engineers: RSEs können als zentral verwaltete Anlaufstelle Wissenschaftler*innen bei der Erstellung ihrer Forschungssoftware unterstützen, da sie über Erfahrung und Fachwissen verfügen, welche über Jahre aufgebaut werden müssen und Promovierenden in der Regel nicht einmal nach mehreren Jahren der Forschungsarbeit zur Verfügung stehen. Tätigkeiten, mit denen RSEs Wissenschaftler*innen unterstützen können, sind beispielsweise die individualisierte Betreuung von Softwareprojekten, Bereitstellung und Administration softwarebezogener Dienste sowie die Organisation und Durchführung von Schulungen. Sie werden in Abschnitt 4.3 detaillierter vorgestellt.

4 Maßnahmen und Umsetzung

In diesem Abschnitt werden Lösungsideen skizziert, um den Herausforderungen aus Abschnitt 2 und den Anforderungen aus Abschnitt 3 gerecht zu werden. Wir stellen zunächst die grundlegende Konzeption und Struktur vor, anschließend die Einbettung bestehender und Einrichtung neuer Stellen und ihre Finanzierung. Dem folgen geplante Aufgaben und schließlich gehen wir auf Kommunikation und Dissemination ein.

4.1 Konzeption und Struktur

Organisationseinheit und Stellenpool

Wir befürworten die Einrichtung einer Organisationseinheit „Forschungssoftware-Entwicklung“ am FoKUS und eines zugeordneten Stellenpools von „Research Software Engineers“ (RSEs). Als forschungsstarke Universität können wir der Wichtigkeit der von uns entwickelten und genutzten Forschungssoftware damit Ausdruck verleihen. Das FoKUS erscheint uns als natürlicher Platz zur Bündelung der Aktivitäten und kann in diesem Zuge zum „Kompetenzzentrum für Forschungsdaten und -software“ ausgebaut werden. Der Organisationseinheit steht beratend ein RSE Advisory Board zur Seite, das aus Stuttgarter Forschenden besteht.

Ausbildung und Karriereweg einer*s RSEs

Angesichts der Tatsache, dass RSE noch ein sehr junges Berufsbild ist, dem kein fertiger Werkzeugkasten mit bereits in einem Universitätsstudium gelehrten Methoden zur Verfügung steht, müssen erfahrenere Wissenschaftler*innen als RSEs gewonnen werden; idealerweise solche, die in ihrer Promotion sehr nahe an

einer Forschungssoftware gearbeitet haben. Je nach Person und Tätigkeit bieten sich im Wesentlichen drei Karrierewege an, die wir mit unserem Konzept unterstützen wollen. Zum ersten soll durch kontinuierliche Weiterqualifikation und einer damit einhergehenden Übernahme zukunftsorientierter und verantwortungsvoller Tätigkeiten der Verbleib auf Dauerstellen attraktiv gestaltet werden. Zum zweiten bieten sich durch RSE-Tätigkeiten in Kooperation mit unterstützten Forschenden Möglichkeiten zum Publizieren an, was die wissenschaftliche Reputation einer*s RSEs steigert und eine weitere Karriere in der Wissenschaft ermöglicht. Schließlich können zum Beispiel Ingenieur*innen durch eine mehrjährige RSE-Tätigkeit den letzten Schliff Softwareentwicklung für eine erfolgreiche Laufbahn in der Wirtschaft erhalten.

Ausgestaltung des Stellenpools

Eine wichtige Rolle werden zu erarbeitende Richtlinien für die Ausgestaltung und Finanzierung des Stellenpools spielen, siehe dazu auch den nächsten Abschnitt. Diese sind eng gekoppelt an ein zu erstellendes Konzept für die Priorisierung und Aufwandsabschätzung und -verteilung zur angemessenen Zuteilung von RSEs auf Arbeitsplätze und Projekte. Wir sehen hier Bedarf und Einsatzmöglichkeiten sowohl für universitätszentrale Stellen als auch für die Ansiedlung in größeren Arbeitsgruppen oder auch auf Fakultätsebene. Naturgemäß können sich die Inhaber*innen zentraler Stellen um disziplinagnostische übergreifende Themen wie Versionsverwaltung, automatisiertes Testen und Containerisierung kümmern und Brücken zwischen einzelnen Disziplinen und Arbeitsgruppen schlagen. Ein*e einer Arbeitsgruppe zugehörige*r RSE kann vorrangig für die in der Gruppe entwickelte Software zuständig sein und zusätzlich für kleinere fachnahe Gruppen Support leisten. Hier lässt sich von entsprechendem Domänenwissen der Kollegen profitieren, so würde zum Beispiel ein*e RSE für wissenschaftliches Rechnen eher am IPVS angesiedelt als ein*e Visualisierungs-RSE, die*der naturgemäß am VISUS beschäftigt wäre. Von zentraler Wichtigkeit ist die Vernetzung der RSEs über die Organisationseinheit am FoKUS, die einen entsprechenden Austausch und Wissenstransfer gewährleistet.

Prinzipiell benötigt der Stellenpool eine gute Mischung aus erfahrenen und langfristig gebundenen RSEs und regelmäßigen neuen Mitarbeitenden. Während erstere die Nachhaltigkeit der Softwareentwicklung und bereitgestellter Werkzeuge und Dienste garantieren, ermöglichen neue Mitarbeitende mit der sich schnell verändernden Technologie Schritt zu halten. Ein regelmäßiger Wechsel von Wissenschaftlern aus den Forschungsgruppen heraus in den RSE-Stellenpool verstärkt zudem die Vernetzung beider Gruppen.

4.2 Stellen und Finanzierung

Wir stellen zunächst die grundsätzlichen Finanzierungsmöglichkeiten für RSE-Stellen vor und schlagen anschließend nächste Schritte für die Einrichtung solcher Stellen an der Universität Stuttgart vor.

Finanzierungsmöglichkeiten für RSEs

Wir sehen vier grundsätzliche Möglichkeiten für die Finanzierung von RSE-Stellen, die wir im Folgenden kurz erläutern wollen:

1. Der Overhead von Drittmittelprojekten, also der Zukunftsfonds,
2. explizit beantragte Personenstunden in Drittmittelprojekten,
3. gezielte RSE-Ausschreibungen und
4. Haushaltsstellen in Arbeitsgruppen.

Von äußerster Wichtigkeit ist für uns hierbei eine Auflösung des Konflikts zwischen zeitlich begrenzten Forschungsgeldern und der Notwendigkeit für unbefristete Stellen, um konkurrenzfähig zu anderen Universitäten und Forschungseinrichtungen sowie der Industrie sein zu können. Erfahrung ist zudem eine essenzieller Bestandteil von Software Engineering, welcher langfristige Beschäftigung unabdingbar macht. Durch Pooling der Stellen und Mittel soll ermöglicht werden, unbefristete Stellen durch wechselnde und gemischte Quellen zu finanzieren. Hier ist zu bedenken, dass die Konzepte zur Definition von Stellenbeschreibungen und die Bestimmung von Entgeltgruppen unter Umständen nicht mit dem bestehenden Tarifvertrag vereinbar sind

und angesichts der Konkurrenz zur Industrie angepasst werden müssen. Da die Stellen zur Infrastruktur zählen, kommt das Wissenschaftszeitvertragsgesetz nicht zur Anwendung.

1. Förderorganisationen vergeben einen Teil der direkten Projektmittel zusätzlich als Overhead, die zwischen der Universität und dem antragstellenden Institut aufgeteilt werden. Wir schlagen vor, einen geringen Prozentsatz dieses Overheads zur dauerhaften Finanzierung einer RSE-Koordinationsstelle und zur Anschubfinanzierung einer weiteren zentralen RSE-Stelle zu verwenden, siehe dazu den nächsten Abschnitt. Bei den über die letzten Jahre hinweg realisierten Drittmittelerträgen alleine aus DFG-Projekten (rund 50 Mio. EUR jährlich) würden rund 100 TEUR dauerhafter Finanzierungsbedarf nur 0.2% dieser Gesamterträge bzw. 1.2% des Overheads ausmachen. Konkret sollten die Mittel aus dem Zukunftsfonds der Universität bereitgestellt werden.
2. In Projektanträgen, welche die Entwicklung von Forschungssoftware nach sich ziehen, sollen entsprechende Personenmonate zur Finanzierung von RSE-Aufgaben beantragt werden. So kann ein antragstellendes Institut eine feste Anzahl von Arbeitsstunden aus dem RSE-Pool buchen und entsprechend für die Kosten aufkommen. Größere Verbundprojekte können Mittel im Rahmen eines zentralen INF-Projektes beantragen.
3. Förderorganisationen erkennen immer mehr die Notwendigkeit nachhaltiger Forschungssoftware-Entwicklung und legen entsprechend ausgewiesene Förderprogramme auf. So organisierte die DFG bereits zwei Ausschreibungen: Eine für Software-Nachhaltigkeit im Jahr 2016[10], die wahrscheinlich in eine zweite Runde gehen wird, und eine für Qualitätssicherung im Jahr 2019[11], die für Institute, die bereits durch die erste Ausschreibung gefördert wurden, nicht offen war. Es ist zu erwarten, dass in Zukunft noch mehr Programme aufgelegt werden. Gerade durch ein bereits etabliertes RSE-Konzept erhöhen sich unsere Chancen, als Stuttgarter Forschende und Dienstleistende in solchen Ausschreibungen erfolgreich zu sein.
4. Analog zu Laborleitern in experimentell ausgerichteten Arbeitsgruppen sollte eine Arbeitsgruppe, die maßgeblich Forschungssoftware entwickelt, eine unbefristete RSE-Stelle anbieten können, um die Nachhaltigkeit und Qualität der Software sicherzustellen. Hier bedarf es wiederum flexibler Maßnahmen, die eine Mischfinanzierung und Entfristung solcher Stellen erlauben.

Nächste Schritte

Basierend auf den grundlegenden Finanzierungsüberlegungen sehen wir folgende nächste Schritte zur Besetzung des RSE-Pools.

- Initiale Besetzung des Pools mit den bereits finanzierten RSE-Stellen der DFG-Software-Projekte, ReSUS und der NFDI4Ing. Auch wenn die Arbeit dieser RSEs natürlich dem jeweiligen Projekt dienlich sein muss, so kann auf jeden Fall ein regelmäßiger Austausch stattfinden für die Evaluierung von Anforderungen und Exploration möglicher Synergieeffekte. Initiale Besetzung des RSE Advisory Boards mit den Antragstellenden der DFG-Software-Projekte und Bestimmung eines*r kommissarischen RSE-Koordinators*in. Diese Schritte können sofort umgesetzt werden.
- Einrichtung und Besetzung eines*r dauerhaften RSE-Koordinators*in. Ansprechpartner*in für alle Stuttgarter RSEs, Veranstaltung von Treffen, Entwicklung von Schulungsprogrammen, Bericht zu übergeordneter Stelle usw. Finanziert aus dem Zukunftsfonds der Universität Stuttgart.
- Einrichtung und Besetzung einer*s zentralen RSEs. Zuständig für zentrale Dienste wie Repositorien, Containerisierung, Testsysteme, Veröffentlichung usw. Anschubfinanziert aus dem Zukunftsfonds, dauerhaft durch Beantragung und Pooling entsprechend ausgewiesener Gelder in Drittmittelanträgen.
- Zuordnung weiterer RSE-Stellen aus existierenden Arbeitsgruppen zum Pool. Die explizit mit Forschungssoftware-Entwicklung betrauten Stellen aus, zum Beispiel, der erwähnten DFG-Ausschreibung, bilden nur einen sehr kleinen Anteil der Bemühungen an der Universität Stuttgart ab. Durch eine zentrale Koordination können Expertisen gebündelt und Synergien erzielt werden.

[10] Nachhaltigkeit von Forschungssoftware.
[11] Qualitätssicherung von Forschungssoftware durch ihre nachhaltige Nutzbarmachung.

4.3 Aufgaben und Tätigkeiten

Die Aufgaben und Tätigkeiten der RSEs sind vielfältig und reichen von der Unterstützung einzelner Forschender bei der Erstellung von Skripten zur Datenprozessierung über die Analyse und Optimierung des Designs bestehender Codes bis hin zur automatisierten Containerisierung heterogener Software-Stacks für die Reproduktion komplexer Simulationsläufe. Sie lassen sich grob in drei Bereiche eingliedern: Individualisierte Betreuung von Softwareprojekten, Bereitstellung und Administration softwarebezogener Dienste, Organisation und Durchführung von Schulungen.

Individualisierte Betreuung von Softwareprojekten

Die RSEs unterstützen Stuttgarter Forschende beim Design von nachhaltiger, d.h. wartbarer und erweiterbarer Software. Forschungssoftware wird im Allgemeinen zunächst mit sehr engem Fokus entwickelt, etwa für eine Publikation. Durch den inhaltlichen Zusammenhang mehrerer Forschungsarbeiten zumindest innerhalb einer Promotion ergibt sich häufig eine nicht-systematische Weiterentwicklung des bestehenden Codes, welcher durch Überfrachtung mit Funktionalität kurzfristig hohen Wartungsaufwand verursacht. Software sollte von Grund auf so konzipiert sein, dass sie mehrfach erweitert werden kann, wofür jedoch Kenntnisse in Software-Engineering und Erfahrung notwendig sind. Darüber hinaus sollen RSEs die Forschenden im Aufsetzen von Build- und Packaging-Routinen unterstützen. Diese technischen Arbeitsschritte profitieren stark von langjähriger Erfahrung und sind unabdingbar, um Forschungssoftware einfach nutzbar zu machen. Zuletzt können RSEs durch aktive Mitarbeit in Review-Prozessen die Softwarequalität erhöhen.

Bereitstellung und Administration softwarebezogener Dienste

- Versionskontrolle ist für Software-Entwicklung essenziell, ebenso ein Ticket-System zur Nachverfolgung von Fehlern. Beides wird zur Zeit durch die GitHub-Enterprise-Installation am TIK abgedeckt, benötigt aber noch eine Strategie zur Einbindung externer Nutzer, welche gegebenenfalls Zugriff auf private Repositorien benötigen und sich an der Entwicklung in öffentlichen Repositorien beteiligen können sollen.
- Ebenso essenziell für die Nachhaltigkeit von Forschungssoftware ist ihre Dokumentation, angefangen bei der Annotierung des Quellcodes, über Wikis und Handbücher bis hin zu Webseiten. Auch hierfür gibt es Werkzeuge und Dienste, die von RSEs zentral bereitgestellt und gewartet werden sollen.
- Aufsetzen, Betrieb und Wartung von Continuous-Integration (CI) Umgebungen. Da diese zentral zur Verfügung gestellt werden sollen, muss die Verfügbarkeit dieser Umgebung sichergestellt werden. Nutzende sollen durch Dokumentation, Beispiele und Hilfestellung sehr schnell in die Lage versetzt werden, mit Ihrer Software von der Continuous Integration zu profitieren.
- Die zur Sicherung der Stabilität und Zuverlässigkeit von Forschungssoftware notwendigen Tests sollten auf einer möglichst breiten Auswahl an Hardware/Betriebssystem-Kombinationen im Kontext eines CI-Workflows durchgeführt werden. Die benötigte Rechenzeit kann pro Software und Kombination in der Größenordnung von Stunden liegen, da der aktuelle Zustand kompiliert und ausgeführt werden muss. Ein weiterer Finanzierungsbedarf ergibt sich hierbei durch die Bereitstellung von Infrastruktur mittels Pooling vorhandener und Beantragung neuer Ressourcen.

Organisation und Durchführung von Schulungen

Die RSEs konzipieren bedarfsgerechte Schulungen, die Forschende befähigen, Software nachhaltig zu entwickeln und bereitzustellen. Diese werden von den RSEs selbst durchgeführt, möglicherweise mit Beteiligung Stuttgarter Lehrender oder auch externer Dozent*innen. So wurde von SimTech-RSE R. Diestelkämper im Sommersemester 2020 ein Seminar „Programming Skills for Simulation Science“ für Promovierende der Graduiertenschule SimTech erfolgreich durchgeführt. In diesem Rahmen wurde auch im November 2019 ein erster Software-Carpentry-Workshop abgehalten. Das Angebot soll entsprechend der Bedürfnisse der Stuttgarter Forschenden weiter ausgebaut und langfristig zum Teil auch in die Masterausbildung integriert werden.

Ein wesentlicher Bestandteil sind Schulungen zum Umgang mit Programmier- oder Skriptsprachen, je nach Anwendungsgebiet. Die meisten Sprachen sind um gewisse Paradigmen aufgebaut, wodurch auch die möglichen Einsatzzwecke beeinflusst werden. Ebenso unterscheiden sich alle stark, was existierendes Ökosystem und Best Practices zum generellen Aufbau von komplexeren Anwendungen angeht. Von besonderem Interesse sind aber auch fortgeschrittenere Konzepte, die für die Mitwirkung in größeren Software-Projekten vermittelt werden sollten, wie z.B. Continuous Integration und Continuous Delivery, fortgeschrittene Werkzeuge zur Versionskontrolle, kollaborative Arbeitsplattformen und statische Code-Analyse. Hier ist in enger Abstimmung mit bestehenden Angeboten der Fakultäten und des HLRS ein ergänzendes Portfolio aufzubauen.

4.4 Kommunikation und Dissemination

Die Stuttgarter RSE-Aktivitäten sollen Vorbildcharakter auf nationaler und internationaler Ebene einnehmen. Dafür ist eine gute Kommunikations- und Disseminationsstrategie unerläßlich. Dies reicht von der Beteiligung bei kooperativen Veröffentlichungen bis hin zur Gründung eines Stuttgarter RSE Chapters.

Publikationen

Die Umsetzung der aufgeführten Maßnahmen und geplanten Tätigkeiten soll in transparenter Art und Weise dokumentiert und in geeigneten Formaten veröffentlicht werden. Dazu gehört auch die Mitwirkung an kooperativen Veröffentlichungen. So wurde Anfang 2020 ein de-RSE-Whitepaper unter Beteiligung von fünf Stuttgarter Autor*innen geschrieben und veröffentlicht[12]. Ein weiteres erklärtes Ziel ist die Anerkennung von Forschungssoftware als vollwertige Publikation an der Universität Stuttgart. Damit einher gehen zu entwickelnde Qualitätskriterien, die in eine entsprechende Publikationshandreichung aufgenommen werden sollen. Darüber hinaus soll ergänzend zur Forschungsdatenpolicy eine Forschungssoftwarepolicy erarbeitet werden.

Outreach: Workshops, Tutorials und RSE-Konferenzen

Als Teilnehmer: RSE-Konferenzen und -Workshops sind notwendig, um die laufenden Bemühungen der Universität Stuttgart um die Entwicklung nachhaltiger Software zu kommunizieren, um neue Werkzeuge und Dienste kennenzulernen sowie um mit der RSE-Community in Verbindung zu treten. Solche Konferenzen finden regelmäßig in Deutschland statt[13], wobei die wichtigste die jährliche deRSE-Konferenz ist. So entstand das bereits erwähnte Whitepaper in Folge eines halbtägigen Workshop auf der deRSE19-Konferenz. Für die Teilnahme an solchen Veranstaltungen sollten geeignete Reisemittel zur Verfügung gestellt werden.

Als Organisator: Die Organisation von Workshops, Tutorials und RSE-Konferenzen sollte ebenfalls in Betracht gezogen werden. Workshops können zum Beispiel genutzt werden, um in Stuttgart entwickelte Software zu fördern und eine Community von Benutzenden und Mitwirkenden aufzubauen. Zur Veranschaulichung: Das ESPResSo-Projekt organisiert seit 2010 jährlich einen 5-tägigen Workshop in Stuttgart[14]. Dieser Workshop macht die Software nicht nur für neue Benutzende zugänglicher, sondern bietet auch die Möglichkeit, sich mit den am Code Mitwirkenden entweder persönlich oder per Videokonferenz auszutauschen und aus langjährigen Benutzenden Mitentwicklende zu machen. Auch die in Abschnitt 4.3 angesprochenen Schulungen können bei entsprechend zur Verfügung stehenden Kapazitäten zu Workshops ausgebaut werden, um mit anderen RSEs in Kontakt zu kommen. Eine RSE-Konferenz in Stuttgart könnte nicht nur von Interesse sein, um unsere Software vorzustellen und die Anliegen der deutschen RSE-Gemeinschaft zu kommunizieren, sondern auch für die Industrie, um ihre RSE-bezogenen Dienstleistungen zu fördern (GitLab, Amazon Cloud, Microsoft) und neue Talente zu rekrutieren (Bosch, etc.). Eine solche Konferenz könnte teilweise von

[12] Anzt H, Bach F, Druskat S et al. *An environment for sustainable research software in Germany and beyond: current state, open challenges, and call for action* [version 1; peer review: 1 approved, 1 approved with reservations]. F1000Research 2020, 9:295. 10.12688/f1000research.23224.1

[13] Liste von vergangenen Veranstaltungen in 2017-2019.

[14] Liste von vergangenen Veranstaltungen in 2010-2020.

Open-Source-Akteuren und Akteuren der Industrie finanziert werden. So wurde die deRSE19-Konferenz von Microsoft Deutschland GmbH, Amazon Web Services, GitLab, dem R-Konsortium und dem Python Software Verband e.V. gesponsert[15].

Schaffung eines lokalen RSE Chapters in Stuttgart: Der Verein de-RSE e.V. wurde 2019[16] als nationales Chapter der uk-RSE (Society of Research Software Engineering Association) gegründet, um das Bewusstsein für die spezifischen Bedürfnisse der deutschen Gemeinschaft der RSEs zu stärken. Die de-RSE war das erste nationale Chapter; weitere Chapter erschienen dann in anderen europäischen Ländern wie den Niederlanden, Schweden und Norwegen. Die uk-RSE hat ein Netzwerk von 30 lokalen RSE-Chaptern und zwei regionalen RSE-Chaptern. Die de-RSE forderte die Schaffung eines solchen Netzwerkes in Deutschland im Jahr 2019. Im Februar 2019 wurden zwei lokale Chapter in Münster[17] und München gegründet[18], und zwei regionale Chapter wurden in Berlin/Brandenburg im Oktober 2019[19] and in Jena im März 2020 eingerichtet[20]. Die bereits beschriebene Stärke in der Forschungssoftware-Entwicklung und der daraus abgeleitete Anspruch macht Stuttgart zu einer natürlichen Wahl für ein lokales RSE Chapter. Die Eintrittsbarriere ist dank des Open-Source-Charakters der de-RSE-Organisation niedrig, so dass eine Gründung zeitnah umgesetzt werden kann.

5 Fazit

Wissenschaftler*innen und Förderorganisationen haben die Wichtigkeit einer nachhaltigen Entwicklung und Nutzung von Forschungssoftware für den Forschungsprozess erkannt. An der Universität Stuttgart befinden wir uns gegenwärtig in einer hervorragenden Ausgangslage, diese Wichtigkeit auch institutionell zu reflektieren. Mit der Einrichtung einer Organisationseinheit „Forschungssoftware-Entwicklung“ und damit einhergehenden Maßnahmen zur Schaffung und Finanzierung entsprechender RSE-Stellen kann die Universität eine Führungsposition in diesem Bereich erlangen und ihre Forschenden in diesem wichtigen Zukunftsthema umfassend unterstützen.

[15] deRSE19 Sponsoren und deren Preise. Alle Preise zzgl. 19% USt.
[16] 2017 RSE-Treffen, 2019 Vereinsgründung.
[17] https://de-rse.org/chapter/ms/
[18] https://de-rse.org/chapter/muc/
[19] https://de-rse.org/chapter/berserc/
[20] https://de-rse.org/chapter/jeRSE/