www.forschung-und-lehre.de

31. Jahrgang | 7,50 €

Forschung & Lehre

3 | 24

ALLES WAS DIE WISSENSCHAFT BEWEGT

ZUKUNFT DER ARBEIT | *ab Seite 160*

Großer Akademischer Stellenmarkt
| *ab Seite 212*

Superwahlen 2024
Herausforderungen an die Demokratie
| *ab Seite 178*

Infrastruktur
Die Bedeutung von Forschungs-software
| *ab Seite 186*

Gewalt an Hochschulen
Einblicke in das Ordnungsrecht
| *ab Seite 192*

[GLHR24] L. Grunske, A.- L. Lamprecht, W. Hasselbring, B. Rumpe:
Research Software Engineering - Forschungssoftware effizient erstellen und dauerhaft erhalten.
In: Forschung & Lehre - Zukunft der Arbeit, Band 24(3), pp. 186-188, Feb. 2024.

Research Software Engineering

Forschungssoftware effizient erstellen und dauerhaft erhalten

| LARS GRUNSKE | ANNA-LENA LAMPRECHT | WILHELM HASSELBRING | BERNHARD RUMPE | **Viele Forschungsprojekte an Universitäten sind ohne entsprechende Software nicht mehr denkbar. Software entwickelt sich zur relevanten Infrastruktur, die gepflegt, weiterentwickelt und gewartet werden muss. Mit Research Software Engineering (RSE) sollen geeignete Rahmenbedingungen geschaffen werden. Handlungsempfehlungen im Überblick.**

Der Begriff „Forschungssoftware“ (engl. „research software”) bezeichnet Software, die während des Forschungsprozesses oder für einen Forschungszweck erstellt wird. Forschungssoftware ist heute für viele wissenschaftliche Aktivitäten zwingend erforderlich. Sie kann zum Sammeln, Verarbeiten, Analysieren und Visualisieren von Daten, zur Erkennung von Zusammenhängen und zur Modellierung komplexer Phänomene und zur Durchführung anspruchsvoller Simulationen vom Material- über das Zell- und Organverhalten, soziale und ökonomische Beobachtungen, über das Wetter, das Klima der Erde bis hin zu Galaxienhaufen verwendet werden. Forschungssoftware spielt daher heute in fast allen Fächern eine entscheidende Rolle für die Forschung.

AUTOREN

Foto: privat

Lars Grunske ist Professor für Software Engineering an der Humboldt-Universität zu Berlin.

Foto: privat

Anna-Lena Lamprecht ist Professorin für Software Engineering an der Universität Potsdam.

Foto: privat

Wilhelm Hasselbring ist Professor für Software Engineering an der Universität zu Kiel.

Foto: privat

Bernhard Rumpe ist Professor für Software Engineering an der RWTH Aachen.

50 Jahre Software Engineering

Software Engineering (SE) hat sich in fast allen Universitäten und Fachhochschulen als eigenständiges Forschungsgebiet etabliert. Dabei haben die Professorinnen und Professoren durch ihre Forschung ein umfassendes Verständnis über die systematische und ingenieurtechnische Softwareentwicklung erarbeitet und dies nachhaltig in der Industrie etabliert. Dieses Wissen ist in Teilbereichen des Software Engineering wie etwa Anforderungsmanagement, Architektur, Design, Modellierung, Testen, Entwicklungsprozesse und angewandte formale Methoden organisiert, die sich weit über die Programmierung hinaus erstrecken.

Das Gebiet des Software Engineering entwickelt sich dennoch kontinuierlich weiter, weil neue Technologien neue Arten von Software und damit neue Herausforderungen für das Software Engineering mit sich bringen: Software ist sehr heterogen und reicht von eingebetteter Software und autonomen Steuerungen bis hin zu Desktop- und KI-Systemen, Geschäftssoftware und auch Forschungssoftware. Dabei sind die Probleme immer die gleichen:

- Wie lässt sich sicherstellen, dass die Software richtig und korrekt funktioniert?
- Wie kann die Qualität der Software sichergestellt werden?
- Wie lässt sich Software effizient entwickeln?
- Wie kann Software weiterentwickelt und langfristig nutzbar erhalten werden?
- Wie lassen sich Zeitvorgaben und Budgetbeschränkungen einhalten?
- Wie kann Software verallgemeinert werden, um mehr Nutzerinnen und Nutzer zu finden?

Die Lösungen und die sich daraus ergebenden Entwicklungstechniken sind in den verschiedenen Teilaktivitäten der Softwareentwicklung jedoch zumeist sehr unterschiedlich, denn unter anderem die Ausgangssituation, die Art der Software, die Komplexitätstreiber, die benötigten Qualitätsmerkmale, der Kontext, in dem die Software eingesetzt werden soll, sowie die regulatorischen Vorgaben unterscheiden sich stark. Die Software Engineering Community hat durch ihre Forschung schon viele Innovationen angeschoben, die oft auch breitere Nutzung finden. Dazu gehören zum Beispiel Wikis (die Grundlage der Wikipedia), agile Entwicklungsprozesse, Open Source (als Vorlage für Open Science) und eine Vielzahl an Werkzeugen zur Automatisierung in der Produktentwicklung, Informationsgewinnung mit Entwicklungs-Dashboards, für kollaborative Arbeitstechniken, für Versions- und Variantenmanagement und noch einiges mehr. Variantenmanagement mit Produktlinien, explizites Anforderungsmanagement und modellbasierte Entwick-

lung wurden zwar im SE nicht erfunden, aber auf Software übertragen, massiv ausgebaut und zur industriellen Praxisreife gebracht. Software Engineering spezialisiert sich immer öfter auf konkrete Anwendungsdomänen, die dann ganzheitlich und eigenständig behandelt werden. Dazu gehören unter anderem Automotive Software Engineering, Informationssysteme, Avionik, Eingebettete Software, Cloud Software Engineering und nun auch Research Software Engineering (RSE).

Research Software Engineering als Schnittstellendisziplin

RSE adressiert die Anwendung von Methoden, Techniken und Best Practices des Software Engineering bei der Entwicklung von Forschungssoftware. Im Rahmen des RSE zeigen sich bei der Entwicklung von Forschungssoftware mehrere Besonderheiten: Diese ist typischerweise hochspezialisiert und erfordert sowohl in der Entwicklung als auch in der Anwendung entsprechende wissenschaftliche Expertise. Weil die Erkenntnisse fortschreiten und damit sich Forschungsfragestellungen und Versuchsbedingungen signifikant verändern, ist Forschungssoftware außerdem änderungsintensiv. Weil die Nutzer- und Entwicklercommunity mit der Zeit wächst und sich weltweit organisieren muss, verändern sich Vorgehensweise, technische und organisatorische Randbedingungen, potenziert sich die Vielzahl unterschiedlicher, teilweise in Konflikt stehender und allzu oft nicht präzise ausformulierbarer Anforderungen etc.

Daraus entstehen spezifische Herausforderungen für die Softwareentwickung: Die Vorgehensweise der Anforderungserhebung im RSE ist völlig anders zu organisieren, weil diese untrennbar mit dem Forschungsprozess verzahnt ist, was in Folge massive Auswirkungen zum Beispiel auf die agile Entwicklung, das oft notwendige Refactoring der Architektur oder die Qualitätssicherung durch automatisiertes Testen hat.

Forschungsspezifische Qualitätskriterien beinhalten Transparenz und Reproduzierbarkeit der berechneten Ergebnisse und die Nachnutzbarkeit der Software selbst und damit Robustheit, Konfigurierbarkeit und Variabilität sowie die Evolutionsfähigkeit der Software. Für eine flexibel nachnutzbare Software werden Techniken aus der Softwarearchitektur, der Framework- und der Produktlinien-Theorie benötigt. Ebenso wichtig, aber weitgehend fehlend, ist die idealerweise automatisierte Konsistenzsicherung zwischen dem wissenschaftlichen Modell im veröffentlichten Papier und der oft hoch optimierten Software zu deren Umsetzung.

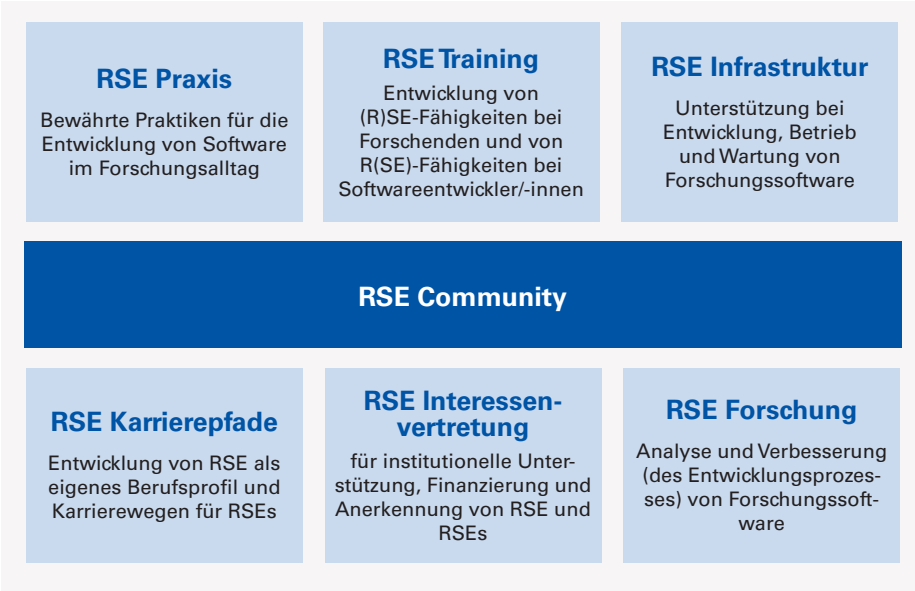

Abbildung: *Sieben Hauptbereiche sind innerhalb des RSE wichtig.*

Forschungssoftware wird heute zu selten von professionellen Entwicklerinnen und Entwicklern erstellt, sondern üblicherweise durch Forschende selbst, die als Teil ihrer wissenschaftlichen Tätigkeit auch Software schreiben. Die Softwareerstellung erfolgt häufig zunächst in einem lokalisierten Projekt, mit zunehmendem Erfolg dann in heterogenen, internationalen Teams, deren Zusammensetzung häufig wechselt und deren einzelne Mitglieder nur für einen begrenzten Zeitraum zur Verfügung stehen. Heterogene Ziele der Beteiligten, steigende Arbeitsteilung und ein oft fehlender organisatorischer Überbau mit Richtlinienkompetenz machen spezielle Organisationsformen für die Orchestrierung, Priorisierung, Harmonisierung und den architekturellen Zusammenhalt der Software notwendig. Dabei ist die Weiterentwicklung und Pflege von Forschungssoftware von oft nur kurzfristig einwerbbaren Projektmitteln der Forschungsförderung abhängig. RSE muss diese und weitere Besonderheiten in seiner Entwicklungsmethodik aufgreifen. RSE basiert sehr stark auf der klassischen Softwaretechnik und integriert als Schnittstellendisziplin weitere Elemente der Informatik, der jeweiligen Fachwissenschaften und Open Science. Getreu dem Motto “Better Software, Better Research” strebt RSE die kontinuierliche Verbesserung speziell der Entwicklungsprozesse für Forschungssoftware an, weil damit auch die Qualität der Forschungssoftware selbst verbessert werden kann. Damit trägt RSE substanziell zu effektiveren Forschungsprozessen und besseren Forschungsergebnissen bei. Innerhalb des RSE sind die sieben Hauptbereiche der Abbildung relevant:

Die **RSE Praxis** adressiert technische und organisatorische Fragen der Softwareentwicklung im Forschungsalltag. Dazu gehören mindestens die Kenntnis und Anwendung von Best Practices aus Software Engineering, Bibliotheken, der Umgang mit Entwicklungswerkzeugen, Lizenzen oder die Umsetzung der FAIR-Prinzipien für Forschungssoftware.

RSE Training behandelt die Notwendigkeit, Forschenden eine adäquate Teilmenge an Software Engineering-Fähigkeiten beizubringen, während beteiligte Softwareentwicklerinnen und -entwickler ein Grundverständnis der jeweiligen Forschungsprozesse benötigen, um im RSE-Bereich arbeiten zu können. RSE ist nicht nur Programmieren, weshalb erste „angewandte” Informatikstudiengänge die Möglichkeit bieten, ein RSE-Profil zu entwickeln. Aktuell ist jedoch ein Großteil der RSE-Ausbildung autodidaktisch, und daher sind die Softwareentwicklungs-Fähigkeiten der beteiligten RSE regelmäßig noch optimierbar.

RSE Infrastruktur umfasst die Unterstützung für die Entwicklung, den Betrieb und die Wartung von Forschungssoftware. Dies beinhaltet zum einen die technische Infrastruktur wie Versionskontrollsysteme, Datenspeicher und Computing-Cluster, aber insbesondere auch signifikante personelle Unterstützung zur Bewältigung der anfallenden

Aufgaben im gesamten Software-Lebenszyklus. In Deutschland dürfte zukünftig die Nationale Forschungsdateninfrastruktur (NFDI) und das darin eingebettete NFDI x CS (NFDI for and with Computer Science) erste technische Komponenten dafür bereitstellen.

RSE Karrierepfade sind nicht standardisiert und verlaufen oft nicht geradlinig. Viele Entwicklerinnen und Entwickler von Forschungssoftware sind eigentlich als Forschende angestellt (insbesondere als PostDocs), andere arbeiten als Technikerinnen und Techniker an Forschungssoftwareprojekten mit oder sind in zentralen Einrichtungen zur Unterstützung der Softwareentwicklung beschäftigt. Gerade an Universitäten ist für eine nachhaltig verfügbare Software-Infrastruktur ein tragfähiges und ausgebautes Dauerstellenkonzept notwendig.

RSE Interessenvertretung arbeitet daher an der Verbesserung der institutionellen Unterstützung, Finanzierung und Anerkennung von Forschungssoftware und ihrer Entwicklerinnen und Entwickler, um der Relevanz von Forschungssoftware auch in der Wissenschaftskultur, in der Personalplanung, in Bewerbungs- und Beförderungsverfahren, in Förderinstrumenten, in Publikationsstrukturen, in der Curriculumentwicklung und Kapazitätsplanung Rechnung zu tragen.

RSE Forschung betrachtet das RSE selbst als Forschungsgegenstand. Ziel dieser Forschung ist es, Forschungssoftware und ihre Entwicklungsprozesse besser zu verstehen, um auf dieser Basis Methoden und Werkzeuge zu ihrer Verbesserung zu entwickeln. Es besteht hier ein erhebliches Potenzial, zumal das Thema in der Forschung bisher wenig Beachtung gefunden hat und viele offene Fragen mit der zunehmenden Digitalisierung der Wissenschaft an Bedeutung gewinnen. Wie oben schon angedeutet braucht RSE eigene Methoden, Techniken und Werkzeuge für die Anforderungserhebung, die Qualitätssicherung, die Konsistenzsicherung, die Codegenerierung, FAIR-Datenspeicherung und vieles mehr sowie eine empirische Begleitung.

RSE Community beschreibt schließlich die Förderung der Vernetzung von Personen und Organisationen, die in irgendeiner Form im RSE aktiv sind. Auf internationaler Ebene spielen dabei die Research Software Alliance (ReSA) sowie das International Council of RSE Associations eine wichtige Rolle. In Deutschland sind der Verein de-RSE und die Fachgruppe RSE der Gesellschaft für Informatik (GI) aktiv.

Handlungsempfehlungen

Aus der Relevanz von Forschungssoftware für den Wissenschaftsstandort Deutschland ergeben sich unter anderem die folgenden Handlungsempfehlungen:

1. Das Wissenschaftssystem insgesamt muss besser auf die Entwicklung, die Weiterentwicklung, und den langfristigen Betrieb von Forschungssoftware vorbereitet werden. Dabei geht es nicht nur um die Effizienz der Software (das macht High Performance Computing), sondern um die Effizienz der Entwicklung, die Korrektheit der Ergebnisse und die langfristige Nutzbarkeit mit der sich daraus ergebenden Fähigkeit zur Weiterentwicklung.

2. RSE erfordert organisatorische Maßnahmen durch die Universitäten bzw. Wissenschaftseinrichtungen, typischerweise bestehend aus

- Training beziehungsweise Ausbildung von Forschenden, um eine proaktive Rolle im RSE einzunehmen, zum Beispiel in Form von Kursen bis hin zu (Micro-)Degrees,
- On-The-Job-Coaching in konkreten Vorhaben und Projekten, um die langfristigen Ziele effizient zu erreichen, zum Beispiel bei Architektur und Design, Technologiewechseln, Refactoring der Software oder Sicherung der Korrektheit,
- Etablierung eines geeigneten Leitfadens zur Softwareentwicklung.
- Etablierung von Forschungssoftware als langfristig verfügbare, gepflegte Infrastruktur mit entsprechendem Investment,
- Einrichtung von RSE-Zentren in Universitäten beziehungsweise Wissenschaftseinrichtungen, unabhängig von Rechenzentren und Bibliotheken.

3. Das Wissenschaftssystem sollte eine stärkere Wertschätzung für RSE in Form von Dauerstellen und Karrierepfaden etablieren sowie die Anerkennung von Software als Forschungsoutput auch für die klassischen wissenschaftlichen Karrierewege.

4. Die Leiterin oder der Leiter eines Forschungsinstituts sollte ein Arbeitsklima und Arbeitsumfeld organisieren, in dem Softwareentwicklung positiv gedeiht und wertgeschätzt wird. Dabei sind agile Softwareentwicklungsmethoden auch organisatorisch zu etablieren, gerade auch weil sie sich von typischen Meilenstein-orientierten Prozessen unterscheiden.

5. Die Fördermittelgeber müssen Software-Großprojekte mit Infrastruktur analog zu einem Forschungsschiff oder speziellen Teleskop und nicht nur IT-Hardware als Grundlage für Forschungen etablieren und finanzieren. Entwicklung durch RSE und Betrieb durch IT-Center sollten wie auch in der Industrie organisatorisch getrennt sein.

6. Letztlich müssen die Software Engineering-Forschenden die Spezifika von RSE noch besser verstehen lernen und damit die RSE besser unterstützen, also Forschung zur RSE als eigenständige Teildisziplin des Software Engineering betreiben. Das führt dazu, dass forschungsstarke Universitäten RSE-Professuren etablieren, die konsequenterweise auch gleich die Schulung und das Coaching-On-The-Job sowie strategische Aktivitäten der Softwareentwicklung übernehmen.

Insgesamt wird so eine Kultur der digitalisierten Forschung mithilfe spezialisierter Forschungssoftware etabliert und es werden auch in Zukunft effizient Forschungsergebnisse erreicht, die effektiv in soziale, gesellschaftliche, wirtschaftliche oder medizinische Innovationen übersetzt werden können.

»Die Weiterentwicklung und Pflege von Forschungssoftware ist von oft nur kurzfristig einwerbbaren Projektmitteln der Forschungsförderung abhängig.«

»Forschungssoftware wird heute zu selten von professionellen Entwicklerinnen und Entwicklern erstellt, sondern üblicherweise durch Forschende selbst.«