



Markus Diesmann, Julia Kowalski, Bernhard Rumpe

Research Software an der RWTH Aachen

In the last ten years, research software engineering has emerged as a crucial capability to an innovative scientific community. Efficient development of high-quality software beyond mere prototypes is often essential for transforming ideas into impactful results, translating hypotheses into knowledge, and implementing this knowledge in applications for the benefit of society and industry. Successful and effective research, development, and teaching at a university like RWTH now rely heavily on software development that is designed for sustainability, efficiency, and technical compatibility for integrated research networks. These activities also require the capabilities to use high-quality software in a creative, solution-oriented, and reproducible manner. Developing high-quality research software is essential for the success and sustainability of science and will increasingly impact teaching and learning as well.

Software nimmt in fast allen Bereichen der Natur- und Ingenieurwissenschaften eine Schlüsselrolle ein. Mittlerweile existieren grundsätzlich verschiedene Software-Arten, beispielsweise Algorithmen für die Simulation thermofluid-mechanischer Multiskalenprozesse, Methoden zu Konstruktion und Optimierung komplexer technischer Systeme, Skripte zur automatisierten Prozessierung extrem umfangreicher Datensätze aus Messkampagnen, Algorithmen zur Steuerung und Regelung komplexer Anlagen und Roboter, Wissenssysteme zur schnellen Identifikation von Lösungen bei selten auftretenden Ereignissen oder Störungen, Digitale Zwillinge zur integrierten Verwaltung und Kontrolle physikalischer, chemischer oder biologischer Forschungsexperimente, assistierende KI-basierte Systeme und nicht zuletzt operative Grundfunktionalitäten in Betriebssystemen, Datenspeichern und Kommunikationsinfrastrukturen. Die Verfügbarkeit und eine weit über den Prototypen hinausgehende Qualität der Software sind oft Voraussetzung dafür, aus Ideen Wirkung zu entfalten oder Hypothesen in Erkenntnisgewinn weiterzuentwickeln, diesen in Anwendungen umzusetzen

und damit der Gesellschaft und der Industrie verfügbar zu machen. Ohne Software würde es keine Navigation geben, es würde kein modernes Auto fahren oder gebaut werden können, keine Wettervorhersage oder Frühwarnung vor Naturgefahren existieren, keine moderne medizinische Diagnostik oder Energieversorgung möglich sein – unser Lebensstandard würde in dieser Form schlichtweg nicht existieren.

Erfolgreiche und wirkungsvolle Forschungs-, Entwicklungs- und Lehrtätigkeit an einer Universität wie der RWTH Aachen ist daher ohne eine auf Nachhaltigkeit, Effizienz und technische Kompatibilität für integrierte Forschungsverbünde ausgelegte Softwareentwicklung, und ohne die Kompetenz zur kreativen, lösungsorientierten und reproduzierbaren Nutzung hochwertiger Software, nicht mehr möglich. Offenheit und Zugänglichkeit von Software im Sinne des FAIR Paradigmas (Findable, Accessible, Interoperable, Reusable) sind weitere Faktoren, um einen software-integrierten Innovationsprozess und wissenschaftlichen Erkenntnisgewinn zu gewährleisten.

Weil Simulation heute als dritte Säule des Erkenntnisgewinns neben Experiment und Theorie etabliert ist und Experimente immer datenlastiger und rechenintensiver werden, macht die Bedeutung von Software und ihrer Entwicklung in den Natur-, Sozial- und Ingenieurwissenschaften derzeit eine ähnliche Entwicklung durch, wie die Softwareentwicklung im industriellen Kontext in den vergangenen 20 Jahren bereits durchlebt hat: Der Entwicklungsbereich eines Automobilherstellers wird heute von Software dominiert. Der Vorstand der Volkswagen AG hat beispielsweise Volkswagen als Software-Konzern bezeichnet. Software kann aufgrund ihrer Komplexität nicht mehr als Beiwerk behandelt werden, ihre Entwicklung muss organisatorisch durchdacht sein sowie technisch unterstützt und behandelt werden.

Software Engineering wurde bereits 1968 als wissenschaftliches Forschungsfeld gestartet und in den nachfolgenden Jahren in allen Informatik-Ausbildungen etabliert, um Komplexität, Effizienz, Nachhaltigkeit sowie die Qualitätsanforderungen adäquat zu adressieren. Der mittlerweile gut konsolidierte und in regelmäßiger Weiterentwicklung befindliche

„Software Engineering Body of Knowledge“ (SWEBOK) unterteilt seine Aktivitäten in 15 Kapitel, die sich mit Anforderungen, Entwurf, Konstruktion, Testen, Wartung, Konfigurations- und Engineering Management, Entwicklungsprozessen, Modelle und Methoden, Qualität, Professional Practices, Entwicklungsökonomie sowie Computing, mathematischen und Engineering Grundlagen des Software Engineering beschäftigen. Das klassische Programmieren gehört zur Konstruktion und betrifft daher nur einen kleinen Teil des Wissens. Viele Innovationen, wie etwa Wikis, Versionskontrolle, Variantenmanagement, Ticketsysteme oder agile Entwicklungsmethoden verdanken ihren Ursprung dem Software Engineering. Um die spezifischen Herausforderungen zur Entwicklung von Forschungssoftware zu adressieren, hat sich zunächst in den angelsächsischen Ländern als Spezialisierung des Software Engineering das „Research Software Engineering“ (RSE) herausgebildet. RSE bezeichnet ein Fachgebiet, das die Prinzipien des Software Engineering mit den Bedürfnissen und Zielen der natur- und ingenieurwissenschaftlichen Forschung vereint. Ähnlich wie andere

Spezialisierungen des Software Engineering (etwa Automotive Software Engineering) existiert RSE nicht isoliert, sondern kombiniert seine Methoden und Best Practices, um die Domänenforschung durch hochwertige Software und Softwarekompetenz effektiver und wirkungsvoller werden zu lassen.

RSE hat sich in den letzten Jahren zu einem der wichtigsten – jedoch nicht sehr lauten – Zukunftsthemen in der wissenschaftlichen Gemeinschaft entwickelt. RSE reicht von der Entwicklung von „Fire-and-Forget“-Skripten, die von Einzelpersonen geschrieben werden und für den einmaligen Gebrauch gedacht sind, bis hin zu koordinierten, institutionsübergreifenden Softwareprojekten, die viele Forscher zusammenbringen, um an einem gemeinsamen Ziel zu arbeiten. Zunehmend mutiert dabei Software, die zunächst als Forschungsgegenstand entwickelt wurde, zu wiederverwendbarer und damit qualitativ hochwertiger Software als Infrastruktur. Software bekommt so ein immenses Potenzial, Innovationen und wissenschaftliche Entdeckungen schneller voranzutreiben. Allerdings ist aus dem Software Engineering bekannt, dass Wiederverwendbarkeit vorbereitet und organisiert werden muss – und auch welche Maßnahmen dafür nötig sind.

Ein gemeinsames Merkmal von RSE ist die Fokussierung auf einen bestimmten Anwendungsbereich und die damit verbundenen Auswirkungen auf alle Phasen des Software-Lebenszyklus. Beispielsweise ist der Prozess der Anforderungserhebung stark mit dem wissenschaftlichen Forschungsprozess verzahnt und die Korrektheit der Software relativ zu den veröffentlichten wissenschaftlichen Papieren beziehungsweise den darin enthaltenen Modellen zu sichern. Deshalb passen klassische Paradigmen des Software

Engineering aus der Industrie oft nicht zur gelebten Praxis einer auf Forschung und Innovationsleistung ausgelegten Universität. Softwareentwicklung muss sich hier nahtlos in die jeweiligen Forschungsprojekte eingliedern, welche oft von internationalen, interdisziplinären Kooperationen, hoher personeller Fluktuation und leider vor allem fragmentierten Finanzierungsstrukturen geprägt ist. Gleichzeitig möchte man in der Forschung, also auch in der damit einhergehenden Softwareentwicklung, agil auf aktuelle Entwicklungen reagieren. Aufgrund dieser Anforderungen und der zunehmenden Komplexität der Algorithmen, der Integration von Datenmengen, KI-Techniken und des Wunsches nach energetischer Sparsamkeit rechenintensiver Prozesse ist der Stand der Praxis im klassischen Software Engineering für die Bedürfnisse der forschungsorientierten RSE noch deutlich ausbaubar. Wie in anderen Bereichen auch, passen die bestehenden Software-Engineering-Methoden, -Werkzeuge und -Best Practices nicht genau und müssen angepasst werden, um bessere Software und damit bessere Forschung zu ermöglichen.

Die RWTH ist dabei, sich hier noch besser zu positionieren. So wurde in der ersten großen Konferenz am 17. Mai 2023 eine RSE-Strategie diskutiert. Definiert wurde eine koordinierte Strategie für Nachhaltigkeit und Exzellenz in RSE, die aus einer Reihe von inhaltlichen, organisatorischen, technischen und auch Ausbildungs-Maßnahmen besteht und sowohl in Exzellenzclustern als auch der Gesamtstrategie der RWTH Aachen ein wichtiger Eckpfeiler sein wird. Zentral ist auch die stattgefundene Sensibilisierung für das Thema über die gesamte Universität hinweg. Ein weiterer Eckpfeiler ist die Etablierung

einer Sammlung an Best Practices der Softwareentwicklung an der RWTH, indem RSE in die Lehre integriert wird. Darüber hinaus sind eine institutionelle RSE-Methodik, Coaching-On-The-Job und Tooling Support durch Experten, also Software Engineers, Architekten und Projekt Engineers im Aufbau. Komplex bleibt die Frage, wie interne und externe Finanzierungsmöglichkeiten nachhaltig geschaffen werden können. Hier ist neben der Universität insbesondere das Forschungssystem gefragt, um langlebige Software-Infrastruktur valide und gesund zu erhalten.

Es ist mittlerweile klar, dass RSE-assoziierte Herausforderungen die gesamte Wissenschaft im globalen Kontext betreffen. Deshalb ist die RWTH dabei, insbesondere institutionsübergreifend mit der Jülich Aachen Research Alliance (JARA), der Euregio und den vielen RWTH-nahen Forschungsverbänden eine gemeinsame RSE-Strategie zu erarbeiten und weiterzuentwickeln.

Diese Ausgabe der RWTH THEMEN zeigt eine Auswahl an Softwareprojekten, welche RSE-Paradigmen umsetzen. Dies beinhaltet Beiträge zur Softwareentwicklung selbst, in Form von Artikeln zur Robustheit numerischer Modelle gegenüber Parameteränderungen, der Methodik zur Nutzung expliziter, domänenspezifischer Modellierungssprachen, generische Optimierungswerkzeuge, Software zur Datenintegration unter Wahrung des Datenschutzes und der Nutzung fachübergreifende Softwareumgebungen für spezifische Modellklassen. Darauf aufbauend werden Anwendungen für spezifische Fachdomänen, wie der Neurowissenschaft, der Materialwissenschaft, der Geologie bis hin zur Luftfahrt beschrieben. Umrahmt werden diese

Beiträge von dem Artikel „Research Software Engineering – Vergangenheit, Gegenwart und Zukunft“, der die mittlerweile zehnjährige Geschichte und Entstehung des RSE beschreibt, siehe Seite 8, und dem Beitrag „Wissenschaftstheoretische Reflexionen zu Research Software Engineering – Konzeptuelle Analyse von Forschungssoftware“, der wissenschaftstheoretische Reflexionen vornimmt, siehe Seite 80.

Die Beiträge zeigen, welche Arten von Software an der RWTH entwickelt, gewartet, aktualisiert und für die Öffentlichkeit beziehungsweise die Fachcommunity frei verwendbar zur Verfügung gestellt wird und oft zur gemeinsamen Weiterentwicklung und Integration neuer Forschungserkenntnisse und Ideen geführt hat. Denn Softwareentwicklung in der Wissenschaft kann sich über Jahrzehnte erstrecken. Sie wandelt sich im Erfolgsfall typischerweise nach einer initialen, projekt getriebenen Phase von einer „Fire-and-Forget“-Software zu einer grundsätzlichen Infrastruktur, die nicht mehr von einer einzelnen Gruppe betreut sondern in Verbänden von universitären Professuren und Forschenden an nationalen Forschungseinrichtungen weiter genutzt und entwickelt wird. Ist eine Software erfolgreich, mutiert sie zur Infrastruktur, was einerseits einen Umbau in Bezug auf Anwendbarkeit und Robustheit, methodische Änderungen zur Weiterentwicklung im Konsortium erfordert und andererseits auch eine große Verantwortung induziert.

Darüber hinaus zeigt diese Ausgabe der RWTH THEMEN, dass Software nicht nur zum Treiber der Wissenschaft geworden ist. Vielmehr sind sich die Forschenden bewusst, dass Software den wissenschaftlichen Prozess selbst verändert und als dritte Säule

des Erkenntnisgewinns neue Wege induziert, wie über ein wissenschaftliches Problem nachgedacht wird.

Im Sinne des institutionsübergreifenden Austausches sind wir froh, im JARA Center for Simulation and Data Science (CSD) Erfahrungen über die Fachdisziplinen hinweg von der Großforschung bis zur universitären Forschung austauschen zu können. So entsteht ein detailliertes und vollständiges Bild, was RSE ausmacht, wie es an einer Exzellenzuniversität gelebt, gelehrt und an zukünftige Generationen weitergegeben werden kann. Der Erfolg bei der Entwicklung von Forschungssoftware ist kritisch für den Erfolg und die Nachhaltigkeit der Wissenschaft als Ganzes und hat in Zukunft auch noch deutlich mehr Einfluss auf die Lehre. Wir sehen daher ein Zukunft, in welcher RSE nicht länger als neuartige Entwicklung angesehen wird, sondern RWTH-weit als gelebte Praxis in allen Disziplinen selbstverständlich geworden ist.

Autoren

Univ.-Prof. Dr.rer.nat. Markus Diesmann leitet das Lehr- und Forschungsgebiet Computational Neuroscience und ist Leiter des Instituts für Computational and Systems Neuroscience (IAS-6) am Forschungszentrum Jülich.

Univ.-Prof. Dr.sc.habil. Julia Kowalski ist Inhaberin des Lehrstuhls für Methoden der Modellbasierten Entwicklung in den Computergestützten Ingenieurwissenschaften.

Univ.-Prof. Dr.rer.nat. Bernhard Rumpe ist Inhaber des Lehrstuhls für Software Engineering (Informatik 3).
