

Research Software Engineering @ DSC

Wilhelm (Willi) Hasselbring

Software Engineering, Kiel University, Germany
<https://se.informatik.uni-kiel.de>

School of Electronics & Computer Science, University of Southampton, UK
<https://www.southampton.ac.uk/research/areas/electronics-computer-science>

Kiel, July 11th, 2024



Kiel University
Christian-Albrechts-Universität zu Kiel



University of
Southampton

Research Software Engineering

Forschungssoftware effizient erstellen und dauerhaft erhalten

| LARS GRUNSKÉ | 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 erforder-

lich. 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.

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 zu meist 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-

www.forschung-und-lehre.de

31. Jahrgang | 7,50 €

Forschung & Lehre

3 | 24

RSE Praxis

Bewährte Praktiken für die Entwicklung von Software im Forschungsalltag

RSE Training

Entwicklung von (R)SE-Fähigkeiten bei Forschenden und von R(SE)-Fähigkeiten bei Softwareentwickler/-innen

RSE Infrastruktur

Unterstützung bei Entwicklung, Betrieb und Wartung von Forschungssoftware

RSE Community

RSE Karrierepfade

Entwicklung von RSE als eigenes Berufsprofil und Karrierewegen für RSEs

RSE Interessenvertretung

für institutionelle Unterstützung, Finanzierung und Anerkennung von RSE und RSEs

RSE Forschung

Analyse und Verbesserung (des Entwicklungsprozesses) von Forschungssoftware

AUTOREN



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



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



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



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

What is Research Software?



[Chue Hong et al. 2022]

Research Software

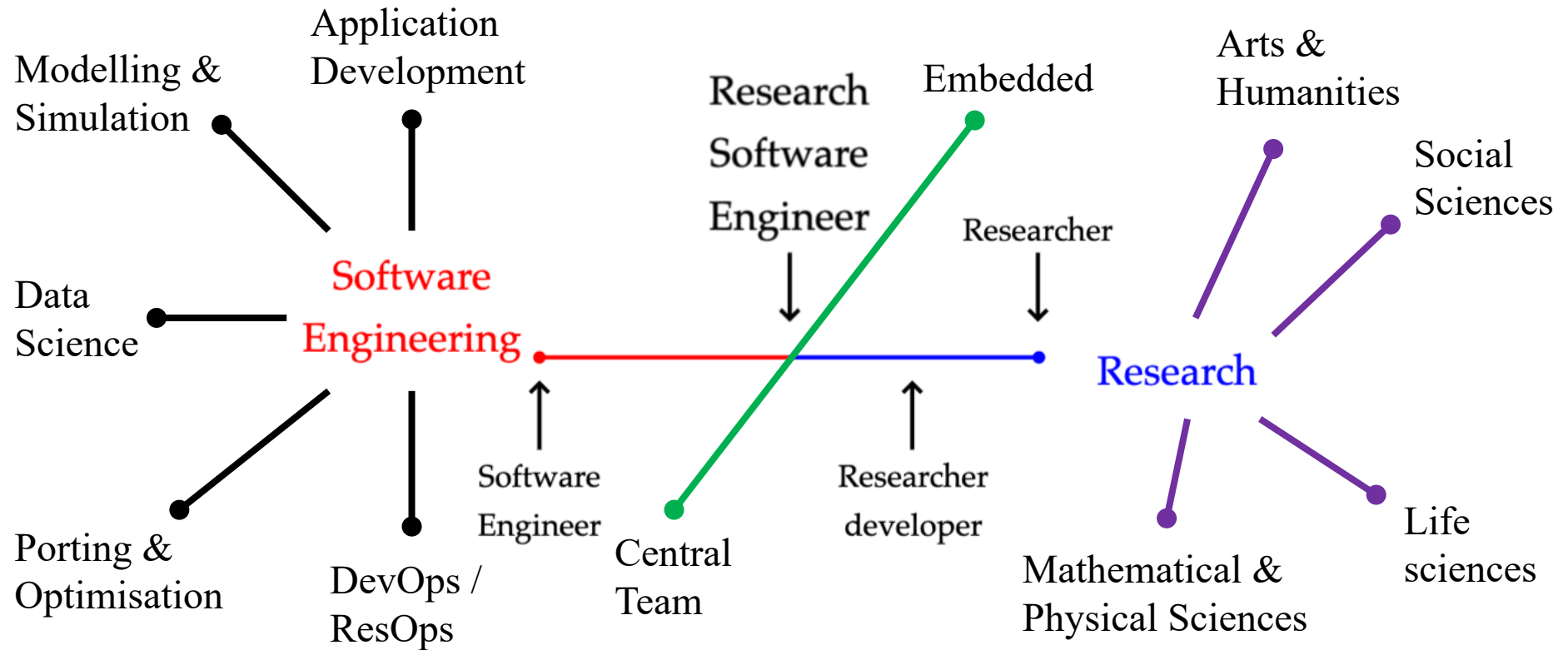
created during the research process or for a research purpose

Software in Research

used for research but not created during or with research intent

Research software should be **FAIR** [Hasselbring et al. 2020b] and **open** [Hasselbring et al. 2020a].

What is a Research Software Engineer?





<https://www.software.ac.uk/>



<https://rsgsoton.net/>



Research IT



Research Software Engineering

The Research Software Engineering (RSE) team enhances the University's capacity to produce high quality research software by collaborating with researchers to create correct, efficient, readable, reliable and sustainable code.

<https://research-it.manchester.ac.uk/>



Research software engineering

EPCC is one of the founding groups involved in research software engineering. As a lead site for the [Software Sustainability Institute](https://www.software.ac.uk/), it was instrumental in the development of research software engineering as a professional area of research. EPCC collaborates with researchers on topics related to research software policy and practice.

<https://www.epcc.ed.ac.uk/research/research-themes>

Examples for Technology Research Software

The logo for Kieker, featuring the word "Kieker" in a blue, sans-serif font. The letter "i" is stylized with a blue dot and a blue underline that extends to the right.

<https://github.com/kieker-monitoring>

Kieker: A monitoring framework for software engineering research

[Hasselbring and van Hoorn 2020]

The logo for ExplorViz, featuring the word "ExplorViz" in a blue, sans-serif font. The letter "o" is replaced by a small globe icon.

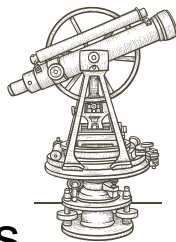
<https://github.com/ExplorViz>

ExplorViz: Research on software visualization, comprehension and collaboration

[Hasselbring et al. 2020c]

The logo for Theodolite, featuring the word "Theodolite" in a dark brown, cursive font.

<https://www.theodolite.rocks>



The Theodolite Scalability Benchmarking Framework

[Henning and Hasselbring 2021, 2022]

German Special Interest Group funded in 2023 “Research Software Engineering”

Interdisciplinary forum for:

- Software Engineering Researchers
- Research Software Engineers



Task Forces:

- Categories of Research Software ←
- RSE Advocacy Strategy
- RSE Community Events
- RSE Online Community
- RSE Research
- RSE Software Development Guidelines
- RSE State of Nation Report

<https://fg-rse.gi.de/> (German)
[Hasselbring et al. 2024]

Outlook: RSE Research



Newcastle, Sept 03-05, '24

<https://rsecon24.society-rse.org/>

Research Software Engineering

Software Engineering Research

Research Software Engineering Research
aims at understanding and improving how software is developed for research.

RSE Research, in short.

[Felderer et al. 2023]

References

- [Chue Hong et al. 2022] N. P., Chue Hong, et al. (2022). FAIR Principles for Research Software version 1.0. (FAIR4RS Principles v1.0). Research Data Alliance. DOI <https://doi.org/10.15497/RDA00068>
- [Felderer et al. 2023] Felderer, M., Goedicke, M., Grunske, L., Hasselbring, W., Lamprecht, A. L. und Rumpe, B.: “Toward Research Software Engineering Research”. 2023. DOI <https://doi.org/10.5281/ZENODO.8020525>.
- [Hasselbring et al. 2020a] W. Hasselbring, L. Carr, S. Hettrick, H. Packer, T. Tiropanis: “Open Source Research Software”. In: *Computer*, 53 (8), pp. 84-88. 2020. DOI <https://doi.org/10.1109/MC.2020.2998235>
- [Hasselbring et al. 2020b] W. Hasselbring, L. Carr, S. Hettrick, H. Packer, T. Tiropanis: “From FAIR Research Data toward FAIR and Open Research Software”, *it - Information Technology*, 2020. DOI <https://doi.org/10.1515/itit-2019-0040>
- [Hasselbring et al. 2020c] Hasselbring, W., Krause, A., Zirkelbach, C.: “ExplorViz: Research on software visualization, comprehension and collaboration.” In: *Software Impacts*, 6, 2020. DOI <https://doi.org/10.1016/j.simpa.2020.100034>.
- [Hasselbring et al. 2024] Hasselbring, W., Druskat, S., Bernoth, J., Betker, P., Felderer, M., Ferenz, S., Lamprecht, A. L., Linxweiler, J., Rumpe, B.: “Toward Research Software Categories”. *arXiv e-prints* . DOI <https://doi.org/10.48550/arXiv.2404.14364>
- [Hasselbring and van Hoorn 2020] Hasselbring, W., van Hoorn, A.: “Kieker: A monitoring framework for software engineering research.” In: *Software Impacts*, 5, 2020. pp. 1-5. DOI <https://doi.org/10.1016/j.simpa.2020.100019>
- [Henning and Hasselbring 2021] Henning, S., Hasselbring, W.: “Theodolite: Scalability Benchmarking of Distributed Stream Processing Engines in Microservice Architectures.” In: *Big Data Research*, 25 (100209), 2021. pp. 1-17. DOI <https://doi.org/10.1016/j.bdr.2021.100209>
- [Henning and Hasselbring 2022] Henning, S. und Hasselbring, W.: “A configurable method for benchmarking scalability of cloud-native applications.” In: *Empirical Software Engineering*, 27 (6), 2022. p. 143. DOI <https://doi.org/10.1007/s10664-022-10162-1>