Future challenges for Research Software Engineering

Florian Goth [1], **Leyla Jael Castro** [2], **Gerasimos Chourdakis** [3], **Simon Christ** [4],
Jeremy Cohen [5], **Nicholas A. Del Grosso** [6], **Jean-Noël Grad** [7],
Magnus Hagdorn [8], **Toby Hodges** [9], **Jan Linxweiler** [10,11],
Frank Löffler [12,11], **Michele Martone** [13], **Jan Philipp Thiele** [10,14]

[1] Institut für theoretische Physik 1, University of Würzburg, 97074, Würzburg, Germany
[2] ZB MED Information Centre for Life Sciences, Cologne, Germany
[3] Institute for Parallel and Distributed Systems, University of Stuttgart, Stuttgart, Germany
[4] Leibniz University Hannover, Department of Cell Biology and Biophysics, Computational Biology, Germany
[5] Imperial College London, London, UK
[6] Institute for Experimental Epileptology and Cognition Research, Uniklinikum Bonn, Germany
[7] Institute for Computational Physics, University of Stuttgart, Germany
[8] Geschäftsbereich IT, Charité Universitätsmedizin Berlin, Germany
[9] The Carpentries, USA
[10] Technische Universität Braunschweig, Germany
[11] de-RSE e.V.—Society for Research Software in Germany
[12] Competence Center Digital Research, Friedrich Schiller University Jena, Germany
[13] Independent researcher, Germany
[14] Weierstrass Institute, Berlin, Germany; Leibniz University Hannover, Institute of Applied Mathematics, Scientific Computing, Hannover, Germany

Abstract: It is 2025 and the research software community is, admittedly, still working on establishing and growing Research Software Engineering (RSEng) as a domain in its own right, with the hope of ensuring that it gets the recognition it deserves. As the field develops, the role and value of Research Software Engineers (RSEs) becomes clearer. With the growing importance and scope of software engineering in research, researchers who code now begin to fill dedicated, specialised RSE roles crucial for the sustainable development of long-term research software projects that can rapidly innovate. Looking into the future, participants of the deRSE25 conference discussed how RSEng could evolve in response to the rapid changes in the broader social, technological, and organisational landscape. This paper presents the main discussion points of the conference participants, imagining a future in which RSEs have a well-defined and well-integrated role in research institutions, working as engineers of ever more complex research software projects, taming a growing range of methods and tools, and guiding researchers through technical and ethical considerations related to research software.

Keywords: trends in RSEng, educational advancements, emergent academic jobs, academic reorganisation

1 Introduction

Research Software Engineering is still a developing field, with increasing integration into various aspects of research. In Germany, a formal call for creating a sustainable environment for research software was only published a couple of years ago [ABD+21], while the foundational competencies of an RSE have only recently been defined in detail [GAB+24]. Central RSE units are in the process of being imagined [KCF+25], while an RSE Master's is in the early stages of design [DCG+25].

In this opinion paper, we try to imagine the state of our field and community in the mid-term future, focusing on future challenges. While future projections are standard practice in software engineering (SE) [KH23, KKKN21, HD23, Bos16, Boe11], so far future projections are far less common for the RSEng domain.

Here, we present the results of a workshop discussion (notes are archived here [GCH+25]) at the deRSE25 conference in Karlsruhe[1], without claiming to represent the entire deRSE community. The material presented here is based on the output of the workshop participants, but extended with the authors' ideas. Since deRSE25 was co-located with SE25, the annual conference for the Software Engineering community in Germany, the discussion was also open to software engineers, but no list or profile of the participants is available. The workshop hosts felt, that the majority of participants were from the German RSE community.

The workshop was held as an open discussion of 45 minutes between the participants. As can be seen from the workshop material [GCH+25], we prepared some slides to steer the discussion towards the time frame of 2035. For a start we prepared questions that we openly asked to the audience, while also allowing the participants to meander towards questions of their interest. The prepared questions stem from internal discussions among the workshop hosts. It started from discussions whether the recently defined competencies of RSEs [GAB+24] are just snapshots of our time, or whether they can constitute something that stands the test of time. Then we asked ourselves how the community of RSEs would interact with some emerging trends. The collected material was further discussed among the authors in follow-up meetings, structuring and refining the ideas, adding new ideas as described in each of the following sections.

Nevertheless, by disseminating the discussed ideas, we aim to engage the community in creating a future worth living and to advance the conversation about how RSEng should, or could, develop. Since this paper focuses on the outputs of the deRSE25 conference, we present our thoughts and perspectives in a largely German-focused context. At the same time, we recognise, the wider international growth of RSEng as a very important development for people undertaking software development within the research community. Much of our work and our discussions are also building on wider perspectives for the future of RSEng that are currently under investigation among RSEng communities in other countries. While taking a German-focused view in this paper, we therefore aim to highlight resources and materials from the wider international community where we feel this is relevant.

It is already difficult to structure the spectrum of RSE tasks and responsibilities today, given the spectrum of research software [HDB+24]. It is even more challenging to structure the as-

[1] deRSE25, session "The End of RSEng? Challenges and Risks for RSEng": https://events.hifis.net/event/1741/contributions/14026/

pects that will become relevant for the RSE community in the future. In order to structure the workshop discussion for this publication, we decided to consider the evolution of the term RSE and its definition, the growth and development of the RSE community, the impact of automation and generative artificial intelligence in RSE work, the impact of complexity and usability in research software, ethical and social considerations, the expected transformations in academic institutions, and the potential impact of alternative career paths for RSEs and new groups of RTPs (*Research Technical Professionals*). This roughly mirrors the questions asked to the audience, where the entire workshop content is condensed into these topics.

This work supplements other opinion pieces developed in parallel in the SE community [CAHJ25, Soc24, BCC+24, OMS+24, DEC+25, CSH22], which where not known to the workshop participants; remarkably, many of the projections made by the participants overlap with predictions made by the SE community in "Research Software Engineering in 2030" [KH23]. We conclude the paper with a summary and with our wishes and recommendations.

2 Aspects of this Future

RSEng in 2035 A

In order to list what has been achieved in a certain aspect within the RSE community of the future of 2035 we use these "RSEng in 2035" boxes.

Looking ahead over a 10-year time frame, we expect that the RSE community will have contributed to making research software and its outputs more sustainable, robust and maintainable. We hope that research software will have become an accepted research artefact on an equal footing with data publications and traditional text publications. This is already happening in a number of ways, particularly through movements such as DORA [DOR] and CoARA [COA] which are advocating for a change in the way that research is assessed and outputs recognised. However, we want to ensure that one of the core achievements made by the RSE community in this time frame is putting the infrastructure and training opportunities in place, alongside the necessary advocacy, to ensure that researchers receive core software skills as part of their scientific training. Developing software tools and scripted workflows is becoming an increasingly important part of everyday research activities. Therefore, it is vital to improve digital skills of all researchers. This will also allow RSEs to focus on other more advanced and specialist areas. In this context, we do not foresee the demise of the RSE as an independent role. However, we do see the work of RSEs developing and shifting as the research community changes [CAHJ25].

At the same time, we do recognise that the application of core software development best practices to enhance and address problems with research codes, that can often make up the bulk of current RSE work, will change. As noted above, these are skills that we would expect any researcher who writes code to have in 10 years time. The rapid development of artificial intelligence (AI) and large language models (LLMs) means that researchers are likely to have access to tooling that can assist them in writing robust, sustainable code, lowering the barrier to developing code, or at least changing the required skill set to some extent. They can also be expected to have an understanding of frameworks to support testing, packaging and deployment of their code as well as integration with central domain specific infrastructures. Again, all these skills will, we anticipate, be gained as part of training infrastructure that will provide domain scientists with software development expertise as part of their core scientific training.

So where do RSEs fit into this picture? Already today, RSEs are providing technical skills and input to research activities in a range of different areas, including: high-quality, robust architecture and design for research software; seamless integration with research and data infrastructures; development of software for novel and emerging architectures; application of specialist numerical and statistical methods within software; optimisation of codes for use on high-performance computing infrastructure; and green computing - efficient implementations of algorithms/methods and use of hardware. These specialist skills that RSEs provide take a significant amount of time to develop and maintain, which is why we do not expect domain researchers to also become experts in a range of advanced RSEng topics.

Story 1: Introduction

To provide a more concrete setting, imagine a world ten years from now – not too overwhelmingly far into the future, but distant enough for some changes to have occurred. Over this time frame Kim [ABD+21] and Kay [GAB+24], our fictional RSEs who we have used to illustrate specific scenarios in our earlier work, were able to ride the waves of their career in RSEng and are now pondering early retirement on a sun blessed beach. But the world has moved on. When they look back, what will they see as problems that have been solved by then? Which tasks are still relevant, where they can be happy that a younger generation is now taking up the baton and carrying forward their work? And what are the issues that have newly emerged during this time? In order to bring this future to life, these two personas will accompany us through the rest of this paper and we will shed light on aspects of their future in respective small story boxes.

2.1 Definition and Evolution of RSEs

RSEng in 2035 B

The term RSE is established in the minds of university boards and leadership, as a valuable member in the academic framework. New RSEs are trained through a specialised Master's programme, and their support is valued by researchers. Researchers are still taught basic software skills, but the content of this lecture has drastically changed.

During the discussion the workshop participants found that the digital skills of domain scientists will increase, and software will become easier to interact with, reducing reliance on RSEs to train newcomers on the most basic tasks. This means, that today's workshop content of proper versioning and data management could become obsolete by using more intuitive tools. Introductory courses can then focus on more higher-level concepts. User training sessions might focus on more advanced topics that cater to a smaller audience, or involve mentoring more frequently.

While the RSE core competencies [GAB+24] might remain unchanged, there might be a shift in the responsibilities of RSEs and thereby requiring a realignment of the priorities in training. Regarding the definition of RSEs, there will still be a spectrum of competencies, from full professional RSEs to domain scientists who develop research software. We note a similar discussion took place regarding how SE roles had to adapt to rapid changes brought by automation, cloud services and new project management methodologies [MOL+19].

What will *not* change over that time frame is the RSE's ability to remain flexible and embrace change, and keep an eye on the wider landscape, picking up new tools and skills as needed. Communication and collaboration skills will remain essential to engage with domain scientists and help them find technical solutions.

Looking at the competencies on a broader scale, especially the education aspect deserves some attention.
The Covid-19 pandemic acted as a catalyst for the digital transformation of higher education [BØLD22]. Students and early-career researchers are now more familiar with digital learning tools, which lead to a shift in the responsibilities of teachers. To quote from the original study, "with so many digital resources at hand, the task of the lecturer will be fewer lectures, and to act more as a facilitator of resources, and to monitor activities and results over time." [BØLD22]

Today, the Carpentries now offer specialised software workshops for data scientists[2], librarians[3], and soon High-Performance Computing (HPC) practitioners[4]. RSEng-relevant open educational resources and Master's programmes are tracked in the Learning and Teaching RSE database[5] and UK SSI resources hub[6].

Story 2: The Definition of RSEs

Kim and Kay chuckle when they look at their original job descriptions from 2020: "build tools for scientists, manage data, write reusable code." Now, their roles are unrecognisable: half AI whisperer, half research ethicist, and part-time diplomat negotiating between synthetic and human collaborators. They now not only manage software, but also the personalities of autonomous lab agents and negotiate data-sharing agreements with international AI consortia. Despite the tech upheaval, their deep knowledge of scientific workflows and the messy, human way science actually gets done remains indispensable. When labs falter under algorithmic bias or unclear model provenance, it is Kim and Kay who transform the chaos back into meaningful order. They understand that science does not move in straight lines, and that trust between collaborators, institutions, and even machines has to be built, not coded. Their job titles have changed, but their role as bridges between logic and life remains the same. In a future of shifting code and constant reinvention, it is their understanding of people that never went out of date.

2 https://datacarpentry.org
3 https://librarycarpentry.org
4 https://www.hpc-carpentry.org
5 https://de-rse.org/learn-and-teach/
6 https://www.software.ac.uk/resource-hub

2.2 Development of the RSE Community

RSEng in 2035 C

National RSE communities are increasingly collaborating with local partners, fostering stronger networks across countries. Regular international conferences further amplify these connections, promoting the global exchange of ideas. The introduction of RSE Master's programmes, coupled with a shift in culture towards valuing research software, has significantly elevated its profile. This growing recognition has led to a surge in membership of national societies, reflecting the increasing importance of RSEng in the research landscape.

In recent years, various RSE-related institutes and advocacy groups have been formed, such as the UK Software Sustainability Institute[7], the US Research Software Sustainability Institute[8], the European Virtual Institute for Research Software Excellence[9], and the EOSC Opportunity Area Expert Group 7 on Research Software[10]. We expect these initiatives to help increase the visibility and accelerate the professionalisation of the RSE role. In addition, a multitude of RSE communities, self-organised at the national and regional level, argue for more viable RSE career paths in academia. It can be expected that RSE positions become more easily accessible, possibly through centrally-funded long-term positions at research institutions, such as in the bwRSE4HPC service[11]. The community of RSEs will get a significant boost in their acceptance if RSE Master's programmes become available at various universities. With increased numbers, the visibility of the RSE community to industry will be enhanced.

Story 3: Development of RSE community

Kay stands proudly at the podium of the first International Research Software Engineering Conference held in the Southern Hemisphere, a historic moment for both her career and the global RSE community. Having moved to South Africa, she speaks passionately about the unique challenges researchers in the Global South face: limited access to high-performance computing, unreliable funding, and the persistent digital divide. Her keynote highlights how local innovation thrives despite constraints, often out of necessity rather than choice. The audience, a diverse mix of global experts, listens intently as she calls for equitable collaborations, not just inclusion as an afterthought.

7 https://www.software.ac.uk
8 https://urssi.us
9 https://everse.software
10 https://eosc.eu/opportunity-area-exp/oa7-research-software
11 https://www.bwrse4hpc.de/68.php

2.3 Impact of AI and Automation

RSEng in 2035 D

RSEs understand how AI models work and AI-RSE specialists know how to tune and extend them. With these skills they can help researchers reap their benefits, while clearly knowing their limits.

Generative AI tools, such as GitHub Copilot [Fri21], are being added to software development tools [AA25]. These AI-powered code assistants promise to increase productivity by automatically generating code, and writing test cases and documentation [BHS25]. Software developers use them to reduce the number of keystrokes, focus on higher-level design rather than on idiosyncrasies, and recall syntax without having to consult the documentation [LYM24]. Code assistants have the potential to help novice or even non-expert programmers write code [FA24] thus enabling non-experts to participate in research and software creation, and fostering greater collaboration across disciplines. However, when these tools are used in practice, many users report increased cognitive load and frustration stemming from the difficulty to prompt the system effectively and from debugging the generated code [STK+25]. Many factors can influence the adoption of AI tools [Rus24]. The potential of AI to be a democratising technology can only be realised when the algorithmic divide along with the digital divide is addressed [Yu20]. We also note that members of the SE community have called for improving LLM literacy by adapting SE curricula [KKLM24].

Besides the direct impact in research software and further research output, we see a number of risks. The workshop participants suggested that the emergence of AI and automation might reduce critical thinking and imagination. We address this in Subsection 2.4. Another risk is that the widespread use of AI tools could make community resources such as e. g. Stack Overflow and Wikipedia (which are used to train AI models) poorer due to fewer incentives to maintain them. However, a recent survey suggests these communities experience less activity from new users and an increased average complexity of the questions posted [BLC24]. The use of AI assistants in these communities could also compromise LLMs via 'model collapse' when AI-generated content becomes part of the training data [SSZ+24].

Further risks that were later discussed include an intensified replication crisis, making research less accessible, and loss of trust in science. Due to the non-deterministic nature of generative AI, we expect AI-assisted simulations and LLM-assisted data analysis to become less reproducible. This is not an entirely new issue, and similar concerns exist in the HPC community, where reproducing simulation data and data analysis workflows presents tough challenges [AH24]. Professional RSEs could safeguard reproducibility by designing, evaluating, and monitoring appropriate workflows. When applicable, standardised and version-controlled models should be preferred over self-developed models, and self-developed models should be treated like any other research output (i.e. publish the trained model, training data and training workflow). Future research will require significant computing and storage resources, as well as useful data and models. Therefore, it will be more likely to be driven by institutions that are able to provide these. Finally, an increasing risk is the loss of trust in science, in particular due to the automation of the peer-reviewing process [Nad25] and the malicious use of hidden prompts [Gib25]. As researchers increase their reliance on AI assistants, RSEs will be freed from repeatedly crafting similar individual elements, but will be spending more of their time designing and reviewing software written by other humans and AI tools. To address these risks, the RSE of the future should not only steer every part of the research software development cycle (ensuring dissemination, archiving, and reproducibility), but also contribute to endangered community resources and infrastructure that will keep research accessible to the wider scientific community.

As AI continues to evolve, it will further blur the lines between human and machine roles, making research and software development faster. With the general availability of AI, the interaction of RSEs with researchers will change at a fundamental level. Whereas previously RSEs have been points of knowledge that could translate human language formulated problems into technical descriptions and then provide solutions to them, it can be foreseen that AI will drive this low-key interaction to a highly specialised level of knowledge. RSEs will still act as points of knowledge by assisting users in integrating AI tools in their development workflows, understanding the reasoning of AI models, and more generally, by improving AI literacy [AA25]. AI-RSE specialists can contribute to the emerging fields of explainable AI for SE (XAI4SE) [MBD+23] and AI for RSE (AI4RSE) [FBB+25].

Story 4: The interaction of RSEs with AI

Kim develops advanced scientific software in tandem with an AI agent that codes, debugs, and simulates experiments alongside her. The AI proactively suggests optimisations, generates reproducible workflows, and adapts to Kim's coding style and domain-specific needs. With this assistance, Kim's research output is amplified while maintaining full control over the scientific direction and ethical oversight.

2.4 Impact of Complexity and Usability in Research Software

RSEng in 2035 E

With a palette of readily reusable software components, and with their skills in managing hierarchical systems of complex software components with clear interfaces, RSEs can now build reliable, maintainable systems of a scale and capability hitherto not possible.

The workshop participants found that, in earlier learning stages, training may abstract away complexities that will always be present in research. For this reason, the participants fear that the capability of researchers to deal with complexity of software frameworks and libraries will reduce over time. As a result, RSEs will need to invest more effort in making the interfaces of individual research software components less complex, as the user experience expectations of researchers will continue to grow. Participants expect an over-reliance on generative AI tools as an extreme manifestation of this issue. No concrete examples were given on which complexities are now hidden behind abstractions.

We acknowledge but would also like to challenge the observations of the participants. We believe that "we are standing on the shoulders of giants": By reducing the complexity of individual software components, we are now able to develop systems of complexity and capability hitherto not possible. This divide-and-conquer approach allows the scientific community to address ever larger and more complex problems. When it comes to teaching, it is true that students need to learn how to handle this complexity. They need to develop a deep understanding of the individual components. However, it is also important to learn how to manage complexity by building hierarchical systems of well-encapsulated, reusable components. It is clear that research will need both RSEs that can dive deep into single projects of significant complexity, as well as RSEs that can manage complex systems of almost black-box components. The future of RSE education should aim to prepare students for both roles. We note members of the SE community independently reached the same conclusions and advocate for the creation of an 'Abstraction Engineering Body of Knowledge' (AEBoK)[BCC+24].

Story 5: The benefits of abstraction

Years ago, Kim had to distribute the software library libkim as a source code package with a custom build system and detailed instructions on how to build it on some common systems, investing significant time in developing and maintaining this auxiliary toolkit and documentation. Around the same time, Kay was developing the research analysis tool kay-bio, which however only provided an application programming interface that required writing complex code even for simple and common analysis tasks. Nowadays, Kim is able to start directly from a template based on established toolkits and workflows, while kay-bio accepts configuration in a simpler, standardised format. Generative AI assistants can guide new users in preparing and debugging their analyses. Meanwhile, new RSEs can very easily integrate libkim, kay-bio, and a multitude of other tools from the research software corpus into a powerful analysis platform that allows researchers to get deeper insights from their research data in just a few clicks.

2.5 Ethics and Social Considerations

RSEng in 2035 F

The principles of Good Scientific Practice [Deu22] have been a cornerstone in the conduct of research. However, their relevance hinges on constant updates in light of how research is performed. By 2035, accepted rules on the use of AI in research tasks and software development have been achieved. Open Science and Open Research are by then well-established in academia, and supported by respective open research software strategies of the political actors. A strong culture of data awareness and protection has been sustained and is actively maintained across the research community.

The ethics and values of the RSE are grounded in their practice. They commit to the health, safety and welfare of the public and act in the interest of society, their employer and their clients [GAB+24].

Sometimes these values may conflict with the aims of their peers or institution. For example, their PI might insist on quickly developing some software without adhering to current best practices. A code of ethics and professional conduct for RSEs should be developed by institutions together with RSEs themselves. This can then form the basis for institutional policies and guidelines.

The work being undertaken by RSEs may well have direct, or indirect, ethical implications in areas such as social engineering [s2e25, SJS+24], climate [LAB+23] and more. Like all areas of research and engineering, RSEs are faced with the dual use dilemma [BH] where artefacts they develop can be used for military purposes. As engineers, RSEs take the responsibility for the software they develop. This includes an awareness of ethical issues that may arise. RSEs need to be aware of these issues and develop their own "red lines" which need to be supported by organisational and professional guidelines. With the advantages of Subsection 2.3, ever-more AI created artefacts will be included in research. As a result of this, we expect a discussion within societies about their acceptable use.

Similarly to how data collected by scientists should adhere to the FAIR principles [WDA+16], research software developed by RSEs should adhere to the FAIR4RS principles [BCK+22]. Members of the SE community have also called for their field to embrace Open Science best practices[OMS+24, DEC+25].

Story 6: The greater world of Kim and Kay

10 years have passed since Kim and Kay started on their journey of being RSEs. Back then, at the universities of Eden and Utopia, life was more orderly and harmonious. Kim is currently designing the software service stack at the university. She increasingly struggles to find compelling Open Source alternatives that enables them to keep complete control of their data rather than relying on software and tools that are backed by large corporate entities. So far, she still has the support of the university board, since it values strategic autonomy. Kay had left academia to work in an industrial R&D department because of a more reliable employment. There she gained valuable experience working in an industrial setting. However, she was very happy when an opportunity arose to work in a central RSE department with an open-ended contract. Apart from her main responsibility of developing and maintaining software for an affordable water monitoring system used in the global south, she now also teaches undergraduates digital literacy.

2.6 Transformations in Academic Institutions

RSEng in 2035 G

Research software is a central part of scientific practice, on the same level as data and publications. This is supported by the first generations of RSE Master's graduates that find their way into research departments, and accelerate the pace of science. Similar to RSEs, who are widely known and understood by university boards in 2035, academic staff members at any level are aware of the concept of RSE units. Universities actively use the presence of an RSE Department as a means to differentiate themselves from other universities. Some central RSE units have local RSE representatives, or even small RSE groups that handle tasks within a department, overseeing specific projects [KCF+25]. Specialised clusters have formed, which pool the skills of specialised RSEs with RSE units specialising in certain tasks like data management, HPC, or visualisation. At universities which create such RSE units, it can be observed that researchers naturally accept them as part of their available services. Additionally, researchers interact seamlessly with infrastructures for using research software. Replacement of staff reaching retirement, combined with the intensified research interest in RSE, has led to the creation of more RSE professorships.

Academia, while claiming to be innovative [Gru19], has organisationally often adopted time scales for change that are more reminiscent to institutions thinking on the time scale of societal generations. During the next decade many positions at universities and research institutions will be vacated as the generation of the baby boomers come to retirement age. The RSE community should plan ahead and use this opportunity to fill some of these positions with RSEs and other RTPs. This generational change opens up the opportunity for an active change within the research landscape. A simple possibility would be to realign newly appointed professorships with a focus towards software development within their domain. A second, more progressive step would be to actually appoint new cross-faculty professorships. Already today the presence, of initiatives like *the Hidden Ref* [The25] hints at a potential for specialisation of research tasks. This gives rise to new job descriptions within the academic system such as specialised RSEs or other RTPs. Some specialisation trends have already been observed in recent years, such as the HPC-RSE and AI-RSE, which are quite recent developments, and whose appeal is stimulated by large public and private investments in HPC facilities and AI technologies. We foresee new RSE specialisations emerging, e. g., in the fields of quantum computing, such as the Quantum-RSE, or in digital research infrastructure. We note the SE community has also independently expressed interest in creating an *infrastructure engineering* role [Soc24] and funding bodies seem willing to promote such career pathways [UK 25, Eur24]. Taking these trends together the generational change should be used as an opportunity to reshape organisational structures to better reflect the work people in research are actually doing. The RSE community should take an active role in shaping this discussion.

While every faculty could hire the specialists they require, Universities benefit more from pooling these specialists into RSE units and offering their services to researchers in a structured way. This compartmentalisation will help researchers innovate in their research, with the support of sustainable software made by or with the help of RSEs. These RSE units should be endowed with enough trust by the university boards, translated into stable funding. With longer-term funding and more trust to pick up longer term projects, some RSEs could focus on code modernisation, clean-up, and modularisation, which could pay off in faster and easier development in the future. This is an opportunity that would not be possible if relying solely on short-term funded, isolated RSEs. We hope that research institutions will have recognised the importance of research software and the need to support the role of RSEs.

Besides all the improvements, the workshop participants still expect some challenges to persist, and new challenges to emerge. Due to the rise of dedicated RSE units, networking people and teams, federating infrastructures, as well as organising and sharing knowledge will become important. Structuring these new units and joining their administrative support will also help the units to grow. A particular example is hiring: While a standardised RSE Master's degree will make it easier to develop the next generation of RSEs, organisations could develop standardised hiring processes to acquire the skill set that they actually need. Budgeting and administrative hurdles will also probably still exist and hinder the formation of permanent and well-funded RSE teams. However, we expect the growing (perceived) importance of research software to convince research councils and funders to provide reliable streams of funding.

Story 7: Changes to the Universities

Kim found a job years ago at one of the first RSE units in Arcadia. After a couple of years, Kim moved to the university of Avalon which was interested in setting up their own department for RSEs in order to not fall behind the university of Arcadia. Kim builds upon the model set forth by the University of Arcadia based on her valuable experience. Since the university of Avalon has to follow different legal and administrative rules, not everything can be straightforwardly adapted. Life would be so much easier if at least university regulations were homogenised.

2.7 Impact of Alternative Career Paths for Researchers on the Structure of Research Institutes

RSEng in 2035 H

As the RSE becomes established, we are also seeing the emergence of a wider range of more specialised, technical roles within the research community, i.e. other (digital) RTPs. The legal requirements for employing people have become flexible enough to allow for the employment of more RTPs. Work outside of the publication record is more readily accepted and enables researchers to build careers at the interface of academia, industry, and policy (e. g. in think tanks, NGOs, or organisations that do knowledge transfer).

RSEs can either be embedded in research groups (belonging to the mid-level university staff), or be members of larger RSE units [KCF+25]. The participants had the opinion that focusing on embedded RSEs would make onboarding new RSEs to existing projects easier. This, however, comes with additional funding challenges, as long-term funding can be more difficult to acquire in lower tiers of a research organisation (i.e., at departmental or individual research group level). Third-party funding can mitigate this issue in the mid-term. However, the participants considered central RSE groups more likely in the future, due to the fact that there are already examples and activities from various universities in this direction (examples include the universities of Jena, Heidelberg, and Göttingen, as well as several universities in the UK where the concept of dedicated RSE groups began to emerge around 2013). We expect funding in central groups to evolve from initially temporary positions to permanent positions, as the groups prove themselves to be increasingly valuable and, hence, sustainable. Embedded RSEs and central groups will need to network with each other, and potentially request and coordinate support from external RSEs. Within Germany at least, the distinction around whether these positions are classed as research staff or technical staff has so far created some discussion. The authors hope that, 10 years from now, this discussion has been resolved towards a solution that enables the employment of RTPs more flexibly, and thereby recognising that the distinction between research staff and technical staff is too restrictive. A similar discussion is currently ongoing in the UK with initiatives like *the Technician's Commitment* [tec], and *the Hidden Ref* [The25], and finding a resolution with having new job titles, pay grades, and career options according to the level of work that you do. The authors expect these initiatives will have spread and will have found a wider resonance over the world by 2035.

Story 8: New career paths

Ken, though trained as a physicist, found his passion not in theory, but in keeping the quantum computers humming. He is the go-to expert for troubleshooting entangled qubits and stabilising delicate cryogenic systems. His unique blend of physics knowledge and hands-on problem solving earned him a new title among his peers: the first Quantum RTP. In a world where theory meets reality, Ken quietly ensures that the future of quantum computing stays on track.

3 What is in this Future for RSEs?

In the previous sections, we have tried to list some of the trends that we foresee. The material presented here is based on the outputs of a workshop, and extended with the authors' ideas. Nevertheless, there are some things that we declared out of scope, such as the ongoing digital transformation in wider society. Digital competencies in the general population continue to increase, making the jump to software development easier than before. But this digitalisation will also lead to changes in the way societies interact and therefore will translate into changes in how academia perceives itself. We have also not sufficiently delved into the interaction with industry. There are also some trends that we believe will stay with us over the coming years. The differentiation of job profiles in academia, that has started with profiles such as the RSE will continue. Societies will be facing new questions, arising from technologies such as generally available AI systems.

We anticipate that the standard of research software will have risen across the board, enabling easier reuse of prior software outputs in new research. We hope that a cultural change is taking place to recognise research software as an integral part of science that it is widely recognised and rewarded across academia and is supported by political actors.

4 Conclusion

We have tried to elaborate on the state of the future ten years from now. It turned out to be a difficult endeavour, since even the full breadth of today's trends is not yet foreseeable. We have tried to shed some light on the social, technical and organisational aspects. In summary, RSEs will very likely be doing different work to what they are doing today, but their core task of enhancing the scientific output of researchers – other human beings – will stay the same. They will still develop software artefacts together with scientists, since better software leads to better research, and better research leads to better software.

Acknowledgements: We thank all the participants of our workshop at deRSE25 in Karlsruhe for their valuable input! FG acknowledges funding from the Deutsche Forschungsgemeinschaft(DFG, German Research Foundation) through the SFB 1170 "Tocotronics", project Z03 - project number 258499086 as well as financial support by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy through the Würzburg-Dresden Cluster of Excellence on Complexity and Topology in Quantum Matter – ct.qmat (EXC 2147, project-id 390858490). JC acknowledges support from the UKRI-EPSRC-funded STEP-UP project under grant EP/Y530608/1.

Bibliography

[AA25] M. Alenezi, M. Akour. AI-driven innovations in software engineering: a review of current practices and future directions. *Applied Sciences* 15(3), 2025. doi:10.3390/app15031344

[ABD+21] H. Anzt, F. Bach, S. Druskat, F. Löffler, A. Loewe, B. Y. Renard, G. Seemann, A. Struck, E. Achhammer, P. Aggarwal, F. Appel, M. Bader, L. Brusch, C. Busse, G. Chourdakis, P. W. Dabrowski, P. Ebert, B. Flemisch, S. Friedl, B. Fritzsch, M. D. Funk, V. Gast, F. Goth, J.-N. Grad, J. Hegewald, S. Hermann, F. Hohmann, S. Janosch, D. Kutra, J. Linxweiler, T. Muth, W. Peters-Kottig, F. Rack, F. H. Raters, S. Rave, G. Reina, M. Reißig, T. Ropinski, J. Schaarschmidt, H. Seibold, J. P. Thiele, B. Uekermann, S. Unger, R. Weeber. An environment for sustainable research software in Germany and beyond: current state, open challenges, and call for action [version 2; peer review: 2 approved]. *F1000Research* 9(295), 2021. doi:10.12688/f1000research.23224.2

[AH24] B. Antunes, D. R. Hill. Reproducibility, replicability and repeatability: a survey of reproducible research with a focus on high performance computing. *Computer Science Review* 53:100655, 2024. doi:10.1016/j.cosrev.2024.100655

[BCC+24] N. Bencomo, J. Cabot, M. Chechik, B. H. C. Cheng, B. Combemale, A. Wąsowski, S. Zschaler. Abstraction Engineering. ArXiv preprint 2408.14074, Aug. 2024. doi:10.48550/arXiv.2408.14074

[BCK+22] M. Barker, N. P. Chue Hong, D. S. Katz, A.-L. Lamprecht, C. Martinez-Ortiz, F. Psomopoulos, J. Harrow, L. J. Castro, M. Gruenpeter, P. A. Martinez, T. Honeyman. Introducing the FAIR Principles for research software. *Scientific Data* 9(1):622, Oct. 2022.
doi:10.1038/s41597-022-01710-x

[BH] C. Bobier, D. J. Hurst. Clarifying Dual Use Research of Concern. *Philosophy & Technology* 37(4):139.
doi:10.1007/s13347-024-00827-8

[BHS25] L. Banh, F. Holldack, G. Strobel. Copiloting the future: how generative AI transforms Software Engineering. *Information and Software Technology* 183:107751, 2025.
doi:10.1016/j.infsof.2025.107751

[BLC24] G. Burtch, D. Lee, Z. Chen. The consequences of generative AI for online knowledge communities. *Scientific Reports* 14(1):10413, May 2024.
doi:10.1038/s41598-024-61221-0

[Boe11] B. Boehm. *Some future software engineering opportunities and challenges*. Pp. 1–32. Springer Berlin Heidelberg, Berlin, Heidelberg, Germany, 2011.
doi:10.1007/978-3-642-15187-3_1

[BØLD22] B. Bygstad, E. Øvrelid, S. Ludvigsen, M. Dæhlen. From dual digitalization to digital learning space: Exploring the digital transformation of higher education. *Computers & Education* 182:104463, 2022.
doi:10.1016/j.compedu.2022.104463

[Bos16] J. Bosch. Speed, data, and ecosystems: the future of software engineering. *IEEE Software* 33(1):82–88, 2016.
doi:10.1109/MS.2016.14

[CAHJ25] N. P. Chue Hong, S. Aragon, S. Hettrick, C. Jay. The future of research software is the future of research. *Patterns* 6(7):101322, 2025.
doi:10.1016/j.patter.2025.101322

[COA] The Coalition for Advancing Research Assessment (CoARA).
https://coara.eu/

[CSH22] A. Carleton, F. Shull, E. Harper. Architecting the future of software engineering. *Computer* 55(9):89–93, 2022.
doi:10.1109/MC.2022.3187912

[DCG+25] J. Dehne, S. Christ, F. Goth, J.-N. Grad, M. Hagdorn, J. P. Thiele, H. von Waldow. Research Software Engineering for Natural Sciences. GitHub, 2025.
https://github.com/the-teachingRSE-project/RSECurriculum4NaturalScience

[DEC+25] S. Druskat, N. U. Eisty, R. Chisholm, N. P. Chue Hong, R. C. Cocking, M. B. Cohen, M. Felderer, L. Grunske, S. A. Harris, W. Hasselbring, T. Krause, J. Linxweiler, C. C. Venters. Better architecture, better software, better research. *Computing in Science & Engineering* 27(2):45–57, 2025.
doi:10.1109/MCSE.2025.3573887

[Deu22] Deutsche Forschungsgemeinschaft. Guidelines for Safeguarding Good Research Practice. Code of conduct version 1.1, Apr. 2022.
doi:10.5281/zenodo.6472827

[DOR] The Declaration on Research Assessment (DORA).
https://sfdora.org/

[Eur24] European High-Performance Computing Joint Undertaking. Paving the way for the EuroHPC Federation Platform. Dec. 2024.
https://www.eurohpc-ju.europa.eu/paving-way-eurohpc-federation-platform-2024-12-19_en

[FA24] M. Q. Feldman, C. J. Anderson. Non-expert programmers in the generative AI future. In Cecchinato et al. (eds.), *Proceedings of the 3rd Annual Meeting of the Symposium on Human-Computer Interaction for Work (CHIWORK '24)*. Pp. 1–19. Association for Computing Machinery, New York, USA, June 2024.
doi:10.1145/3663384.3663393

[FBB+25] S. Farshidi, K. Bennin, ö. Babur, J. Sallou, A. Kassahun, B. Tekinerdogan. Advancing research software engineering with AI: a research framework. Research Square preprint, 2025.
doi:10.21203/rs.3.rs-7178452/v1

[Fri21] N. Friedmann. Introducing GitHub Copilot: your AI pair programmer. GitHub blog, June 2021.
https://github.blog/news-insights/product-news/introducing-github-copilot-ai-pair-programmer/

[GAB+24] F. Goth, R. Alves, M. Braun, L. J. Castro, G. Chourdakis, S. Christ, J. Cohen, S. Druskat, F. Erxleben, J.-N. Grad, M. Hagdorn, T. Hodges, G. Juckeland, D. Kempf, A.-L. Lamprecht, J. Linxweiler, F. Löffler, M. Martone, M. Schwarzmeier, H. Seibold, J. P. Thiele, H. von Waldow, S. Wittke. Foundational Competencies and Responsibilities of a Research Software Engineer [version 1; peer review: 2 approved]. *F1000Research* 13(1429), 2024.
doi:10.12688/f1000research.157778.1

[GCH+25] F. Goth, G. Chourdakis, M. Hagdorn, J. P. Thiele, J. Cohen, M. Martone, T. Hodges, M. G. Barrios Sazo. de-RSE25 BoF: The End of RSEng? Challenges and Risks for RSEng. July 2025.
doi:10.5281/zenodo.16541820
https://doi.org/10.5281/zenodo.16541820

[Gib25] E. Gibney. Scientists hide messages in papers to game AI peer review. *Nature* 643(8073):887–888, 2025.
doi:10.1038/d41586-025-02172-y

[Gru19] Grundsätze der staatlichen bayerischen Hochschulen - zum Umgang mit Befristungen nach dem WissZeitVG und zur Förderung von Karriereperspektiven für den wissenschaftlichen Nachwuchs. 2019.
https://www.stmwk.bayern.de/download/13009_Grundsaetze_Befristungen_Fassung_2019_mErgDKuSG.pdf

[HD23] X. Hu, H. K. Dam. Future of Software Engineering @ ICSE 2023. In *2023 IEEE/ACM international conference on software engineering: future of software engineering (ICSE-FoSE)*. Pp. 1–3. 2023.
doi:10.1109/ICSE-FoSE59343.2023.00005

[HDB+24] W. Hasselbring, S. Druskat, J. Bernoth, P. Betker, M. Felderer, S. Ferenz, A.-L. Lamprecht, J. Linxweiler, B. Rumpe. Toward Research Software Categories. 2024.
doi:10.48550/arXiv.2404.14364

[KCF+25] D. Kempf, R. Caspart, B. Flemisch, F. Goth, J. Linxweiler, F. Löffler, P. M. Schäfer, R. Speck, A. Struck, M. J. Ankenbrand, L. J. Castro, I. Ehlert, J.-N. Grad, M. Hagdorn, A. Loewe, M. Schlottke-Lakemper, U. Schmitt, P. S. Sommer. Developing central Research Software Engineering units in German research institutions. GitHub, 2025.
https://github.com/DE-RSE/2023_paper-RSE-groups

[KH23] D. S. Katz, S. Hettrick. Research Software Engineering in 2030. In *2023 IEEE 19th International Conference on e-Science (e-Science)*. 2023.
doi:10.1109/e-Science58273.2023.10254813

[KKKN21] F. Khan, R. L. Kumar, S. Kadry, Y. Nam. The future of software engineering: visions of 2025 and beyond. *International Journal of Electrical and Computer Engineering (IJECE)* 11(4):3443–3450, Aug. 2021.
doi:10.11591/ijece.v11i4.pp3443-3450

[KKLM24] V. D. Kirova, C. S. Ku, J. R. Laracy, T. J. Marlowe. Software engineering education must adapt and evolve for an LLM environment. In *Proceedings of the 55th ACM technical symposium on computer science education (SIGCSE 2024), volume 1*. Pp. 666–672. Association for Computing Machinery, New York, USA, Mar. 2024.
doi:10.1145/3626252.3630927

[LAB+23] L. Lannelongue, H.-E. G. Aronson, A. Bateman, E. Birney, T. Caplan, M. Juckes, J. McEntyre, A. D. Morris, G. Reilly, M. Inouye. GREENER principles for environmentally sustainable computational science. *Nature Computational Science* 3(6):514–521, June 2023.
doi:10.1038/s43588-023-00461-y

[LYM24] J. T. Liang, C. Yang, B. A. Myers. A large-scale survey on the usability of AI programming assistants: Successes and challenges. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*. ICSE '24, pp. 1–13. Association for Computing Machinery, New York, USA, Feb. 2024.
doi:10.1145/3597503.3608128

[MBD$^+$23] A. H. Mohammadkhani, N. S. Bommi, M. Daboussi, O. Sabnis, C. Tantithamthavorn, H. Hemmati. A systematic literature review of explainable AI for software engineering. ArXiv preprint 2302.06065, 2023.
doi:10.48550/arXiv.2302.06065

[MOL$^+$19] E. Meade, E. O'Keeffe, N. Lyons, D. Lynch, M. Yilmaz, U. Gulec, R. V. O'Connor, P. M. Clarke. The changing role of the software engineer. In Walker et al. (eds.), *Systems, software and services process improvement*. Pp. 682–694. Springer International Publishing, Cham, 2019.
doi:10.1007/978-3-030-28005-5_53

[Nad25] M. Naddaf. AI is transforming peer review – and many scientists are worried. *Nature* 639:852–854, 2025.
doi:10.1038/d41586-025-00894-7

[OMS$^+$24] E. OliveiraJr, F. Madeiral, A. R. Santos, C. von Flach, S. Soares. A vision on Open Science for the evolution of Software Engineering research and practice. In *Companion proceedings of the 32nd ACM international conference on the Foundations of Software Engineering (FSE 2024)*. Pp. 512–516. Association for Computing Machinery, New York, USA, July 2024.
doi:10.1145/3663529.3663788

[Rus24] D. Russo. Navigating the complexity of generative AI adoption in software engineering. *ACM Transactions on Software Engineering and Methodology* 33(5), June 2024.
doi:10.1145/3652154

[s2e25] Security and Software Engineering Research Center (S^{2}ERC). 2025.
https://iucrc.nsf.gov/centers/security-and-software-engineering-research-center/

[SJS$^+$24] H. Siadati, S. Jafarikhah, E. Sahin, T. B. Hernandez, E. L. Tripp, D. Khryashchev, A. Kharraz. DevPhish: exploring social engineering in software supply chain attacks on developers. ArXiv preprint 2402.18401, 2024.
doi:10.48550/arXiv.2402.18401

[Soc24] V. Sochat. Infrastructure engineering: a still missing, undervalued role in the research ecosystem. ArXiv preprint 2405.10473, May 2024.
doi:10.48550/arXiv.2405.10473

[SSZ$^+$24] I. Shumailov, Z. Shumaylov, Y. Zhao, N. Papernot, R. Anderson, Y. Gal. AI models collapse when trained on recursively generated data. *Nature* 631(8022):755–759,

July 2024.
doi:10.1038/s41586-024-07566-y

[STK+25] A. Simkute, L. Tankelevitch, V. Kewenig, A. E. Scott, A. Sellen, S. Rintel. Ironies of generative AI: understanding and mitigating productivity loss in human-AI interaction. *International Journal of Human–Computer Interaction* 41(5):2898–2919, Mar. 2025.
doi:10.1080/10447318.2024.2405782

[tec] The Technician Commitment.
https://www.techniciancommitment.org.uk/

[The25] The hidden ref. 2025.
https://hidden-ref.org/

[UK 25] UK Research and Innovation. Digital research infrastructure programme. 2025.
https://www.ukri.org/what-we-do/creating-world-class-research-and-innovation-infrastructure/digital-research-infrastructure/

[WDA+16] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. T. Evelo, R. Finkers, A. Gonzalez-Beltran, A. J. G. Gray, P. Groth, C. Goble, J. S. Grethe, J. Heringa, P. A. C. 't Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S. J. Lusher, M. E. Martone, A. Mons, A. L. Packer, B. Persson, P. Rocca-Serra, M. Roos, R. van Schaik, S.-A. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M. A. Swertz, M. Thompson, J. van der Lei, E. van Mulligen, J. Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao, B. Mons. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data* 3(1):160018, Mar. 2016.
doi:10.1038/sdata.2016.18

[Yu20] P. K. Yu. The algorithmic divide and equality in the age of artificial intelligence. *Florida Law Review* 72(19-44):331–89, 2020.
https://ssrn.com/abstract=3455772