



[us-rse.org](http://us-rse.org)

# Research Software Engineers

## Creating a Career Path— and a Career

in collaboration with



# TABLE OF CONTENTS

Introduction .....	3
RSE: A Role Evolves .....	8
The RSE's Role .....	9
Evolving Perceptions of the RSE .....	10
How to Build a Career as an RSE .....	13
Prepare: Basic Skills .....	15
Practice: Opportunities Abound .....	17
Engage: RSE Groups and Conferences .....	19
Why Research Software Engineering? .....	23





## INTRODUCTION

Scholarly research typically starts with a question, evolves into a project, and sometimes delivers results that radically change our world.

The Large Hadron Collider, for example, has altered our understanding of matter and the universe, while the World Wide Web has revolutionized how people across the world connect, do business, and access information. But whether it produces reality-altering results such as these or problem-specific outputs—say, an eye-tracking study of threat-related attention bias in anxious people—most scholarly research relies on the same key resource: software.



LOOKING AHEAD

## A Community of People Who Make Research Software Happen

The US Research Software Engineer Association (US-RSE) is a community-driven effort that brings together people who write and contribute to research software within the US. We aim to build a connective, supportive, and diverse community of individuals from universities, laboratories, knowledge institutes, companies, and other enterprises. We strive to advocate for our members, raise awareness of the importance of the RSE role, and actively improve diversity within the RSE profession.

Explore more at  
<https://us-rse.org>.

Software underpins research, facilitating everything from experiment design and data analysis to simulations of complex phenomena and the presentation of research results. Despite software's central role, however, many project members who build and manage research software are often self-taught coders who know or learn only enough about software development to create the needed tool. Few such coders have experience with software best practices and how to ensure that the code they build is robust and sustainable over time.

This scenario is rapidly changing, however, due to a global movement of research software professionals. More than a decade ago at a Software Sustainability Institute (SSI) workshop at the University of Oxford, a group of these professionals coined a job title—research software engineer (RSE)—to describe their work

and launched a movement. Their goals were and are to not only illuminate the RSE's critical role in scholarly research, but also to create a defined career path in a research world that increasingly relies on specialized software skills.

“Computation now permeates everything, so the future of research is in the hands of RSEs,” says Caroline Jay, a professor of computer science at the University of Manchester, where she also is head of research in the School of Engineering. “I hope to see job opportunities for RSEs become more diverse as software is valued more, including as an output of research in its own right.” This, says Jay, will feed “even greater opportunities—and a very real need—for people with RSE skills to take on leadership positions.”

The benefits of pursuing the emerging RSE career path are numerous, both for

researchers who like to code and for mainstream software engineers looking to work in an environment that thrives on exploration, flexibility, and creativity.

Rinku Gupta, a senior research software specialist at Argonne National Laboratory, has 20 years' experience creating scientific research software on high-performance computing (HPC) platforms. Working at the intersection of software development and research, Gupta says, has allowed her to contribute to cutting-edge scientific research that has the potential to impact the world and change lives.

"It feels great to be a part of a team doing research that seems to be pushing the boundaries of knowledge and developing innovative solutions to complex problems," Gupta says. "I also appreciate the diversity of work that comes with being an RSE. Every project is unique and presents its own set of challenges, which keeps me engaged and motivated. I'm constantly learning and acquiring new skills and knowledge in different areas of scientific computing and computer science, which has been invaluable for my professional development."



Computation now permeates everything, so the future of research is in the hands of RSEs.

## Meet the Experts



Caroline Jay / Software Sustainability Institute and University of Manchester

Caroline Jay is a Professor of Computer Science and the Research Director of the Software Sustainability Institute, where she investigates and advocates for the role of software in furthering open and reproducible science. She is Head of Research in the School of Engineering at the University of Manchester, a Chartered Psychologist, and a Fellow of the Alan Turing Institute, using an understanding of human behavior to improve the way we design Artificial Intelligence. Caroline works across a diversity of areas, including creative media, medicine, and environmental science. In 2022, Caroline was named one of the Guardian and Women in Engineering Society's Top 50 Women in Engineering, for her work advancing health technology.

Daniel S. Katz / NCSA and University of Illinois Urbana Champaign



Daniel S. Katz is Chief Scientist at the National Center for Supercomputing Applications (NCSA) and Research Associate Professor in Computer Science, Electrical and Computer Engineering, and the School of Information Sciences at the University of Illinois Urbana Champaign. His interests include policy issues, such as citation and credit mechanisms and practices associated with software and data, organization and community practices for collaboration, and career paths for computing researchers. He is a senior member of the IEEE and ACM, co-founder and current Associate Editor-in-Chief of the Journal of Open Source Software, co-founder of the US Research Software Engineer Association (US-RSE), and co-founder and steering committee chair of the Research Software Alliance (ReSA).

## Meet the Experts



Jeremy Cohen / Imperial College London

Jeremy Cohen is an Advanced Research Fellow in the Department of Computing and Director of Research Software Engineering (RSE) Strategy at Imperial College London. His current work focuses on improving practices and processes around the development and use of software within the research community, with a view to improving the quality, sustainability, and reproducibility of research outputs. Jeremy has a Computer Science background and a Ph.D. in computing from Imperial College London. He has recently completed a 5-year RSE Fellowship funded by UK Research and Innovation's Engineering and Physical Sciences Research Council. Jeremy started, and leads, Imperial College's local Research Software Community and the regional Research Software London community, which provides skills development and networking opportunities for researchers, RSEs, and other Research Technology Professionals based at institutions in London and the South East of England.

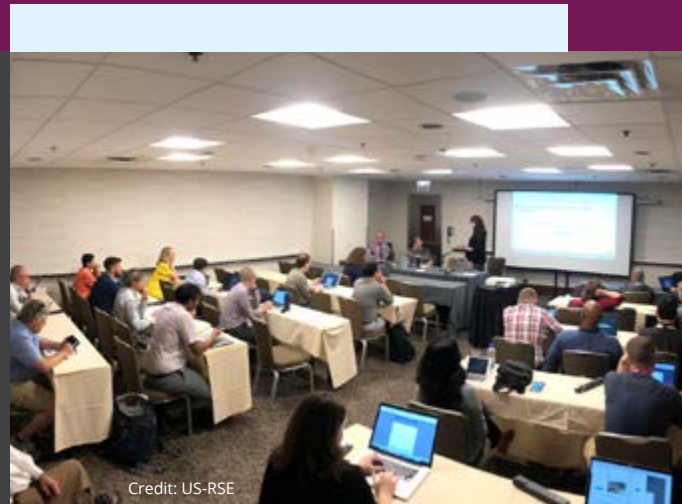
Rinku Gupta / Better Scientific Software



Rinku Gupta is a Senior Research Software Specialist with over two decades of experience working with scientific research software used on high performance computing platforms. She has several years of experience as a project manager and manages large teams and multi-institution-based exascale computing projects. Ms. Gupta serves as an Editor-in-chief of the Better Scientific Software website (BSSw.io), which serves as a central hub for the scientific community to address pressing challenges in software productivity, quality, and sustainability. She is well-known in the area of research software engineering for her leadership, and a recognized figure for her consistent work on promoting the Research Software Engineers (RSE) movement, both within Argonne, and the broader supercomputing community.

# RSE: A Role Evolves

Many people creating software for scientific research today continue to be solely self-taught and focused only on immediate project requirements.



Credit: US-RSE

Jeremy Cohen, advanced research fellow in the Department of Computing and director of RSE strategy at Imperial College London, says that one result of this ad hoc approach is that people outside the research arena sometimes perceive the quality of research software to be less than stellar.

“With the professionalization of research software development and the greater focus on best practices and ensuring reproducibility of results, things are changing rapidly,”

says Cohen. “RSEs have contributed significantly to supporting this change and will continue to do so.”

Today, RSEs and RSE groups at universities are increasingly common, with RSEs participating in research projects across academic disciplines, from physics and aeronautics to humanities and the social sciences. Private, government, and nonprofit research labs also support the RSE role, and many have existing or emerging RSE groups.



## The RSE's Role

As embedded members of research teams, RSEs write, manage, and maintain the software for scholarly research, whether by building a tool from scratch or adding needed features to existing software. RSEs apply software best practices, including testing and documentation, to ensure that the software is sustainable and that a project's results are reproducible. Soft skills are also critical, as communicating with research team members and scientists from a different domain (e.g. biology, chemistry, and physics) throughout the project is essential to ensure that the software meets the

project's needs, which typically evolve over time.

“When people with an enterprise background join a research team, they often expect there to be a level of order that just isn't realistic—for example, that diverse datasets from completely different academic groups will all look exactly the same,” says Jay, adding that the software creation process is also often “exploratory and iterative. We may not know the end goal of the project in advance, or [we may] have to pivot quickly.”

Developing software for research teams differs from traditional software engineering in other ways as well, including

The software creation process is also often “exploratory and iterative. We may not know the end goal of the project in advance, or [we may] have to pivot quickly.

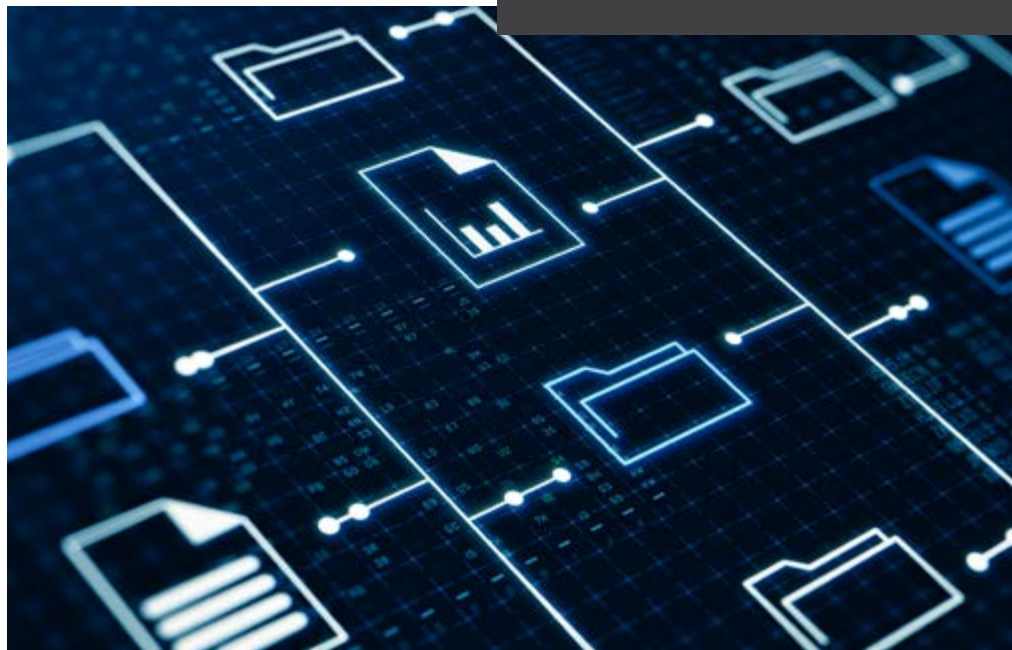


that it is often built not for use by others but rather to generate results, process data, or simulate phenomena to support outputs for publication in research papers.

In addition to developing and communicating about software on a range of research projects, RSEs might also conduct training across their campuses or labs, and advocate for quality software and the practices that produce it. The RSE role and career can thus offer variety and creativity, as well as a focus on helping society and generating new knowledge that traditional software engineering can rarely match.

## Evolving Perceptions of the RSE

Despite the increasing recognition of research software and its importance, many misconceptions about the RSE role remain. Daniel S. Katz, cofounder of the US Research Software Engineer Association (US-RSE), is chief scientist at the National Center for Supercomputing Applications at the University of Illinois Urbana



Champaign, where he is also a research associate professor in several departments, including the Department of Computer Science. Among the ongoing misconceptions about RSEs that Katz sees are that people assume that RSEs “are code monkeys, just programming what someone else tells them to” or, in contrast, that they are “typical software engineers, using full software engineering best practices in all cases.”

So, while awareness of research software engineering and its contribution to research continues to grow globally, much more change and growth in awareness is needed—including at a basic level: “More is needed from industry in terms of recognizing RSEs and using the RSE name,” says Katz. “There might be 300,000 RSEs in the world, but only 10,000 or so identify as RSEs now.”

As Katz notes, titles are still varied for people who create research software and range

from simply “graduate student” to “software engineer” to “research programmer.” However, the RSE title is becoming increasingly common, particularly in the UK and the US. According to the [2022 RSE International Survey](#), which includes data from a total of 589 respondents who create—or manage teams that create—research software in more than 40 countries, “research software engineer” was by far the most common job title among respondents in the UK, the US, and the Netherlands.

Katz says that as more organizations recognize the RSE name and skill set, more opportunities will arise from researchers who see the value that these professionals bring to projects. Likewise, as universities increasingly recognize RSEs, they are more likely to support RSE groups and thus increase RSE job opportunities. Gupta agrees,

adding that heightened awareness will add prestige to the RSE career track and differentiate it from traditional software engineering.

“This recognition and acceptance of the RSE career track can open up new doors and opportunities for the entire RSE community,” says Gupta. “Moreover, increased

recognition and acceptance of the RSE profession could lead to younger generations in colleges becoming more aware of this career track and aspiring to it. This could lead to a domino effect of increased talent, training, rewards, and recognition for individuals and the RSE community as a whole.”



Increased recognition and acceptance of the RSE profession could lead to younger generations in colleges becoming more aware of this career track and aspiring to it.

# How to Build a Career as an RSE



The RSE role “falls in between that of a researcher and a software engineer. It blends some aspects of each role,” says Cohen, “and the extent to which individual RSE roles may look more like that of a researcher or a traditional software engineer vary significantly.”

# How to Build a Career as an RSE



As a result, he says, both researchers and professional software engineers may have trouble fully identifying with the role of an RSE—and erroneously assume that it isn't open to them as a career option. In fact, people with both backgrounds bring great strengths to the table.

With research software engineering, the specifics of the career path depend on where you're starting: those who are already creating research software may need further education in software best practices, while software engineers may need education in the scholarly research process in general or in a specific field. For students interested in an RSE

career, Gupta suggests starting with software.

"Focus first on building your software engineering skills by working on programming languages—after all, you are going to be developing and writing code—and learning about software development best practices," she says, then following up this best-practices education by actually creating a software from inception to end-of-life. "These are universal skills that can be useful whether you're working in industry, academia, or research-lab environments."

As with any career, the key to entering or advancing as an RSE involves preparation, practice,

and engagement with the thriving RSE community.

## Prepare: Basic Skills

RSEs need solid software development and project skills, including skills in software design, testing, and documentation. Among the common programming languages used in research are Python, C++, R, SQL, JavaScript, C, Fortran, and MATLAB. Also, RSEs are embedded in research projects where they must translate project needs into software requirements—and translate complex technical concepts into language that nontechnical stakeholders can easily understand—so communication and collaboration skills are crucial. Finally, RSEs need an agile mindset and sound project management skills.

“An important aspect of the work that prospective RSEs may overlook is the ability to deal with ambiguity when it comes to concrete software requirements, product definition, and metrics of success for the end product,” says Gupta. “In research projects, the goals and outcomes may not always be well-defined or may change as the project progresses. RSEs need to be able to adapt to these changes and work collaboratively with researchers to determine the best course of action. This requires strong communication skills, critical thinking, and the ability to work independently and as part of a team.”

Many of these skills can be honed on the job—whether your current job is in traditional software engineering or in a research environment. For students, internships at research organizations or RSE groups in academia or the

research industry offer similar opportunities and also let them begin to network, establish relationships, and learn about research practice. In fact, all prospective or advancing RSEs can take advantage of the growing RSE awareness, which can lead to project overload for established RSE groups—and create skill-learning opportunities.

“The current growth and popularity of research software engineering means that many teams have more opportunities for work than they are able to take on,” says Cohen. “If you already have good programming

skills, you may be able to help out on active open source projects and provide a valuable additional resource for the team whilst gaining great experience along the way.”

Taking advantage of such opportunities can expand your network and your understanding of diverse research areas, which can offer further benefits and open new doors.

“It’s important to be able to communicate with people in a variety of research disciplines, and understand their goals and problems,” says Katz. “Such discussions are often how new collaborations start.”

In research projects, the goals and outcomes may not always be well-defined or may change as the project progresses. RSEs need to be able to adapt to these changes and work collaboratively with researchers to determine the best course of action.





## Practice: Opportunities Abound

While on-the-job and internship experiences can be essential, contributing to open source projects or taking advantage of in-person or online training can also increase your knowledge and your value in the job market.

## Open Source Opportunities

Participating in open source projects or starting your own is an excellent way to develop the skills you need to establish or elevate your RSE toolkit. Resources for people new to the open source world include the following:

- Open source enthusiast Hillary Nyakundi offers a comprehensive overview of open source project work, including the various roles and [how to get started](#), as

well as a list of [open source projects](#) with detailed descriptions.

- Becky Arnold, an RSE at University of Sheffield and an SSI fellow, offers another [overview of contributing to open source projects](#).
- The UpGrad online education site includes a list of the [top eight projects for beginners](#).

Many options exist for finding open source projects at all levels. The following are a few to consider:

- The LINUX Foundation lists open source projects, [sortable by sector](#).
- Pangloss Labs lists open source [projects focused on addressing global challenges](#), ranging from climate change to hunger.
- The Eclipse Foundation offers a [searchable database of open source projects](#), as well as guides on how

to contribute and how to start your own open source project.

- Rocket Chat recommends “[16 open source projects to contribute to in 2023](#),” organized by programming language.

## Training and workshops

The abundance of training programs and workshops available can help you to build your strengths further and strengthen areas in which you have little to no experience. They're also a great place to meet peers and find support along the path. The following are some starting points to explore workshop and training opportunities:

- US-RSE has a comprehensive list of [training videos](#), [tutorials](#), and [training opportunities](#)—including links to other lists of resources by experience level.
- [INTERSECT](#), funded by the US National Science Foundation, offers links to training materials by topic and its own RSE training.
- The Carpentries Incubator has a beta version of its [Intermediate Software](#)



Credit: US-RSE

# How to Build a Career as an RSE



Credit: US-RSE

Development course, which focuses on skills and best practices for working as part of a research team.

- CodeRefinery is a training network that offers workshops, lessons, and self-study paths that provide essential knowledge of software, computing, and data for researchers.
- The Society of Research Software Engineers has a resources page with links to training and essential skills development opportunities.
- Forge RSE offers guidance on RSE skills development pathways.

## Engage: RSE Groups and Conferences

One of the best ways to understand your own RSE path and its next steps is by talking with—and especially, listening to—a range of developers and researchers, and reflecting on how you might apply what you learn from their career path and work experiences.

As Katz notes earlier, such conversations can help you to better develop crucial RSE skills that are often overlooked, including “being able to talk about research software” and

to talk with people across research disciplines. Developing these habits will also help you understand how to generalize software solutions across research problems and thereby build stronger general tools and practices.

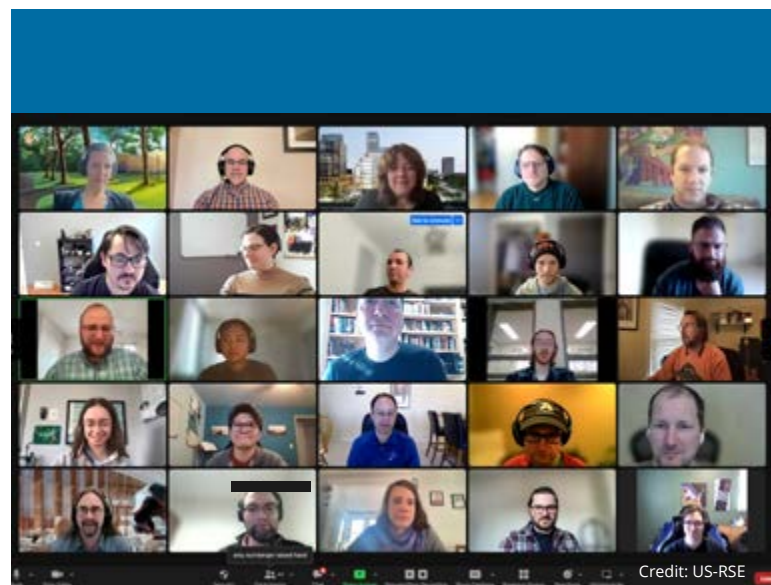
Two ways to ensure that you're having these conversations regularly is by engaging with the RSE community, such as by joining an RSE association or group, or by attending RSE workshops and conferences. You can also engage with your colleagues and ask them about their goals and their software issues.

## RSE Organizations

Research software communities exist at international, national, regional, and local levels. These organizations bring together RSEs who work in diverse domains and are in various career stages across industry, academia, and lab settings.

Their role in elevating research software engineering as a profession—as well as helping elevate and educate RSEs and RSE hopefuls—is pivotal and growing.

“Through these organizations, RSEs are developing a sense of belonging, to some degree,” says Gupta, adding that they also increase participation in RSE conferences and workshops, which in turn increase recognition of the RSE career track and create “working groups/task forces which can tackle research software



Credit: US-RSE

engineering challenges and come up with solutions.”

The Society for Research Software Engineering, which was formed in 2019 from the previous national UK RSE Association, is one entity looking to provide international support to RSEs. While the Society has developed out of a UK-focused community, membership is open to anyone in the world “who supports the Society’s aims of furthering researcher software engineering.”

National groups coordinate national research software activities and advocacy for RSEs. Examples here include France’s well-established [DevLOG network](#) as well as RSE groups that emerged following the 2012 Oxford meeting, including [Australia/New Zealand](#), [Germany](#), and [the Netherlands](#)

[US-RSE](#) was cofounded by Katz and a number of US participants in the 2018 UK RSE Association’s International

Leadership Conference. US-RSE focuses on four overarching goals: community, advocacy, resources, and diversity, equity, and inclusion. In addition to its working groups, jobs board, and resources page, US-RSE hosts regular community calls and a [national conference](#).

Regional groups are also forming, particularly in the UK, to bridge the gap between large national organizations and the smaller local RSE groups or communities that are increasingly found in universities or nonacademic research labs.

Among the regional groups in the UK are

- [Research Software London](#)
- [RSE Midlands](#)
- [Community of Edinburgh Research Software Engineers](#)
- [Supercomputing Wales](#)

Regardless of size, RSE groups offer support to RSEs and raise the profile of research software.

Most also offer opportunities for experienced and aspiring RSEs to get involved and share experiences and lessons learned, as well as learn about research software standards and practices, and RSE job and training opportunities.

“Research software communities often survive through support from volunteers helping to keep things running and organizing events,” says Cohen. “While large national communities may have some sort of financial support, this is often not the case for local communities within institutions. If you get in touch and offer to help, the people running the community will be delighted to hear from you and only too happy to have someone enthusiastic and interested to get involved and help out.”

## Conferences

Following are a few of the annual RSE and RSE-related conferences, along with dates

and locations for upcoming events:

- [Collaborations Workshop](#)
- [Annual Conference for Research Software Engineering](#)
- [US Research Software Engineer Association \(US-RSE\) Conference](#)

There are also RSE workshop at other conferences, such as the International Conference for High Performance Computing, Networking, Storage, and Analysis (SC'xy) and IEEE eScience.



# Why Research Software Engineering?

Being an RSE can offer greater variety and creativity than building software in other industries; it also offers opportunities to collaborate with people who are solving some of the world's most challenging problems.

"Being an RSE has given me the opportunity to work on meaningful projects, collaborate with brilliant researchers, and constantly challenge myself to grow and learn," says Gupta. "It's a fulfilling and rewarding career path that I feel fortunate to be a part of."

A career in research software engineering can also let you merge your knowledge of and passion for other domains with your software engineering work and give you a unique skill set, whether that passion is for chemistry or history or any other among myriad research areas.

“Don’t underestimate the value of your domain knowledge,” says Cohen. “While technical RSE skills are highly transferable and RSEs may switch between projects in different fields, there is also strong demand for people who have training in specific fields ... [which can] make it easier for you to understand researchers’ requirements in areas that you’re familiar with and help you to build an effective collaboration.”

Although all of these benefits are sizable, many RSE professionals

will tell you that the greatest benefit of working in the RSE world is the community.

“The RSE community has worked to develop a strong ethic of being open and collaborative,” Cohen says, adding that this includes “sharing of technical skills, ideas, and perspectives through community training efforts that help both new and more experienced community members to grow and maintain their skills.”

[Many] RSE professionals will tell you that the greatest benefit of working in the RSE world is the community.







[us-rse.org](http://us-rse.org)

