April
2017

Research Software Engineers: State of the Nation Report 2017

Most research would be impossible without software, and this reliance is forcing a rethink of the skills needed in a *traditional* research group. With the emergence of software as the pre-eminent research tool used across all disciplines, comes the realisation that a significant majority of results are based, ultimately, on the skill of the experts who design and build that software.

The UK has led the world in supporting a new role in academia: the *Research Software Engineer* (RSE). This report describes the new expert community that has flourished in UK research, details the successes that have been achieved, and the barriers that prevent further progress.

Attendees at RSE Conference 2016

About the RSEN

The Research Software Engineer Network (RSEN) was formed to support Research Software Engineer activities in the UK.

In addition to providing administrative and financial support to grow the RSE community and help it achieve sustainability, the RSEN provides an annual summary - a state of the nation report - about RSE activities.

The RSEN is funded by the Engineering and Physical Sciences Research Council (EPSRC) through grant EP/N028902/1.

About this report

This report was written by Alys Brett, Michael Croucher, Robert Haines, Simon Hettrick, James Hetherington, Mark Stillwell and Claire Wyatt.

DOI: 10.5281/zenodo.495360

Contents

Executive summary

Most research would be impossible without software, and this reliance is forcing a rethink of the skills needed in a *traditional* research group. A survey of 15 Russell Group universities found that 92% of researchers used research software, 67% reported that it was fundamental to their research, and 56% said they developed their own software [1]. With the emergence of software as the pre-eminent research tool used across all disciplines, comes the realisation that a significant majority of results are based, ultimately, on the skill of the experts who design and build it.

The work of software experts in academia is rarely recognised, and they suffer poor conditions of employment relative to other research roles. Since 2012, a community of these experts has grown around a campaign to raise awareness of the people who build the software used in research. These people have worked under many titles, but many now identify as *Research Software Engineers*.

A vital, yet hidden, role

There are many software experts in academia (page 6), but there is no formal career path to support them.

Without a career path to recruit into, anyone wishing to employ a software expert in academia must overcome restrictions related to human resources, finances and funding policies - not to mention a culture in universities that tends to overlook the importance of software (page 8). A hotchpotch of solutions has been contrived to circumvent the lack of a career path, but this has forced software experts into a transient and unrecognised community. This severely limits access to software expertise. Indeed, despite their reliance on software, many researchers find it phenomenally difficult to employ and retain a software expert.

Without access to the appropriate skills, we cannot expect the development of research software that is reliable and reproducible. By supporting Research Software Engineers, we can significantly increase access to software engineering expertise in UK academia and advance research across all disciplines.

The growth of a new expert community will advance research

Over the last four years, the UK has led the world in recognising the importance of research software engineering. This has been the result of a grass-roots campaign, initiated and still supported by the Software Sustainability Institute [2], but since coordinated by the research software engineering community itself (page 10).

The campaign has witnessed the rapid growth of an active community of almost 800 Research Software Engineers, generated significant international interest as other countries look to emulate the UK's approach (page 11), seen the proliferation of a new type of group in academia that increases access to software expertise (page 17), and led to the creation of a fellowship that builds on these successes (page 19).

To maintain our position as a world leader in this field, we must convince all research stakeholders of the vital link between reliable software and reliable results, and we must make the employment of software experts straightforward and transparent by implementing a career path for Research Software Engineers (page 24). By providing the expertise needed by *modern* research groups, we will promote the development of well-engineered software that will increase the scope, productivity and reliability of research.

Timeline

Mar 2012

The term **Research Software Engineer** is coined at the Collaborations Workshop.

Late 2012

The new model of Research Software Group is **launched at UCL**.

"The Research Software Engineer" paper published at Digital Research 2012.

Jan 2013

The Software Sustainability Institute begins its **campaign for recognition and reward** of Research Software Engineers.

Aug 2013

"Top Ten Reasons to be a Research Software Engineer" receives **4,000 views** on first day of publication.

Sep 2013

After nine months raising recognition of RSEs, a workshop is organised to bring them together.

A unanimous vote at the workshop leads to the formation of the UK RSE Association.

Early 2014

Launch of the **UK RSE Association**.

160 members of UK RSE Assocation attracted in first three months.

Late 2014

UK RSE Association becomes democratic.

Elections take place at first AGM.

Research Software Group founded at **University of Manchester**.

Jun 2015

The EPSRC's RSE Fellowship attracts **211 applications** for three fellowships on offer.

Late 2015

Research Software Groups founded at **University of Cambridge, University of Southampton** and **Francis Crick Institute.**

440 members of UK RSE Assocation

Early 2016

Research Software Group founded at **University of Sheffield**.

Mid 2016

Invitations to speak about UK's RSE work to Canada and US.

"Top Ten Reasons to be a Research Software Engineer" hits **20,000 views**

Sep 2016

First RSE Conference sells out months in advance and attracts **202 attendees** from **14 countries**.

Election of new committee to run UK RSE Association.

Late 2016

Research Software Group founded at **Alan Turing Institute**.

735 members of UK RSE Assocation.

Research relies on software experts

Researchers solve problems that are highly specific to their field of expertise. These problems cannot be solved by the straightforward application of off-the-shelf software, so researchers develop tools that are tailored to their exact needs.

The most popular software packages used in research are all, broadly speaking, programming languages [3]. These allow researchers to develop their own programs to solve problems specific to their research. Some of these programs develop into substantial software packages with millions of lines of code, but to concentrate only on these headline-grabbing examples is to overlook the ubiquity of software in research. Millions of programs are used by researchers to analyse and transform their data. These are the real workhorses of research, and without them the vast majority of results could not be produced. Who writes these programs if there is no career path for a software expert in academia?

"Millions of programs are used by researchers...

...Who writes these programs if there is no career path for a software expert in academia?"

The current approach to recruiting software expertise is a hotchpotch of different solutions that have been developed to meet the disparate needs of local human resources and finance departments, university culture and restrictions from funders. This lack of consistency has created an unrecognised and tenuous existence for the people who develop software in academia, and this severely limits the number of people who can help researchers benefit from software.

More than just software engineering

The investigatory methods used in research constantly evolve as problems are uncovered and new hypotheses are tested. This makes research wholly unsuitable to an *over the fence* approach to software development (where an expert builds software to meet a specification written by a researcher). By the time the software is written, the problem has often changed beyond recognition. What's more, researchers are not trained to write software specifications, and they are rarely aware of the latest software or practices that could advance their work. To be effective, software development in research should be approached, not as a one-off transaction, but as a partnership between researcher and software expert. A significant majority (70%) of Research Software Engineers hold a PhD [4], so they assimilate easily into research teams and benefit from a preexisting understanding of research objectives and incentives. A partnership of this form allows the code to evolve with the research, with the expert guiding the research into the use of more productive technology, and ensures that the software meets the reliability and reproducibility standards that we expect from any research tool.

This does not mean that all research groups should recruit a Research Software Engineer on a permanent basis. Different groups will have different requirements. Large, technologically dependent groups will need to employ many Research Software Engineers, small groups with only sporadic periods of software development may look to share research software engineering expertise or to work with centralised groups of Research Software Engineers.

Open research needs software

Software can document the research process to aid reproducibility. By opening their methods, models and analyses to others, we can accelerate the rate at which we gather knowledge and make discoveries. Software is helping to change research culture by promoting the benefits of openness. Yet a great deal of research software remains unpublished [5]. This is a huge loss: software has uses that extend far beyond a single publication [6]. To paraphrase, whilst a paper could not exist without software, the software can exist without the paper.

It is difficult to work openly without the right software skills. Some researchers are not confident in the accuracy or engineering of their software, and some are concerned about the overhead of time required to develop software *properly* [7]. This is particularly troubling: if a researcher is not confident in their own software, how can they be confident in their results? Increasing access to software expertise would help solve these problems, and would increase both the openness of a project and confidence in its conclusions.

Many problems; one solution

There are many problems that limit access to software experts in research, but the majority of these can be solved with one relatively straightforward change: the creation of a new career path.

The lack of a formal career path means that it is difficult - if not impossible - to recruit software experts, and if they are recruited they tend to be associated with a career path that is not related to their work. This creates a paradox: the expert is tasked with writing software, but their career is judged by wholly different metrics. Ultimately this limits career advancement and makes employment inconsistent. As a result, the community of software experts is currently somewhat transient. This makes it difficult to retain valuable staff members - a huge loss for research - and means that few software experts can attain a position of seniority that would help in raising recognition of the role. Software experts do not command overheads on funding bids so the university that employs them cannot recover the full economic cost for the position. This a powerful disincentive for being open about the employment of software experts. With a simple change of job title from *Research Software Engineer* to *postgraduate researcher*, a bid author will increase income for their university. Many funders understand the importance of software experts, but their review panels, which are drawn from the research community, have a bias towards favouring the inclusion of researchers on bids rather than software experts.

"a career path for Research Software Engineers would increase access to skills that are vital to modern research"

The solution to these problems lies in making a simple step: we must create a career path for Research Software Engineers. People who are recruited into the role could then be judged against metrics that relate to the work they actually perform. A career path, and the metrics needed to measure how well it is being followed, makes promotion possible, and this would help universities retain the best staff. Legitimising the role would help persuade researchers of the benefits of providing access to software expertise, and this would help persuade review panels that software experts should be recruited. Universities would be quick to accept software experts if they could recover full economic costs for the positions.

By changing the employment of software experts into a legitimate and transparent process, a career path for Research Software Engineers would increase access to skills that are vital to modern research whilst making research more reliable, more efficient and more reproducible.

The name is new; the role is not

There are many software development jobs in the research community. Analysis of jobs advertised on jobs.ac.uk (the de facto job site for UK research) indicates that 4-7% require skills in software development. If this were the case over the entire UK research community [8], it's possible that as many as 14,000 roles exist in UK academia that require expertise in software development.

Coining the name *Research Software Engineer* led to the growth of a hitherto unrecognised community of software experts in academia. But the new name was merely a catalyst; the role has existed for many years. There was no agreement on what to call these positions before the RSE campaign began in 2012 and, as a consequence, hundreds of different titles were used. This lack of clarity in something as basic as the name of the position harms recruiters by making it difficult to advertise effectively, and harms job applicants by making it difficult to find suitable positions.

After taking part in their first Research Software Engineer event, we see people imbued with a strong sense of recognition, validation and empowerment. They often feel that they have found their *tribe*.

How are software experts currently employed in universities?

Programmers of some description have existed in academia for as long as computers have been used to produce results. In universities, the demand for software expertise is often met by postgraduate students and postdoctoral researchers. Handing the development of this most important tool to some of the most intelligent people in the country seems like a safe choice, but there are limits to anyone's ability to take on new skills. Most researchers have no training in software engineering, which raises serious concerns about the reliability of the software they develop.

> "They will spend their days writing software but - paradoxically - their career will be judged against research they don't conduct"

Since no formal career path exists for a software expert, most researchers turn to what they know best - research - and recruit into a postdoctoral position. Whilst this sleight of hand solves the access-to-skills problem for the researcher, it creates a serious problem for the person inhabiting the role. They will spend their days writing software but - paradoxically - their career will be judged against research they don't conduct. In effect, this makes career advancement impossible.

The most common route into what we now call research software engineering begins with an early career researcher who shows interest in software development - or has it thrust upon him or her. Over time these people acquire expertise and some discover that they prefer working with research software over so-called *pure* research. They become the go-to person for software in their group, school or faculty. However, this much-needed expertise does not benefit their career, it merely reduces the time they have to conduct research and write papers. Although there will always be demand for their skills, the potential software expert will find it near impossible to advance their career or secure a permanent contract within academia.

Large research groups and specialists, like high-performance-computing (HPC) centres, exhibit a significant demand for software experts. However, the lack of a formal career path presents obstacles even to these more substantial groups. Unsuitable career paths are often manipulated in an attempt to meet the demands of the group, the university and the software expert, but they rarely succeed in producing a system that rewards all parties.

There is competition for our experts

Academia needs software to advance research, but it is industry that understands the value of software experts. Companies like Google, Apple and Facebook would not exist without software expertise, and competition within industry for leading software experts is intense. According to reed.co.uk [9], the average salary for a UK software developer in industry is £53,000, with just under a third of developers gaining a salary of more than £60,000. Compare this with academia where the majority of Research Software Engineers are paid in the range of £30,000-34,000 [10], and the question becomes: why would anyone choose a career in academic software development?

Fortunately, salary is not the only consideration when choosing a career. Software experts choose a life in academia for the same reason that academics do: a passion for discovery and the freedom to choose how that discovery takes place [11]. Academia cannot compete on salary, but this does not mean that universities should ignore the superior financial incentives offered by industry. Universities should look to significantly improve their own incentives, starting by recognising the role of software experts in academia.

"why would anyone choose a career in academic software development?"

National laboratories provide a good example of how universities could benefit from changing their employment practices. The laboratories are not under the same constraints as universities so they have greater flexibility in the roles they can recruit. Organisations like the Alan Turing Institute, Culham Centre for Fusion Energy, Diamond and Francis Crick Institute employ software experts into a formal career path. As a result, these centres appear to benefit from greater staff retention and a greater range of seniority than found within universities.

Established roles are not suitable

Various solutions have been proposed to employ software experts without changing university career policies, but these approaches tend to prioritise minimising change over maximising suitability.

The IT career path is understood within academia and has been suggested as suitable for software experts. There is a tendency to use the term *IT* to refer to anyone who uses a computer, but this is a major simplification. Software experts and IT staff both use computers, but only in the same way that surgeons and chefs both use knifes. Even if the IT career path were suitable, the centralisation of IT means that most researchers would be blocked by HR policies from recruiting into that career. There is a notable exception: a few universities, such as Manchester and UCL, support a specialist *research IT* career path that represents the work of a software expert, but these universities are in the minority.

There has been some discussion of recruiting software experts into technician career paths, but this is fraught with difficulties. Software experts contribute to research on a strategic level that (unfairly) the label *technician* does not imply. Technicians are less well paid even than software experts in academia. Since university HR attempts to balance pay across everyone in a career path, this would tend to make software experts' pay even less competitive than it is currently.

The above proposals are not suitable for the simple reason that the metrics, incentives, pay scales and career progression of software experts differ substantially from other established roles in academia. The only way to solve this problem is to implement a new career path designed specifically for Research Software Engineers.

The growth of a community

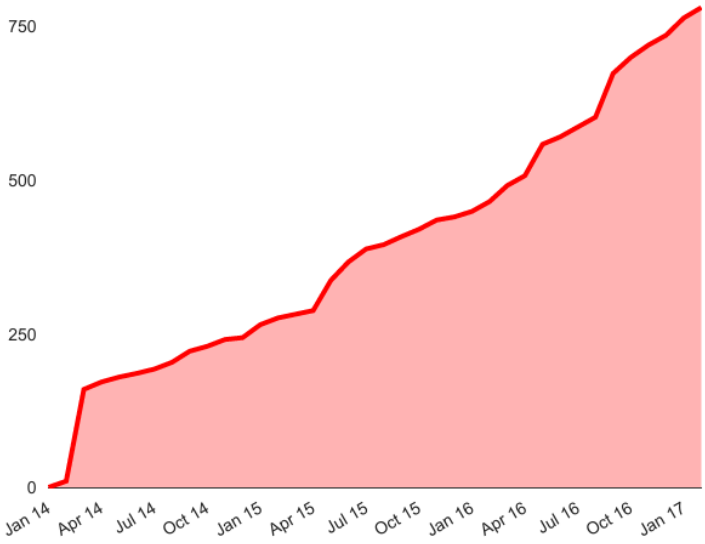
The community of Research Software Engineers has grown quickly since its beginnings in March 2012 [12]. At that time, a group met at the Collaborations Workshop [13] and discussed the problems caused by the lack of a career path for software experts in academia. They realised that software developers lacked not just recognition, they lacked a name. This led to the birth of the term *Research Software Engineer* which the group members agreed to adopt to raise recognition of the role. A summary of the discussions was presented at a later conference [14].

In January 2013, the Software Sustainability Institute created a policy team to study software use and campaign for better software practices in research. It was realised that improving access to software experts had the potential to have a wide-ranging and highly positive impact across every aspect of software use in research, and for this reason the policy group began a campaign to support Research Software Engineers.

The community comes together

Following a drive to increase awareness of the RSE role, a workshop in 2013 brought over 50 people together who now identified as RSEs. Most RSEs were isolated at this time, which meant that they often believed that their position was unique in academia. The workshop revealed to them a nascent community, one with which they could share experiences and expertise. There was considerable excitement about the potential of this community, and this fact alone drove the campaign forwards.

It was agreed to formalise the community by founding the UK RSE Association [15], which occurred in January 2014. The Association gained 50 members in the first week, and since that time membership has risen steadily to 780 in February 2017 - with no signs of growth abating (see figure).



Membership of the UK RSE Association

The UK RSE Association was led in its first year by a group of volunteers and chaired by a representative from the Software Sustainability Institute. In 2015, to improve the sustainability of the Association, elections were held and control was handed over to a committee drawn from its members. The most recent elections were held in September 2016. This saw a significant rise in the number of applications to join the committee, which is evidence that RSEs are invested in the community founded to represent them.

The RSE Fellowship

In acknowledgement of rising interest in research software engineering, the EPSRC funded an RSE Fellowship [16] in 2015. It provided five years of funding to allow the successful applicants to conduct their research software engineering and set up new Research Software Groups. Demand was intense: 211 people applied for the three places that were on offer. This led the EPSRC to increase the available funding and award seven Fellowships [17] to people around the UK. Funding the RSE Fellowship may turn out to be a revolutionary step in the growth of Research Software Engineering.

New groups are founded

In 2013, a new group was pioneered at UCL [18] to provide a home for Research Software Engineers. Based on its success, the following years saw groups created at universities in Bristol, Cambridge, Manchester [19], Sheffield [20] and Southampton [21], and at the Alan Turing Institute and the Francis Crick Institute. There are now 20 groups that provide research software engineering services at research organisations across the UK [22].

There is a significant market for research software engineering: all of the Research Software Groups report demand that far outstrips the availability of their staff. For this reason, growth in the number and size of Research Software Groups should be expected in the coming years. However, these groups are far from easy to establish since there is no career path into which staff can be recruited. Pioneers of Research Software Groups have benefitted from existing Research IT career paths, or have simply laboured against university policies to develop workarounds. Despite the obvious demand within the research community for their services, there is a significant risk that the growth of these groups will be curtailed unless they are supported.

"growth in the number and size of Research Software Groups should be expected in the coming years"

Research Software Groups are discussed in detail on page 17.

The first RSE Conference

The growing international interest in RSEs led to the decision to hold an RSE conference in 2016 [23]. Not only did this succeed in bringing together 202 RSEs from 14 different countries, the conference generated enough income to assure the following year's conference and it sold out over a month before the event took place.

The RSE community was intent on the conference being a success: almost all publicity was gained through the actions of the UK RSE Association's members. Feedback from the conference was excellent, with 95% of attendees expecting to attend the next conference and 100% reporting that they would recommend the conference to others.

The RSE Conference is discussed in detail on page 19.

International collaboration

The problems that impede research vary little across countries. The UK RSE Association includes members from 20 countries, with the largest groups in the US (25 members) and Germany (16 members). Members of the UK RSE community have been active in promoting the benefits of research software engineering in a series of invited talks across the world. This has led to a number of collaborations with international partners.

The US-based WSSSPE project [24] is an international community-driven organisation that promotes sustainable research software. It shares many of the goals of the Software Sustainability Institute which supports the RSE campaign, and for this reason there has been considerable interchange of ideas between these organisations. A number of UK RSEs have talked at WSSSPE events in order to share expertise with the group and learn from US-based RSEs. The WSSSPE project seeks to support the work of RSEs in the US, which culminated in the project co-locating its annual workshop with the RSE Conference in 2016. This was the first time that the WSSSPE workshop has taken place outside of the US.

CANARIE is a Canadian funder responsible for delivering digital infrastructure and driving its adoption, and supporting research software and Research Software Engineers. CANARIE's Research Software Program [25] attended the first UK RSE AGM in 2014, and invited speakers to discuss the UK's RSE community at their annual research software developers' workshops in 2016 and 2017. CANARIE both sponsored the 2016 RSE Conference and sent a delegation to it, and in March 2017 began a survey of Canadian RSEs in collaboration with the Software Sustainability Institute which conducts the surveys in the UK.

"the UK has taken a world-leading position in recognising and advancing research software engineering"

Internet2 [26] is an advanced technology community founded by leading higher education institutions in the US. Internet2 are investigating ways to support *Cyberpracticioners* (a US term for an RSE) in US research, especially in the HPC sector. Internet2 invited a speaker from the UK RSE community to present a summary of the UK's RSE campaign at a workshop [27] which created the framework for a much larger initiative, the *Cyberpractitioner Project*.

The MSDSE (Moore-Sloan Data Science Environments) [28] is a joint programme funded by the Moore Foundation and the Sloan Foundation that supports cross-disciplinary academic data scientists in the US. There are many similarities between the role of the Data Scientists and Research Software Engineer: both are new technology-based roles that are growing increasingly important to research. The MSDSE sent a delegation to the RSE Conference in 2016 and invited speakers from the UK RSE community to attend their 2016 summit and discuss career paths in academia [29].

Software Carpentry [30] is an international organisation dedicated to teaching researchers best practices in research computing. Many RSEs are qualified Software Carpentry instructors and teach the course as a means to increase the level of software engineering skills in the research community. Software Carpentry has often used its international reach to help the RSE community disseminate news across the world.

Following the RSE conference in 2016 and discussions with key personnel in the RSE community, a group of German RSEs founded a sister organisation to the UK RSE Association to support German RSEs. The group *de-RSE* [31] was founded in January 2017.

A brighter future for software in research

The campaign for Research Software Engineers is now entering its fifth year. A lot has changed during this time. A large, active and engaged RSE community has grown, there has been widespread support of RSEs across research stakeholders, access to software expertise has been increased, and the UK has taken a world-leading position in recognising and advancing research software engineering. There is still no widely adopted career path for Research Software Engineers across UK academia, but this goal has moved significantly closer.

Career case studies

Research Software Engineers come from a variety of backgrounds. These case studies provide some insight into this community and the routes that lead people into the profession.

Tania Allard: a Junior Research Software Engineer

Tania recently completed a PhD in Materials Science. Her work contained a significant computational element: the project required the amalgamation of multiple hardware and software technologies. In a recent post [32], she discusses many of the problems associated with academic software. Poorly documented, badly written code and a lack of version control led to her wasting an "incredible amount of time" on the software aspects of the project.

"The support and recognition provided by a central RSE group allowed [Tania] to develop her skills and career"

Her experiences of fixing software issues in her own lab led Tania to consider pursuing a career in research software engineering. While considering her move to the profession, she noted that the "RSE community in the UK is relatively small." This observation refers to the small number of groups that currently offer full-time RSE positions that are recognised as such. This is a concern of many in the community since, among other things, it limits worker mobility.

Tania currently works within the central Research Software Group at The University of Sheffield. The support and recognition provided by a central RSE group allowed her to develop her skills and career while simultaneously making significant contributions to both academic and RSE communities.

For a young research professional such as Tania, the move to full-time research software engineering in academia is essentially a leap of faith. The lack of *typical* academic metrics that the role all but guarantees will make it difficult for her to pursue a traditional academic career. If the RSE movement fails to deliver on the promise of a UK-wide network of suitable positions with an associated career structure, her academic career will end almost as soon as it begins.

Louise Brown: an EPSRC Research Software Engineering Fellow

Louise worked as a software engineer for 25 years and has a PhD in Mechanical Engineering. She worked on the TexGen project, a large and respected software package that models the geometry of textile structures. Although she worked as a software engineer, her job title was *Research Fellow*.

Louise's main issue was a lack of career progression. The work she did as a software engineer gave

"performance was measured by a set of criteria that bore no resemblance to what [Louise] was employed to do"

her little chance of fulfilling the criteria for Senior Research Fellow. In short, her performance was measured by a set of criteria that bore no resemblance to what she was employed to do. She writes "I don't fit the normal *money-in, papers-out* model of many academics" and noted that she was turned down for promotion due to a lack of publications.

Like many RSEs, Louise's work underpins research. Without the software she produces, the research of her academic collaborators would not be possible. Despite this vital role, it is the research, and never the software, that gets both the publication and the credit.

This state of affairs changed dramatically when Louise was awarded one of the new EPSRC Research Software Engineering fellowships. A promotion quickly followed and Louise has been able to dedicate herself fully to the TexGen project and to disseminating her software engineering knowledge among academia.

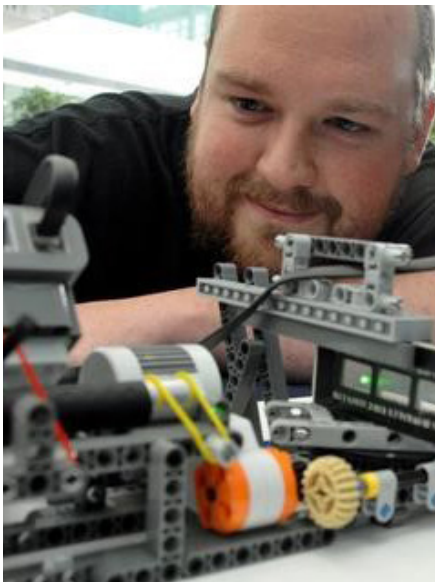
Louise's work has always demonstrated excellence, evidenced by the volume of academic publications that her software has made possible. The EPSRC RSE Fellowship demonstrated that such work is valued by a major funding body. In turn, this was quickly valued more strongly by her university.

Mark Basham: a Research Software Engineer at a national laboratory

Mark completed his PhD in Computational Physics in 2004, and since then has spent most of his career working (both knowingly and unknowingly) as a Research Software Engineer. For the last ten years he has worked in the Data Analysis Group at the Diamond Light Source, the UK's national synchrotron facility. Mark moved to this role after assisting experimental colleagues by writing custom data-analysis code at university. When the opportunity presented itself to work with such a diverse set of experimental methods, with all the associated challenges, he leapt at the chance.

The massive increase in data volume and the rising complexity of experiments means that software plays a critical role at Diamond. A large part of Mark's role is to make sure that core software is written in a sustainable and extensible way, and that the data analysis and acquisition algorithms developed by researchers at the experimental stations are not lost when they move on. These goals made him aware of the Software Sustainability Institute, and especially interested in the RSE movement.

Mark now leads a small team of scientists and RSEs specialising in image processing, and his more senior position allows him the opportunity to manage and design large projects to develop software and hardware on multiple experimental stations to help improve the quality and efficiency of the experiments conducted at Diamond Light Source.



"a large part of Mark's role is to make sure that core software is written in a sustainable and extensible way"

Robert Haines: a Research Software Group Leader

After completing a degree in computer science in 2000, Rob was attracted into academic software engineering by the promise of varied and interesting work. He started work with the Manchester Visualization Centre at the University of Manchester, and in 2003 he accepted a position at the RealityGrid project. "I was called a Research Assistant, but my work was software engineering" writes Rob.

The next few years saw Rob working on a range of projects at the University of Manchester: SPICE, Northwest Grid, GENIUS, Taverna and BioVeL. It was during this time that he first experienced the differences in how software experts are treated in research. "A lot of these projects won awards and publications in top journals, but the software engineers who were instrumental to their development were rarely even acknowledged". Some projects developed an inclusive approach that welcomed software experts as part of the team, and working for these projects became very attractive.

"A lot of these projects won awards...

...but the software engineers who were instrumental to their development were rarely even acknowledged"

Rob became one of the pioneers of the RSE campaign when he took part in the first discussions about the role at the Collaborations Workshop in 2012. After the workshop he ignored his formal job title and began referring to himself as a Research Software Engineer. Rob was elected to the UK RSE Association Committee in 2014 and retained his seat on the committee at the 2016 elections.

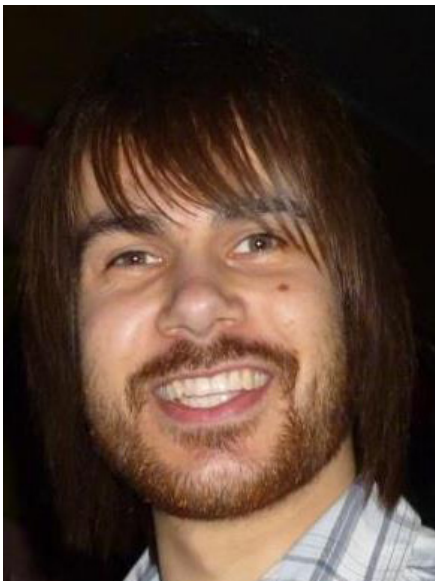
In 2015, the University of Manchester became one of the first universities to adopt the Research

Software Group model for providing software expertise to researchers. Rob has led the group since its beginning, and has more than doubled its size to its current 16 full-time RSEs. In 2016 alone, the group worked on over 40 projects for researchers at the University of Manchester. The group also supports research applications across campus "our projects range from five minutes to get your code working, to five years to include an embedded RSE in a research group". The group is responsible for providing software engineering training and held over 10 courses in 2016 reaching more than 1000 researchers. Rob is also an honorary lecturer, teaching software engineering to undergraduate and masters students.

Rob believes that Research Software Groups have a fundamental role to play in research "RSGs are important because they make our skills available to all researchers - not just those that can afford to hire a full-time RSE". But it is more than just the availability of skills "RSEs care about the correctness of science through the reproducibility of code, and in sustaining software so it lives beyond a single paper publication."

Gary Macindoe: an ex-Research Software Engineer

It takes bravery to actively choose to become a Research Software Engineer. Few universities have centrally supported Research Software Groups and despite significant advances over the last few years, the role is unrecognised by most university HR departments. This leads potential recruits to ask themselves "How do I know there will be somewhere for me to go next? Am I staking my career on this idea?"



"It takes bravery to actively choose to become a Research Software Engineer"

We turn to the longest established central RSE group to attempt an answer: UCL's RSE group was founded five years ago and now contains 10 members. Gary worked there in the past and writes "There are very few software engineering positions available where one would be able to gain a similar breadth of experience."

He now works as a Senior Software Engineer for the BBC and believes that his time in a Research Software Group, and the experience that this gave him, was integral to his success in applying for his new role.

The Research Software Group

Most software experts are, and will continue to be, employed on specific research projects. Alongside the campaign to recognise Research Software Engineers, a new model has emerged to organise software expertise within a university: the Research Software Group.

Better access to skills, more reliable employment

These groups of permanently employed Research Software Engineers collaborate with researchers to build and maintain research software. They have both a service function, working to support researchers within their institutions, and a research function, collaborating with the academics they support to win grants and author research outputs. This identity, with RSEs seeing themselves as both a scholar and an enthusiastic service for their academic colleagues, is critical to success.

"a new model has emerged to organise software expertise within a university"

The model that has succeeded is based on mixed funding: a core leadership team is funded through a *top-slice* of research funding. This provides staff for management and coordination of the group, some free-to-access programming services, training in scientific programming, and a stable baseline to allow long-term support to grants. Additional RSEs are then funded through paid-for services, either direct income on a day-rate, or through grant income, where anything from 10% to 50% full-time equivalent (FTE) of an RSE is costed into a research grant. This allows a gradually growing team to develop, according to demand for services, by carefully managing the resulting income. The additional RSEs are appointed as permanent members of staff or on a *permanent but subject to continued grant income* basis.

This model was pioneered in UCL, where it has seen, over a five-year period, a team with three centrally funded posts grow to include six grant-funded posts. Additional university-based Research Software Groups have since emerged in some of the country's leading research-intensive universities [22]. It is worth noting that in 1990, long before the current interest in research software engineering, the University of Edinburgh trialed a similar group which has grown into the EPCC [33], a self-sustaining centre employing just under a hundred people. Ensuring that more universities can benefit from this kind of success is one of the goals of creating Research Software Groups.

The benefit of centralisation

Why might such a shared group be more effective than individual RSEs dedicated to individual research projects?

High quality staff can be attracted by providing a shared and stable home for RSEs with many interesting projects available. By establishing a service function, the creation of high-quality, impactful code can be the sole goal of the RSEs, who are not distracted by the competing career demands experienced by researchers. By aggregating demand at a wider level than the individual research group, RSE resources can be made available to projects that do not have sufficient need to hire a permanent RSE of their own. Even for research groups with only the occasional need for a full-time RSE, such short-term roles are better integrated within an RSE group. This provides a more attractive long-term position for the RSE and hence increases the likelihood that the position will be filled by a good candidate. Crucially, such groups also provide a nucleus for a shared culture of excellence in program-

ming practice. Isolated software experts can often *go dark*, developing code without input or review from peers.

Where should Research Software Groups be placed within universities? Multiple models are being explored. Some place the group centrally in their Research Computing organisations, alongside support for High Performance Computing. Others place RSE groups within individual faculties or departments. A central group allows a larger volume of work to be aggregated, reducing risk from variations in demand, while a more local group allows RSEs with the appropriate research background to focus on work within their field. In the long-term, we believe a hub-and-spoke model, with both central and departmental teams working together and sharing projects, will serve the community best.

Research Software Engineering activities

A number of activities have grown alongside the campaign to recognise Research Software Engineers. These range from fellowships to interest groups and each helps build and support the community.

There is always a danger that the sudden growth of a community will be accompanied by a proliferation of competing activities. However, the RSE community has benefitted from a healthy overlap in roles, with people taking part in a number of leadership positions across different activities. This has fostered good communication across activities and prevented duplication of effort. There has also been a sustained level of interest in the community, with pioneers from the start of the campaign continuing to contribute towards activities. This has created a strong social history of the campaign and, most importantly, ensured support for newcomers which will help them become the next generation of RSE leaders.

The Research Software Engineer Fellowship

In 2015, the EPSRC funded a call for Research Software Engineer (RSE) Fellowships [16]. This provided five years of funding to "exceptional individuals with combined expertise in programming and a solid knowledge of the research environment". The Fellowship was designed to support the RSE role and lead to the development of a stronger, more sustainable RSE community.

The RSE Fellowship call was incredibly popular, with 211 applications received for the three places originally on offer. After reviewing the applications, funding was found to support seven fellowships which began in 2016. The RSE Network conducted and published interviews with the new Fellows in April 2016 [17].

Less than a year into their fellowships, the cohort began achieving successes. New RSE groups have been formed at Sheffield and Bristol and the fellows are firmly embedded within the community. They have taken leading roles in the national RSE committee, the RSE support for Tier-2 HPC infrastructure and in championing the RSE cause in their own institutions.

Even the mere existence of the program has been impactful, with several institutions starting to create RSE-friendly structures in order to accommodate future calls. One of the fellows, Mike Croucher, notes that "On winning the fellowship, everything changed. Academics became more interested in including us in grants, the University became more interested in supporting RSEs and providing them with a career structure and other institutions, both in the UK and beyond, started inviting me to help them begin RSE groups in their area."

The next RSE Fellowship call is planned for 2017-2018 [34].

The First Conference of Research Software Engineers

In early 2016, it was decided to build on growing interest in research software engineering by running a conference for Research Software Engineers. With the help of seed funding from the EPSRC (via their support of the RSE Network [35]), the world's first conference for Research Software Engineers took place on 15-16 September 2016.

The conference was a phenomenal success, mainly in the fact that it attracted many new people to the community. One of these new recruits summarised the experience of many in their feedback "This might have been my 30th conference, but it was the first where I felt thematically 100% at home and understood".

The conference attracted 202 people from 14 different countries. There was significant international interest. For the first time in their history, the US-based WSSSPE community [24] hosted its annual workshop outside of the US to co-locate with the conference, and large delegations attended from

Home locations of attendees at RSE Conference 2016. Map data © 2017 Google.

CANARIE [25] and the Moore-Sloan Data Science Environments [28]. Significantly, the conference provided the venue for a group of German attendees to meet and form their own German RSE association [31].

The conference attracted eight sponsors including Microsoft Research (Gold sponsor) and Intel (Silver sponsor). It received 38 talk proposals, double the number that could be accommodated, and ran 15 workshops in which a broad range of expertise was disseminated.

Keynotes were presented by Matthew Johnson, Head of Agile Projects at Microsoft Research, Cambridge, and by Professor Susan Halford, Director of the Web Science Institute at the University of Southampton. Matthew talked about the wide range of projects that RSEs undertake at Microsoft and gave examples of how work by RSEs has driven innovations in products across all aspects of their product range. Susan discussed the rise of new methods and expertise around data and how this has unsettled the research boundaries in the 21st century. The final plenary talk was presented by Caroline Jay, a Senior Lecturer in Software Engineering, who discussed the importance of software in the research process and the vital role that RSEs have in designing, developing and sustaining it.

Efforts to promote gender diversity appear to have been successful. The UK RSE community appears to be 11% female (see following section), but 18% of conference attendees were female. Gender diversity was a key issue during the organisation of the conference. A diversity report published after the conference stated that alongside the improved representation from the community, the organising committee was "remarkably gender balanced for the domain" [36]. It is difficult, and takes considerable time, to change demographics, but it would appear that the RSE Conference is moving in the right direction.

Following the event, 47% of attendees responded to a post-conference survey [37]. The majority of attendees came from a background in Physical Sciences (30%), Computer Sciences (18%) and Biological Sciences (17%), but it also attracted attendees from a wide range of disciplines (albeit at lower numbers). The overall feedback score from attendees was 4.3 out of 5, with 95% of respondents stating they would attend the conference again and 100% stating they would recommend it to colleagues.

The next RSE Conference is scheduled to take place on 7-8 September 2017 [38].

What do we know about UK RSEs?

To gain a better understanding of the RSE community, the Software Sustainability Institute ran a nationwide survey of Research Software Engineers in January 2016 [4]. The survey was distributed to all UK RSE Association members, and 335 responses were collected from UK-based RSEs [10].

How many Research Software Engineers?

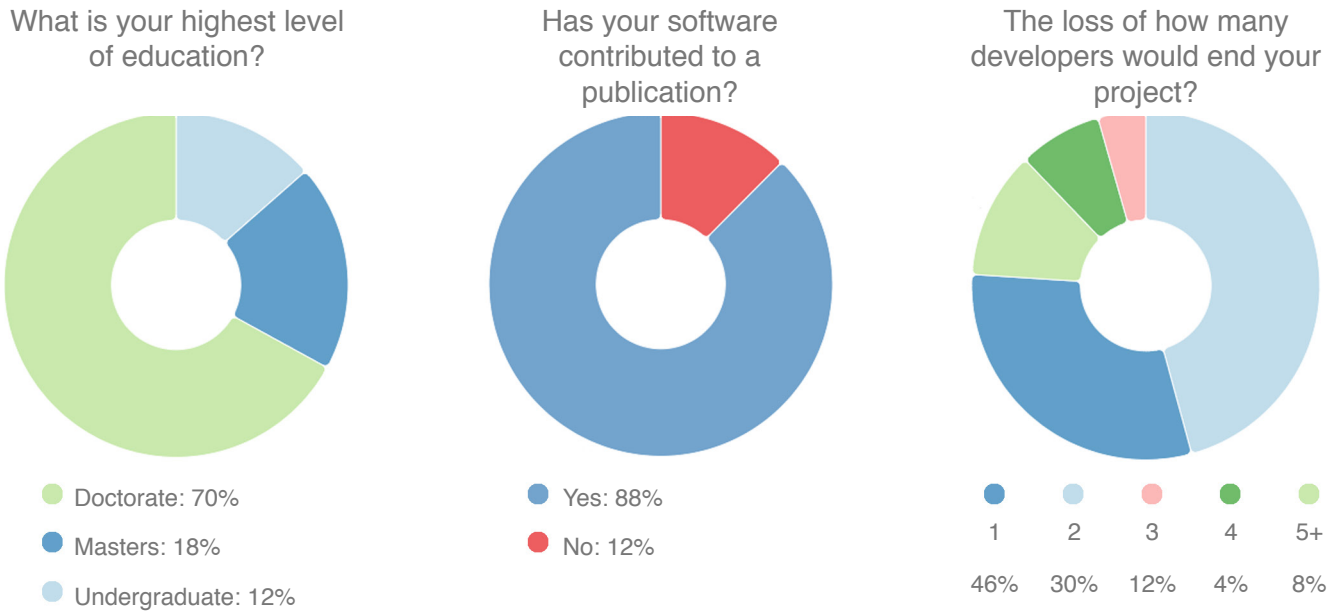
The lack of a career path for Research Software Engineers means that software experts are hidden in other roles, and this makes it difficult to discover how many RSEs exist. We can apply some limits, at least in the UK, thanks to data we have collected on the community.

In February 2017 there were 780 members of the UK RSE Association and growth over the last three years has been steady at around 200 members each year. Based on these figures, it appears safe to assume that there are at least 1,000 RSEs in the UK.

Upper limits are somewhat more difficult to apply. Analysis of jobs advertised on jobs.ac.uk (the de facto job site for UK research) indicates that 4-7% of jobs require skills in software development. If this were the case over the entire UK research community [8], it's possible that as many as 14,000 software-development roles exist in UK academia in 2015/16.

It would appear there are between 1,000 and 14,000 RSEs in the UK. This is a wide range, and opinion within the RSE Community is that the actual number of RSEs lies somewhere in the mid-range of this spread. A more definitive answer will be provided later in 2017, and hence available in the 2018 State of the Nation Report, when the Software Sustainability Institute concludes its study of the academic jobs market.

Demographics



RSEs have an intimate knowledge of research: 70% hold a doctorate as their highest level of education. They mainly derive from a background in the physical sciences and computer science.

The peak age range of the community is between 35-44 years old. RSEs tend to be younger than other academics: only 4% of RSEs are between 55 and 64 years old, compared to 14% of academics in that age range [39]. This difference could be explained by the fact that software-dependent roles in re-

search are relatively new, hence the RSE community has not yet reached its final age distribution, or that RSEs reach a ceiling in their career and choose to leave academia, or by a combination of both these factors. The apparent lack of older RSEs may be a cause for concern and requires further investigation.

Only 11% of survey respondents were female, which agrees with experience of an RSE community that is predominantly male. This is a significant imbalance. If research software engineering were an undergraduate subject, it would report a lower level of female participation than any other subject [40]. The clue to this large imbalance is most likely as described above. Research Software Engineers mainly derive from the physical sciences and computer sciences, both of which have well-publicised problems with gender imbalance, and they generally hold a PhD, and these subjects are found to become even more male dominated beyond undergraduate study.

The RSE community is open about its gender-balance problem and is focusing on increasing female participation in all aspects of the community's work: from events to the committees that run the community. These short-term goals are progressing well: the RSE Conference attracted a higher proportion of female attendees (18%) than are present in the RSE community in general, and the ten-member UK RSE committee is 30% female, which is another step forward. The UK RSE Association is working on writing job descriptions and advice for recruiters that is intended to improve diversity of applicants. By concentrating on these short-term goals, the hope is that there will be an improvement in gender balance across the RSE community in the long term.

Recognition

It is generally believed that the RSE community does not receive a level of recognition commensurate to its contribution to research. This argument is supported by the survey results which found that 88% of RSEs contributed to a result that appeared in publication, but 24% of this group were not acknowledged for their work.

Paper publications might not be the most appropriate mechanism for acknowledging the work of RSEs, but that does not mean that their work should go unacknowledged until a better mechanism is found. It would be a major controversy if almost a quarter of researchers were not acknowledged for their contribution to research. Why should the situation be any different for Research Software Engineers?

"It would be a major controversy if almost a quarter of researchers were not acknowledged for their contribution to research"

Vulnerability of software

Software becomes vulnerable in many ways, but one frequent reason for the failure of software is the departure of staff who develop and maintain it. The survey investigated how well research projects were prepared to cope with the departure of staff, and the results show a significant vulnerability.

RSEs were asked for two metrics on their most important software project: the bus factor and the existence of a technical handover plan. The bus factor is a measure of how many developers a project can lose (i.e. be hit by a bus) before the project fails. The larger the bus factor, the safer the project. A technical handover plan describes a software product in enough detail to allow a new RSE to get up to speed with the software and start developing it further. Without a technical handover plan, a new RSE will have to spend many months forensically reviewing the software to discover how it works. This is both time-consuming and a needless waste of resources.

It was found that 45% of RSEs work on projects with a bus factor of 1. A higher bus factor would be preferable, but there is a practical reason for this low figure: many projects would struggle to employ

more than one RSE. More worrying, 78% of RSEs work on a project with no technical handover plan. Unlike the low bus factor, there is no reason why a project should not invest effort into writing a technical handover plan for their most important piece of software. The lack of a technical handover plan is a serious vulnerability, and becomes even more concerning when it correlates with a low bus factor: 40% of respondents work on a software project with no technical handover plan and a bus factor of 1.

Future surveys

To understand how the RSE community changes over time, further surveys are scheduled for 2017 and 2018, with the expectation that they will continue for some time. The results of the 2016 survey attracted attention from a number of countries who are planning to run their own RSE surveys in 2017.

How we can improve access to software expertise in research

The UK has led the world in recognising the importance of research software and the people who build that software. If we wish to maintain that position, we must build on our successes.

The changes required to improve access to software expertise are not onerous, but changing academic culture take time. While we may have to wait for software experts to be valued the same as researchers, there are changes we can make that will more quickly improve access to software expertise and, in doing so, advance research.

Increasing recognition

All research stakeholders should aim to increase awareness of the reliance of research on software.

We must continue to increase recognition of the vital role that software plays in research, and the need for experts who can develop and maintain reliable software.

Implementing a career path for Research Software Engineers

We must implement a career path for Research Software Engineers across UK academia.

To fulfil the demand for software expertise in academia, we must make the employment of Research Software Engineers straightforward and transparent by implementing a career path. RSE Groups have begun to share job descriptions for RSEs of different level of seniority, which can become the basis of a career path. Across all RSE organisations we have accumulated contacts covering many UK research organisations. We will use these local contacts to push adoption of the RSE career path. Once the benefits of improving access to RSEs are shown at these leading universities, the remaining universities will seek to emulate them.

Support from universities

Universities should simplify the employment of Research Software Engineers and support Research Software Groups.

Universities can increase their rate of discovery by increasing research productivity. A researcher may labour for months performing tasks that can be performed in seconds, and with far fewer mistakes, by bespoke software. Increasing access to Research Software Engineers is a cost-effect method of improving the productivity and reliability of research. The universities who are first to support software experts will be those who entice the best experts.

Universities who wish to access this benefit can do so by recognising the RSE position and implementing policies that provide researchers with the flexibility needed to recruit these roles. In the short term, this may mean allowing researchers to recruit into positions not currently described in HR policy, and accepting that these positions do not yet command research overheads.

Universities should support the creation of more Research Software Groups. These provide software expertise for many researchers across a university and, with time, they become financially self-sustaining. At best, universities should provide financial support for the staff members needed to start a new Research Software Group. At worst, universities should ensure that they do not actively dissuade the creation of a Research Software Group through hiring and financial policies that are incompatible with the needs of these groups.

Support from research funders

Research funders should look to include software experts in the preparation and execution of funding calls, and they should explicitly support Research Software Engineers in their guidance for applying to, and reviewing, funding applications.

Research funders understand that improving access to software expertise will advance research, so they have been some of the keenest supporters of the RSE campaign. However, some funding policies unduly limit access to Research Software Engineers. Funding calls that are likely to include a significant reliance on software should aim to recruit software experts as reviewers. There are many senior RSEs who have the necessary experience in research, the funding process and software engineering best practice.

Although it is researchers themselves who review funding applications, and hence decide on the merits of including software expertise on a bid, it is the funders who provide guidance to these reviewers. Funders should provide guidance that explicitly supports recruitment of a Research Software Engineer on research grants that rely on software.

References

1. Hettrick, S., Antonioletti, M., Chue Hong, N., Crouch, S., Inupakutika, D., & Parkinson, T. *It's impossible to conduct research without software, say 7 out of 10 UK researchers*. https://www.software.ac.uk/blog/2016-09-12-its-impossible-conduct-research-without-software-say-7-out-10-uk-researchers. 4 December 2014.

2. https://www.software.ac.uk

3. Hettrick, S. *Quick and dirty analysis of the software being used in research: Python, Matlab and R*. https://www.software.ac.uk/blog/2016-09-12-quick-and-dirty-analysis-software-being-used-research-python-matlab-and-r. 12 August 2016.

4. Philippe, O., Chue Hong, N. & Hettrick, S. *Preliminary analysis of a survey of UK Research Software Engineers*. 2016. 4th Workshop on Sustainable Software for Science: Practice and Experience.

5. Peng, R. D. *Reproducible research in computational science*. Science, 2011. 334(6060), 1226-1227.

6. De Souza, M., Haines, R. & Jay, C. *Defining Sustainability through Developers' Eyes: Recommendations from an Interview Study*. 2nd Workshop on Sustainable Software for Science: Practice and Experience. 2014.

7. Jay, C., Sanyour, R. & Haines, R. *Not everyone can use Git: Research Software Engineers' recommendations for scientist-centred software support (and what researchers really think of them)*. 2016. First RSE Conference.

8. HESA staff overview. https://www.hesa.ac.uk/data-and-analysis/staff

9. Reed salary checker. https://www.reed.co.uk/average-salary/it-telecoms/software-developer

10. RSE Survey 2016 data set. https://github.com/software-saved/RSE-Survey-2016

11. Cannam, C., Gorissen, D., Hetherington, J., Johnston, C., Hettrick, S., and Woodbridge, M. https://www.software.ac.uk/blog/2016-10-06-ten-reasons-be-research-software-engineer. 23 August 2013.

12. Hettrick, S. *A not-so-brief history of Research Software Engineers. https://www.software.ac.uk/blog/2016-11-24-not-so-brief-history-research-software-engineers*. 17 August 2016.

13. Collaborations Workshop 2012. https://www.software.ac.uk/cw12

14. Baxter, R., Chue Hong, N., Gorissen, D., Hetherington, J., & Todorov, I. *The Research Software Engineer*. http://digital-research-2012.oerc.ox.ac.uk/papers/the-research-software-engineer. September 2012.

15. http://rse.ac.uk

16. https://www.epsrc.ac.uk/funding/calls/rsefellowships/

17. Croucher, M. *Interviews with the EPSRC Research Software Engineering Fellows*. http://www.walkingrandomly.com/?p=6037. 18 April 2016.

18. Hetherington, J. *Introducing the new UCL Research Software Development Team*. https://blogs.ucl.ac.uk/research-software-development/introduction. 3 October 2012.

19. http://www.itservices.manchester.ac.uk/our-services/research/engineering

20. http://rse.shef.ac.uk

21. http://rsg.soton.ac.uk

22. http://rse.ac.uk/community/research-software-groups-rsgs

23. http://www.rse.ac.uk/conf2016

24. http://wssspe.researchcomputing.org.uk

25. https://www.canarie.ca/software

26. http://www.internet2.edu

27. http://meetings.internet2.edu/2016-07-cyberpractitioner-workshop

28. http://msdse.org

29. http://msdse.org/summit

30. https://software-carpentry.org

31. http://www.de-rse.org

32. Allard, T. *A new member of the team: Tania Allard*. http://rse.shef.ac.uk/blog/tania_allard. 29 February 2017

33. https://www.epcc.ed.ac.uk/

34. https://www.epsrc.ac.uk/files/funding/calls/future-calls201718/

35. EPSRC grant: EP/N028902/1

36. Jones, C. *Diversity report from the first conference of Research Software Engineers*. https://www.software.ac.uk/blog/2017-01-10-diversity-report-first-conference-research-software-engineers. 19 December 2016.

37. Hettrick, S. *RSE Conference feedback: what did people think?* https://www.software.ac.uk/blog/2016-10-19-rse-conference-feedback-what-did-people-think. 10 October 2016.

38. http://rse.ac.uk/rse-conference/

39. https://www.hesa.ac.uk/data-and-analysis/staff/overviews?breakdown%5B%5D=582&year=2

40. https://www.hesa.ac.uk/data-and-analysis/students/courses

41. https://www.epsrc.ac.uk/research/facilities/hpc/computingcentres

42. http://gow.epsrc.ac.uk/NGBOViewPanelROL.aspx?PanelId=1-414KW8&RankingListId=1-414KWJ

43. http://rse.ac.uk/community/local-champions

Appendix A: glossary and summary of RSE organisations

A number of organisations have been created to support Research Software Engineers. The community benefits from good communication across the different organisation largely down to a healthy overlap in roles, with people taking part in a number of leadership positions across different activities. Communication is also supported by the RSE Network, which actively seeks to improve ties across the RSE community.

The majority of these organisations are completely reliant on voluntary effort. Although this shows that the RSE community is passionate about supporting itself, it is simply not sustainable to rely on volunteering to support a critical section of the research community.

RSE Fellowship

In 2015, the EPSRC funded a call for Research Software Engineer (RSE) Fellowships to support the RSE role and lead to the development of a stronger, more sustainable RSE community. The RSE Fellowship call was incredibly popular, with 211 applications received for the three places originally on offer. After reviewing the applications, funding was found to support seven fellowships which began in 2016.

Details can be found on page 10.

The UK RSE Association

By February 2017, the UK RSE Association reported 780 members. It was founded in January 2014, and is the first organisation in the world to be created to represent Research Software Engineers.

The UK RSE Association acts as the focal point for the RSE community. It runs a website, mailing list and slack channel through which RSEs can discuss issues, advertise jobs, organise training and determine the future of the RSE campaign.

The Association was originally led by the Software Sustainability Institute, but became a democratic organisation run by its own community after a committee election held in September 2014. The second election, held in September 2016, saw a growth in applications to stand for the committee, a growth in the voting community and new positions on the committee for important subjects such as industrial contact and diversity.

Details can be found on page 10.

Research Software Groups (RSGs)

A Research Software Group is a group of Research Software Engineers who work for a university or research organisation. Not all projects are large enough to reliably employ a full-time RSE. Research Software Groups pool RSEs into a single group and allow researchers to hire an RSE only when someone is needed.

This represents a significant win-win for the research community. Researchers can access RSE skills without the worry of sustaining an RSE through periods of lower funding or less demand, and RSEs receive more consistency in employment because it takes more than the failure of a single project to end their employment.

Details can be found on page 17.

Research Software Engineer Leaders network (RSEL)

The founders of the first Research Software Groups created the RSE Leaders network to help share experiences of running these groups and advice on how to do it better. The RSEL currently includes members from 25 universities and research organisations. It is open not just to established groups, but also to people who aim to set up a new group. In this way, it supports the creation of new groups and helps prevent them from repeating past mistakes.

Tier-2 Champions

In 2016, the EPSRC invested in renewing the Tier-2 layer of HPC centres [41], and awarded £20 million to six HPC centres across the UK [42]. The EPSRC encouraged centres to explore novel technologies, and to provide research software engineering support to users to enable them to explore those technologies.

The centres are working together with the RSE Network and UK RSE Association to create a Tier-2 champions scheme with the aim of preventing duplication of effort. The RSEs in the scheme will share knowledge and, most importantly, make HPC training and access available to the whole UK research community.

Research Software Engineer champions

RSE champions are local contacts who volunteer to promote Research Software Engineering and raise awareness of the RSE community at their university or research organisations. They host seminars, represent RSE interests on local committees and some run regular, informal RSE meetings.

There are currently 18 RSE Champions based at organisations across the country, from the British Geological Survey to the University of St. Andrews [43].

Research Software Engineers Network (RSEN)

The authors of this report represent many of the early founders of the community of Research Software Engineers and have each contributed their time to the campaign or the UK RSE Association. In 2015, the authors were successful in a bid for funding to help support core RSE activities [35]. The funding was mainly directed to employ an RSE Coordinator, who helps organise and support all of the above organisations, and to provide seed funding for the RSE Conference and RSE website. The RSE Network is also responsible for this annual report on the RSE Community.