Research software at the University of Southampton

December 6, 2019

1 Introduction

Most research would not be possible without software. It is the enabling technology behind major advances, from decoding the human genome to the discovery of the Higgs boson, it lies at the heart of strategically important technologies, such as artificial intelligence, and it is a tool of constant use and fundamental importance in day-to-day research. Despite its pivotal role, support for software and software skills within academia has not kept pace with the rapid adoption of software as a research tool. This is a major barrier that inhibits research and puts results at risk.

A 2014 study [1] found that 92% of UK researchers use software and 69% report that it is fundamental to their research. It is not possible to disentangle the reliability of the software used to generate results from the reliability of the research itself. Unreliable software leads to unreliable results that cannot be trusted. Trust cannot be established in poorly engineered software, because it is too difficult to understand whether the software is reliable. Without trust, researchers will not build on existing software and will instead recreate it themselves: an avoidable duplication of effort and a waste of resources.

In 2019, the Southampton Research Software Group [2] conducted a survey of research staff to identify the relationship between software and research at the University of Southampton. Software use is near ubiquitous across all but one faculty, researchers report that software is vital to their work, a third of researchers develop their own software, and around a third who develop software are interested in commercialisation. Worryingly, around two thirds of staff who develop software do not believe they have had sufficient training to develop reliable software.

Software has an almost unparalleled power to advance research and enterprise. Much of this potential is likely to remain unrealised at the University, unless researchers are given access to the necessary software skills and expertise.

2 Major findings

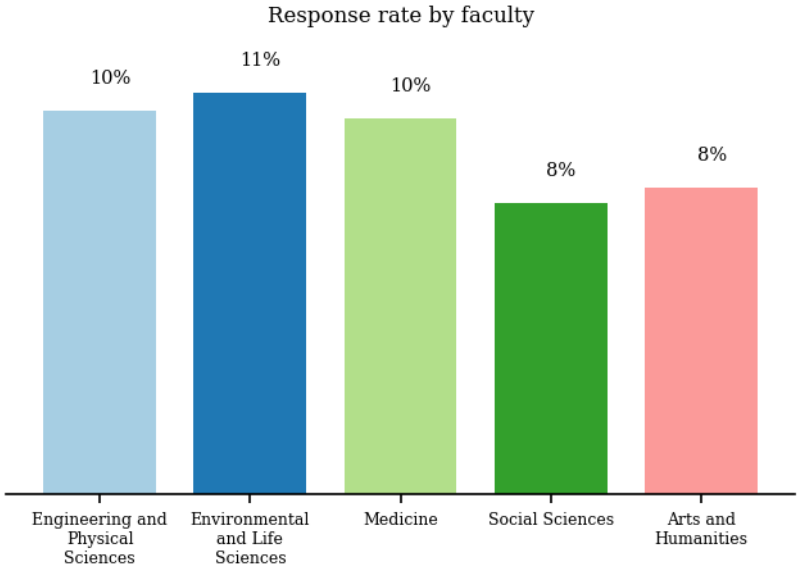
2.0.1 Of researchers at the University of Southampton:

1. 95% use software in the generation of results they expect to appear in a publication.
2. 73% report software as "vital" to their research.
3. 33% develop their own software and, worryingly, of these software developers 64% do not believe they have received sufficient training to develop reliable software.
4. 47% expected to develop software as part of a funding proposal, but despite this fact, less than half of these included costs for software development in the proposal.

5. 67% report that hiring a Research Software Engineer from a central pool would be a suitable or perfect way of recruiting software expertise into their group.

2.1 Methodology

The survey was sent to all research staff via an email invitation to the mailing lists for all staff employed on an ERE (Education, Research and Enterprise) contract and all PhD students. The emails were sent on a faculty-by-faculty basis with the following response rate:



The survey asked 16 questions about the researcher and their use of software. It received 594 usable results which are included in this analysis.

The appendices to this report include information about the number of responses to each question (not all were mandatory), and further information about the respondents such as the respondents' job titles and funders.

2.2 Reproducibility, licences and citation

All resources related to this survey are stored on Github and freely accessible to anyone. The repository [3] contains:

1. A pdf of the original survey
2. The anonymised data collected by the survey
3. The code used to analyse the software
4. This report on the results of the analysis

The above have all been published under an open licence (BSD 3-clause for the code, CC by attribution for the survey and the data).

If you reproduce any of the results of text from this study, please cite: * Brown A, Graham J, Grylls P, Hettrick SJ, Mangham S.W., Robinson J & Wyatt C. "Research software at the University of

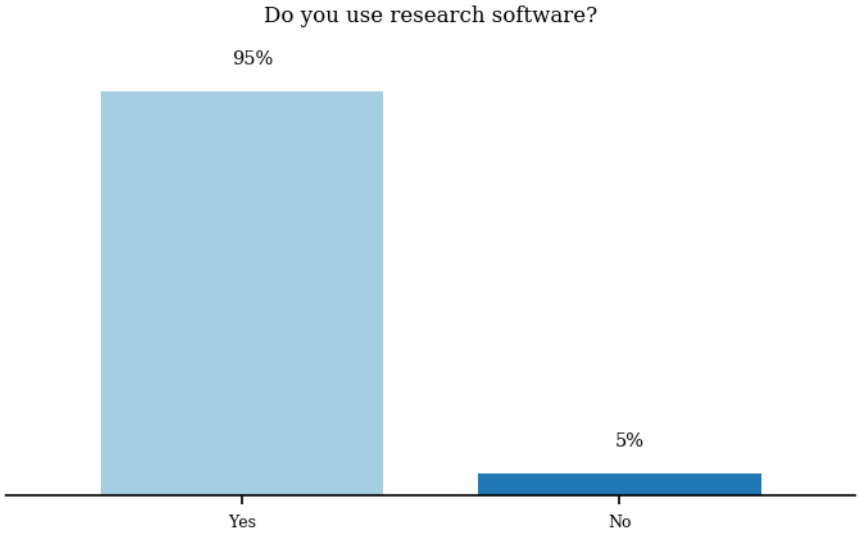
Southampton". 6 December 2019.

3 What is research software?

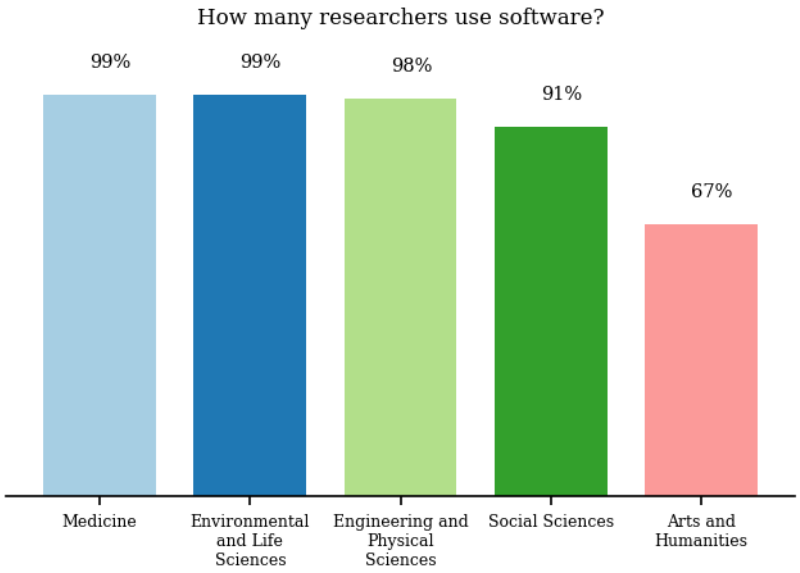
"Software" is an umbrella term that disguises a huge amount of complexity and variance. The applications for software are equally complex, from innovations in algorithms to an online service, from the control of a genome sequencer to the analysis of electronic health records, and each is associated with its own particular development requirements. In this survey, we defined "research software" as:

> "any software you have used in the generation of a result that you expect to appear in a publication. This might be anything from a few-line script to clean some data, to a fully fledged software suite. It includes code you have written yourself and code written by someone else."

4 Use of software and importance to research

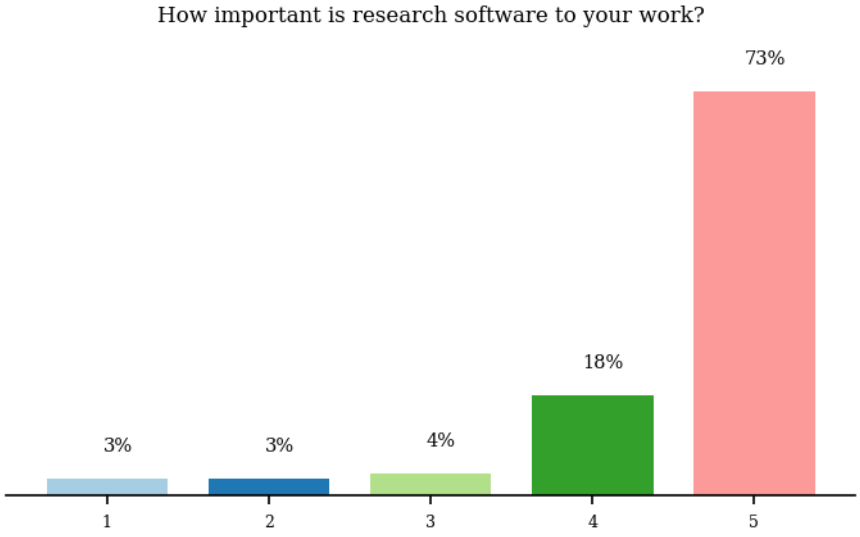


The vast majority of researchers at the University use research software. Only 5% report that they do not, but even this small number seems too large. More insight is gained if we segment the data across faculties.

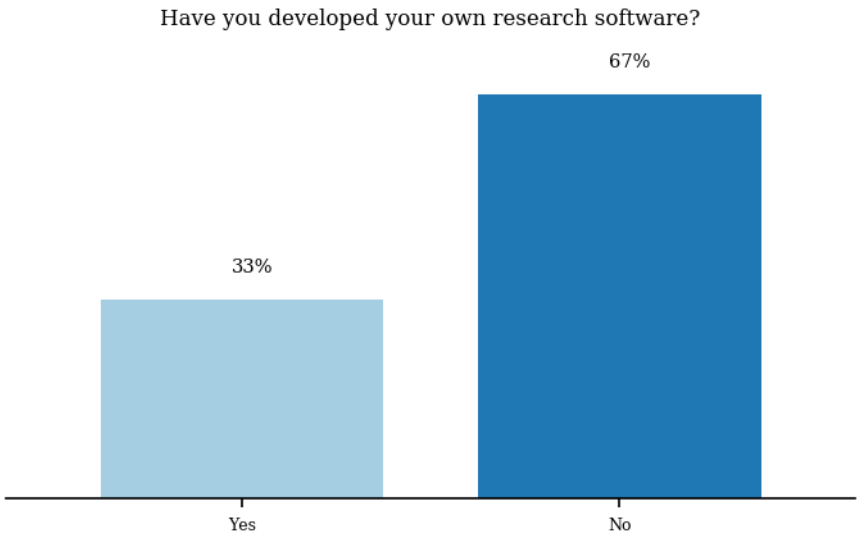


In Engineering & Physical Sciences, Environmental & Life Sciences and Medicine, the use of software is near 100%. Software use drops slightly in Social Sciences - but it is still above 90%. In Arts and Humanities only 67% report using software in their research, which is understandable in this discipline, although it is noted that humanities researchers rely heavily on software. The mean percentage of researchers who use software in their research across the university, neglecting the contribution from Arts & Humanities, is 97%.

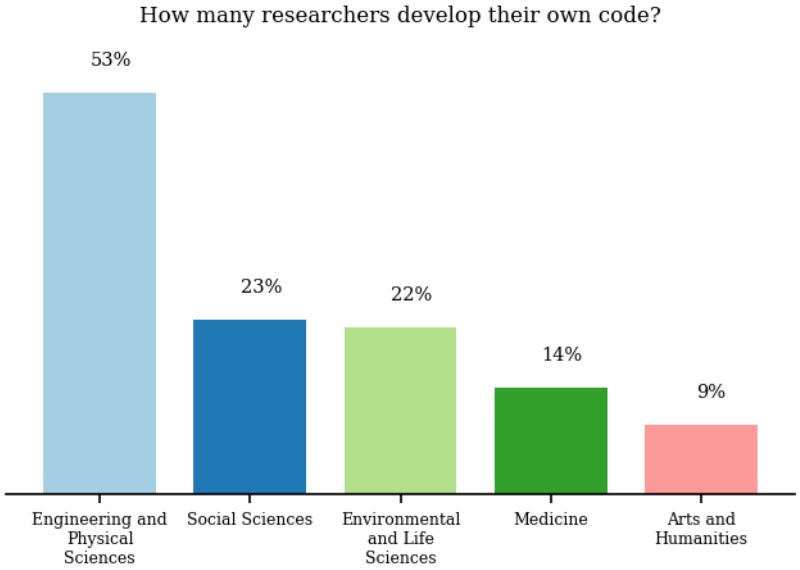
Use of software provides insight into its prevalence, but only hints at its importance to research. To investigate this issue further we asked researchers to rate how important research software was to their research (in which a rating of 1 was "not at all" important and a 5 was "vital").



Software is of fundamental importance to the research conducted at the University of Southampton: 95% of researchers use software and 73% state that it is vital to their research.



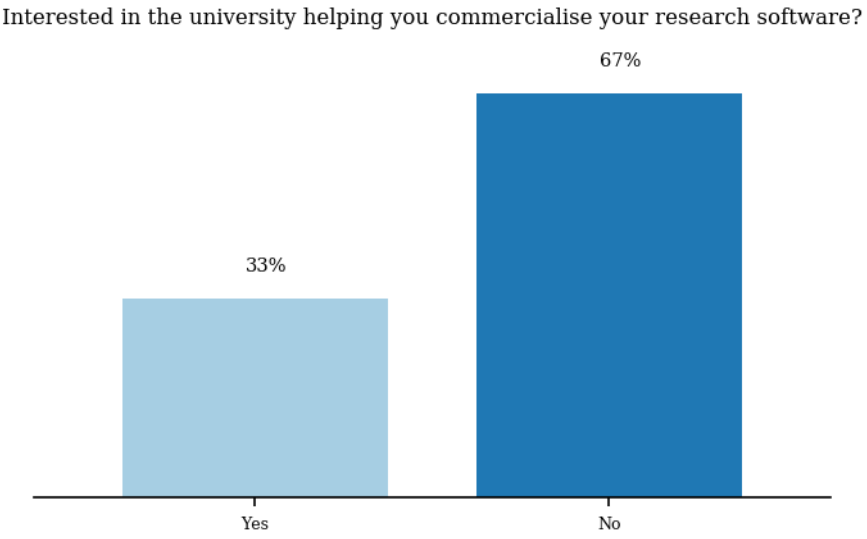
Around a third of researchers at the University develop their own software. This is not surprising: the specific problems faced in research tend to require bespoke solutions. However, the number of researchers who develop their own software varies widely across faculties.



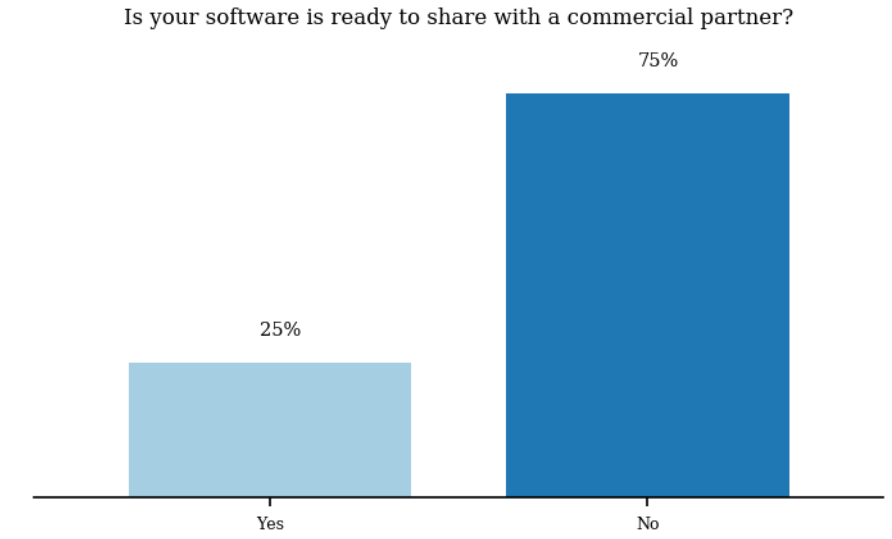
Engineering & Physical Sciences, Environmental & Life Sciences, Medicine and Social Sciences report similar levels of *software use*, but researchers within Engineering & Physical Sciences are far more likely - over double - to write their own code. It seems unlikely that Engineering & Physical Sciences researchers are facing challenges that require significantly more bespoke software development than researchers from other faculties. It is a generally held opinion that researchers from EPS background are more likely to have experienced training in programming at some point in their education. If this is true, and further investigation is required, then this may explain why EPS researchers are more likely to develop their own software. Whatever the cause, there lies in this result a huge opportunity. training researchers from faculties that present high software use

and low software development could have a significant effect on their speed, scale and reliability of their research.

5 Potential for commercialising software



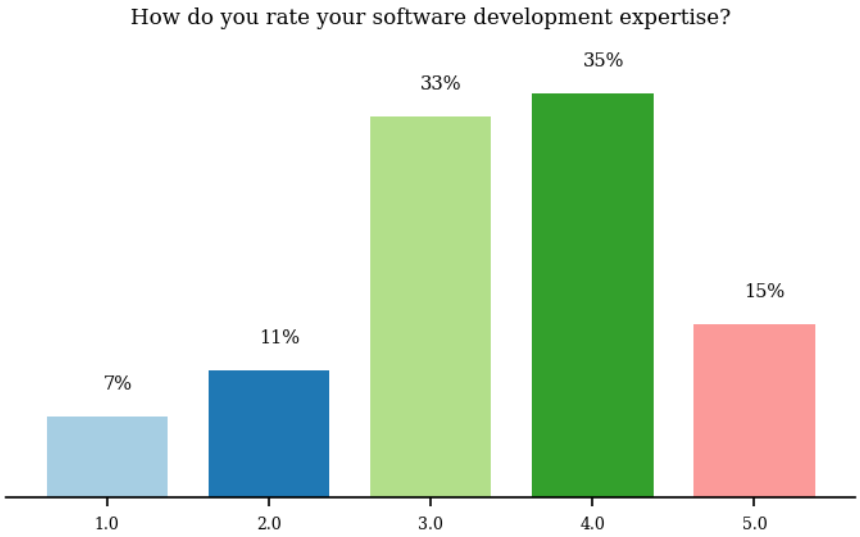
A third of respondents who develop their own code are interested in commercialising their software. Extrapolating this result over the University suggests that almost 700 people could have software that they would like to commercialise. With such a large number of potential software to draw from, there would appear to be significant potential for successful commercialisation.



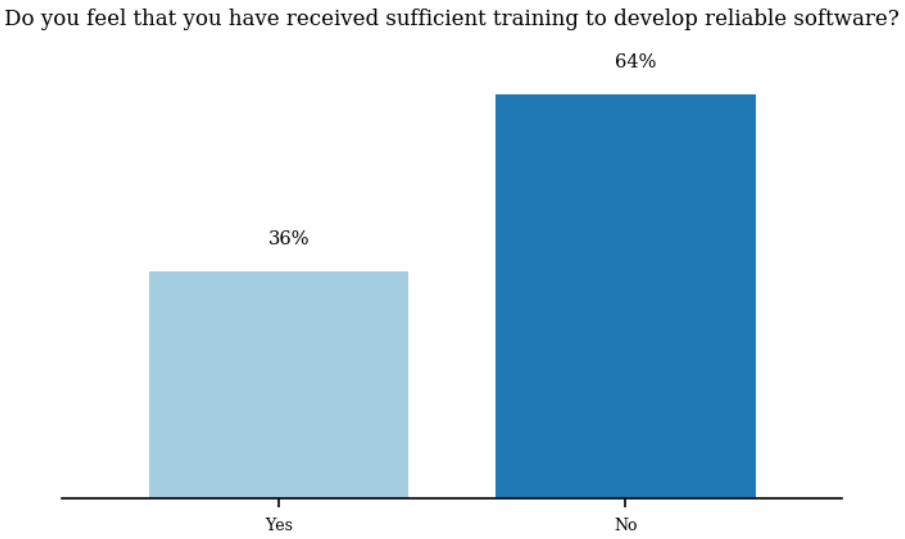
75% of people who wish to commercialise their code do not feel it is ready to share with a commercial

partner, which is a strong indicator that this software is unfit for commercialisation in its current state. Regardless of the commercial promise of the concept behind the software, commercial success can only be achieved if the software is robust and reliable. Providing access to software engineering skills could create a significant number of opportunities to commercialise software developed at the University.

6 Do researchers have access to the software expertise they need?

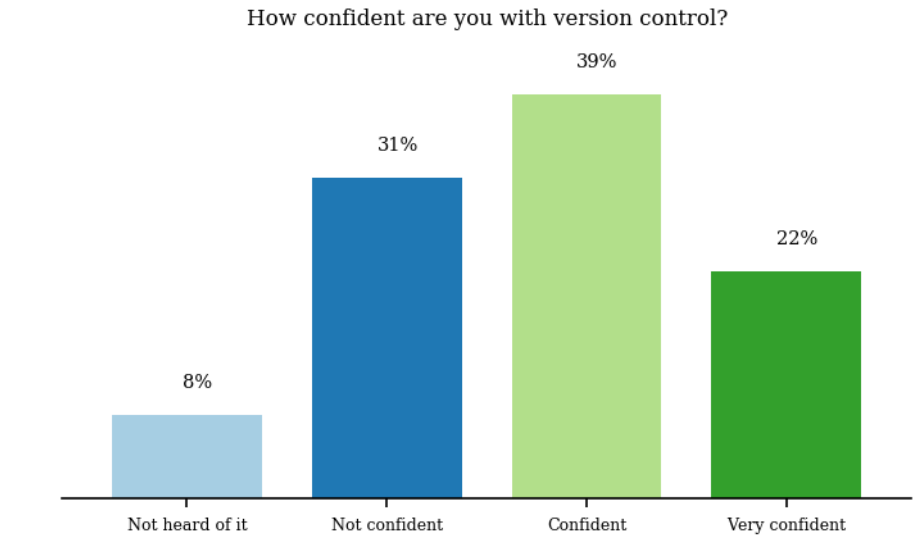


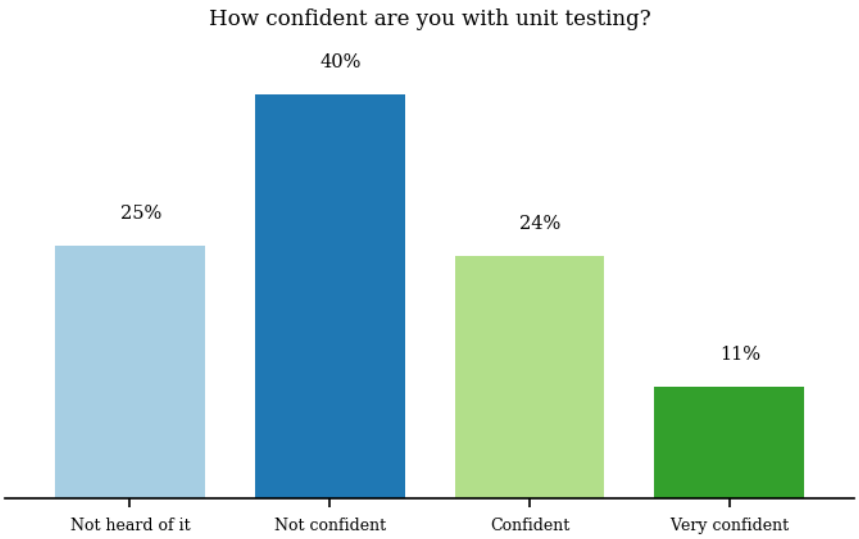
Researchers who develop software were asked to judge their software-development expertise on a scale from 1 (described as a "beginner") to 5 (described as a "professional"). The majority response (35%) was recorded in the 4 out of 5 category. 15% of staff described their expertise as "professional", and only 18% judged themselves as having poorer than average software-development expertise. At first sight, this result is grounds for some optimism. However, judging one's own expertise can be highly subjective. We investigated further with questions on training and on the understanding of a few basic software-engineering techniques.

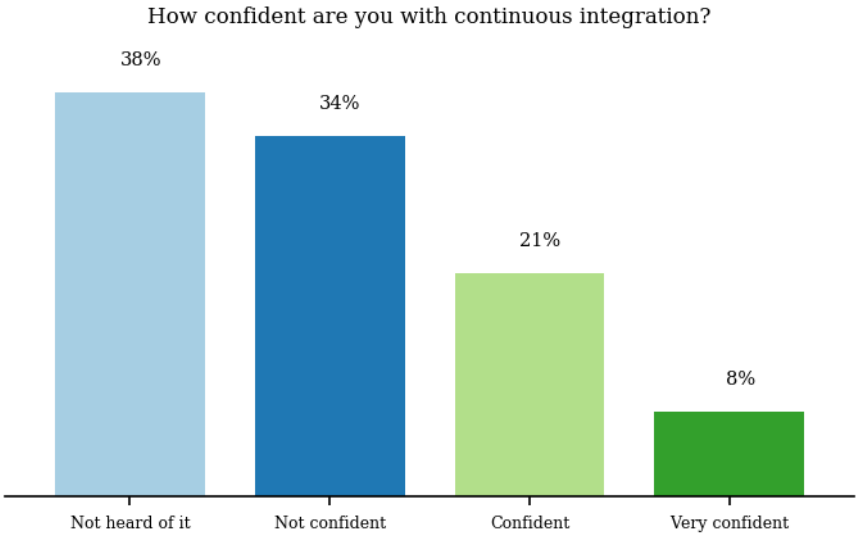


Almost two-thirds of researchers who develop software do not believe that they have received the training needed to produce reliable software. This raises concerns about the publications based on results generated by homegrown software. This question also raises doubts about the level of development expertise recorded in the previous question, e.g. a third of researchers who evaluated their development expertise at the "professional" level reported that they had not received sufficient training to develop reliable software.

To remove some of the subjectiveness from the development-expertise question, we asked further questions about three of the most basic software-engineering techniques: version control, unit testing and continuous integration. Any software engineer would be confident with these techniques.



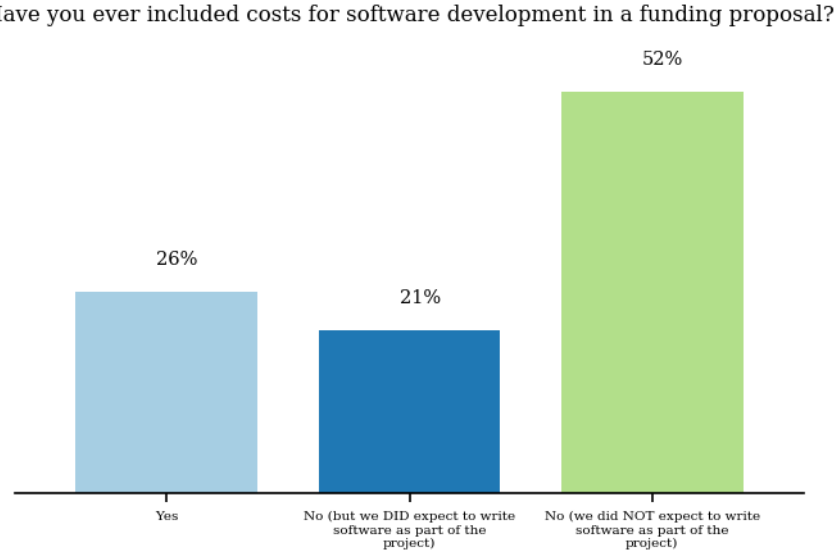




Version control is certainly the most well-known technique, with a majority of researchers reporting that they are confident or very confident with the technique. Yet one in ten of the researchers who develop code have not heard of this most basic software-engineering technique. As we move to more advanced (yet still basic) tools, confidence levels drop with only a quarter of staff confident with unit testing and the majority of staff not having even heard of continuous integration. In conclusion, despite the reasonably high level of self-assessed software development expertise, it seems reasonable to state that a significant number of researchers who develop their own software lack the skills to do so reliably.

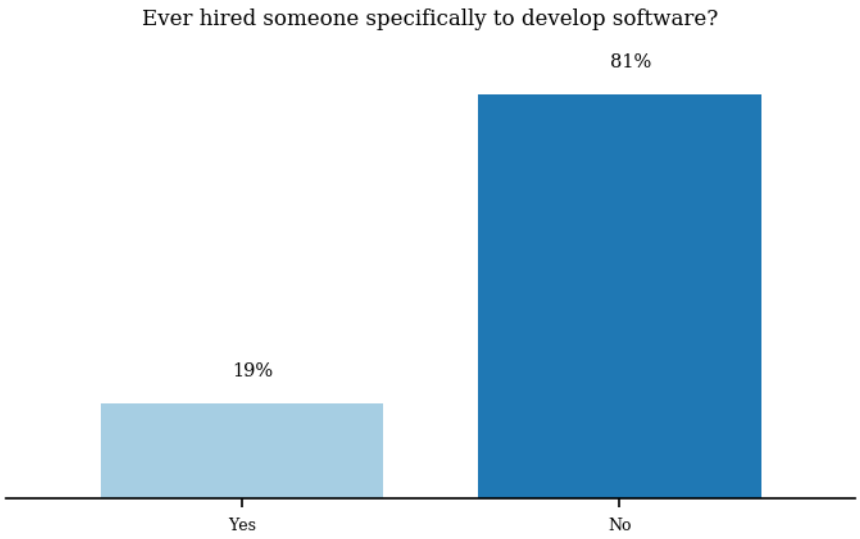
7 Access to software-engineering expertise

Research groups relies on a broad range of skills, many of which researches have to acquire themselves through training - where it exists - or through trial and error. It is simply not possible for all researchers to master of all the skills needed in modern research. The obvious solution is to recruit the skills needed from an expert. To investigate this phenomenon, we asked whether researchers had included costs for software development in their funding proposals and whether they had hired staff to develop software for them.



Respondents who were not involved in writing funding proposals were dropped: leaving 280 responses on which to conduct the analysis.

Just over a half of respondents did not expect to write software in their research, which is in conflict with the response to the early question which found that 67% of researchers developed their own software. Around a quarter of researchers had included costs for software development in their bids, but interestingly around a fifth of researchers expected to develop software but had not included costs for doing so in their bids. This is likely caused by the expectation that postgraduate students and postdoctoral researchers will pick up any software development that is required (although it may also be due to the incorrect belief that research funders do not look kindly on software development costs in research bids). Regardless of the cause, it cannot be good practice to expect work to be completed that has not been budgeted for. Furthermore, this result shows that a review of bids won by University researchers will underestimate the amount of software development occuring at the University by around a half.

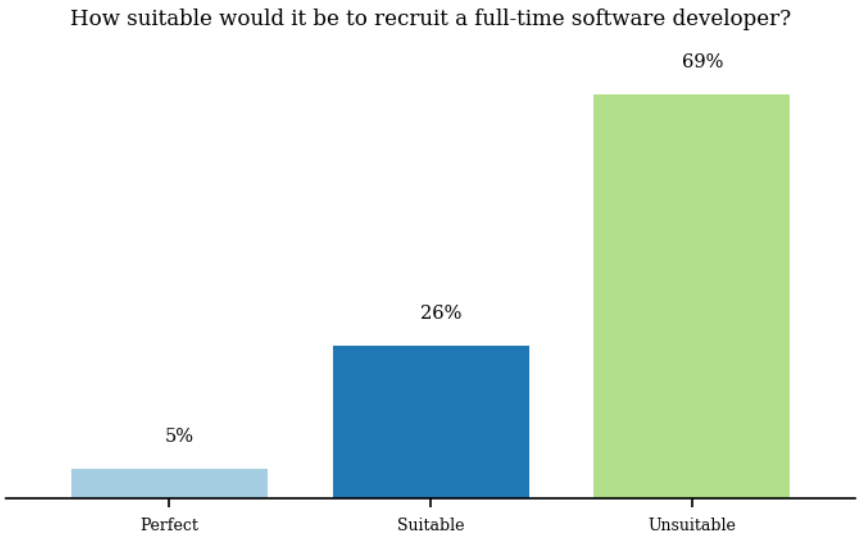


With almost half of researchers at the University expecting to develop software for their research, it seems reasonable to expect a significant number of them to have hired a software expert. However, only 19% of researchers have hired someone specifically to develop software.

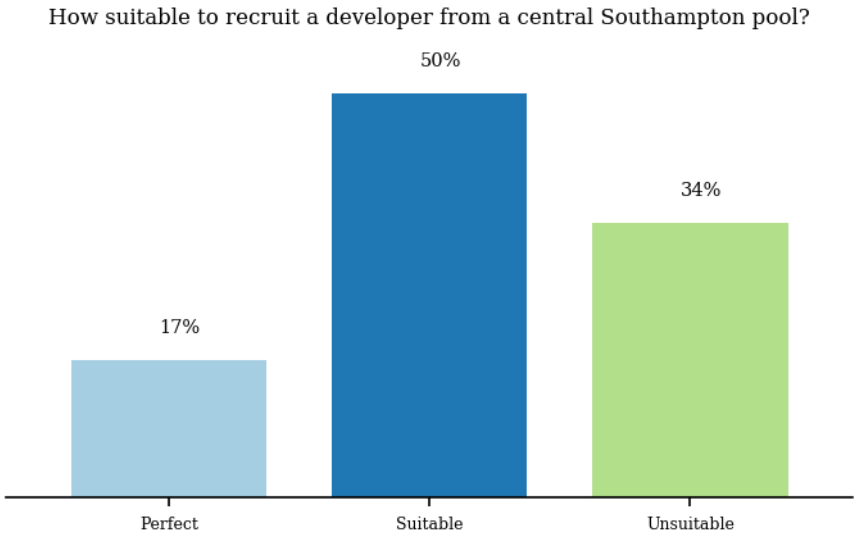
This result indicates that the majority of software development is being conducted by people hired for skills in other areas - most likely postgraduate students and postdoctoral researchers as discussed above. There is an obvious problem of handing the responsibility for software development to people who are not trained to develop software.

8 Preferred model for accessing software expertise

There is significant demand for software engineering expertise at the University. We asked respondents for their preferred model for recruiting this expertise.



Over two-thirds of researchers would find employing a full-time software developer unsuitable - most likely because they lack either the funds to hire a full-time developer or they do not generate enough software development work to keep one employed.



The alternative option within the University of Southampton (and the other 27 UK universities with a Research Software Engineering group) is to hire a Research Software Engineer from an RSE Group. The researcher gains access to a range of professional software engineering expertise, but they do not need to concern themselves with recruitment and - importantly - they pay only for the time they need.

Based on these benefits, it is unsurprising that over 80% researchers find this model for recruiting expertise either suitable or perfect.

9 References

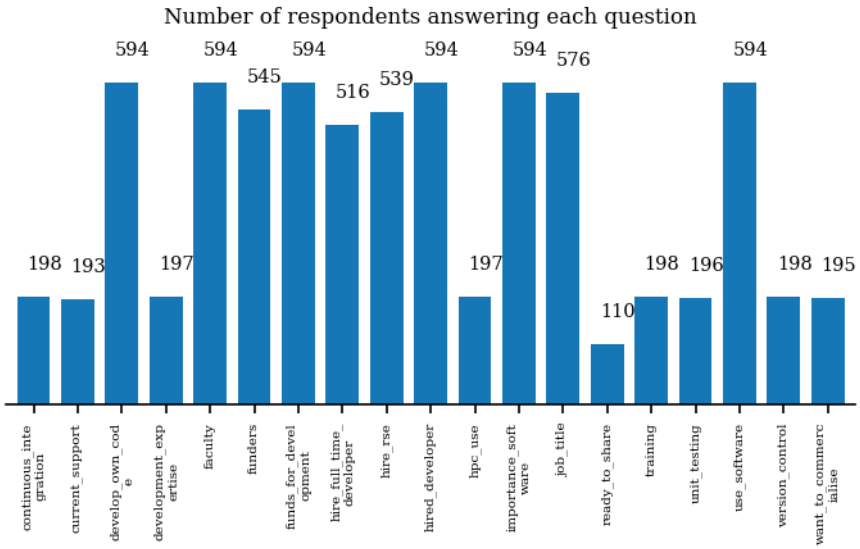
1. National software survey: "S.J. Hettrick, et al, UK Research Software Survey 2014", DOI:10.5281/zenodo.1183562.
2. https://rsg.soton.ac.uk
3. Github repository for this project: https://github.com/Southampton-RSG/soton_software_survey_analysis_2019

10 Appendices

We collected information on the number of responses to each question and asked further questions in the survey to understand the demographics of the respondents, their use of HPC and their rating for software support.

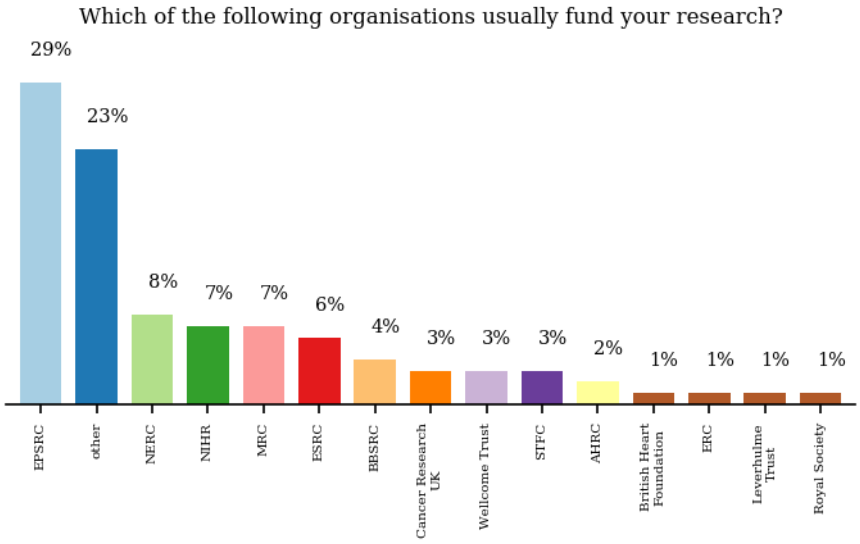
10.1 Appendix A

Number of responses to each question:



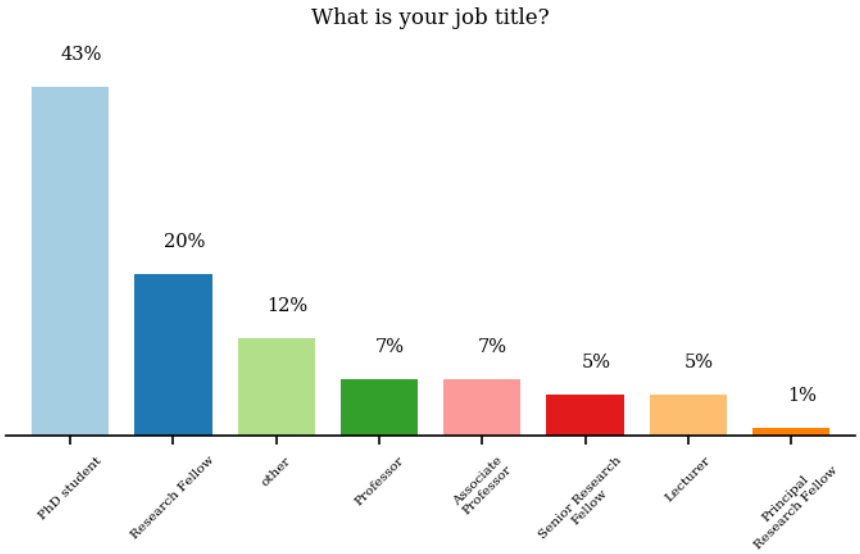
10.2 Appendix B

Origin of research funding of respondents:



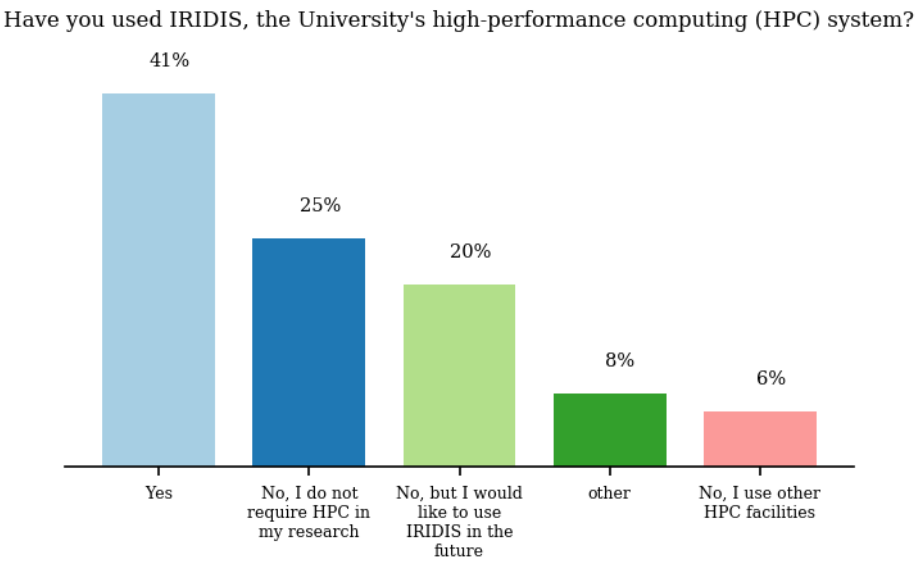
10.3 Appendix C

Cleaned job title of respondents:



10.4 Appendix D

We investigated whether researchers who develop software had ever used the University's HPC system:



10.5 Appendix E

We investigated how researchers who develop software felt about the University's current level of support for software development: