

# **Report on the Research Software Survey at the University of Strathclyde**

## **1. Participants**

Out of the 110 valid responses so far 49% were from the Faculty of Engineering and 46% from the Faculty of Science with the remaining 5% distributed over the Faculty of Humanities and Social Sciences and the Business School.

44% of the respondents were faculty or on a tenure track position. 22% fell into the category Research/KE/Teaching Associate or Assistant, 9% were Research/KE/Teaching Fellows and 18% were PGR students with 6% falling into other categories.

The following departments formed more than 5% of the total responses: Architecture 12%, Chemical and Process Engineering 10%, Computer and Information Sciences 10%, Electronic and Electrical Engineering 8%, Mechanical and Aerospace Engineering 11%, Physics 18%, Pure and Applied Chemistry 6%, Strathclyde Institute of Pharmacy and Biomedical Sciences 7%

*This shows there is a strong interest in this topic across at least two faculties. While appreciate that research software engineering might be not quite as relevant in the faculty of Humanities and Social Sciences and the Business School, we anticipate there is a large, hidden interest in big data applications, which also require professional developers.*

## **2. Software Usage**

Around 95% of the respondents use research software, which was classified as any software they have used in the generation of a result that they expected to appear in a publication, ranging from a script with a few lines to fully-fledged software suites. 87% said that software was vital (64%) or very important (23%) for their work.

*Clearly, research software is a crucial infrastructure for almost 9 out of 10 researchers at Strathclyde. This result reflects total dependence on software and is even higher than the average during a survey across 15 Russell Group universities in 2015, which found that 7 out of 10 researchers said their work would not be feasible without software.*

## **3. Software Development**

Of the respondents 59% develop their own code. The personal expertise in software development is rated by 79% as expert (14%), advanced (29%) or intermediate (36%), whereas the remaining 21% rate their experience as limited (9%) or complete beginner (12%).

We asked how confident the users are with some of the most common software development technologies.

- Version control: very confident (23%), confident (44%), not confident (27%), not heard of it (6%)
- Unit and regression testing: very confident (12%), confident (20%), not confident (42%), not heard of it (26%)
- Continuous integration: very confident (12%), confident (17%), not confident (41%), not heard of it (30%)
- Automated compiling and linking: very confident (17%), confident (33%), not confident (36%), not heard of it (14%)
- Automated documentation generators: very confident (6%), confident (29%), not confident (47%), not heard of it (18%)

Asked whether their research software is ready to be released or used outside their group, 24% responded it has already been released, 18% say it is ready to be released or shared, 58% say it is not ready to be released or shared.

24% say they received sufficient training to develop reliable research, whereas 76% say they have not. 92% say the current level of support from the University of Strathclyde is average (30%), below average (19%) or poor (43%).

88% say hiring an expert developer from a pool for a particular time would be perfect (33%) or suitable (55%). This compares to 55% who said hiring a fulltime developer would be perfect (11%) or suitable (44%).

50% of the relevant respondents who develop their own software would like to commercialise it to some extent.

*The majority of respondents develops their own software. Almost half of the developers rate their expertise as above average. However, when asking about common development tools and practices that are beyond version control systems, the dominating majority turns out to be either not confident or has not even heard about the most basic concepts like e.g. unit and regression testing (66%), continuous integration (71%), automated compiling and linking (50%) and documentation generators (65%).*

*More evidence that personal development expertise could be overrated – a known mindset among software developers – can be found in the 76% who said they have not received sufficient training to develop reliable software. More than 9 out of 10 respondents say the current level of support for research software development at Strathclyde is at best average.*

*The vast majority of respondents could make better use of an expert developer from a centralised pool. Although 55% said a fulltime developer would be also suitable or even better, this figure doesn't reflect the common issues related to short-term recruiting or the*

*generation of continuous funding streams, loss of institutional knowledge and the required depth and breadth of expertise that can only be consolidated across a collaborative team.*

#### **4. Funding**

Of the total 104 participants who responded to this question, the most common answers were EPSRC (64%), Industry (30%), Innovate UK (23%) Leverhulme Trust (12%), BBSRC (9%), Scottish Funding Council (8%) and ERC (7%). Note that multiple answers were possible and the percentage includes as well PGR students.

Asked, whether the participants were involved in writing funding applications and ever included costs for software development, 30% answered yes, they did, and 70% did not. Of those who did not cost a developer in a proposal, 53% expected that software would be an output of the project.

Only 13% of the respondents ever hired specifically a person to write software. 68% never hired a dedicated developer, but 19% said that someone in their group has.

*The lion share of funding applications goes to EPSRC. This is important as it is EPSRC that recognises software as a critical infrastructure and has always emphasised the importance of dedicated developer time on software-intensive grant proposals. Moreover, EPSRC leads this agenda on behalf of all other UK Research Councils. Other highly represented examples in the survey were Innovate UK and BBSRC. Obviously, many PIs expect to write software, but never hired a dedicated developer. These proposals are guaranteed to become less competitive compared to institutions that have a convincing strategy for research software engineering.*

#### **5. HPC Usage**

We asked whether the participants use the ARCHIE-WeSt service. Developing parallelised code is often what researchers cannot do themselves anymore.

12% use exclusively ARCHIE-WeSt, while 6% use exclusively other HPC facilities. 17% use a mix and another 17% would like to use ARCHIE-WeSt in the future. 47% of the respondents do not use HPC at all.

*As often forgotten, HPC is only relevant for around half the respondents. But it remains unclear how many of those who said they don't use HPC facilities require expertise in AI/machine learning or big data frameworks. This would also fall under the HPC umbrella in a wider sense.*