**2024-01-18**

# Alliance Research Software Survey Report

Digital Research Alliance of Canada | Alliance de recherche numérique du Canada

# Table of Contents

# Contributors

The research software survey was designed and developed by the Alliance Research Software (RS) Strategy Working Group (WG):

Girma Bitsuamlak (Western University)

Susan Brown (University of Guelph)

Liseanne Cadieux (The Alliance)

Brian Corrie (SFU)

Carolyn Côté-Lussier (Institut National de la Recherche Scientifique)

Fares Dhane (The Alliance)

Félix-Antoine Fortin (Université Laval)

Kimberley Hartley (The Alliance)

Mark Leggott (The Alliance)

Tamanna Moharana (The Alliance)

Pierre-Olivier Quirion (McGill)

Y.G. Rancourt (The Alliance)

Seppo Sahrakorpi (The Alliance)

Abdel Yousif (University of Calgary)

Qian Zhang (The Alliance)

Authors of the report on the Researcher Software Survey:

Fares Dhane (The Alliance)

Brian Corrie (Simon Fraser University)

Felipe Pérez-Jvostov (The Alliance)

# Key Findings

**Research Software Development:**

Teams involved in research software development often involve multiple types of roles, and doctoral students are often the primary developers, particularly in Sciences and Engineering (SE) and Health Research (HR).

Teams vary in size and composition, with Social Sciences and Humanities (SSH) and HR respondents generally having less experience in software development compared to SE.

Proficiency in programming/coding also varied across domains, with SE respondents typically having more advanced skills.

Respondents from SSH spend less time developing software compared to SE and HR researchers.

Respondents from SE tend to allocate more project time to research software development than other domains.

While version control is common, there is high variability in adherence to standards and best practices. Proper software documentation remains a concern.

**Sharing and Publication of Research Software:**

Respondents share software mainly through platforms like GitHub, but documentation practices are inconsistent.

Many respondents have publications referencing their research software, but awareness of principles for research software development is highly variable.

Most respondents have received software-related training, primarily through self-teaching and online resources.

There's interest in training but less motivation for sustainability-related skills, and participation in software engineering associations is low, especially in SSH.

**Support and Funding for Software Development:**

Access to software development support was considered important across domains, but many respondents lack access to such services.

Funding emerged as a significant barrier to research software sustainability across domains. Institutional support is essential for research software development, but dedicated funding is limited.

Most respondents agreed that facilitating reusability, reproducibility, and open science are important, yet there was no clear agreement as to roles and responsibilities for long-term software sustainability.

**Software Preservation and Challenges:**

While most respondents were familiar with common repositories like GitHub, familiarity with other repositories like Zenodo or DockerHub varies across domains.

Challenges in accessing and using preserved software included financial constraints, lack of expertise, time limitations, and less familiarity with specific repositories.

Peer review of code and support services for documentation and software sustainability were deemed valuable.

# Background and Context

Research software plays a crucial role in modern, data-driven scholarship. Not surprisingly, researcher software is a foundational pillar of digital research infrastructure (DRI), as it is a prerequisite for data analysis and manipulation, and it optimizes researchers' use of compute resources. There has also been a growing recognition of research software as a research output, in that like data, it can be shared to increase reuse, reproducibility, and further knowledge generation. Yet, efforts to understand how Canadian researchers develop, use, and fund their research software activities has been conducted in the context of the broader DRI and have not been able to capture the rapidly evolving practices, or the level of adoption by different research communities. To bridge this gap, the Digital Research Alliance of Canada (the Alliance) conducted an extensive survey of the Canadian research community to better understand the existing gaps and strengths within the Canadian context.

The survey was open between May 10th and June 8th 2023. This report presents the key findings of the survey and is organized in sections: Respondents, Research Software as an Undervalued Output of Research, Research Software Use, Research Software Development, and Research Software Sustainability. To facilitate the reading and cross-checking with the survey, references to specific questions are represented within brackets and italics (e.g., *[B3]*).

# Respondents

A total of 548 researchers responded to the survey. In this context, respondents are individuals who participated in the survey by answering at least one question, regardless of whether they completed the entire survey (152 researchers completed the complete survey). Each question had on average 181 responses. University Faculty accounted for 38% of all respondents (Faculty - Professor, Faculty - Other), with Research Staff (12%) Administration (12%), Research Software Engineer/Expert (8%) being the next three most highly represented groups *[B3]*. The percentage of other respondents and their roles can be seen in Figure 1 *[B3]*. Seventy one percent of respondents stated that they were in a full time permanent position, while 22% stated they were on a fixed-term contract, 3% in a part-time position, and 3% stated Other *[B5]*.

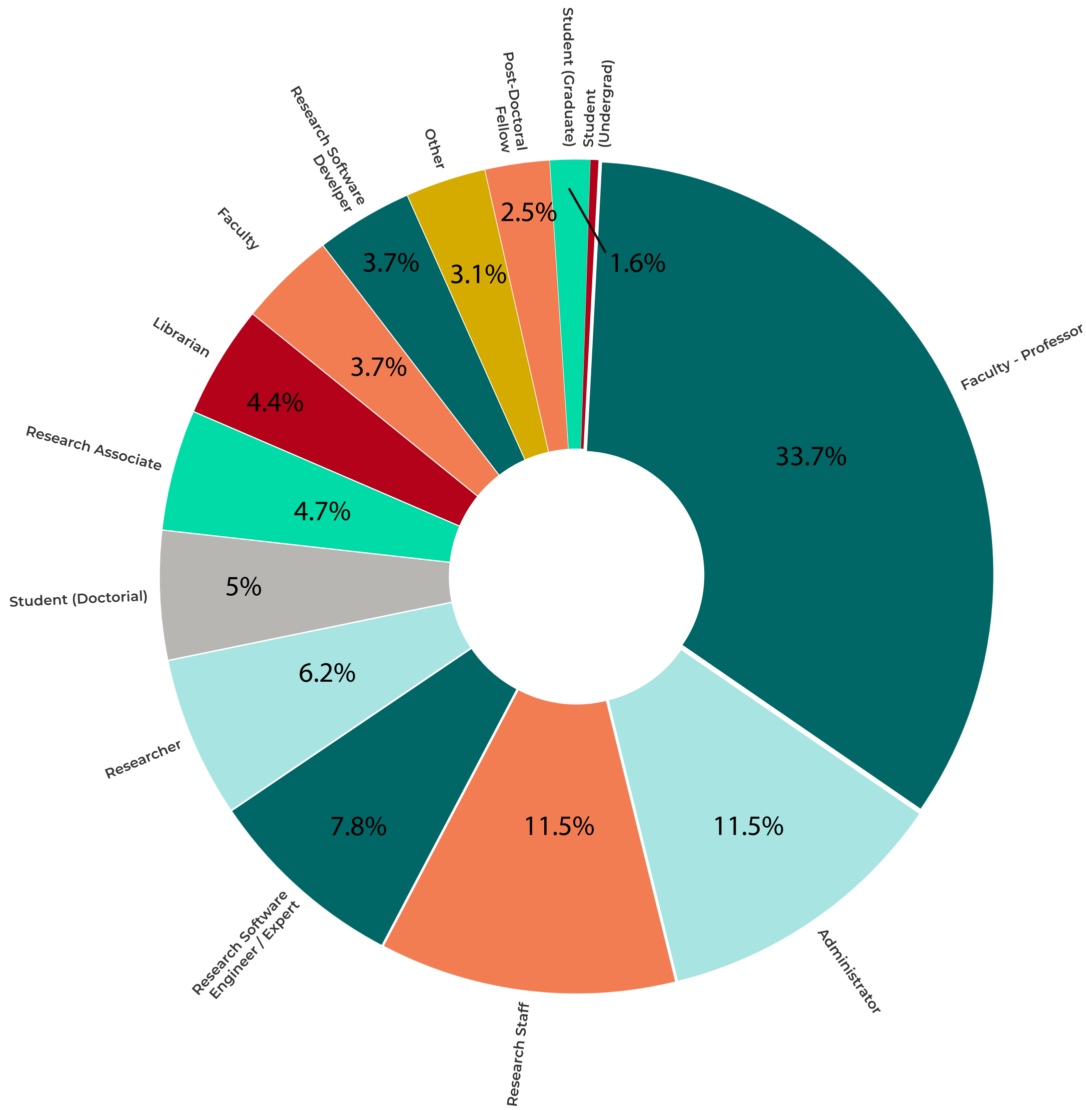



**Figure 1. Question [B3].** Distribution of respondents by role.

Participants were asked to identify themselves based on the Canadian Research and Development Classification (v.1 2020) *[B2].* There was broad participation across Divisions, 30% identified as Natural Sciences; 24% with Medical, Health, and Life Sciences; 17% with Engineering and Technology; 14% with Humanities and Arts; 12% with Social Sciences; and finally 3% with Agricultural and Veterinary Sciences (Figure 2). To facilitate the interpretation and presentation in this report these six categories have been consolidated into three broad groupings. This should allow readers to focus on the broader patterns rather than intrinsic differences between Divisions. The three categories are the following: *Sciences and Engineering* (SE), encompassing Natural Sciences, Engineering and Technology, and Agriculture and Veterinary Sciences; *Social Sciences and Humanities* (SSH), covering Humanities and the Arts, and Social Sciences; and *Health Research* (HR), which included Medical, Health, and Life

Sciences. The distribution of respondents across the three consolidated categories is presented in Figure 3.

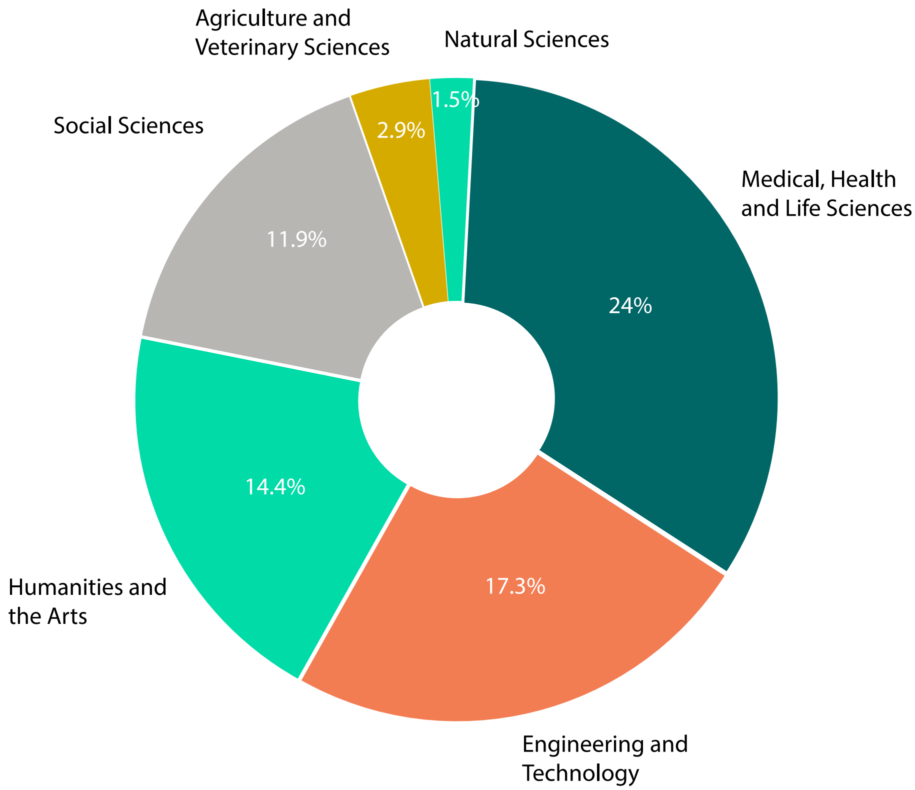



**Figure 2.** Distribution of respondents according to the Canadian Research and Development Classification.

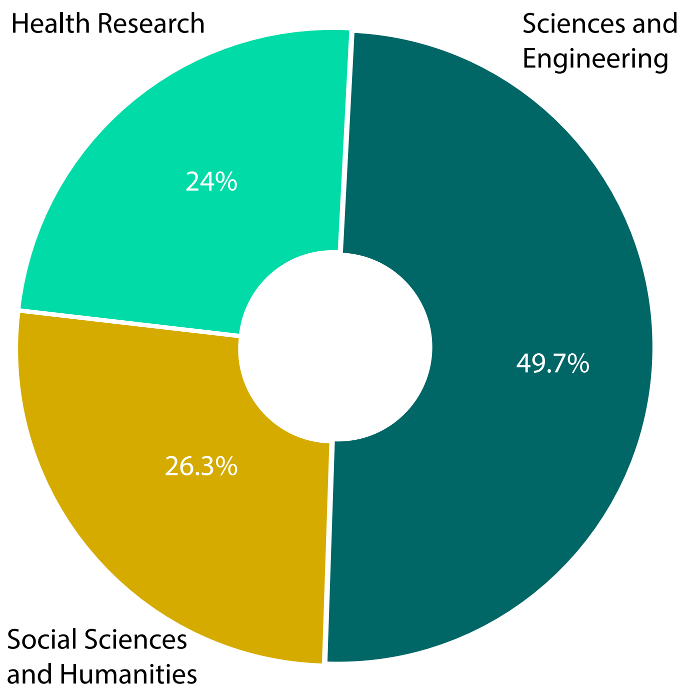



**Figure 3.** Distribution of respondents across the three aggregated research domain categories.

# Research Software as an Output of Research

One of the first questions of the survey was related to the concept of research software and to identify whether the definition currently being used by the Alliance resonates with the understanding different researchers have of research software [*C1*]. The definition provided was "*Research software is defined from a broad perspective as an emerging and complex intersection of tools, disciplines, services, platforms, hardware, resources, and the people (users, researchers, developers, stakeholders, personnel, and communities, etc.) who use and contribute to them.*" The majority of respondents (65%) agreed with the definition provided, yet many did not (35%), either because they found it too vague or they themselves were uncertain. For some, research software encompasses any software tool used in their academic activities, while others specifically associate it with the code developed for their research. Considering the differences in interpretation of what research software is or is not, generalized conclusions from the results presented in this report should take such variation into consideration.

Most respondents (91%) considered the use of research software critical to their research *[C2]*. Fifty percent of respondents considered the development of research software a primary output of their research *[C3]*, with clear differences across the domains (SSH - 36%; SE - 59%; HR -

48%). Among respondents who consider research software as a primary output of research 43.4% were Faculty–Professors, followed by Research Software Engineer / Expert (10%). These findings are in line with similar international results where it was found that 92% of researchers in the United Kingdom and 95% of researchers in the United States rely on research software.[1] The UK survey also found that 56% of researchers in the UK develop their own research software.[2] This underscores the importance of research software in modern research and indicates that research software plays a similar role in Canada as it does in other countries.

# Research Software Use

Many respondents (85%) indicated having access to readily-available research software or that they already know which software meets their needs *[E2]*, thus only spending a small portion of their time searching for research software (i.e., 10% or less). However, there are differences in how researchers of different disciplines find the software they use. For example, researchers in SE and HR primarily rely on scholarly publications and their supplemental materials, while researchers in SSH rely more heavily on their community of experts, workshops and courses, and conferences *[E1]* (Figure 4). Researchers in SE and HR also use source code hosting platforms such as GitHub to discover software, with 37% and 32% respondents in these disciplines using them. Only 14% in the Social Sciences and Humanities mentioned using GitHub, however.

[1] https://www.software.ac.uk/blog/2014-12-04-its-impossible-conduct-research-without-software-say-7-out-10-uk-researchers, https://zenodo.org/record/1183562; Nangia et al., “Track 1 Paper: Surveying the U.S. National Postdoctoral Association Regarding Software Use and Training in Research”, In *Workshop on Sustainable Software for Science: Practice and Experiences (WSSSPE 5.1)*. 2017, https://doi.org/10.6084/m9.figshare.5328442.v3.

[2] https://www.software.ac.uk/blog/2014-12-04-its-impossible-conduct-research-without-software-say-7-out-10-uk-researchers, https://zenodo.org/record/1183562.

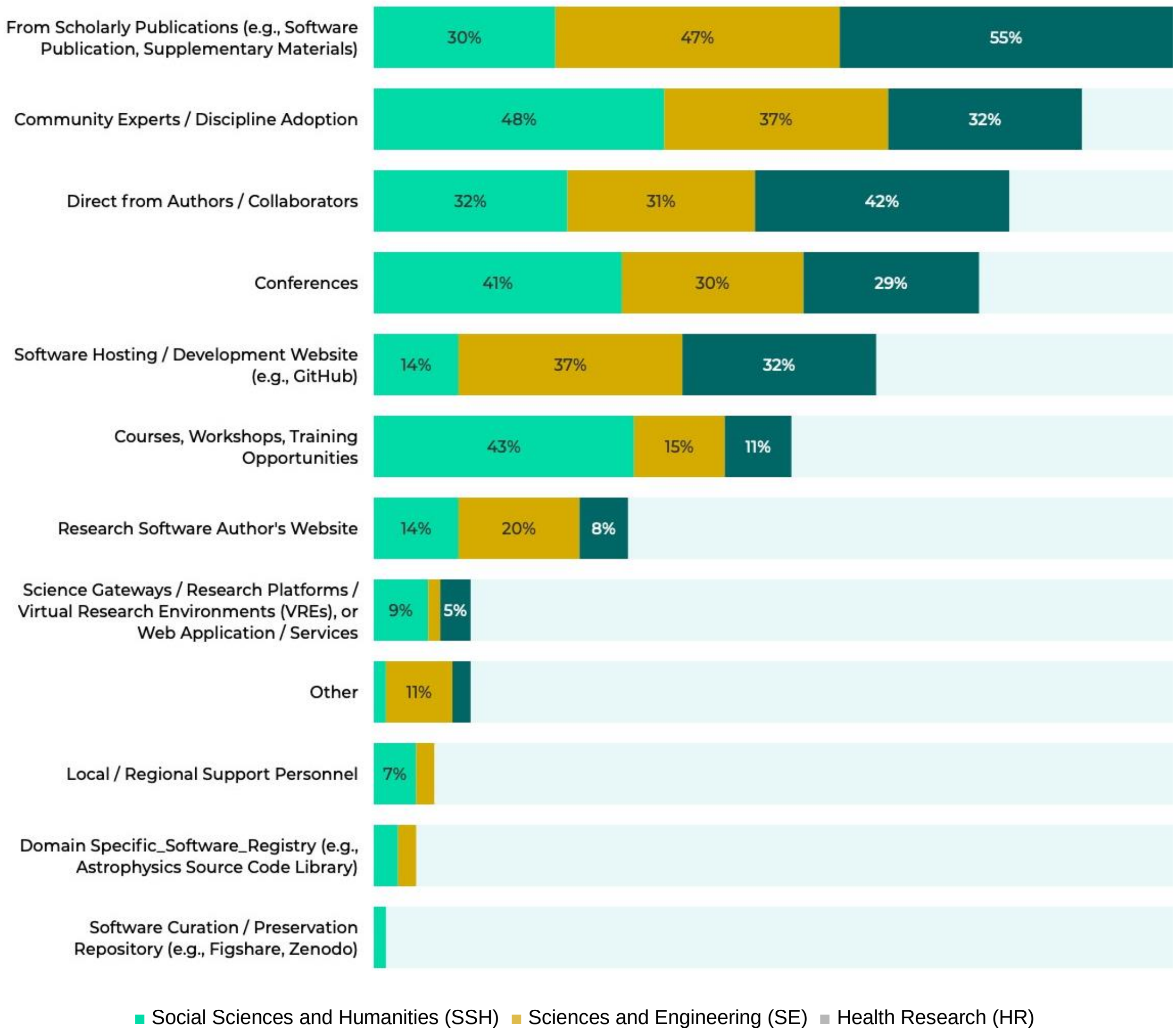



**Figure 4. Question [E1]: Choose the 3 most common means which you have found other people's research software you use in your research.** Responses measured in percentage across Tri-Agency's domains. Social Sciences and Humanities, n = 30; Health Research, n = 28; Sciences and Engineering, n = 84; Unique respondents = 142.

The factors that influence researchers' choice for using a given software tool or platform are very similar across disciplines, and have relatively similar importance *[E14]* (Figure 5). These include user-friendly interfaces and efficient performance, cost (including Open Source software), sustainability, and documentation. Performance, however, was least valuable in the SSH (55%) compared to SE (75%) and HR (82%).

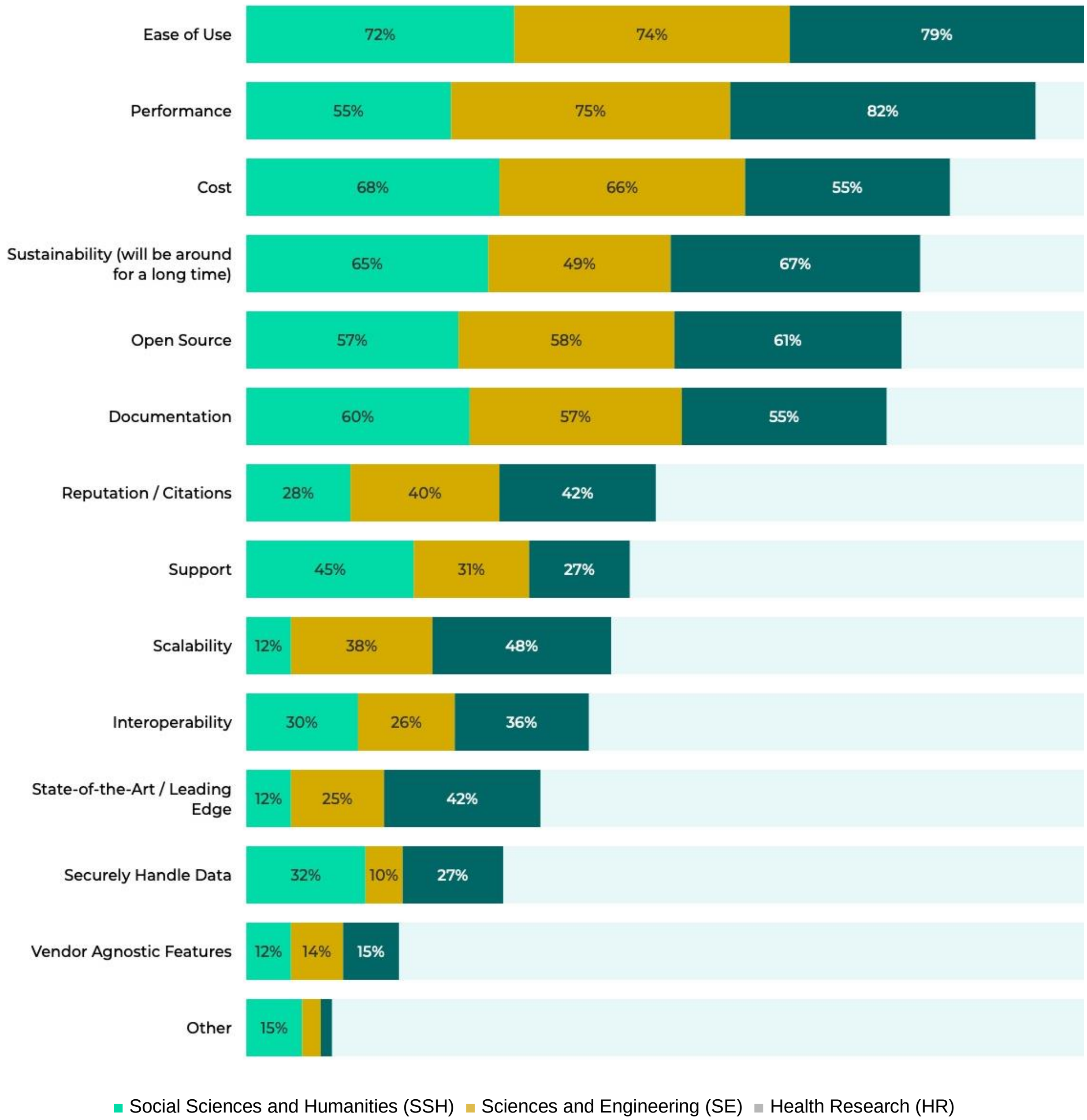



**Figure 5. Question [14]: What factors do you consider in choosing which research software to use?**
Responses measured in percentage across Tri-Agency's domains. Social Sciences and Humanities, n = 40; Health Research, n = 33; Sciences and Engineering, n = 77; Unique respondents = 150.

Respondents reported a range of technical challenges when using other researchers' software *[G1]*, including a lack of documentation (SSH - 59%; SE - 79%; HR - 74%), difficulty in finding software (SSH - 56%; SE - 38%; HR - 45%), difficulty installing it (SSH - 33%; SE - 45%; HR - 58%), issues with compatibility (SSH - 41%; SE - 34%; HR - 45%), lack of training (SSH - 54%; SE - 27%; HR - 39%), and difficulty maintaining the software (SSH - 18%; SE - 26%; HR - 42%) with a small number of respondents (less than 13%) reporting other challenges. Non technical challenges when using other researchers software *[G2]* were also reported; these included lack of documentation (SSH - 51%; SE - 68%; HR - 71%), lack of long term support/reproducibility (SSH - 59%; SE - 49%; HR - 61%), lack of functionality (SSH - 46%; SE - 49%; HR - 52%), a lack of support (SSH - 16%; SE - 25%; HR - 52%), and a lack of language support (SSH - 16%; SE - 7%; HR - 13%), with a small number of other challenges listed.

The most common platforms on which respondents run their research software are mainly their personal desktop/laptop (SSH - 97%; SE - 88%; HR - 68%) and/or lab computers (SSH - 47%; SE - 54%; HR - 59%) *[E3]*. ARC resources (SSH - 21; SE - 54%; HR - 50%, ), cloud resources (SSH - 26%; SE - 39%; HR - 35%), and community cloud resources (SSH - 24%; SE - 25%; HR - 26%) are also commonly used (Figure 6). Similarly, laptop/desktop computers (SSH - 82%; SE - 74%; HR - 55%) and lab computers (SSH - 45%; SE - 51%; HR - 50%) are the main mechanism by which researchers access research software. Remote command line interfaces (SSH - 27%; SE - 62%; HR - 53%) and VRE/Science Gateways (SSH - 20%; SE - 32%; HR - 37%) are additionally widely used across domains, but with command line being much more common in the SE *[E4]*. Fourty four percent of respondents stated they use research software installed and maintained on ARC platforms managed and supported by the Alliance *[E5]*.

Respondents reported that their research software workflows included a range of tools such as version control software (e.g., git SSH - 43%; SE - 67%; HR - 58%), commercial storage providers (e.g., Dropbox SSH - 59%; SE - 43%; HR - 58%), open source data software (e.g., SSH - Numpy 36%; SE - 59%; HR - 47%), and desktop tools (e.g., MS Office - SSH - 61%; SE - 27%; HR - 37%). Other tools were also mentioned but were much less used (i.e., <25%), these included continuous integration platforms (e.g., Travis-CI), cloud collaboration platforms (e.g., OSF), project management tools (e.g., Asana), commercial data software (e.g., Tableau), secure data capture software (e.g., Redcap), and electronic lab notebooks among others *[E8]*.

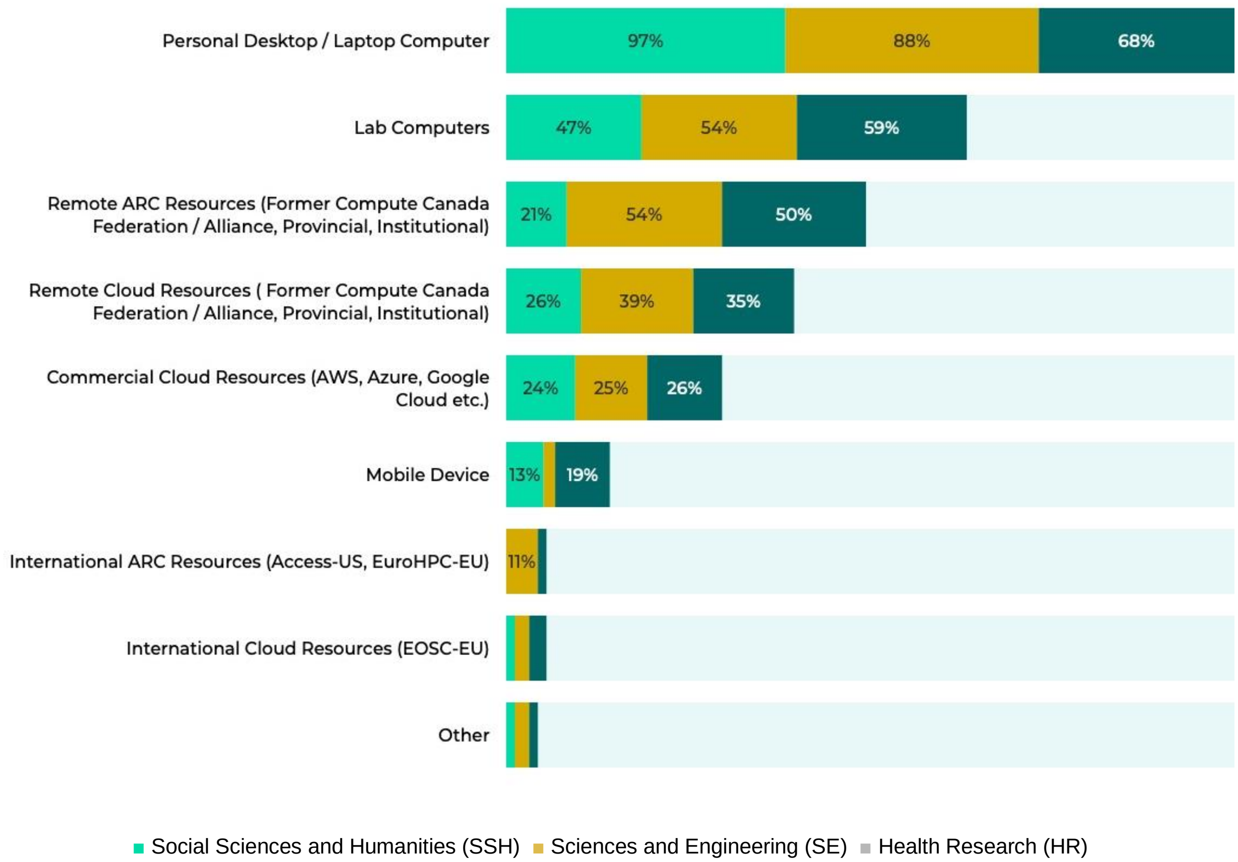



**Figure 6. Question [E3]: On which platform(s) do you use research software?** Responses measured in percentage across Tri-Agency's domains. Social Sciences and Humanities, n = 44; Health Research, n = 38; Sciences and Engineering, n = 91; Unique respondents = 173.

When asked whether respondents cited the research software they used, 76% of them confirmed doing so *[E15]*. Half of those who reference software cite the publication describing the research software while the other half cite the research software directly *[E16]*.

A limited number of respondents (29%) stated that they used research management services provided and supported by the Alliance such as the DMP assistant and the CCDB *[E6]*. Similarly, a limited number of respondents stated that they used research software platforms supported by the Alliance (18%) *[E7]*.

# Support for the use of research software

More than half of the respondents (SSH - 50%; SE - 57%; HR - 55%) stated that they themselves play a role in supporting the research software they use *[E9]* (i.e., those that identified "me" as the ones supporting the research software they use). Almost half of these respondents in the SE and SSH were Faculty–Professors (49% and 43% respectively), both much higher than their overall representation in the general pool of respondents (~33%). In contrast, Faculty-Professors in HR only accounted for 25% of respondents who support research software use (Figure 7). This may suggest that in SE and SSH Faculty-Professors play a prominent role in supporting research software–even more so than dedicated Research Software Engineers and Developers. Whereas in HR this task is more evenly shared by a variety of roles (Figure 8). The amount of Full Time Employees (FTEs) dedicated to installing and managing software, however, remained low across domains with 53% of respondents mentioned spending less than ¼ FTE in such tasks, and only 18% of respondents stated that their teams spent over 1 FTE managing software *[E10]* .

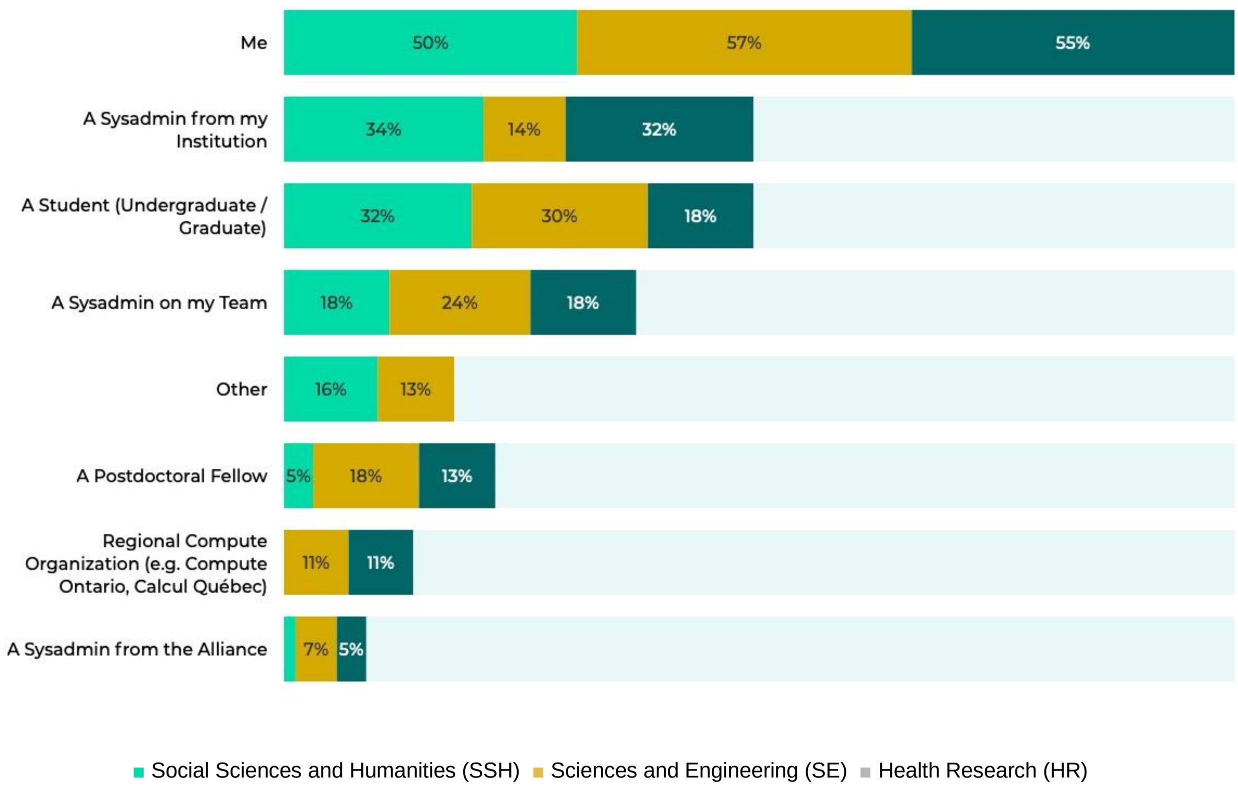



**Figure 7. Question [E9]: Who on your team supports the research software you typically use?** Responses measured in percentage across Tri-Agency's domains. Social Sciences and Humanities, n = 44; Health Research, n = 38; Sciences and Engineering, n = 91; Unique respondents = 173.


**Figure 8.** **Question [E9]: Who on your team supports the research software you typically use?** A breakdown of roles among respondents who answered 'Me' in E9.

The active involvement of researchers in research software support is not without its challenges. Almost half of respondents reported being sometimes prevented from reusing other people's research software due to lack of the necessary resources to install, use, or support it *[E18]*. Other factors that prevent the use of research software include funding, expertise, lack of documentation, and support systems required to effectively install, use, and maintain research software *[E19]*. These challenges emphasize the importance of both active researcher involvement and adequate resource provision for the reuse of research software.

In terms of the type of data stored and analyzed by the respondents, 21% of respondents stated that data was stored/collected by/about indigenous communities, 45% stated that data was stored/collected that involved personal information about individuals, and 28% stated that data was stored/collected that required a high level of security for other reasons.

# Funding the use of research software

Respondents mainly cover the costs of use of research software through their research grants *[E11],* albeit institutional funding is even more important for researchers in SSH. In the SE, funding from research grants, however, was more important than institutional funding (Figure 9).

Researchers in the Social Sciences and Humanities report using personal funds for research software (31%) more often than those in SE (13%) and HR (10%). Of those respondents who reported on how much they spend personally 44% in SSH, 30% in SE, and 19% in HR spend between $0 - $1000 on their research software *[E12]*. These are similar findings to those described for cloud compute, where researchers in SSH also used personal funds to cover the expenses related to commercial cloud (see Alliance Cloud Survey Report 2023).

In terms of research group spending, the same trend is observed, with 23% SE, 19% HR, and 38% SSH respondents spending $1-$1000 on purchasing, licensing, installing, or managing research software resources while 13% SSH , 28% SE, and 19% HR respondents spent $0 *[E13]*. A number of respondents reported spending $5000-$50K on research software (SSH - 10%; SE - 5%; HR - 13%) while a small number reported spending over $50K (SE - 5%; HR - 3%).

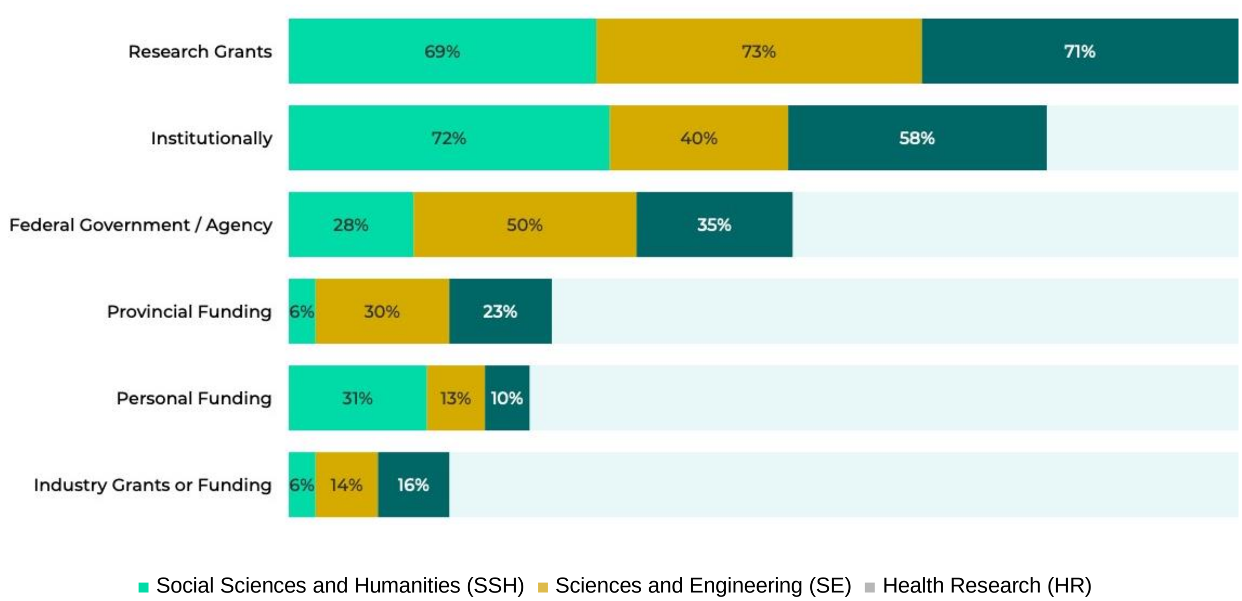


**Figure 9. Question [E11]: How does your group fund the research software that you use?** Responses measured in percentage across Tri-Agency's domains. Social Sciences and Humanities, n = 36; Health Research, n = 31; Sciences and Engineering, n = 70; Unique respondents = 137.

# Research Software Development

When asked about previous experience leading dedicated research software projects, approximately 52% of the respondents, primarily faculty members (26%) and other researchers (12%) identified having either led or are currently leading a research software development project *[D1]*. Although respondents in SE do seem to be more engaged in such activities compared to respondents in other fields (Previously led - Yes = SSH - 38%; SE - 53%; HR - 43%. Are leading - Yes = SSH - 36%; SE - 58%; HR - 45%). The majority of respondents (~70%) play multiple roles within these projects: as designers, researchers, or end users of the research software they develop *[D5]*. Respondents in SE (67%) and HR (56%) have a more active involvement in the development of research software compared to respondents in SSH (38%).

The task of developing research software within given research teams seems to be shared across different members. Doctoral students are for example the most common software developers in the SE and HR, likely because their research requires them to do so, compared to doctoral students in SSH where only 22% were identified doing so. Similarly to research software support, Faculty particularly in SE play a prominent role in software development (SSH - 33%; SE - 43%; HR - 28%). Staff positions traditionally responsible for the professional development of research software such as Research Software Developers and Engineers / Experts play a role within their teams, but to a much lesser extent than graduate students and Faculty (Figure 10).

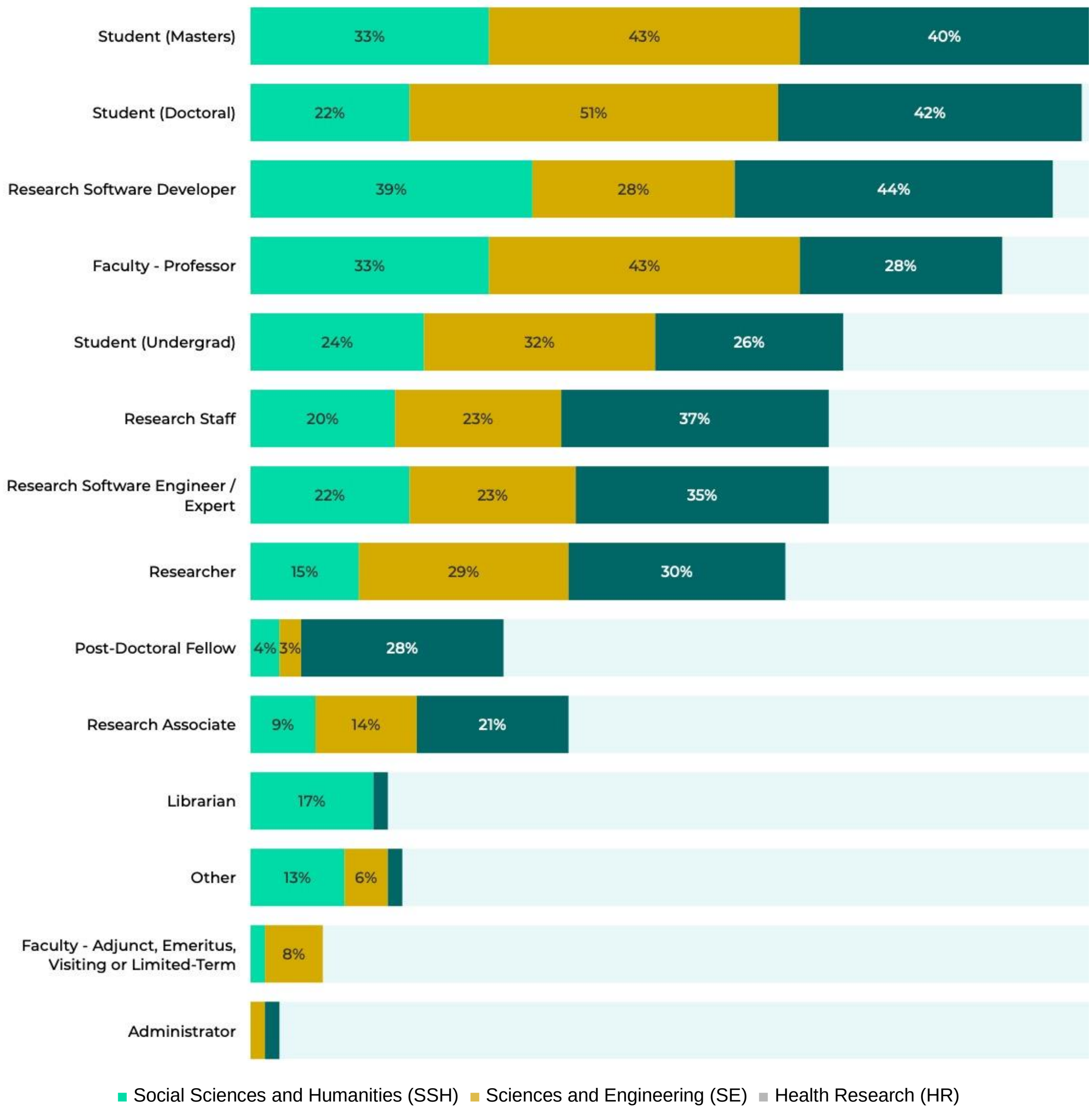



**Figure 10. Question [D6]: Who develops the software in your group?** Responses measured in percentage across Tri-Agency's domains. Social Sciences and Humanities, n = 46; Health Research, n = 43; Sciences and Engineering, n = 98; Unique respondents = 165

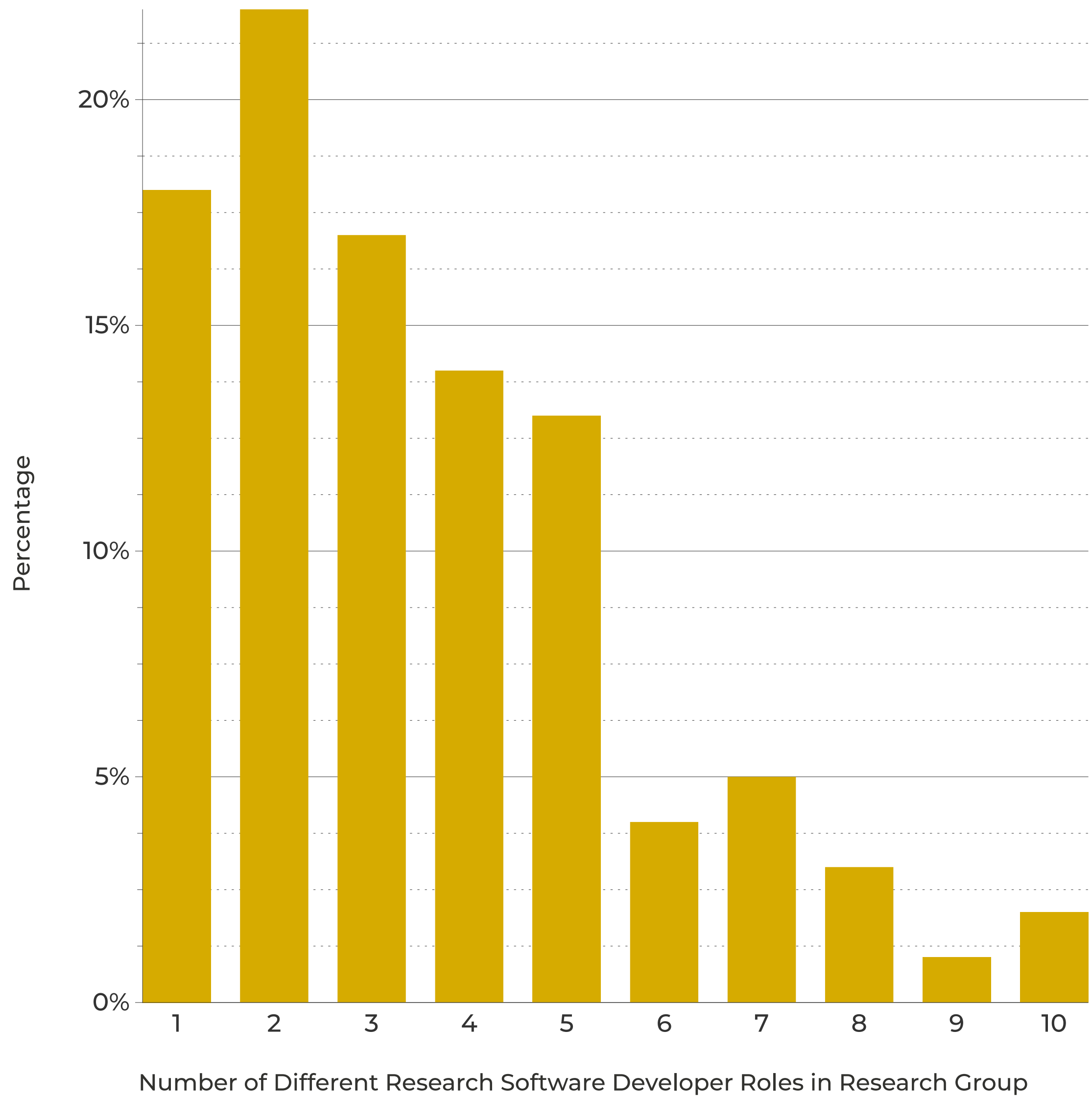



**Figure 11.** The diversity of staff roles (e.g., Faculty, Research Software Engineer, Graduate student etc.) that perform research software development in research groups.

The size and composition of research software development teams varied greatly both between and within disciplines. On average these teams consisted of 3.5 different staff positions (out of 14 positions listed - Faculty, Research Software Engineer, Masters and Doctoral Student, etc. ) with the maximum number of staff positions in a group 10 out of 14 (i.e., large diverse teams), the minimum number of staff positions in a group 1 out of 14 (a single staff position doing all of the development), with 40% of the groups reporting having 1 or 2 developer staff positions on their teams (Figure 11).

The survey revealed that across research domains experience in software development is bimodal. Over 50% of respondents in all domains have over 6 years of software development experience, while the other half less than 5 years. Some important differences between domains, however, were evident, particularly in the least experienced categories. In SSH and HR 33% and 24% of respondents stated that they have less than one year of experience in research software development, compared to only 9% of respondents in the SE *[D4]*. This survey does not allow extraction of the specific reasons for this difference, but the observed difference between the SSH, HR and SE, could be attributed to curricular differences in undergraduate courses and projects, where undergraduates in the SE are more likely to receive training or classes on software development compared to those in HR and SSH.

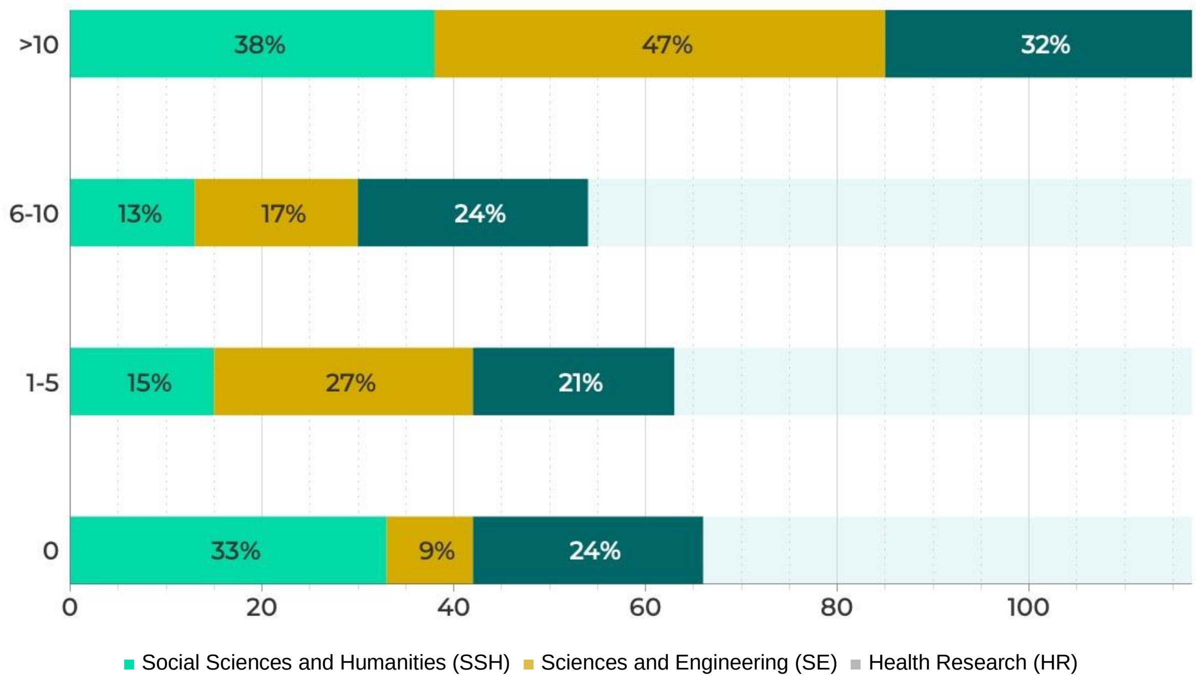



**Figure 12. Question [D4]: How many years of research software development experience do you have?**
Responses measured in percentage across Tri-Agency's domains. Social Sciences and Humanities, n = 39; Health Research, n = 38; Sciences and Engineering, n = 93; Unique respondents = 170.

The years of experience in software development were somewhat related with those of proficiency. When asked to self identify personal level of skills in programming/coding *[D26],* half of SSH respondents identified as having either limited experience or basic knowledge, compared to 11% and 27% in SE and HR respectively. On the other hand, 30% in SSH identified as advanced or expert, compared to 48% in SE and 45% in HR (Figure 13).

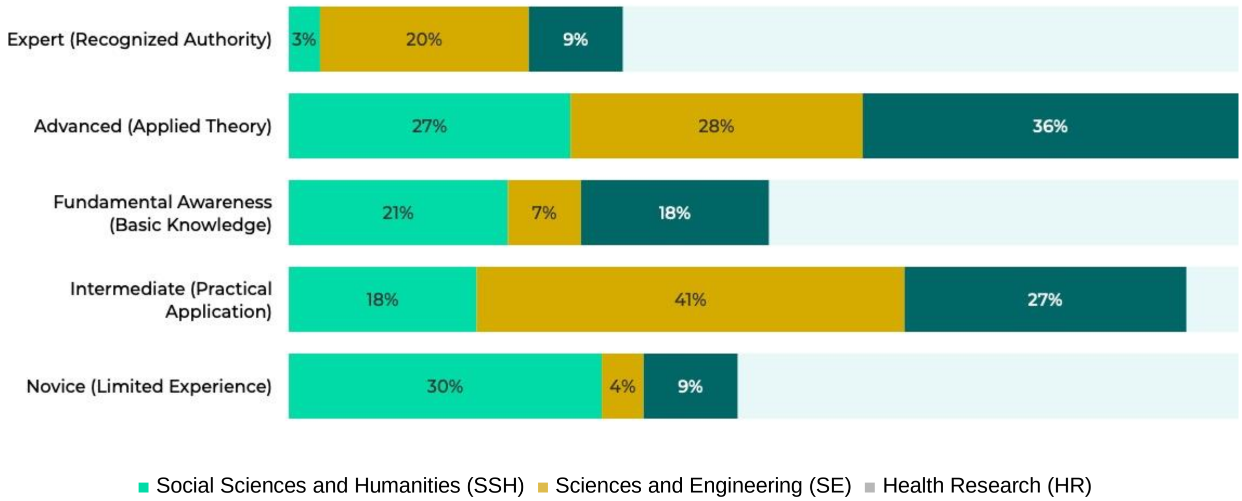



**Figure 13. Question [D26]: How would you describe your personal level of skills in programming/coding?** Responses measured in percentage across Tri-Agency's domains. Responses measured in percentage across TC domains. Social Sciences and Humanities, n = 33; Health Research, n = 33; Sciences and Engineering, n = 90; Unique respondents = 156.

The amount of time that different respondents spend on software development was relatively consistent across research domains–although like with previous questions, researchers in SE tend to spend slightly more time on such tasks. For example, 68% of respondents from SSH spend little to no time in software development tasks (i.e., between 0% - 25% of their time), while 58% of HR respondents and 40% of SE respondents fall into this category *[D27]*). Dedicated staff performing software development (i.e., 75-100%) were common across domains, but were less so in HR (Figure 14). These results suggest that SSH respondents have both less initial training in software development *[D4]* as well as spend less time developing software as researchers *[D27]* compared to SE and HR.

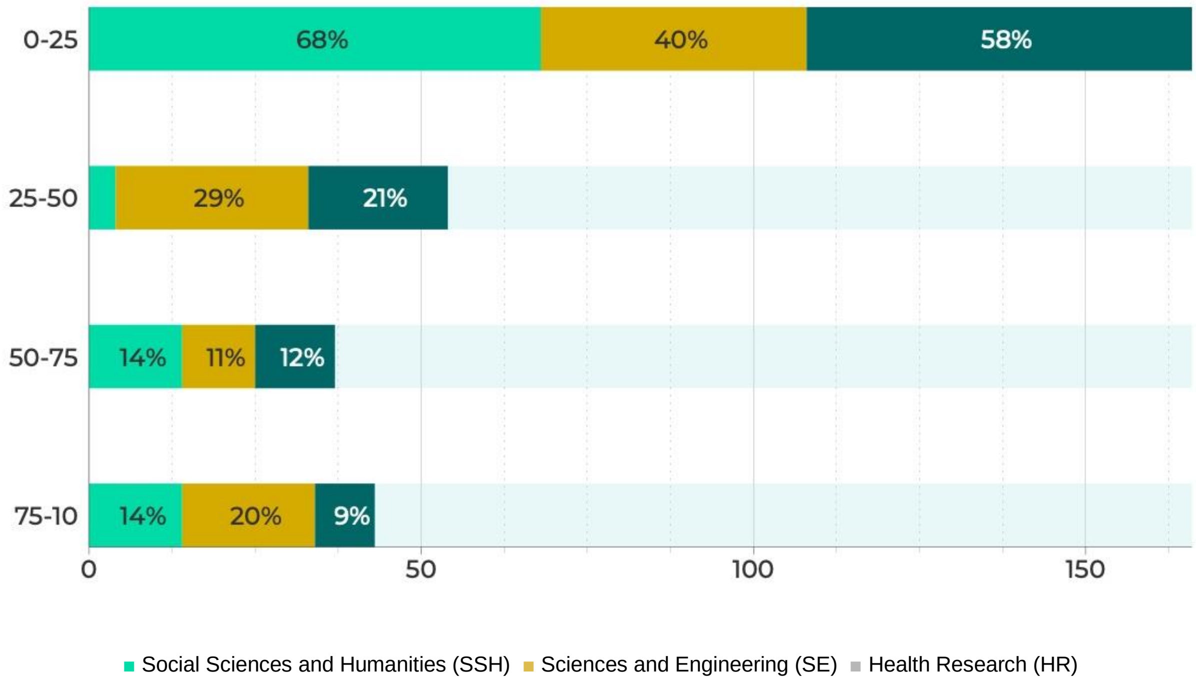



**Figure 14. Question [D27]: How much project time do you spend programming/coding on average per project (% of total project time)?** Responses measured in percentage across TC domains. Responses measured in percentage across Tri-Agency's domains. Social Sciences and Humanities, n = 28; Health Research, n = 33; Sciences and Engineering, n = 80; Unique respondents = 141.

When asked about the amount of time different research groups spent developing software, 30% of respondents stated that their teams had a significant software effort (more than 2 FTE developers), while 33% stated that their teams had a minimal software development effort (< ½ FTE) *[D7]*. Thirty seven percent of respondents described their teams as having a modest software development effort (between ½ FTE and 2 FTEs). Here, respondents in SE appear to allocate slightly greater amounts of time to research software development, compared to other domains (SSH - 24%; SE - 34%; HR - 27%; Figure 15).

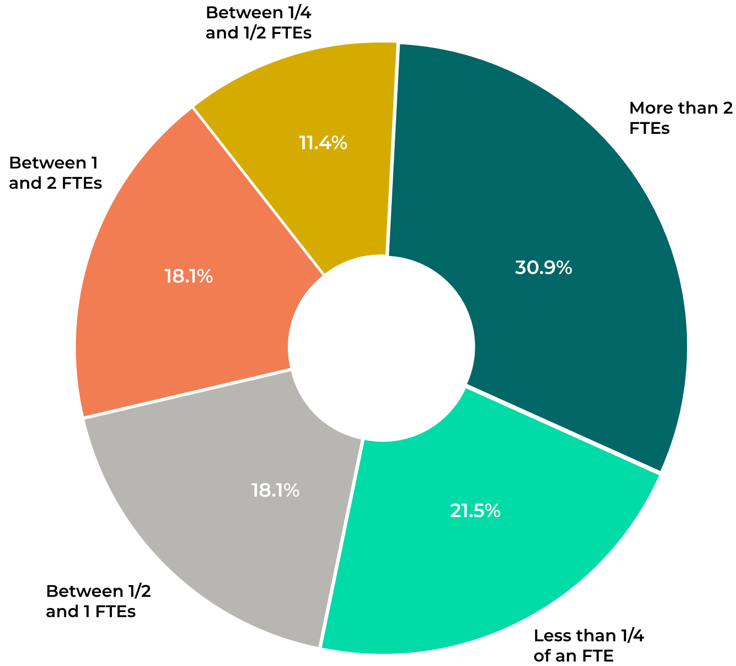



**Figure 15.** Time spent on research software development across team members.

# Types of Research Software

The types of software developed varied significantly, with all 27 software categories from the survey being selected by at least 5 respondents. Python, however, was the most widely used programming language across the three domains (SSH - 43%; SE - 76%; HR - 67%) *[D9]*. However, the findings also highlight the diverse landscape of research software languages, with 21 languages identified as being used by respondents. Respondents mainly develop open source, source code, tools that are run by researchers, and platforms that are used by external users *[D10, D14]*. When asked to provide a list of software that has been developed and actively maintained by their research groups, over 100 respondents provided a list of packages that are actively maintained *[D11]*. One third of respondents stated that they had software packages that they developed, but no longer maintained *[D12]*. Additionally, 54% of the respondents stated that their team was involved in the co-development of, or contribution to the collaborative development of a research software platform *[D15]*.

Respondents stated that version control is widely adopted (81%) for software development purposes *[D23]*. Although standards are only leveraged/used by some research software developers (38%) a significant proportion do not utilize such standards (62%) *[D24]*. 54% of respondents stated that they used security best practices in their software development *[D25]*.

Less than half of respondents consistently document their software (Always Document: SSH - 31%; SE - 38%; HR - 44%) *[D13]*. Given that good documentation is noted as one of the key factors that dictate whether a researcher uses a software package (SSH - 60%; SE - 57%; HR - 55%) *[E14]*, this is an important gap that needs to be addressed.

The practices for managing and sharing research software are primarily influenced by the expectations of the research community and collaborators, with significant impact observed in Social Sciences and Humanities (57%), Science and Engineering (61%), and Health Research (49%) *[D16]*. Funders' policies and institutional policies also influence how research software is managed and shared (funders policies: SSH - 30%; SE - 29%; HR - 33%) with publisher policies having a smaller influence (SSH - 9%; SE - 12%; HR - 19%).

When it comes to sharing research software, respondents commonly rely on open software repositories like GitHub, direct access to the software, and personal or research group websites *[D17]*. These platforms serve as crucial channels for disseminating research software among the academic community.

A considerable number of respondents (54%) stated that there was a research publication that described their research software and 49% included a publication reference of some kind, indicating the scholarly recognition and integration of their research software in their work *[D18]*. Fifty eight percent of respondents stated that they were not aware of the FAIR4RS and other related principles for research software development *[D19],* with 63% of respondents stating that they have never published a release for their research software *[D20]*. 52% of respondents stated that they are able to measure the impact of their software using other measures (not including paper or research software publication citations as described in D18 and D20), with the primary measures being number of users and number of downloads *[D21]*.

# Participating in Research Software Development

The majority of respondents (75%) have received training in using or developing research software *[F1]* but most of them are self-taught (SSH - 84%; SE - 85%; HR - 88%), relying on online resources like StackOverflow and Software Carpentry (SSH - 61%; SE - 72%; HR - 62%) *[F2]*. Additionally, some respondents mentioned engaging in peer-led training or extracurricular courses as alternative means of acquiring software-related knowledge and skills.

Most respondents value training in research software development (72%) *[F4]*, but they are less interested (45%) *[G13]* in training related to software sustainability. This indicates a strong motivation to enhance software-related skills and stay current in the field but less motivation to develop skills to ensure that their software is maintained and usable over time. Most respondents (97%) are not part of a Research Software Engineering Association *[F5]*, but a notable percentage (37%), especially in SE (41%) and HR (38%) expressed interest in joining such communities or programs. In contrast, only 24% of respondents in the Social Sciences and Humanities expressed interest *[F6]*. Those seeking to join a Research Software Engineering Association are primarily interested as a way to learn best practices in research software development, receive training and education, and engage in collaboration and networking activities within the research software engineering community *[F7]*.

## Supporting the Development of Research Software

When considering software development support, 76% of respondents emphasized the importance (very important = 49%; Somewhat important = 27%) of having access to such a service for their research *[C6]*. Despite its importance, 57% of respondents reported not having access to such support (42%) or uncertainty (15%) regarding the availability of it *[C4]*.

For those who did have access (43% of respondents), it is the local institution that primarily provides it. Over 90% of respondents from the SSH, 53% from the SE, and 92% from HR identified institutional support as their primary source of assistance (Figure 16) *[C5]*. This finding highlights the crucial role that institutions play in facilitating the development of research software by offering the necessary resources and expertise to researchers. The Alliance (SSH - 26%; SE - 28%; HR - 24%) and disciplinary communities (SSH - 17%; SE - 38%; HR - 20%) were also identified as providers of software development support.

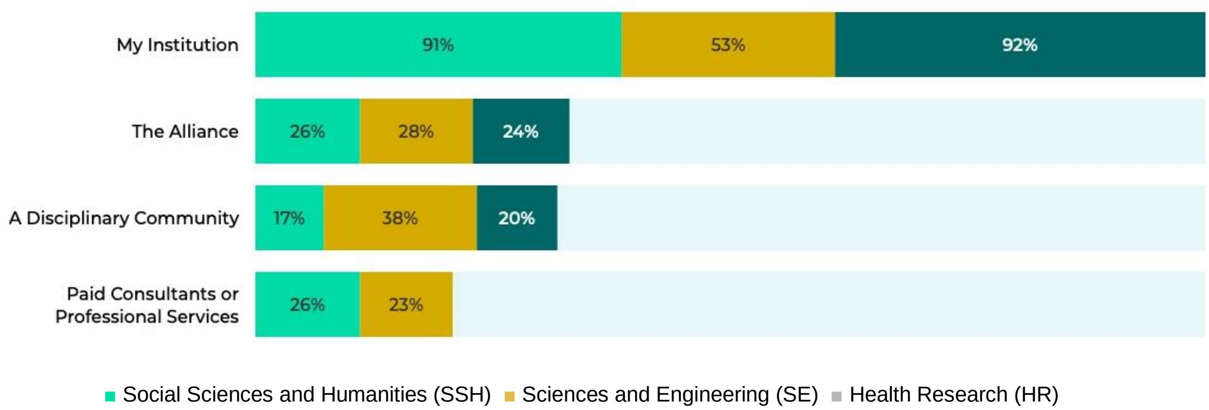
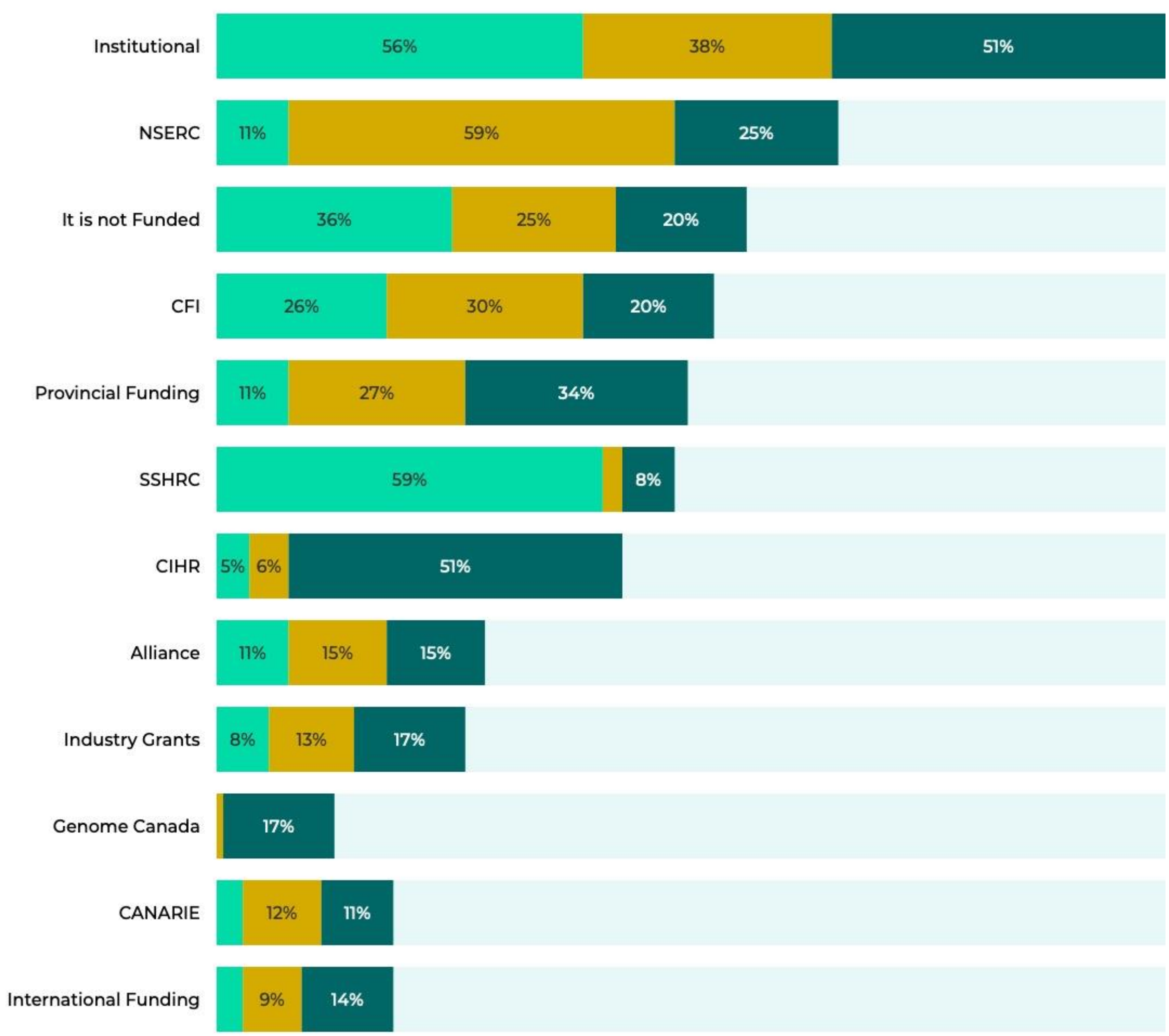



**Figure 16. Question [C5]: Who provides this support?** Responses measured in percentage across TC domains. Responses measured in percentage across Tri-Agency's domains. Social Sciences and Humanities, n = 23; Health Research, n = 25; Sciences and Engineering, n = 53; Unique respondents = 101.

Forty eight percent of respondents stated that there were software platforms or services that would be valuable if supported as a national service *[C11]* and 31% of respondents expressed that they believe the platforms or software services they are currently developing or co-developing would be valuable as national services *[C13]*.

# Funding the Development of Research Software

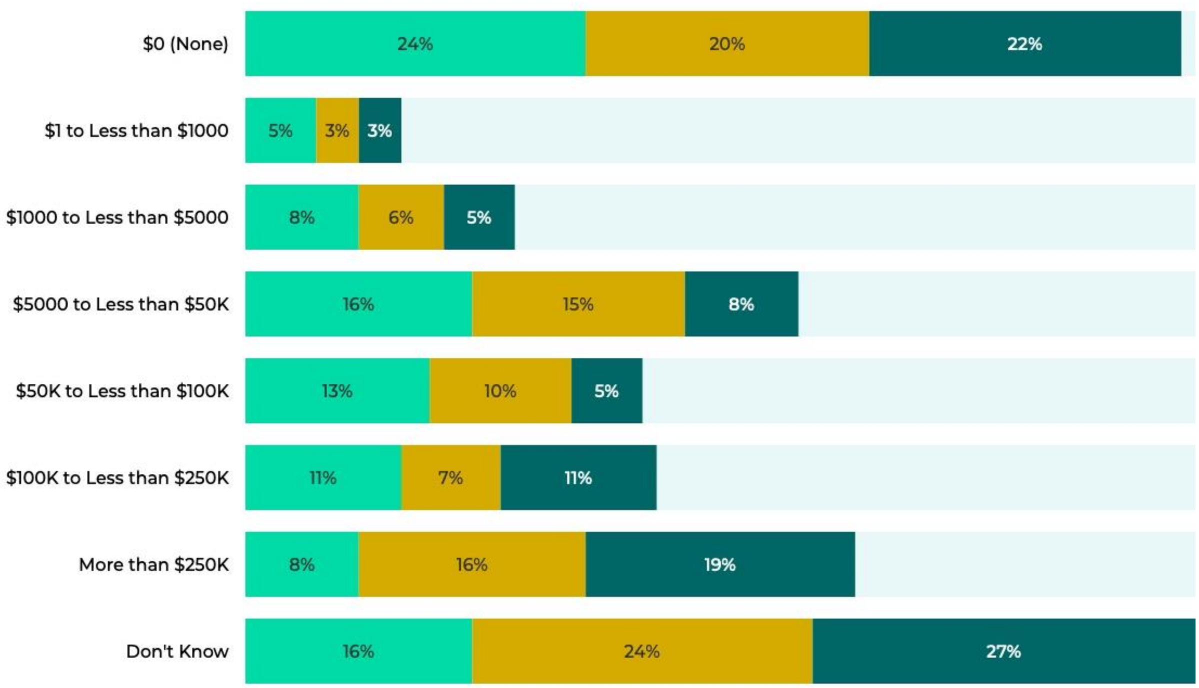
Only 26% of the respondents reported receiving funding from dedicated research software development funding calls, with the majority of respondents (63%) using non-specific funding for their research software development activities *[D2]*. Among those who have received dedicated research software funding and specified the source for that funding, the majority of respondents stated they received funding from CANARIE and the Canada Foundation for Innovation (CFI) *[D3]*. Non-specific funding for research software (spanning both development and use) comes from a variety of sources [*B8*; Figure 17], with institutional being the most important across domains. An important proportion of respondents identified that they do not receive funding for their research software development, and this was most common in SSH with 36%. Tri-Council funding was the most prominent source of funding for respondents within each Council (i.e., SSH - SSHRC; SE - NSERC; HR - CIHR), although some cross funding was also present. Both CFI and provincial governments also appear as prominent research software funders (Figure 17).



■ Social Sciences and Humanities (SSH) ■ Sciences and Engineering (SE) ■ Health Research (HR)

**Figure 17. Question [B8]: How is your group's research software budget funded?** Responses measured in percentage across Tri-Agency's domains. Social Sciences and Humanities, n = 73; Health Research, n = 65; Sciences and Engineering, n = 136; Unique respondents = 274.

Respondents report spending substantial amounts of research funding annually on the development of research software *[D8]*, with 33% of respondents reporting that they spent over $50K/year on software development and 15% of respondents reporting that they spent over $250K/year on software development. 20% of respondents reported spending $0 on software development while 24% of respondents stated that they did not know how much money they spent per year on software development (Figure 18).




■ Social Sciences and Humanities (SSH) ■ Sciences and Engineering (SE) ■ Health Research (HR)

**Figure 18. Question [D8]: Approximately how many dollars (CDN) of research funds did your research group spend over the last calendar year on developing research software resources?** Responses measured in percentage across Tri-Agency's domains. Social Sciences and Humanities, n = 38; Health Research, n = 37; Sciences and Engineering, n = 89; Unique respondents = 164.

Funding was stated as the largest barrier to research software sustainability across all domains *[D30]*, with 78% of SE, 83% of HR, and 67% of SSH respondents stating funding is a critical issue. The finding and retaining of staff was the main other barrier to software sustainability *[D31]* (Figure 19).

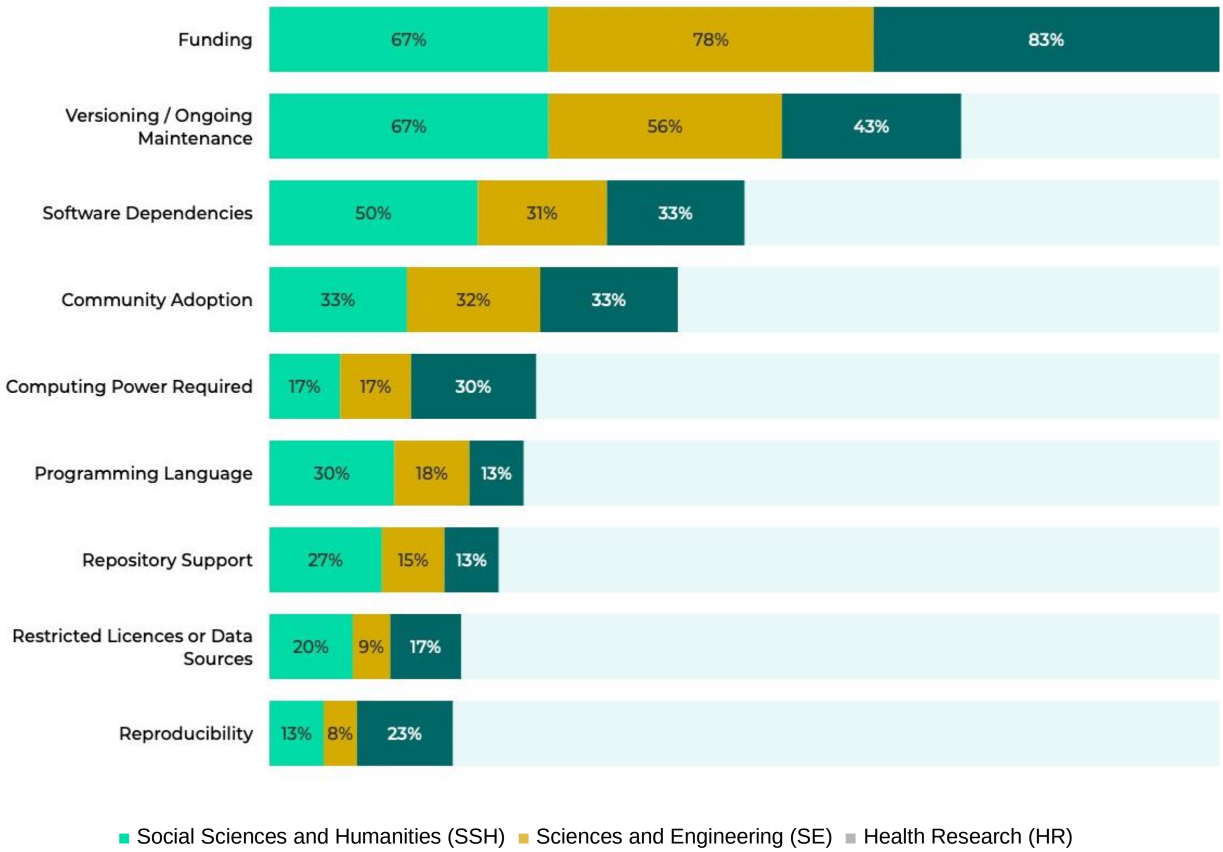



Figure 19. **Question [D30]: When you consider your own software, what do you see as the biggest barrier(s) to sustainability?** Responses measured in percentage across Tri-Agency's domains. Social Sciences and Humanities, n = 30; Health Research, n = 30; Sciences and Engineering, n = 78; Unique respondents = 138.

These findings highlight the limited availability of dedicated funding opportunities for research software development and the challenges faced by many researchers in securing financial support for their software projects.

# Research Software Sustainability

The majority of respondents highly value preserving research software (94%) *[G9]*, but they have differing opinions on who should be responsible for the long-term sustainability of research software. Some suggest that the community of users should be responsible, while others believe it falls on the principal investigator, lead developer, or institution *[G4]*. Respondents agreed that research software sustainability was important for facilitating reusability (SSH - 80%; SE - 79%; HR - 83%), facilitating reproducibility (SSH - 54%; SE - 74%; HR - 87%), supporting open science principles (SSH - 74%; SE - 63%; HR - 63%), and building community (SSH - 63%; SE - 47%; HR 57%) *[G5]*. Over 30% of respondents have benefited from access to preserved software *[G6]* but 25% also stated they were unsure, suggesting that there is a lack of knowledge as to whether software is preserved or not. Similarly, one third of respondents have developed software that is no longer being maintained, with the lack of maintenance primarily due to funding, versioning, and maintenance challenges *[D12]*.

Respondents further face many challenges when using other users' software. The primary obstacles are a lack of financial resources (funding), lack of expertise, and lack of time. Additionally, respondents emphasized the importance of having funds that align with their specific needs and receiving support in areas such as training and maintenance.

Most respondents are familiar with common software repositories for software preservation like GitHub or BitBucket (SSH - 96%; SE - 91%; HR - 87%), but are less familiar with generalist repositories like Zenodo (SSH - 26%; SE - 19%; HR - 35%) or Software Containers repositories like DockerHub (SSH - 26%; SE - 16%; HR - 26%) *[G10]*. Institutional software repositories and the Software Heritage Archive are not extensively utilized.

Regarding software publication and preservation, most respondents across disciplines see value in peer review of code, support in writing documentation, and consultations about software sustainability *[G12]*. However, the importance of these services varies between disciplines. For example, in-person consultation is highly important for Humanities and Social Sciences (70%), while it holds less significance in Sciences and Engineering (38%) and Health Research (22%). On the other hand, running compatibility tests is more important in Health Research (61%) and in Sciences and Engineering (52%) than in Humanities and Social Sciences (33%).

# Conclusion

Research software has become an essential tool of modern research across disciplines. Nonetheless, there are multiple cultural and technical factors that have limited the broader adoption of software engineering best practices (e.g., software development and documentation) and software sharing (i.e., FAIR4RS). The results presented in this survey are aligned with those found internationally, where 95% of respondents use research software and 50% believe that it is a primary output of their research.

The role of developing research software within different groups seems to be shared across different roles or types of respondent. Nonetheless, the most common research software developers were found to be graduate students, rather than professional software developers or Research Software Engineers. Yet, this survey was not necessarily targeted at this group. Indeed, there was very little representation from students (~7%), Postdoctoral researchers/Fellows, and Research Assistants (~7%). On the other hand, a large proportion of the respondents were Faculty – Professors, which allows for a strong understanding of how this group develops, uses, and funds their research software activities. Faculty in the Sciences and Engineering consistently spent more time and played more roles on research software development than those in Social Sciences and Humanities, and Health Research. While this may not be surprising, considering that research in such domains often requires heavy computation and code, it does highlight that Faculty in SE may also spend more time in supporting the research software they develop or use.

Institutions play a prominent role in research software support and funding. Indeed, most respondents rely on institutional software support and those that do not have access to such service, highlighted that this would be very beneficial to them. Similarly, institutional funding for research software was just as important a source of funding for respondents as their respective Tri-Council funding agency. As universities expand their catalogs of IT-vetted and secure software for research purposes, it is likely that they will continue playing an ever-larger role in supporting their Faculty and research communities.

While most respondents agree that preservation and sharing research software is very important, many struggle with the reusability of other people's software. Among the most important factors precluding reuse were lack of proper documentation and support. Yet, these are the same aspects that respondents who developed their own software were struggling with, and many were unaware of best practices for software development such as the FAIR4RS principles. Training and awareness seem to be the most important aspects for cultural change and broad adoption of best practices in software development.