Software
Sustainability
Institute

Report on the AHRC Digital/ Software Requirements Survey 2021:

Where is Investment Needed?

Shoaib Sufi
Software Sustainability Institute | University of Manchester

Emily Bell
University of Leeds | Software Sustainability Institute

Anna-Maria Sichani
School of Advanced Study, University of London | Software Sustainability Institute

February 2023

Authors:

Shoaib Sufi, Software Sustainability Institute | University of Manchester

Emily Bell, University of Leeds | Software Sustainability Institute

Anna-Maria Sichani, School of Advanced Study, University of London | Software Sustainability Institute

Acknowledgements:

We would like to thank the reviewers who contributed towards improving the survey: Adam Crymble (UCL), Nicola Osborne (University of Edinburgh), Lisa Otty (University of Edinburgh), Pip Willcox (The National Archives), Ross Wilson (University of Cambridge), Jane Winters (University of London), and Melodee H. Wood (Loughborough University).

Thanks is also due to those who commented on drafts of the report, as they substantially helped the authors improve the clarity of our methods and findings: James Baker (University of Southampton), Anita Banerji (University of Manchester), Neil Chue Hong (University of Edinburgh), Jez Cope (The British Library), Adam Crymble (UCL), Ginestra Ferraro (King's College London), Patricia Herterich (University of Edinburgh), Lisa Otty (University of Edinburgh), Catherine Smith (University of Birmingham), Rebecca Taylor (University of Southampton), Melissa Terras (University of Edinburgh), Johanna Walker (University of Southampton), and Melodee H. Wood (University of Loughborough).

The survey is based on earlier work led by Simon Hettrick (University of Southampton), and the idea and context was shaped by David De Roure (University of Oxford) and Neil Chue Hong (University of Edinburgh).

We would also like to acknowledge Jacalyn Laird (University of Edinburgh) for her help with the publicity of the survey.

This work has been supported by the Software Sustainability EPSRC, BBSRC, ESRC, NERC, AHRC, STFC and MRC grant EP/S021779/1. Anna-Maria's contribution has been also partially supported through a UKRI Policy and Engagement Fellowship in Digital Research and Innovation Infrastructure AH/W011220/1.

CONTENTS

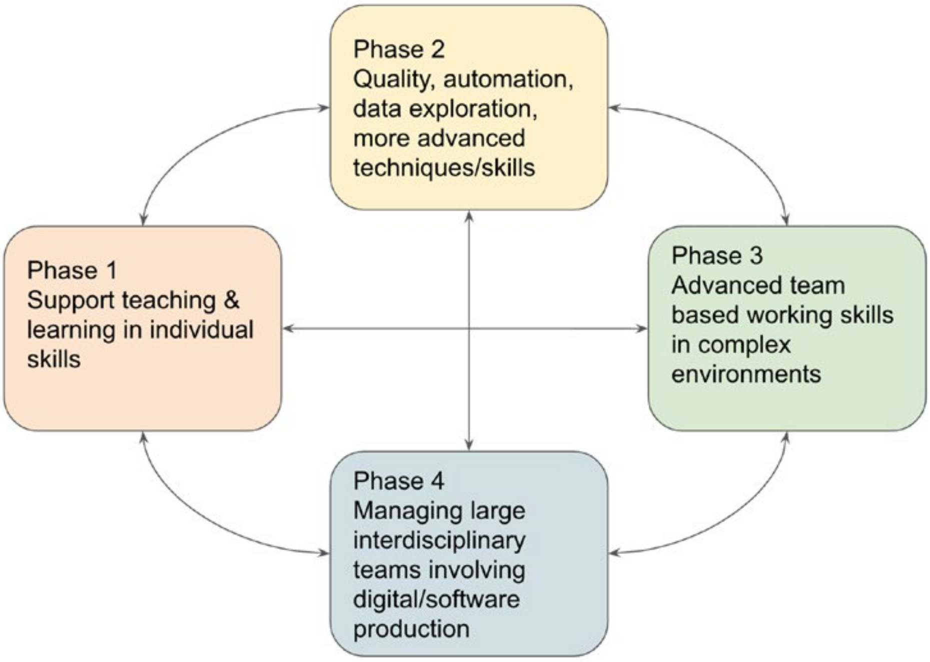
1 EXECUTIVE SUMMARY: KEY POINTS, ACTIONS AND CONCLUSIONS

1.1 SUMMARY

There is a clear demand and need to provide software training, recognition and resourcing for software skills and techniques in the arts and humanities. To meet this need funders should tailor calls to encourage the development of skills and training by and for those at different career stages and support the recognition of software and digital innovation as high value research outputs. Institutions should support software development and provide training in computational techniques and skills, incentivise attending training and support the development of training pathways in relevant techniques and skills. Communities of practice also have a key role in creating and raising awareness of learning opportunities and encouraging sustainable software practices via networking and collaboration.

1.2 KEY POINTS

1. **Knowledge transfer:** AHRC members at different career stages are knowledgeable in tools and practices in the digital/software space that are useful to others in the AHRC community; there should be incentives to create material and teach others to increase knowledge transfer and utilisation. This will avoid unnecessary rediscovery.


Phase 2
Quality, automation, data exploration, more advanced techniques/skills
Phase 1
Support teaching & learning in individual skills
Phase 3
Advanced team based working skills in complex environments
Phase 4
Managing large interdisciplinary teams involving digital/software production

Knowledge transfer between career stages and predominant areas of expertise

2. **Status:** Digital/software outputs should be seen as first class research outputs alongside publications, offering recognition and reward for those who contribute to their production.
3. **Collaboration:** Partnerships between AHRC community members – not only those with a DH focus – and Research Software Engineers should be fostered, to allow gauging of skills, filling of gaps, to increase confidence, and to relieve some of the development burden.
4. **Market intelligence:** There is a lot of information around what practices and tools different career stages use and would like to use next in this report (see section 5.3 Research Software Development and section 6 Results: Open-Text Questions); this can inform funders and community members on use and future directions that need supporting.

1.3 CONCLUSION AND RECOMMENDATIONS IN BRIEF

Detailed conclusions and recommendations are outlined in section 8 Recommendations & Future Steps. Here we give a summary of the key findings in that section.

These findings relate to actions and activities that are needed by three distinct stakeholders in the AHRC community: funders, institutions and communities of practice.

1.3.1 Funders should:

- Tailor funding calls to encourage the development of skills at particular career stages by encouraging collaboration across career stages and cognate disciplines.
- Support the recognition and prestige of software and digital innovation as high-value outputs by:
  - Funding sustainability, training and innovation;
  - Championing such outputs in the Research Excellence Framework (REF[1]).
- Support the Digital Humanities to share their knowledge in digital methods and data with the wider humanities communities and with the Research Software Engineering community.
- Work with other funders (e.g. ESRC, EPSRC, the British Academy and Leverhulme Trust) to adapt or build training for AHRC use cases and needs.
- Support the development of training pathways in techniques and production skills that have been highlighted by the survey:
  - High use techniques: data cleaning, text/data mining and visualisation;
  - Advanced techniques: probabilistic linking, natural language processing and machine learning;
  - High use production skills (e.g. version control);
  - Advanced production skills (e.g. how to work with developers).
- Mirror the 'Innovation Scholars' programme[2] run by the BBSRC, ESRC and MRC, which would be an excellent model for AHRC to learn from. This is another opportunity for the AHRC to partner with other funders, and it would allow the more efficient transmission of knowledge and practice between later and earlier career phases.
- Remove the 'hidden scholarship' element of the development of software by encouraging and legitimising the use of funds to develop and maintain software.

65%

learnt their software engineering on the job or in their own time

Only

35%

rated their software skills at the highest level

Only

29%

of those who developed software felt that they had sufficient training to develop reliable software

56%

worked in groups which hired software developers

[1] Research Excellence Framework 2021

[2] Innovation Scholars: Data Science Training in Health and Bioscience - while this report was being prepared the AHRC put out a call, Embed digital skills in arts and humanities, this is an excellent step forward.

1.3.2 Institutions:

- Need to play their part in encouraging software/digital tools to be viewed as first class research outputs.
- Need to support software development in the research and creative space by providing in-house research software engineering, with AHRC relevant expertise, which can be bought out in research grants.
- Need to provision training in techniques and skills in computational aspects of relevance to humanistic research questions, and provide a programme of activities from beginner to more advanced.
- Need to provide computational infrastructure to support the deployment of digital tools by researchers in the AHRC space.
- Need to build incentive (e.g. buying out time) structures to encourage those at various career stages to build and deliver training to those in other career stages, to support the transfer of knowledge around technical skills, collaborative working, scaling and sustainability.
- Need to incentivise attending training (e.g. count it towards Continuing Professional Development and annual development reviews).

1.3.3 Communities of Practice:

- Need to establish learning opportunities to allow the sharing of knowledge between different career stages.
- Should help seed networking and collaboration to allow the creation of more robust digitally focused bids.
- Should encourage sustainable software practices, become the official supporters/hosts of particular software projects if applicable, and maintain directories of relevant projects to facilitate reuse.
- Should encourage applications to software/digital focused Fellowships (such as the SSI Fellowship Programme[3]) to encourage interdisciplinary networking, discovery, and following and sharing best practices.

88%

said institutional research engineering provision was a perfect or suitable resourcing solution

Over

40%

felt their institutional support for software development was acceptable or poor - only 10% thought it was excellent

38%

of respondents used or were moving to HPC / Cloud computing

[3] SSI Fellowship Programme.

2 MOTIVATION

The Software Sustainability Institute (SSI) (Crouch et al., 2013) champions and supports the use of digital tools and computer-aided methodologies (AKA software) in the research process.

Discussions for this study started in the SSI between Prof David De Roure, Mr Neil Chue Hong and Dr Simon Hettrick in Summer of 2020 concerning gaining a better understanding of the AHRC community's digital attitudes, and requirements around software, learning and resourcing. This was driven by work in the Digital Humanities (DH), and designed in light of the reshaping of digital infrastructure funding that was being envisaged by the AHRC.

The overall aim of this investigation was to ensure that any future calls could support the software needs of the community. There was a recognition that the AHRC community is much more diverse and fragmented than those of other funders. There was also recognition that alongside the Digital Humanities there are other humanities subjects that use software, and additionally the creative industries, digital art and the GLAM (galleries, libraries, archives and museums) sectors. We were therefore keen to get views from those in the humanities who did not see themselves as users/developers of software.

To take this investigation forward, it was decided that a survey would be produced and used to capture views on digital/software tools, experience of development of such tools, practices, learning intentions and preferences around how projects involving digital/software should be resourced.

The final sample taken in the survey was skewed towards those practising the Digital Humanities and it is not representative of the AHRC community as a whole. It is precisely this fact that makes the findings useful; DH are the 'canaries in the coal mine' and their practices today are the practices of the wider AHRC community tomorrow.

3 REVIEW OF RESEARCH ON DIGITAL/ SOFTWARE LITERACY AND USE IN THE ARTS AND HUMANITIES

There have been few studies directly focused on the software needs of the Arts and Humanities community, although several scholars have identified similar concerns in other disciplines and domains, and signposted training and support providers which are also relevant to humanities researchers. In a paper focused on software in the social sciences, Duca and Metzler (2019) reviewed 418 tools, finding that close to 50% were developed in the US, suggesting a greater support from funding bodies and more developed collaborative Digital Humanities networks. More broadly, they noted the more innovative tools came out of academia, but that while 85% of respondents use software (and this is a vast increase since the 2005 Summit on Digital Tools (*Summit on Digital Tools for the Humanities: Report on Summit Accomplishments*, 2006) at the University of Virginia found only 6% of humanities scholars went 'beyond general purpose information technology and use[d] digital resources and more complex digital tools in their scholarship'), only 10% have developed their own software. Significantly, they also noted only 10% of the key people developing tools were women. 13% of the tools surveyed were developed by individuals, highlighting a focus on collaboration; these individually-developed tools had a reduced chance of ongoing maintenance and funding. 30% of the tools surveyed were developed by teams within academic institutions or public organisations; these were usually funded by grants. Their work shows how vitally important it is for funding bodies to provide adequate support not only for the development of new digital projects and software, but for training and sustainability; their study indicated that approximately 50% of the tools surveyed are free to use, meaning almost half are not.

Duca and Metzler identified major providers of training including the Software Sustainability Institute itself, the discipline-specific Digital Methods Initiative, NUMFocus and Pelagios Commons, as well as several national-level centres offering training. Münster and Terras (2020), in a study on visual digital humanities, noted that several respondents entered the Digital Humanities (DH) as software engineers employed for specific projects, and had thus learned the appropriate skills through studying complementary courses (such as a Masters course in Advanced Visualisation and Analysis). Their work also highlighted the importance of self-driven learning (i.e. online tutorials and other resources), and learning by experience. Crymble (2021) has outlined in detail the 'invisible college', an informal network of support that fosters this self-directed learning. A focus on interdisciplinarity, rather than siloed disciplines, was emphasised by Münster and Terras (2020), and they found that researchers in this area come from a wide range of academic backgrounds. These were generally from 'technical disciplines' or humanities, but with a strong interdisciplinary ethos.

In terms of the specific software or areas of software mentioned, Schindler et al. (2022) found increasing mentions of 'Plug-Ins' and a decreasing focus on 'Applications' in their survey of scientific software mentions. The most regularly mentioned software included SPSS, R, Prism, ImageJ, Windows, Stata, Excel, SAS, BLAST and MATLAB, the authors noting a specific interest in R. Duca and Metzler (2019), in a paper focused on software in the social sciences, identified a wide variety of tools and packages used by researchers, with an emphasis on social media, data visualisation, annotation, transcription, text mining, qualitative data, tools for recruiting participants, citizen science, and survey software. However, in an article focused on Digital Humanities more broadly, Gibbs and Owens (2012) found that few researchers interviewed were interested in specific tools for analysing digital archives, beyond 'a few mentions of Zotero (11) and Endnote'. They found 'there was no mention of third-party tools, or of methodologies involving text or data analysis and visualization'. Münster and Terras (2020) noted that data acquisition and management were the most prominent research areas for those working with visual DH. These various studies show that use of software stretches from project management to quantitative and qualitative analysis; the more recent of these studies also show a growing interest in software, though less so in humanities than other areas. More recently, attempts to map the field of code literacy and software needs and skills for Humanities scholars have been initiated at a national (McGillivray et al., 2020) and international level.[4] Institutions such as The Turing Institute (Humanities and Data Science Working Group)[5] and communities of practice such as the Computational Humanities Research Network[6] have been actively working in this direction. However, it is difficult to draw conclusions from studies taking place nearly a decade apart, given that the digital tools and how researchers use them are constantly evolving. There are several journals and digital resources which are focused on disseminating information about new tools, such as the *Journal of Interactive Technology and Pedagogy*[7] and *Programming Historian*[8].

[4] Elli Bleeker, Marijn Koolen, Kaspar Beelen, Liliana Melgar, Joris van Zundert, and Sally Chambers, 'A Game of Persistence, Self-doubt, and Curiosity: Surveying Code Literacy in Digital Humanities' , Journal of DH Benelux 2022, (forthcoming).

[5] The Alan Turing Institute.

[6] Computational Humanities Research discourse.

[7] The Journal of Interactive Technology and Pedagogy

[8] The Programming Historian.

Barriers to software use in the existing literature include lack of training (Giles Bergel et al., 2020) and awareness of existing tools, lack of funding, copyright restrictions, but also sustainability issues, particularly around projects developed by individuals. A knock-on effect of the lack of academic recognition of software development, particularly in the humanities, is issues of sustainability, and even of getting resources to be used, in light of quality concerns around robust peer review processes and transparency about corpora and code. The 2021 REF saw a decline in 'digital' items, suggesting academics may be right to distrust that their colleagues will value such work in REF terms; although this may be an artefact of a higher amount of returns per institution on average. Barriers to software use are therefore also barriers to software creation among humanities researchers.

Several studies have shown that the development of software is not adequately recognised as academic labour, in the humanities and more broadly (Warwick *et al.*, 2006; Schreibman and Hanlon, 2010; Anderson *et al.*, 2016). Another aspect of this discussion concerns the value and the credit to be assigned to tools and tooling. In the first issue of the *Debates in the Digital Humanities* series (2012), Ramsey and Rockwell, developers in Digital Humanities themselves, argued that, despite the widespread anxieties about credit for digital work and an undeniable resistance in certain parts of academia caused by the 'fear of an automated scholarship', 'it would still fall to the builders to present their own activities as capable of providing affordances as rich and provocative as that of writing'. Thus Schindler et al. (2022), in a study of mentions of software in scientific research, note that references are 'usually informal and often incomplete', missing information about developers and versions in particular. This pattern, also evident in humanities research, makes it difficult to trace digital tool use in published humanities research, but again raises issues with funding and training. Similar questions could be asked about those outside of academia building research tools (i.e. independent scholars), and the reliance on goodwill and good digital citizenship over funding and infrastructure that animates much of DH research.

4 METHODOLOGY

4.1 OUR APPROACH

We used a quantitative approach, designing and using one survey instrument which had both fixed choice and open ended questions. We conducted a mixture of univariate and multivariate analysis on the fixed choice questions, producing descriptive statistics with commentary. We also took a quantitative approach to open ended questions to identify and quantify occurrences of themes, and provided commentary on this.

4.2 THE SURVEY

The Software Sustainability Institute (Crouch et al., 2013) ran the digital/software requirements survey of the Arts and Humanities Research Council (AHRC) research community between 1 June and 5 August 2021. We received 118 eligible responses. The survey[9] was undertaken to improve our understanding of the digital practices people undertake, or wish they could undertake, in the arts, humanities and GLAM (galleries, libraries, archives and libraries) communities.

The survey asked about views on digital tools/software, experiences of developing digital tools/software, and practices and preferences for recruiting help with digital tool/software development. It focused on the Arts and Humanities Research Council (AHRC) Research Community, particularly those individuals who are connected to a UK-based institution. The aim is to share these findings with the AHRC to provide a source of community evidence to inform their digital infrastructure funding, and to inform the practices and approach of the Institute and the wider community participating in AHRC area digital/software projects.

We aimed for inclusiveness, actively seeking the views of people who do not develop digital tools/software, and sought input from all roles, including senior decision makers, early career researchers, curators, librarians and software developers in the AHRC remit.

4.3 SURVEY DURATION

The survey was open to the public from 1 June 2021 until 5 August 2021. The analysis of the findings took place from September 2021-May 2022. This was followed by two rounds of draft reviews and associated updates which took place between May 2022 - November 2022.

4.4 NOTE ON ETHICS

This study started out as a market analysis rather than a specific research piece, hence ethical approval was not sought. We do not envisage publication in an academic journal for the results mentioned here, and our aim is a report published in the name of the SSI. When we gathered the data for this study, we did so under the Software Sustainability Institute's Survey's privacy policy,[10] the intent being to make an anonymised version of the data collected available. This entailed removing any personal identifiable information (PII, as defined by the General Data Protection Regulation) such as name and email address, as well as information which could lead to deductive disclosure, such as job titles and institutions.

If we were to run such a survey again we would seek specific ethical approval for the study, as we feel some of the results and commentary around them would enhance the research literature.

4.5 DISTRIBUTION

The survey was sent to different categories of the AHRC community, targeting mainly those using digital/software tools, but also explicitly targeting projects (see 'AHRC investments' in the table below) which had no mention of digital/software tool use. These were contacted on a fortnightly basis from 3 June 2021 to 26 July 2021; there were thus five rounds of active promotion. In cases where an individual contact was actively engaging with us through dialogue via email, we did not keep sending automated reminders.

We asked individuals to fill in the survey themselves, and also promote it to their networks, letting us know where else they felt it should be promoted.

[9] Survey of digital/software requirements for the AHRC research community

[10] Privacy Policy - Surveys, The UK Software Sustainability Institute.

Categories	Notes
Reviewers	These were the individuals (listed in the acknowledgements) who helped us shape the survey
Critical Friends	These were a wider set of individuals who we knew worked in the area and had previous interactions with the SSI
Networks	These were internal and institutional networks relating to the SSI and where the SSI is based
AHRC advisory board	We received permission to contact AHRC advisory board members as our survey was connected to an AHRC funded survey on Software and Data loss[11].
Mailing lists	These were relevant AHRC community-oriented mailing lists (this were mainly suggested by those in the Friends category); examples include the Museums Computer Group, GLAM Labs community list and the Humanist group
Funder	This included the AHRC itself, and a large cluster it had funded (the Creative Industries Cluster – CIC)
AHRC investments	Using Gateway to Research,[12] a sample was taken of current AHRC investments with a skew towards larger investments and those with a clear computational aspect, although there was representation of medium and smaller projects and those without a clear computational aspect. We chose 62 projects from the list of approx. 750 current active projects. A sampling approach was taken, due to the resource limitations available to us with respect to running/promoting the survey

One of the problems encountered was that the survey was set up for individual response. One organisation asked for the questions in document form as they wanted to provide an organisational response, however they did not in the end offer a response. This may have been due to the very individual nature of some of the questions.

4.6 DEFINITION OF (RESEARCH) SOFTWARE

This is the definition we used for research software in the survey; it was refined from the definition used in the 'Survey of software use at the University of Southampton' (Brown et al., 2020) and co-developed with those mentioned in the acknowledgements sections to allow it to be relevant and understood by the AHRC community. It is relevant to how we analysed the data that was supplied (e.g. in terms of what we counted as software):

'Research software' is any software or digital tool that you have used in the course of your research that has helped you produce a research output (e.g. a publication). This might be anything from a short script, such as one written in Python or R, to help you clean your data, web/mobile apps, to a fully-fledged software suite or specialised toolset, whether you access this online or run it on your own computer. It includes code that you have written yourself and code written by someone else, either specifically for your project or a general tool for data, text or statistical analysis. It does not include common software applications used to prepare research publications, such as word processors (e.g. Word, Pages, LibreOffice). It also does not include online databases (e.g. Literature Online and Eighteenth-Century Collections Online). However, the use and/or construction of spreadsheets that perform calculations or transformation automatically according to a set of pre-programmed rules, are considered to be software.

[11] Shaping data and software policy in the arts and humanities research community.

[12] UKRI Gateway to Research Portal.

4.7 QUESTIONS: BASIS, REVIEW, MODIFICATIONS, AND FINAL VERSIONS

The final list of questions, the fixed choice answers, hint text, sections and applicability are available in section 10 Appendix A: Survey. Here we discuss how the questions were reviewed and adapted with input from the reviewers. We then list the questions in brief, detail whether they are fixed-choice or open ended, whether they are optional or mandatory and any applicability criteria.

4.7.1 Discussion on formulation of the questions

The final survey questions constitute the third and final version. The first version was adapted from the 'Survey of software use at the University of Southampton' (Brown et al., 2020). This was sent to a small set of reviewers. We based the survey on this as we essentially wanted to know similar things about the AHRC community.

The second draft of the survey questions was sent to a much broader set of reviewers and most of those credited in the acknowledgements section commented on them. Here are some of the key changes we incorporated:

- The use of digital and digital tools in addition to saying software, as the former terminology is more well known.
- Institutional affiliations were added with the caveat (if applicable), due to the precarity of those who contribute to digital matters in the AHRC space; we also changed the use of 'job title' to 'job role' to reflect the non-official roles people play in the space.
- We added 'Digital Humanities' to the list of disciplines as (rather surprisingly) it was not on the list on the AHRC website.
- We asked which funder people had applied to rather than received funds from, as it was seen as a sensitive issue. The change was more indicative of who a funder's natural community was, and hence more inclusive.
- The definition of software received extensive notes to improve the language, but also to couch the definition in terms that those in AHRC would find recognisable; again, using the term 'digital' as well as 'software', and better explaining what type of spreadsheet use we classed as software.
- We had combined computing tools (e.g. programs), techniques (e.g. types of analysis) and infrastructure (e.g. types of hardware) into one question initially; this was split into three questions on production skills, computing technologies and scale.
- We added a question on types of data being processed, and also incorporated suggestions of techniques from the more common to the more specialist (e.g. 'Data Cleaning', 'Database Design', 'Network analysis','Textual analysis' to 'Working with Geospatial data' and 'Immersive technologies' such as virtual/augmented reality') which were known to be used in the space but missing from the original list provided. We added further options and re-wording here. The list of computing technologies and skills can seem arbitrary, but they were a mix of those the SSI knew in the space, and also additions and modifications added by the reviewers.
- It was suggested we ask about where people learn their skills, as there was an assumption that it was within institutions, but one of the reviewers research pointed towards it being much more informal and part of the 'invisible college'.
- Given the AHRC's broad remit, we mention the GLAM acronym (galleries, libraries, archives and museums) over and above arts and humanities only; in hindsight we should have mentioned the creative sector, as we later realised they were unintentionally excluded by the language we used as they did not see themselves as doing 'research' even though they were funded by AHRC.
- Identification of additional funders that are relevant to the AHRC community.

4.7.2 Survey questions in brief

We start at 'Section 2', as 'Section 1' was the preamble for the survey. We also leave out 'Section 6' which was pertaining to follow up contact (which was collected to aid with a study which followed on chronologically from this investigation, on software and data loss in the AHRC community) and information about a prize draw for those who had participated.

4.7.2.1 Section 2: About you

Question	Type	Mandatory	Applicability
Email	Open	Yes	All
I confirm that I am 18 or over and I am a researcher who self identifies as part of the broad AHRC research community	Open	Yes	All
Institutional Affiliation	Open	Yes	If applicable (not having one was not a barrier to answering the survey)
Research Discipline	Fixed choice	Yes	All
Which of the following organisations have you sought funds for your research from?	Fixed choice	No	All
What is your job/role title?	Open	No	All
What is your career stage?	Fixed choice	No	All

4.7.2.2 Section 3: Research software and you

Question	Type	Mandatory	Applicability
Do you use research software?	Fixed choice	Yes	All
How important is research software to your work	Fixed choice	Yes	All
What do you use research software for?	Open	No	All
Why do you use the research software that you do?	Open	No	All
How could better software practices and solutions improve your area or field of work?	Open	No	All
Have you developed your own research software?	Fixed choice	No	All
What research software have you developed?	Open	No	Intended for those who answered 'Yes' to the previous question

4.7.2.3 Section 4: Your software development

Note: these questions only applied to those who had answered 'Yes' to the question in Section 3, 'Have you developed your own research software?'. It is for this reason that we have removed the 'Applicability' column in this table. Of the respondents, 59 (out of 118 responses) developed their own research software.

Question	Type	Mandatory
How do you rate your software development expertise?	Fixed choice	No
Where have you learnt your software skills	Fixed choice	No
Do you feel that you have received sufficient training to develop reliable software?	Fixed choice	Yes
What types of information/data do you or your team work with?	Fixed choice	No
What computing techniques are you interested in exploring for your research?	Fixed choice	No
What computing production skills are you interested in exploring for your research?	Fixed choice	No
Are there any other computing techniques or production skills that you are interested in exploring for your research?	Open	No
Which statement below best matches the scale of your computing needs?	Fixed choice	No
How would you rate the current level of support for your software-development needs from your institution?	Fixed choice	No

In hindsight there was no reason to restrict the question 'What types of information/data do you or your team work with?' to those who developed software, as users of software would also be dealing with these data types.

4.7.2.4 Section 5: Recruitment of software developers

Question	Type	Mandatory	Applicability
Have you or someone in your group ever hired someone specifically to develop software?	Fixed choice	Yes	All
Have you ever included costs for software development in a funding proposal?	Fixed choice	Yes	All
How suitable would the following models be for your software development needs?	Fixed Choice	No	All

4.8 HOW WE ANALYSED THE THREE MAIN OPEN QUESTIONS

The three main open questions asked in the survey were:

1. What do you use research software for?
2. Why do you use the research software that you do?
3. How could better software practices and solutions improve your area or field of work?

They were asked in the section 4.7.2.2 Section 3: Research software and you and results can be found in section 6 Results: Open-Text Questions.

The use of the terms 'software' or 'research software' was in relation to the definition in section 4.6 Definition of (Research) Software.

Responses to these question were tagged using the computer-assisted qualitative data analysis software (CAQDAS) tool Taguette[13] (Rampin and Rampin, 2021). Sets of responses based on career stage for each of the questions were tagged using an inductive reading of the responses as opposed to using a predefined model. We unified the responses based on words and phrases within the responses rather than the whole response, due to the similar categories and thematic areas covered in many of the responses, using our judgement and experience in the field of software/digital tools to guide us. Using this unitisation strategy means that the total occurrences of themes/categories was larger than the number of responses, as a single response could contribute to multiple themes.

Some words and phrases were tagged with multiple categories and themes, and we described differences and their potential causes between the different career stages.

4.9 OUR SAMPLE AND ITS IMPLICATIONS FOR ANALYSIS

Details about respondents are covered in section 5.1 Respondents' Profile. Here we highlight some of the information about responses and comment on the way in which this impacts our commentary on the data collected.

We received 118 valid responses. 40% of the responses were from those self reporting as part of the Digital Humanities; this is an over representation compared to the AHRC community, although responses from a broad range of over 50 distinct areas were recorded.

In terms of career stage, 70% of those responding were from career Phases 2 (early) and Phase 3 (mid/recognised). This reflects those who were targeted by the survey (i.e. with a focus on those in established networks and funded projects).

Universities with DH centres or labs and/or an established DH programme of study (BA or MA), such as UCL, University of Edinburgh, University of Oxford, University of Sussex, King's College London, University of Cambridge, among others, as well as institutions with a clear digital scholarship focus (The British Library, The National Archives); these 8 (out of 46) institutions represented 41% of the responses.

The skew of respondents towards those in the Digital Humanities space, mid to late career and at established universities with specific Digital Humanities units is reflected in our commentary; this data is not necessarily reflective of the wider AHRC community. However, it does cover those on the leading edge of digital/software matters in the AHRC space, and therefore is a useful sample for foreshadowing future needs in techniques and skills training, infrastructure needs and resource models of software/digital effort.

[13] The TAGUETTE open-source qualitative research tool.

5 RESULTS: FIXED CHOICE QUESTIONS

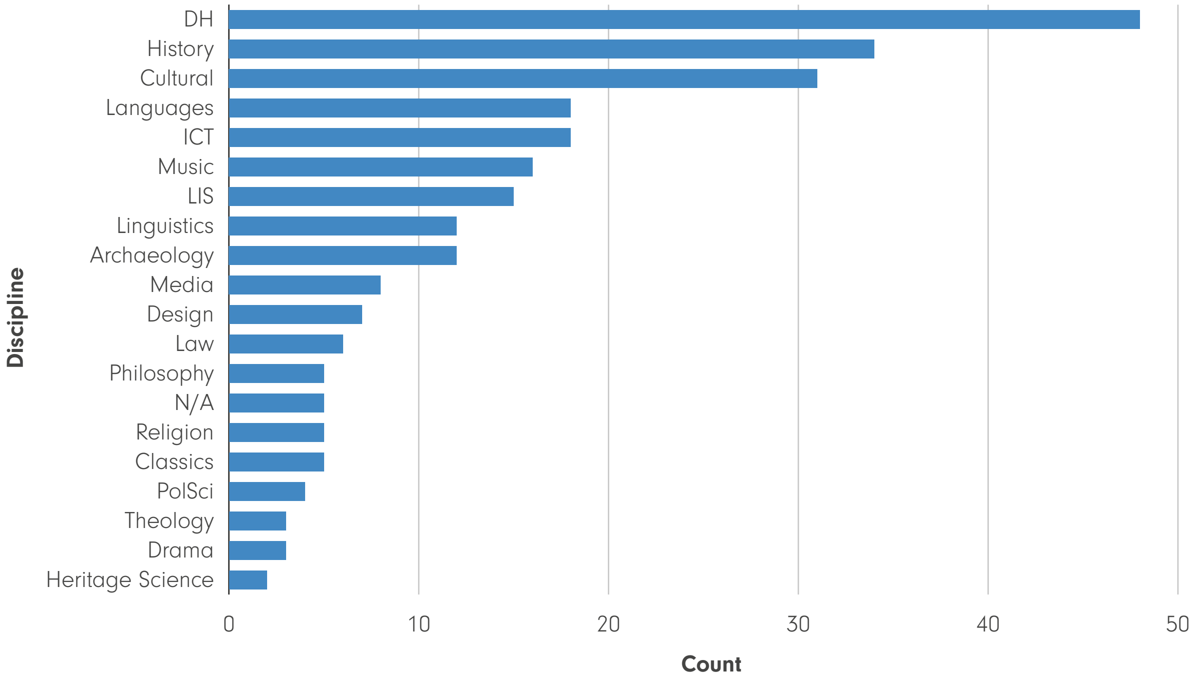
5.1 RESPONDENTS' PROFILE

5.1.1 Research Discipline

This question maps respondents' research discipline by using level one codes from the Primary Research Areas covered by AHRC discipline funding remit.[14] Respondents were asked to use the 'Other' option if their discipline is not listed, and could tick multiple options. We added the Digital Humanities option based on feedback from reviewers of the survey.

Discipline	Count
Digital Humanities (DH)	48
History	34
Cultural and Museum Studies (Cultural)	31
Languages and Literature (Languages)	18
Information and Communication Technologies (ICT)	18
Music and Visual Arts (Music)	16
Library and Information Studies (LIS)	15
Linguistics	12
Archaeology	12
Media	8
Design	7
Law and Legal Studies (Law)	6
Philosophy	5
N/A — I do not do research	5
Divinity and Religion (Religion)	5
Classics	5
Political Science and International Studies (PolSci)	4
Theology	3
Drama and Theatre Studies (Drama)	3
Heritage Science	2

[14] AHRC Disciplines (29th Sept 2020).



A Primary Research Area covered by the AHRC discipline funding remit that did not get any responses was Development Studies.

These were the responses to the 'Other' option. All responses here only received one choice:

Anthropology, Art History, Collections Management, Computer science, Creative Industries, Creative R&D, critical data studies, critical policy studies, Dance, Design Informatics – part design/part computing; Creative industries, Digital Documentation, education, ethnomethodology, Genealogy; company histories, Health and environment, Heritage, Heritage conservation, Heritage science, History of Art/Architecture/Visual & Material Culture, Humanities-led critical sociology, JISC, Psychology, Public Policy, Social anthropology, Sociology (I'm sort of at the ESRC/AHRC bridge, sociology of the future, though I'm currently AHRC-funded), User experience design and audience research.

Comparing these answers to the list of ESRC areas in a similar survey run by the SSI,[15] we can see that education, sociology, psychology and social anthropology are areas of overlap between AHRC and ESRC with regards to their research communities, showing great intersection between disciplines.

The survey asked individuals to 'tick all disciplines' that applied, leading to 287 individual disciplines chosen for the 118 respondents. This highlights a strong degree of interdisciplinarity, in keeping with the existing literature's focus on the importance of collaborative and interdisciplinary working in DH research.

We detail some career stage personas in section 5.1.4 Personas below.

5.1.2 Career Stage

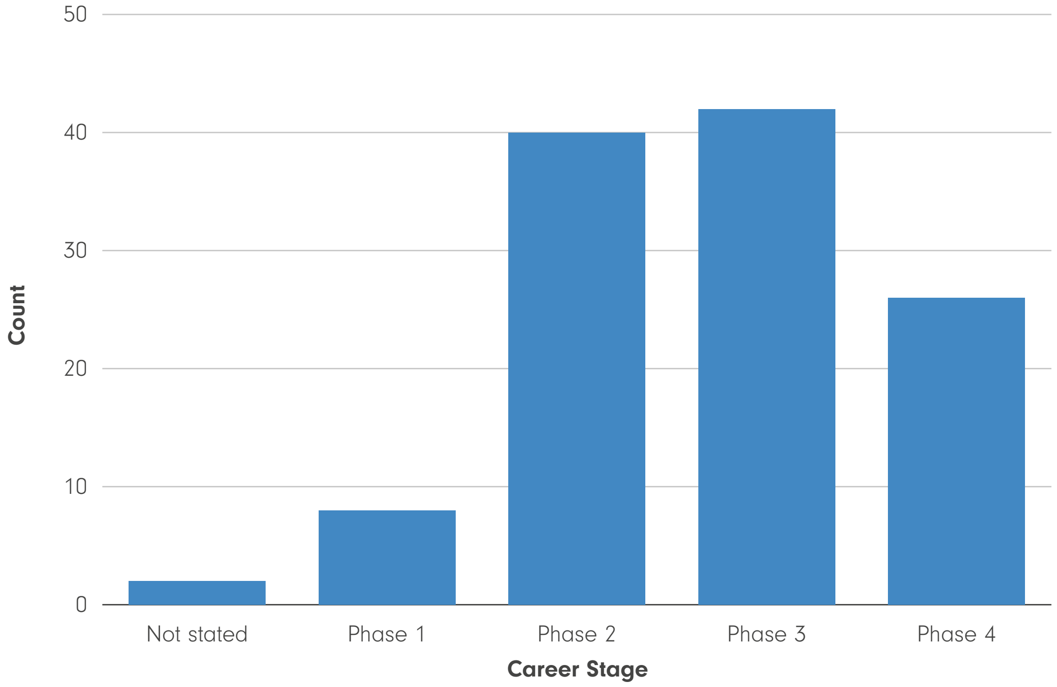
In the table below, we look at the number of respondents per career stage.

The career phases are based on the League of European Universities (LERU) definitions[16]. Modifications were made to include equivalent Research Software Engineering and research related posts. The higher the number, the more senior the post. Participants were asked to look at the definitions and choose the one which matched their situation most closely.

[15] A survey of digital methods and [sic] in the economics and social sciences research areas (Q19).

[16] Possible Research Career Paths in the United Kingdom (ENGLAND).

Career Stage	LERU Description	Count
Not stated		2
Phase 1	Junior (e.g. PhD candidate, Junior Research Software Engineer)	8
Phase 2	Early (e.g. Research Assistant/Associate, first grant holder, Lecturer, Research Software Engineer)	40
Phase 3	Mid/Recognised (e.g. Senior Lecturer, Reader, Senior Researcher, Senior Research Software Engineer, Research Software Group Leader)	42
Phase 4	Established/Experienced/Senior (e.g. Professor, Director of Research Computing, Distinguished Engineer, Chief Data Scientist)	26



We had many more responses from those at a later stage in their career. This is perhaps not surprising, as the focus of the publicity was on targeting those who were more established in the community (project leaders, principal investigators, directors and co-investigators) as well as those working in software/digital-related projects whose involvement would occur beyond the level of a PhD and starting in post doctoral posts.

5.1.3 Institutional Affiliation

Participants were asked which organisation or company paid their salary, or to state one or more organisation/company that they were affiliated with in some way.

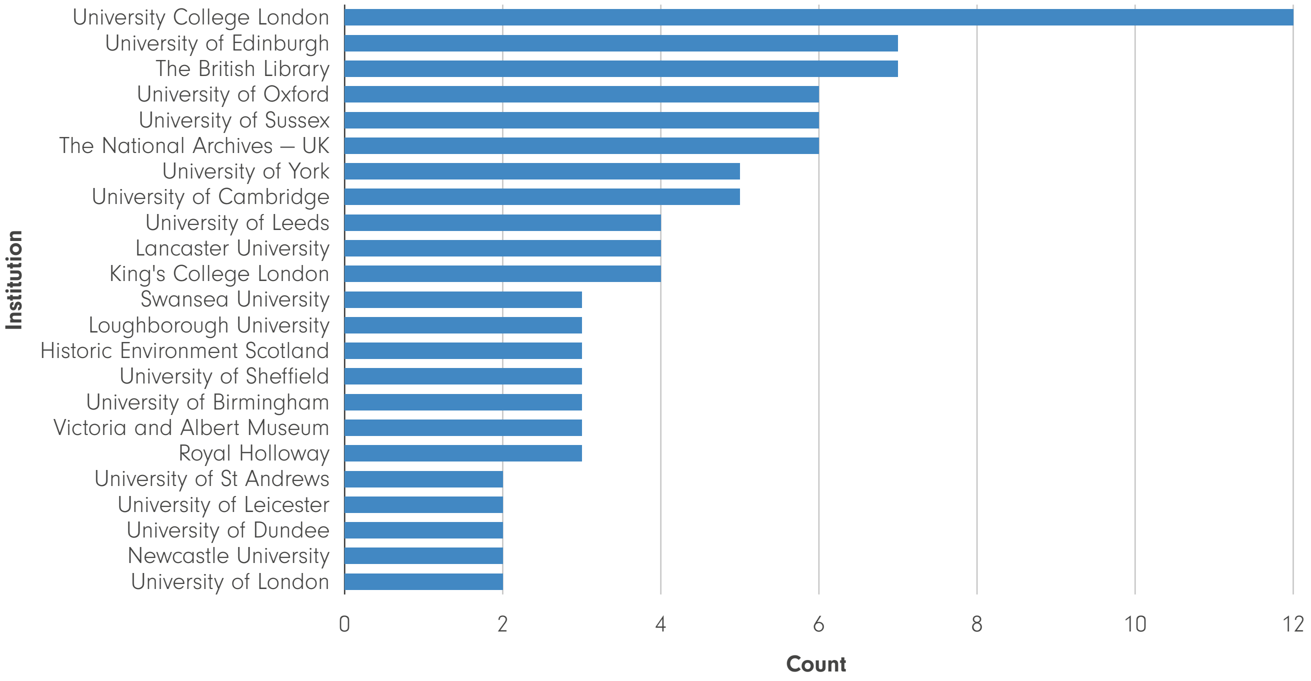
Below is a table of responses where there was more than one respondent:

Institutional Affiliation	Count
University College London (UCL)	12
University of Edinburgh	7
The British Library	7
University of Oxford	6
University of Sussex	6
The National Archives – UK	6
University of York	5
University of Cambridge	5
University of Leeds	4
Lancaster University	4
King's College London	4
Swansea University	3
Loughborough University	3
Historic Environment Scotland	3
University of Sheffield	3
University of Birmingham	3
Victoria and Albert Museum	3
Royal Holloway	3
University of St Andrews	2
University of Leicester	2
University of Dundee	2
Newcastle University	2
University of London	2

There were institutions that received two or fewer respondents, including:

Courtauld Institute of Art, De Montfort University, Durham University, English Heritage, Historic England, Imperial War Museums, Manchester Metropolitan University, Queen Mary University of London, Royal Botanic Gardens, Kew, SOAS, Tate, The Alan Turing Institute, The National Gallery, London, The Open University, University of Bristol, University of Hull, University of Kent, University of Manchester, University of Portsmouth, University of Sheffield, University of the Arts London, UKRI, Yale University.

With 46 UK institutions represented, it would appear that 20% of eligible institutions took part. Although representative, this does not show the whole picture. A complete mapping of the UK research landscape is beyond the scope of this survey, but the institutional profile of the participants offers several useful insights of the landscape. Unsurprisingly, universities with DH centres or labs and/or an established DH programme of study (BA or MA), such as UCL, University of Edinburgh, University of Oxford, University of Sussex, King's College London, University of Cambridge, among others, as well as institutions with a clear digital scholarship focus (The British Library, The National Archives) are prominently represented. This shows a strong interest in research software development in such environments where there has already been success with AHRC funding for digital projects. In addition, there is an increasing interest in research software from institutions with growing digital scholarship agendas, such as Loughborough University, Royal Holloway, University of Birmingham, Swansea University, University of St Andrews and University of Leicester, showcasing a growing need for decentralised funding and training provision for research software.

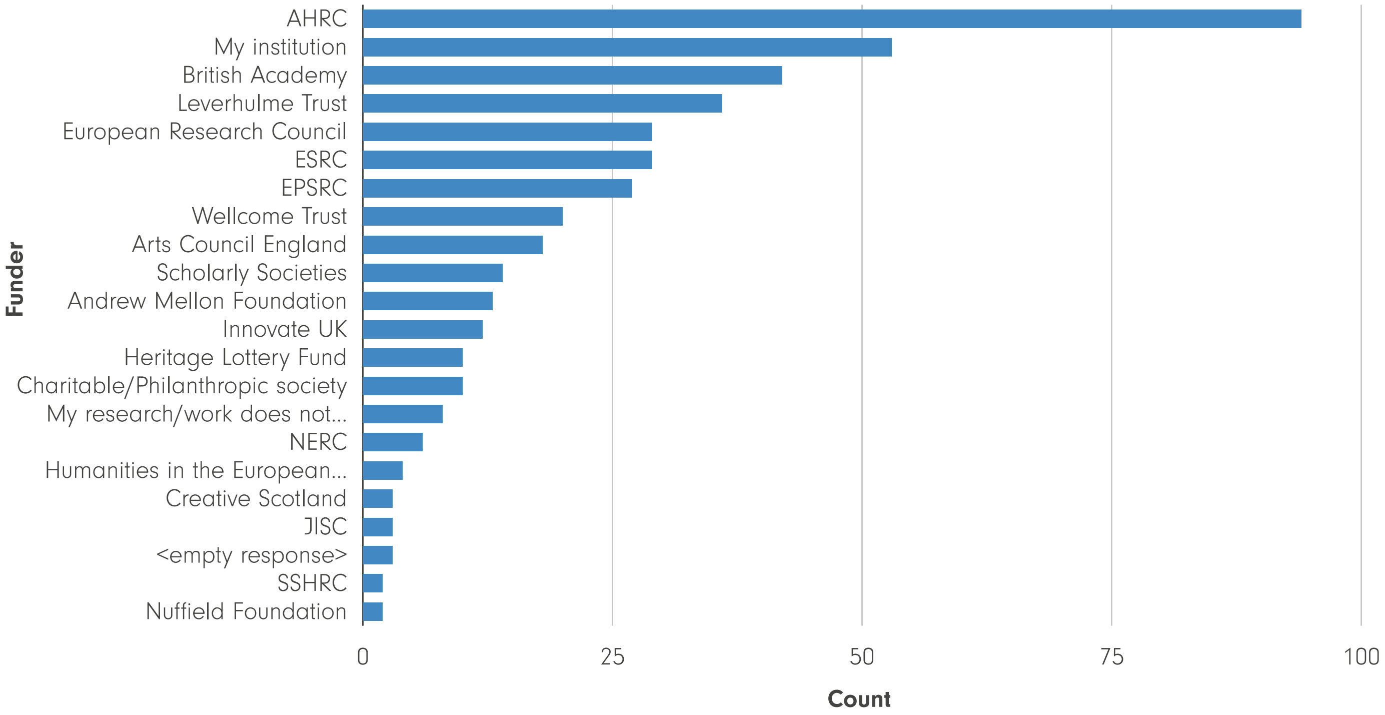


5.1.4 Funding

We asked participants which organisations they had sought funds from for their research. As we targeted the AHRC community, it was expected that most responses would be from those funded by, or seeking funding from, the AHRC or by other UKRI-related council bodies. However, it is interesting to understand the other bodies that the Arts and Humanities community see as natural funders for the type of work that they do, who may be considering (or may also want to consider) their support for software provision and training.

Funder[17]	Count
AHRC	94
My institution	53
British Academy	42
Leverhulme Trust	36
European Research Council	29
ESRC	29
EPSRC	27
Wellcome Trust	20
Arts Council England	18
Scholarly Societies (e.g. Royal Historical Society, Past & Present Society)	14
Andrew Mellon Foundation	13
Innovate UK	12
Heritage Lottery Fund	10
Charitable/Philanthropic society	10
My research/work does not normally seek external funding	8
NERC	6
Humanities in the European Research Area	4
Creative Scotland	3
JISC	3
<empty response>	3
SSHRC	2
Nuffield Foundation	2

[17] Some responses were free text responses; lists items were added manually to occurrences and titles were expanded in every situation where the author's experience suggested the acronym was not established. (e.g. EPSRC and other UKRI funders were not expanded).



Other replies with only one response included:

BBSRC, Botnar Foundation, British Council, British Sociological Association, Edinburgh & South East Scotland City Region Deal (Data Driven Innovation Programme), Elephant Trust, European Commission – H2020, EU/Council of Europe, Feminist Review Trust, HEFCE – Higher Education Funding Council for England, Henry Moore Foundation, Historic England, Institute of Classical Studies, Medical Research Council, National Endowment for the Humanities, Other fellowship schemes in the USA, PALATINE, Paul Mellon Centre, the Royal Society, the Royal Society of Edinburgh, SFC – Scottish Funding Council, Volkswagen Foundation, Wolfson Foundation.

This shows a wide variety of funders also cover the areas supported by the AHRC. In total, around 480 instances were mentioned, meaning that on average each participant chose four funders who they had sought funding from. This picture, on the one hand, celebrates the variety of funders and funding in the UK and further afield, and on the other highlights the precedence of funders such as the AHRC, British Academy, Leverhulme Trust, ERC, and Wellcome Trust in funding interdisciplinary work in the area of digital scholarship.

Surprisingly, none of these funders has committed in any way to a clear recognition of research software as a distinct, high-value output, resulting in a continuous lack of support and training for research software in Arts and Humanities environments, as well as a related lack of provision for software development and sustainability throughout A&H projects, as per section 5.4.2 Costs on a Proposal.

5.1.4 Personas

In this section, we take one respondent from each career phase who has ranked research software as important to their work to look at the kinds of answers that may be typical of a researcher in this group. The Phase 1 response shows a lack of institutional support, but other than the RSE in phase 2, institutional support is not ranked highly, and self-directed learning is emphasised.

A Phase 1 researcher in History at a university with a well-developed Digital Humanities provision ranks software at the highest level of importance in their work. They identify 'text processing and CSV data wrangling' as the primary reason they use software, and have developed their own software to correct OCR text. In spite of this, they rank their own software development expertise at the lowest level, and are self-taught using books, YouTube, short courses, and resources such as *The Programming Historian*. They do not feel that they have received sufficient training. This Historian uses textual, numeric and image data, and selects data cleaning, statistics, text/data mining, geospatial data, and has interests in most other techniques (excluding computer vision, augmented reality and machine learning, which they have no plans to use, and probabilistic linking, which they are not aware of). They use multiple laptops, and rate their current level of institutional support for their software at the lowest level in spite of the institution's investment in this area, highlighting a need for funder support. This also shows a marked disconnect between the importance of software to researchers' work and institutional support.

A Phase 2 researcher at a Russell Group university, who does not apply for funding directly but supports other projects, also ranks research software at the highest level of importance, developing it for their team. They are committed to open-source software, except with XML editors, and have developed several pieces of software that are publicly available. They rank their expertise at 4, but would appreciate more opportunities to test. This research software engineer works primarily with textual data and has had on-the-job time to learn, completing further qualifications, but has also supplemented this with self-directed study in their own time. Rather than a lack of training, they highlight that they are learning and improving over time, and there may be new, better ways to do things. They currently use software for database design, APIs, unit testing, version control, web application development, with a higher number of areas that they have no plans to use than the Phase 1 researcher. They use a desktop/laptop set-up and rank their institutional support at the highest level. They have hired developers and included costs in funding bids, but find outsourcing development unsuitable.

A Phase 3 Archaeology researcher at a different Russell Group institution ranks software importance at the highest level. They use software for data cleaning, data analysis, data visualisation, and automation of simple tasks. Their answers also show a commitment to open source software and transparency, but also a need for flexible tools for routine tasks and more bespoke ones. This is further highlighted in their call for students to be trained earlier in scripting, and more training opportunities for established researchers, to improve the scrutiny of results. They have developed their own software, but rank their expertise at the lowest level. They are largely self-taught, and do not feel they have had sufficient training. They use numerical and spatial data in their research, and currently use software for data cleaning, data visualisation, statistics, working with geospatial data and version control, finding their needs served by a desktop/laptop. They rank their institutional support in the middle of the scale, and have not employed an RSE, though they have included software costs in a funding bid. They view hiring an RSE as an unsuitable option for their work.

A Phase 4 Languages and Literature researcher at a mid-sized university uses research software and ranks it at the highest level of importance. Their projects develop bespoke software, but in their teaching they use off-the-shelf tools such as Palladio, Gephi and Open Refine. In developing software, they are focused on their research needs but also use by others, and specifically mention teaching, with an emphasis on software that is easy to use. They highlight the importance of establishing more communities of practice, and sharing solutions to specific problems more widely. They have not developed their own software, but have overseen teams who have. They find outsourcing software development unsuitable, but hiring an RSE or having access to an institutional pool is perfect for their needs.

5.2 SOFTWARE USE

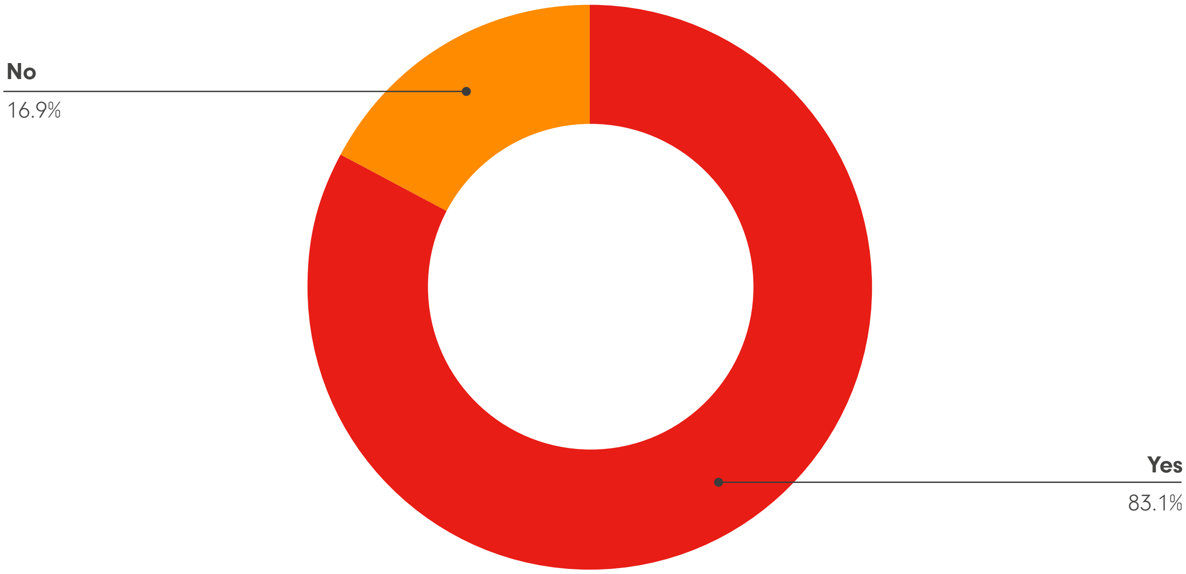
In this section, we highlight the fixed-choice questions that were asked to better understand the respondents' use of software and its importance to them. The open text questions are covered in section 6 Results: Open-Text Questions.

5.2.1 Software Use

Participants were asked the question 'Do you use research software?'. This was a mandatory question with two fixed choice answers ('Yes' or 'No'). See section 4.6 Definition of (Research) Software for the definition of research software provided to respondents.

Use software	Count	Percentage
Yes	98	83
No	20	17

Do you use research software?



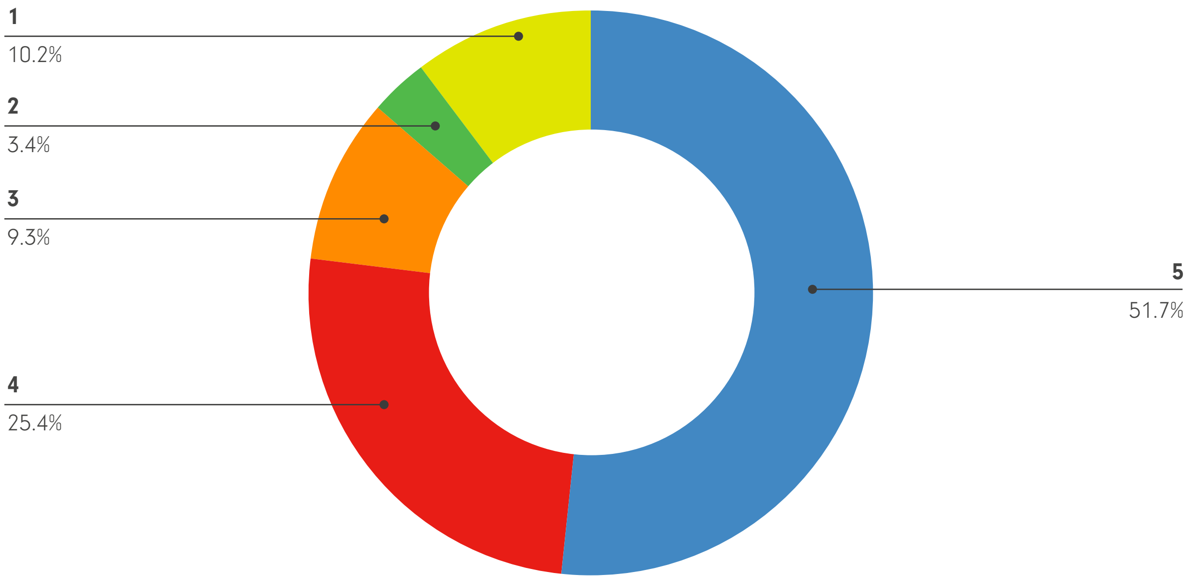
The percentage of those using software (83.1%) closely mirrors Duca and Metzler's study of software use in social sciences (2019), which found that 85% use software. An SSI survey conducted at the University of Southampton in 2019 similarly found 83% of respondents in social sciences used research software (*Software Use in Southampton*, 2019). This suggests that, in spite of what might be perceived as a disciplinary difference (i.e. that social sciences may be more likely to use software because of a greater use of quantitative data, surveys, and statistics), humanities researchers may be using software at a comparative scale. Therefore the kind of training given to humanities scholars merits closer attention, given that methodological training is often a more common core component of degrees in social sciences than humanities; as the literature review demonstrated, much training in this area is coming from third parties rather than institutions or funders.

5.2.2 Software Importance

Participants were asked the question 'How important is research software to your work?' and provided with a Likert scale, with 1 representing the value 'Not at all' and 5 representing the value 'Vital'.

Importance	Count	Percentage
5 (vital)	61	51.7
4	30	25.4
3	11	9.3
2	4	3.4
1 (not at all)	12	10.2

How important is research software to your work?



77% of respondents stated that software is vital or close to vital for their work. It is interesting to note that this is a smaller percentage than those who use software, and around 10% of respondents stated that software was not at all important for their work; perhaps this highlights the fact that there is a portion of the arts and humanities which will remain non-computational or will ignore altogether the use/application of software in their production of digital outputs. Further qualitative investigation on software use related to the career stage in section 6 Results: Open-Text Questions sheds light on where further action and investment is needed in terms of training and support. Although the scholarly value of software as a first class research output in the scholarly ecosystem has been constantly raised by research (Crouch et al. 2013) and by recent initiatives, mainly in the STEM, such as dedicated journals for publishing and reviewing software (The Journal of Open Source Software[18], SoftwareX[19]) and guidelines such as those produced by the SSI[20] and CLARIAH[21], the present survey's findings (section 2 Motivation) shows that software still does not fit comfortably into the funding landscape, especially in the arts and humanities.

[18] The Journal of Open Source Software.
[19] SoftwareX.
[20] SSI Guides.
[21] Clariah.

5.3 RESEARCH SOFTWARE DEVELOPMENT

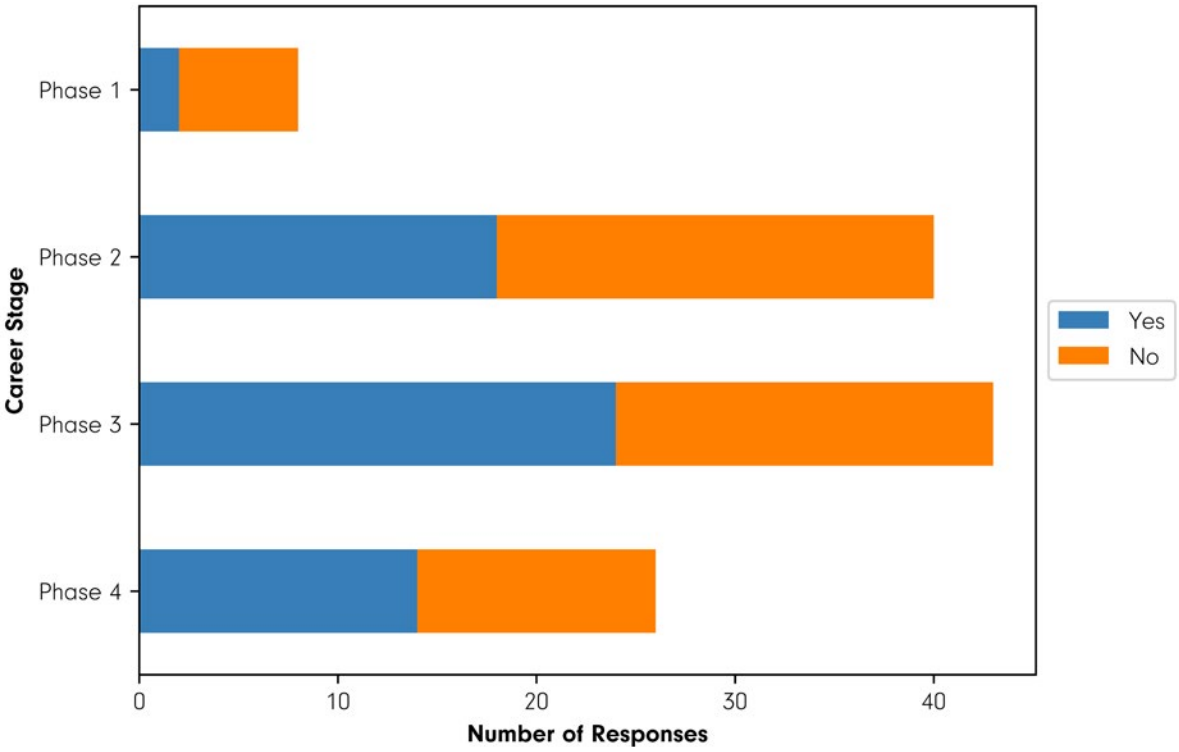
The questions in this section aim to map the existing software development skills and computing capabilities of the survey respondents, as well as to locate potential gaps in training and infrastructure.

5.3.1 Developing Software

Participants were asked, 'Have you developed your own research software?'. This included the definition in section 4.6 Definition of (Research) Software, and we further noted to participants that this includes anything from a script to automate some tasks (e.g. data cleaning) to writing bespoke tools, or contributing to larger pieces of software.

Career stage	Yes	No
Phase 4	14	12
Phase 3	24	19
Phase 2	18	22
Phase 1	2	6

Have developed own research software by career stage



50% of those who responded said that they developed research software, much higher than in comparable studies in other areas such as social sciences (see section 3 Review of Research on Digital/Software Literacy and Use in the Arts and Humanities). This may be because survey participants were a self-selecting group of more senior researchers rather than junior ones, reached through the SSI's advertising and dissemination to networks of primarily DH scholars. It is not necessarily reflective of the humanities community as a whole, mirroring the issues DH itself faces where only those already identifying as DH scholars, for example, attend conferences, subscribe to mailing lists, and read articles aimed at DH scholars, while the majority focus on domain and discipline-specific publications (i.e. within a discipline like English or History, or periods such as Renaissance or Victorian). Our survey participants reflect that DH outreach often fails to cross these silos. Nevertheless, the training, software use and needs of this group will still reflect some trends in the wider field.

The number who responded 'Yes' to whether they developed research software were fairly even (50:50) across the career phases (esp. 2-4).

The other sections in section 5.3 Research Software Development onwards apply only to those who answered 'Yes' to this question (n=59).

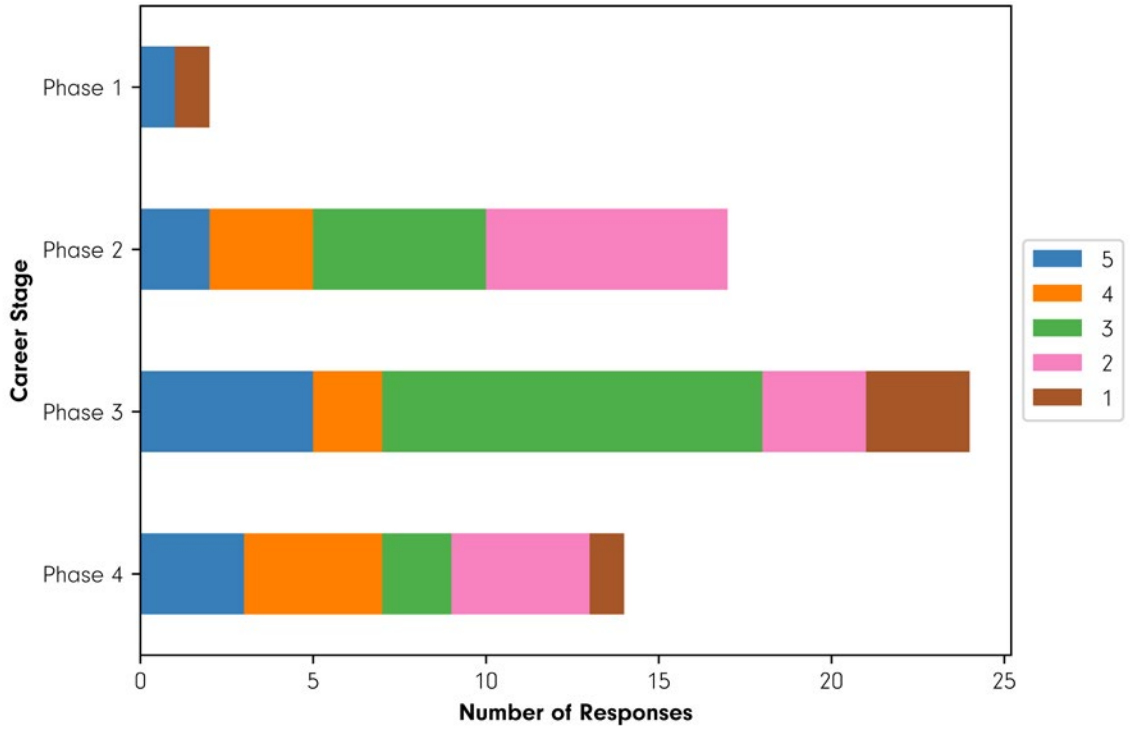
5.3.2 Expertise

Participants were asked the question 'How do you rate your software development expertise?', and provided with a Likert scale with 1 representing the value 'Beginner' and 5 representing the value 'Professional'.

Career stage	5	4	3	2	1
Phase 1	1				1
Phase 2	2	3	5	7	
Phase 3	5	2	11	3	3
Phase 4	3	4	2	4	1
Percentage	19	16	32	25	9

(Note: we excluded the one participant who did not state their career stage, but who had said they developed software, in the table and the graph below).

Expertise by career stage



The fact that responses grouped around the middle, with few ranking themselves as beginners (9%), 57% ranking themselves at 2 or 3, and 35% ranking themselves as 4 or 5 ('Professional') shows a lack of (confidence in) skills while also reflecting the 'getting stuck' phase, especially for arts and humanities scholars: it's easy to learn basic programming, but getting to more advanced skills is difficult for this community due to lack of formal training opportunities. This may also mirror the spread of career stages among participants, but the case remains that with such a self-selected group the middling level of self reported expertise is stark. Nevertheless, those of more advanced career stages use a wider range of software in their work, as we will discuss in relation to the qualitative questions in section 6 Results: Open-Text Questions.

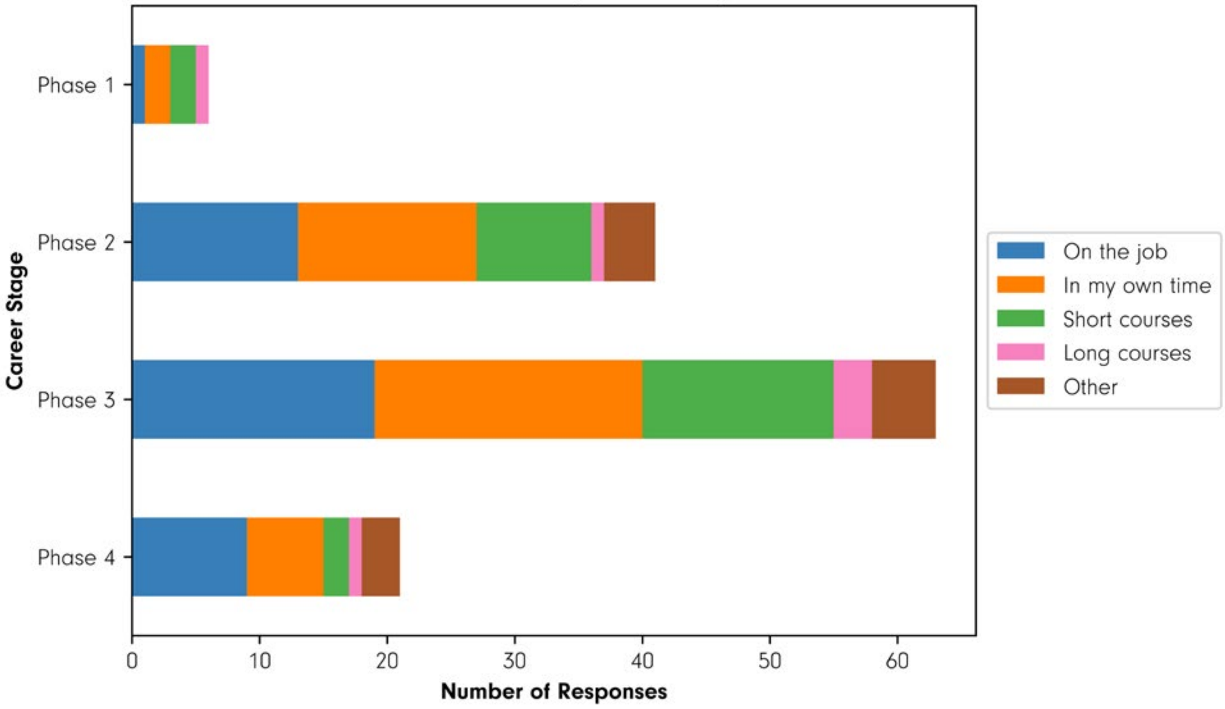
5.3.3 Learning

Participants were asked the question 'Where have you learnt your software skills?', and provided with a semi-closed list of choices.

Original Choice in survey	Short version
On the job (i.e. time allocated in the role to learn)	On the job
In my own time (e.g. books, YouTube)	In my own time
Part time short courses (e.g. Software Carpentry, Programming Historian, MOOCS)	Short courses
Part time long courses (e.g. part time degree or masters)	Long courses
Other	Other: (some examples of responses) • I am a user of software, not a developer • MSc then part time work • Learnt during PhD before starting this role • MSc in Computer Science and 10 years experience working with academics

Career stage	On the job	In my own time	Short courses	Long courses	Other
Phase 1	1	2	2	1	0
Phase 2	13	14	9	1	4
Phase 3	19	21	15	3	5
Phase 4	9	6	2	1	3
Percentage	32	33	21	5	9

How skills learnt by career stage



The results here show that participants largely did not have the 'formal' training offered by courses and degree programmes, and largely learned 'on the job' with a notable group self-teaching. This highlights a gap in the area of institutional support for DH skills development from introductory, intermediate and advanced skills. What's particularly interesting is the lack of short courses taken by those in the early and then most senior career stage. The Improved availability of training at different levels would certainly provide a clearer path to acquiring expertise and filling in knowledge gaps. Also it would be interesting to compare the state of formal training in the AHRC community with those in other communities but this is left for a future investigation.

5.3.4 Sufficient Training

Participants were asked the question 'Do you feel that you have received sufficient training to develop reliable software?' (mandatory) in order to assess software training provisions.

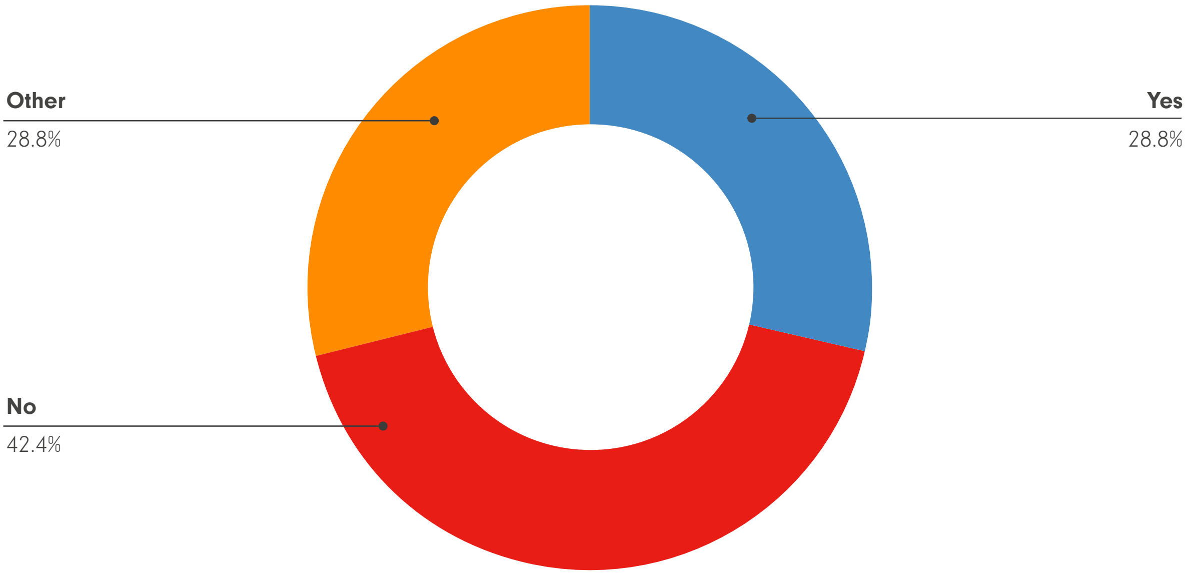
If this question was not relevant to the respondent personally (e.g. if they are a project leader who does not write software), they were requested to use the 'Other' option to let us know the reason.

Yes	16
No	25

We have categorised the responses that were provided under 'Other':

Other (18):	Software production is not a core part of my role (e.g. I lead or contribute in other ways) (7)
	I need further skills (e.g. for production or to make my code generalisable to other problems) (4)
	Yes, but learning is continual (e.g. on the job) (3)
	My skills match my current job needs (2)

Do you feel that you have received sufficient training to develop reliable software?



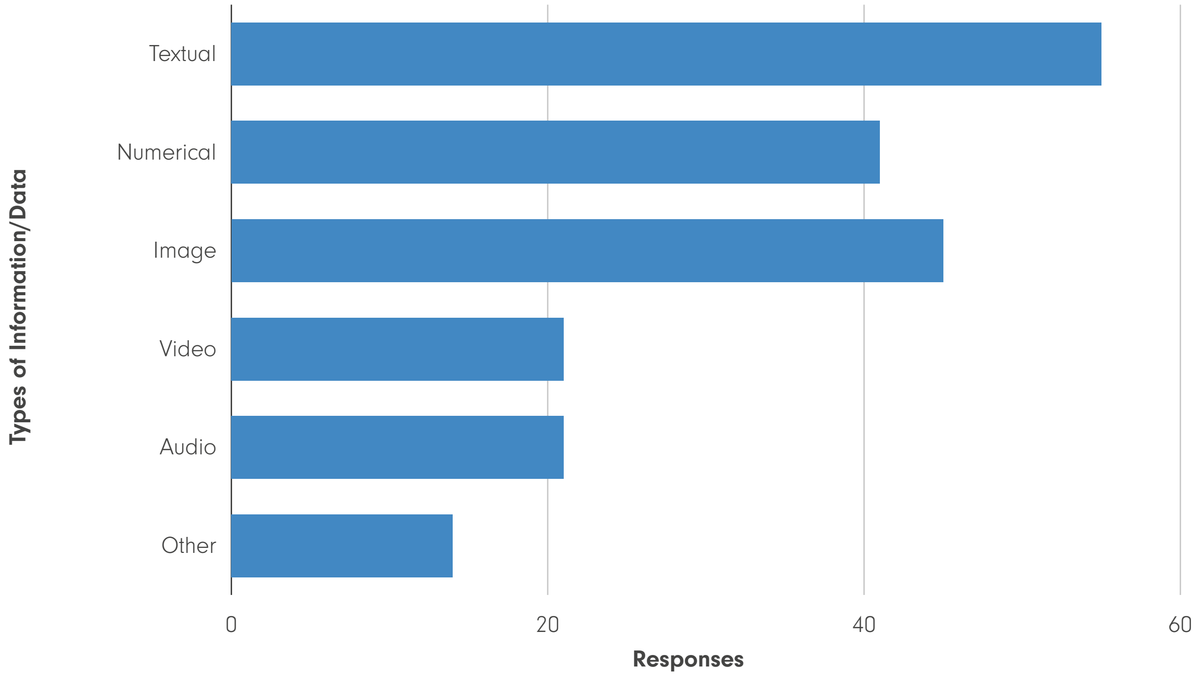
The majority of respondents (25) did not feel they had received sufficient training. In combination with the previous question, which showed that few researchers surveyed had undertaken formal training and were doing it 'on the job', the results here suggest that there is an unmet skills need for DH researchers. The open text answers reinforce this: there is a focus among the answers on staying up-to-date and the need for additional training. There is also an emphasis on collaboration: respondents note that they lead projects/teams and need (or have) a certain level of expertise in software in order to do that which is less than the expertise needed to be the primary creator of the research software they are involved with. Insufficient skills and time to do things 'properly' in a generalisable and reusable way are also highlighted, pointing at the need for greater training and resources to do this important work.

5.3.5 Type of Data

Participants were asked 'What types of information/data do you or your team work with?' in order to understand the data types they are working with to better understand their research practices and software needs. We thematically categorised the 'Other' responses.

Types of Information/Data	Responses
Textual	55
Numerical	41
Image	45
Video	21
Audio	21
Other:	3D (e.g. volumetric, spatial, immersive experiences) (4)
	Taxonomies & Ontologies (3)
	Databases & Tabular data (e.g. relational, non-relational, GIS) (3)
	Complex combination of types (2)
	XML (e.g. encoding of music, metadata) (1)
	Synthetic data

What types of information/data do you or your team work with?



The types of data mentioned here show a focus on textual, numerical and image data. However, video and audio are still significant, and textual data does not dominate as much as might be expected for arts and humanities researchers (this might also be explained by the fact that a large part of our respondents came from outside text-based disciplines, such as 'cultural and museum studies' (31) and 'music and visual arts' (16), showing the possibilities for cross-pollination of ideas with other cognate disciplines in other areas, such as the social sciences. The 'Other' responses highlight the variety and richness of primary data types that are being used by respondents and by implication the sophisticated software/digital systems and processes used to make use of this data.

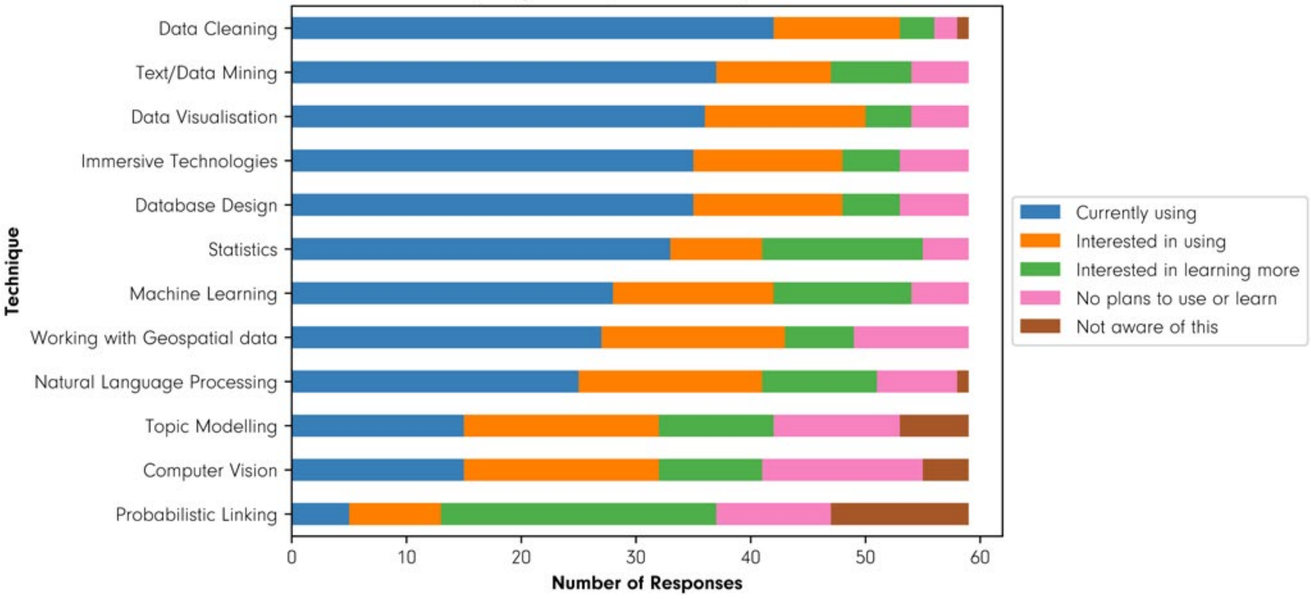
5.3.6 Computing Techniques

Participants were asked 'What computing techniques are you interested in exploring for your research?' (mandatory). In this question, we tried to map current trends in computing techniques, mainly referring to broad areas of advanced data processing techniques and computerised applications usually found or emerging in Arts and Humanities. It is important to mention also that these techniques can be used on their own or in combination to enact a more sophisticated approach to data processing.

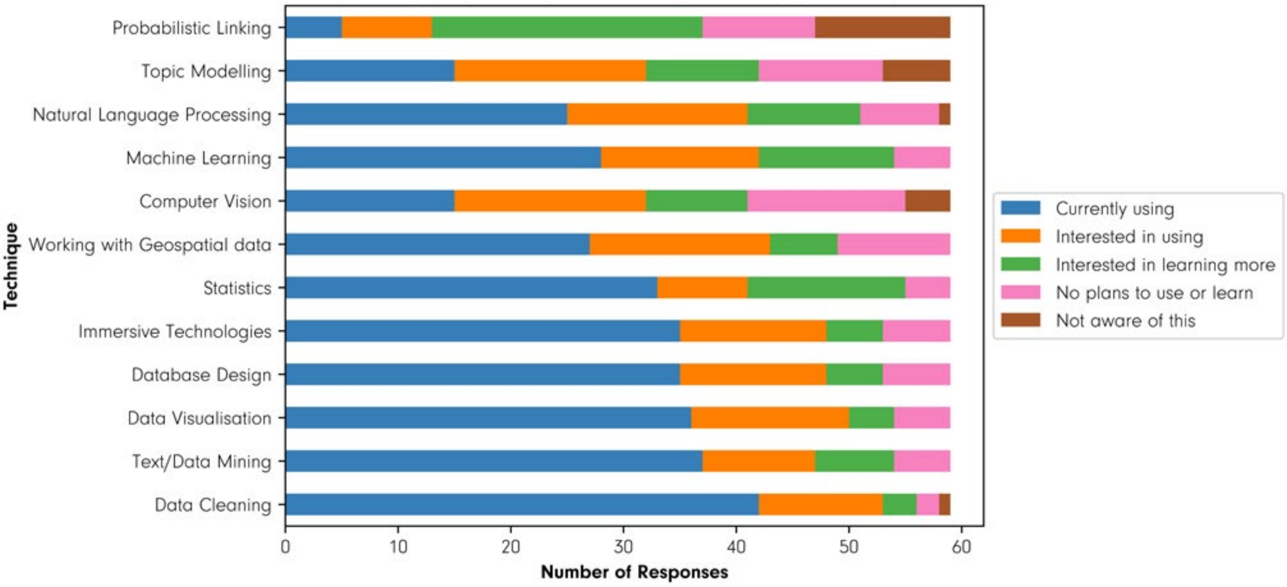
Technique	Currently using	Interested in using	Interested in learning more	No plans to use or learn	Not aware of this
Data Cleaning	42	11	3	2	1
Text/Data Mining	37	10	7	5	0
Data Visualisation (including 3D)	36	14	4	5	0
Database Design	35	13	5	6	0
Immersive Technologies - - Virtual Reality (VR) / Augmented Reality (AR) / Mixed Reality (MR)	35	13	5	6	0
Statistics	33	8	14	4	0
Machine Learning / Artificial Intelligence / Generative Adversarial Networks	28	14	12	5	0
Working with Geospatial data	27	16	6	10	0
Natural Language Processing (NLP)/ Textual Analysis / Sentiment Analysis	25	16	10	7	1
Computer Vision	15	17	9	14	4
Topic Modelling	15	17	10	11	6
Probabilistic Linking	5	8	24	10	12

We shorten the technique names on the y-axis of the graph below to make the graph easier to read.

Computing Techniques: sorted by current use



Computing Techniques: sorted by next usage/learning



There are some clear trends in these results about which computing techniques are most used by researchers: data cleaning, data visualisation, database design, statistics, immersive technologies, geospatial data and text/data mining, with almost no respondents not aware of these techniques (only 1 respondent indicated this response for data cleaning). Computer vision, machine learning, natural language processing, topic modelling and probabilistic linking were indicated as areas our respondents are interested in using and/or learning more, showing a clear training gap and desire for more information about these techniques. All answers showed that awareness amongst this self-selecting group of participants is good, though barriers to using these techniques were not investigated.

The data also shows that more baseline computing techniques are being used at the moment as they are a necessity to do any type of computation work (e.g. data cleaning, text/data mining, visualisation). In terms of what respondents are interested in using or learning next, there is a focus on more advanced techniques (probabilistic linking, topic modelling, natural language processing, machine learning and computer vision). This highlights the base areas in which training is needed — the core curriculum, as it were — but also highlights the more advanced areas to which people want to move next.

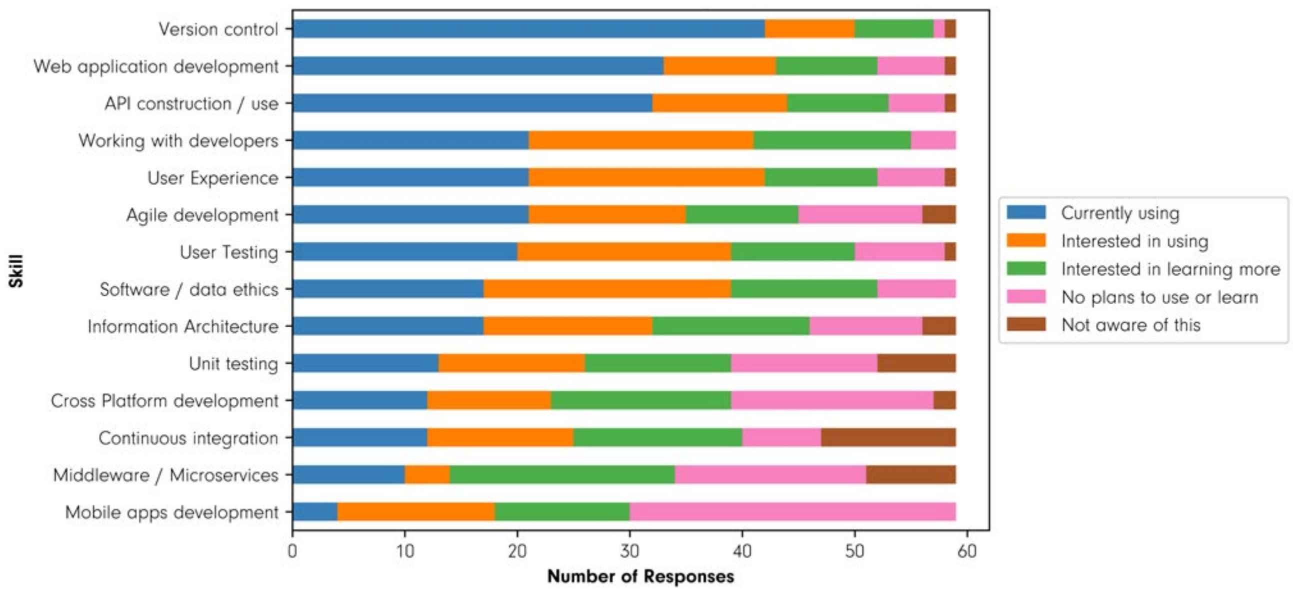
5.3.7 Production Skills

Participants were asked 'What computing production skills are you interested in exploring for your research? (mandatory)'. Through this question, respondents were asked about the repertoire of computing production skills they are interested in applying (currently using or learning to use) in their research, mainly focusing on advanced web application and web development skills as well as specific skills related to software use.

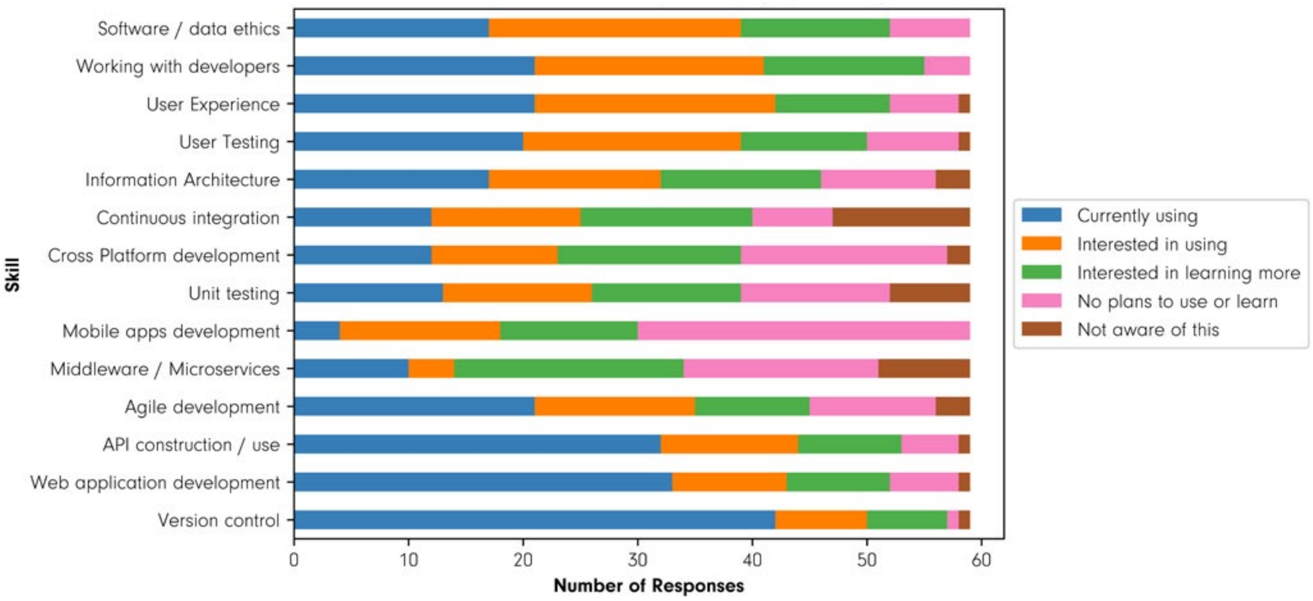
Skill	Currently using	Interested in using	Interested in learning more	No plans to use or learn	Not aware of this
Version control	42	8	7	1	1
Web application development	33	10	9	6	1
Application Programming Interfaces (API)	32	12	9	5	1
Agile development	21	14	10	11	3
User Experience	21	21	10	6	1
Effective skills for working with software developers (e.g. specifying/managing the brief)	21	20	14	4	0
User Testing	20	19	11	8	1
Information Architecture	17	15	14	10	3
Software & data ethics (e.g. bias and implications)	17	22	13	7	0
Unit testing	13	13	13	13	7
Continuous integration	12	13	15	7	12
Cross Platform development	12	11	16	18	2
Middleware/Microservices	10	4	20	17	8
Mobile apps development	4	14	12	29	0

We shorten the skill names on the y-axis of the graph below to make the graph easier to read.

Production skills: sorted by current use



Production skills: sorted by next usage/learning



Participants' production skills mainly concerned version control, web application development and APIs, while there is a growing interest in less technical areas such as agile development, user experience and user testing, and even less use of skills related to more theoretical aspects of software such as software/data ethics and information architecture. More specialised — and thus more expensive — areas of expertise such as mobile apps development and middleware services, are something mainly performed by external developers via outsourcing. In terms of what people would like to use or learn next, there is a focus on strategic data skills in terms of software/data ethics and working with developers, as well as more quality related skills such as user experience and user testing. More technical skills, such as information architecture and continuous integration, also appear, showing that to some extent there is an awareness of the commercial software development trends. This is probably an artefact of the career stage, with later career researchers more likely to have bigger teams working on larger projects.

5.3.8 Other Computing Techniques or Production Skills

Respondents were asked, 'Are there any other computing techniques or production skills that you are interested in exploring for your research?'. These are the open text responses. Note that in some cases the respondent may already have mentioned techniques or skills before we have removed these (there were two cases of machine learning and related technologies).

We categorised the responses into broader techniques or skill categories in the table below (parts of a response could be categorised in more than one technique/category depending on what was mentioned). The 'Examples' column pulls out specific techniques/skills mentioned in the responses.

Classification of open text responses:

Technique or Skill Category	Examples
Better Software Engineering (7)	Version Control (GitHub), Functional testing, following Object Oriented Principles (SOLID), Interoperability, documentation, more resilient programming paradigms (declarative systems & functional programming)
Alternative Computing Model (5)	Minimal computing, Distributed computing, Human computation, Quantum Computing
Reproducibility (4)	Infrastructure as code and containerisation technologies (Docker, Kubernetes, Unikernels, Vagrant)
Scaling up & Resilience (3)	Productionisation, Machine learning on larger datasets
Data: Standards, Processing and Preparation (2)	Warehousing, repositories, modelling, semantic web, linked open data and standards development
End of Project Sustainability (2)	Following the Endings Project Principles[22], Software Preservation
Community (1)	Code development communities
IT Skills (1)	Business Analysis
Digital Assets (1)	Non-Fungible Tokens (NFTs), Distributed Ledger technologies and programmable money.

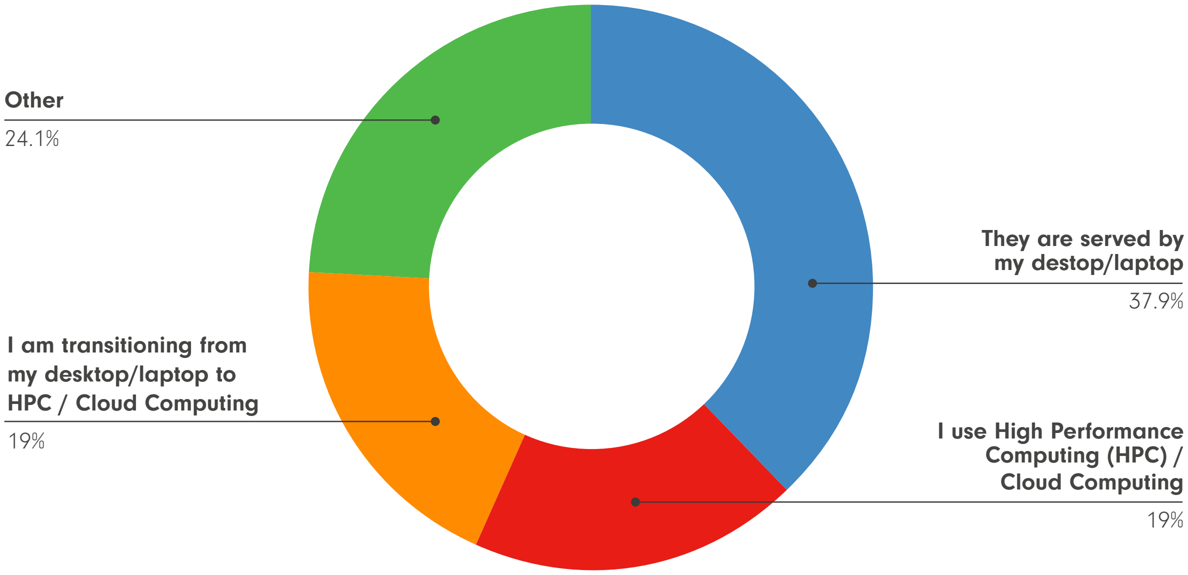
There is a strong focus in responses on improving software engineering skills, i.e. how software is managed and written to make it more robust and reliable. After this there is a focus on different computing models, including the use of low power devices in Minimal Computing, to Distributed Computing, using human beings for some computational steps in Human Computation to the new area of Quantum Computing. It would be interesting to be able to further explore why these choices were made, and what the timescale for these needs was for various respondents. Further down, there is a focus on scaling up and resilience of analysis, and then improved data skills and software sustainability and preservation skills. Code Communities, Business Analysis and Digital Assets also get a mention, highlighting the influence of broader socio-technical trends.

[22] The Endings Project.

5.3.9 Computing Scale

Respondents were asked, 'Which statement below best matches the scale of your computing needs?'.

Which statement below best matches the scale of your computing needs?



The table below summarises the fixed choice responses.

Your computing needs	Responses
They are served by my desktop/laptop	22
I use High Performance Computing (HPC)/Cloud Computing	11
I am transitioning from my desktop/laptop to HPC/Cloud Computing	11
Other:	(see table below)

There were 14 'Other' responses so we break these down further in the table below; some of the responses covered more than one theme.

Your computing needs (other)	Count
High power workstation	5
Cloud Computing	5
GPU Computing	2
High power laptop	2
Server	1
Clusters	1
Cloud storage	1
Hosting	1
Virtual machines	1
Support all levels	1
Containerisation	1
Scaling	1
Many laptops	1

The majority of respondents (22) replied that they are using their own machines (desktop, laptop or even multiple machines) for their software-intensive research and work. A reasonably small number of the respondents (11) are relying on Cloud services and High Performance Computing (HPC), and an equal number (11) are currently transitioning from their own device to a Cloud or HPC facility.

The 'Other' responses drew out an interesting middle ground between the desktop and HPC/Cloud infrastructure. With an emphasis on high-powered workstations/laptops, servers and clusters, this represented a mid-tier of power. In addition, GPU computing was mentioned, with its provision of specialised hardware that can accelerate specific tasks.

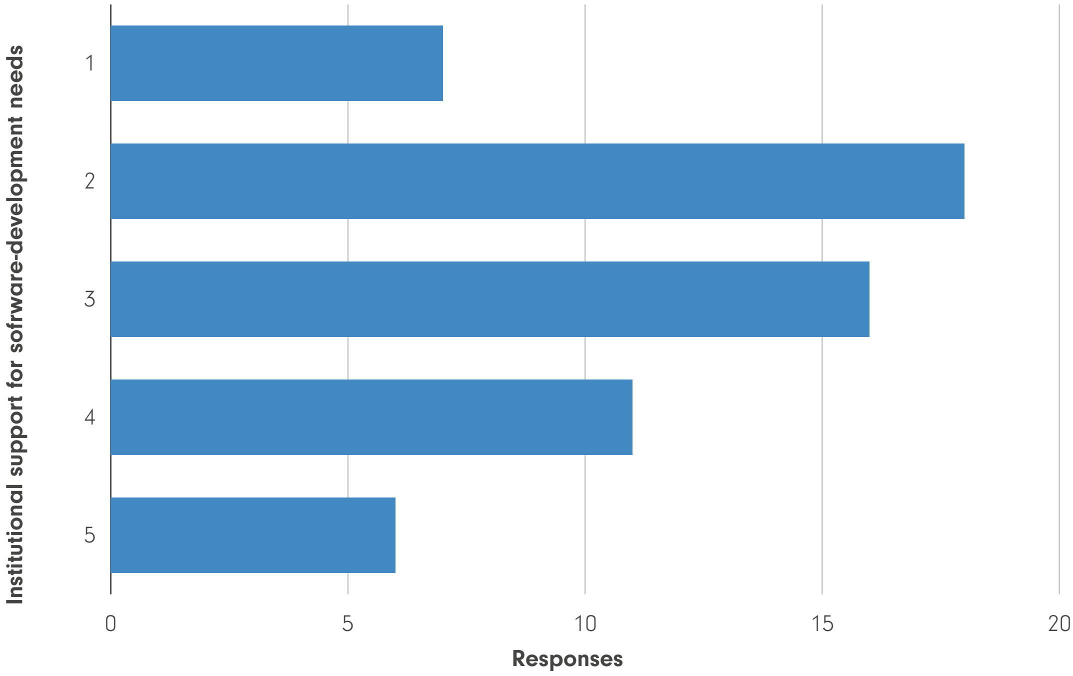
As Arts and Humanities scholars are often dealing with large sets of unstructured data (e.g. historical newspapers, books, election data, archaeological fragments, audio or video content), HPC facilities, cloud computing, more powerful workstations and specialist hardware is now being employed to give the computing power, speed and (in some cases) security now needed to sort through, mine, and better understand, analyse and visualise these heterogeneous datasets – tasks that have outgrown normal desktop PCs and laptops.

5.3.10 Institutional Support for Software

Survey participants were asked 'How would you rate the current level of support for your software-development needs from your institution?'and were provided with a Likert scale, with 1 representing the value 'Poor' and 5 representing the value 'Excellent'. This question maps the existing software development support as well as software development training provision from UK institutions, aiming mainly to assess whether the current institutional support and provision is adequate in terms of quality and quantity for the community.

Support for your software-development needs from your institution	Responses
1 (Poor)	7
2	18
3	16
4	11
5 (Excellent)	6

How would you rate the current level of support for your software-development needs from your institution?



What the responses show with regard to institutional support for the software development needs of respondents to the survey (mainly Digital Humanities researchers, or existing software users) is that there is room for improvement: 7 people consider the institutional support poor, 18 acceptable, 16 good, 11 very good, and only 6 people think that they receive excellent support. The limited institutional provision for software development support is quite telling and shows, on the one hand, the limited training provision for software development for Arts and Humanities use cases, and on the other hand, the need to tailor support for Arts and Humanities projects with digital/software components.

5.4 RECRUITMENT OF SOFTWARE DEVELOPERS

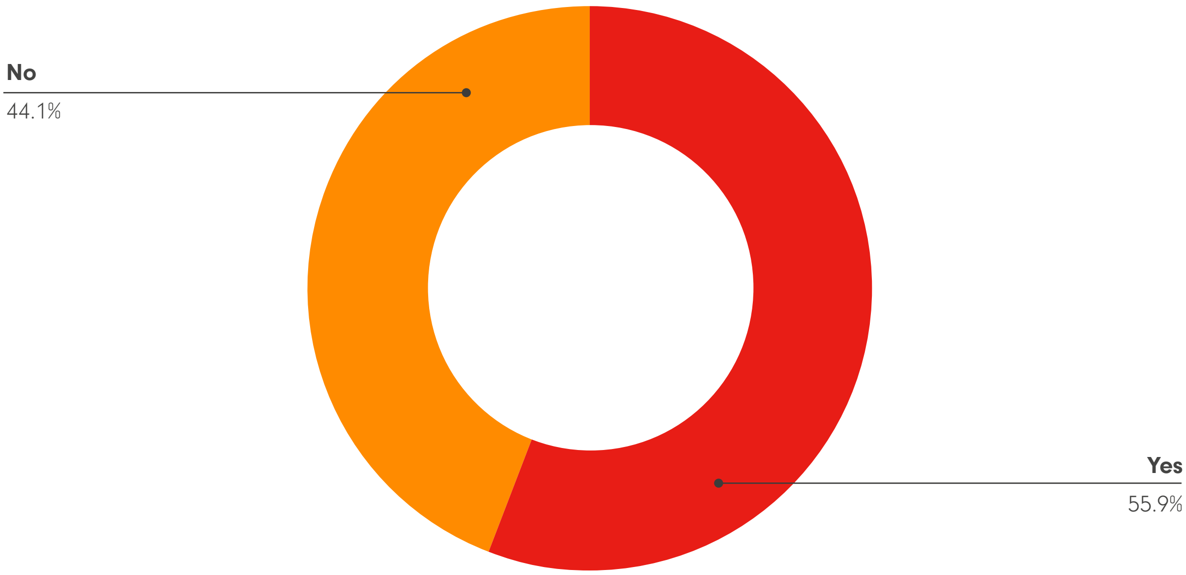
This section contains questions related to the recruitment of software development roles within Arts and Humanities communities and projects. On the one hand we want to trace the current landscape of the need of highly-skilled software developers within the field from a team and/or HR perspective, and on the other hand to claim for a structured, embedded RSE career path within the Arts and Humanities/DH projects.

5.4.1 Hiring

Participants were asked 'Have you or someone in your group ever hired someone specifically to develop software?' (mandatory). We want to see whether software development has been approached as a distinct function and/or role in an Arts and Humanities work environment or projects.

'Have you or someone in your group ever hired someone specifically to develop software?'	Responses
Yes	66
No	52

Have you or someone in your group ever hired someone specifically to develop software?



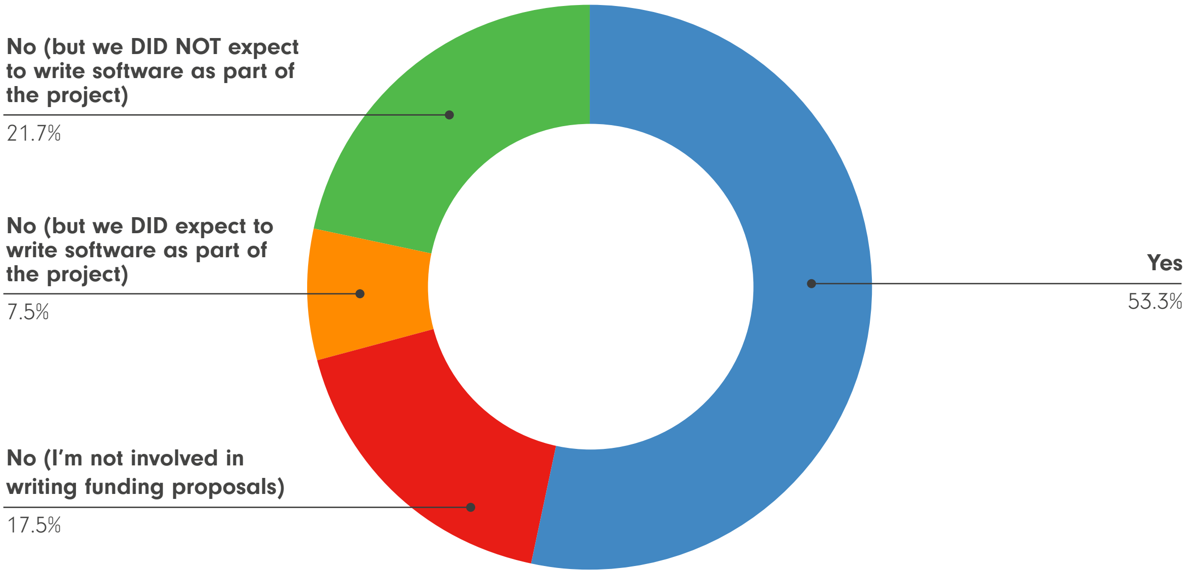
From the answers, 66 from the respondents declare that there was a distinct hire for software development within their group or project, acknowledging thus the emerging trend and need of having a dedicated provision for software development in research environments in the Arts and Humanities. Nevertheless, for 52 participants there was not a clear provision for a dedicated hire for software development within their group, verifying that there still a lot of work to be done in order to ensure there will be a dedicated software development role in Arts and Humanities and DH projects. Such an understanding is crucial for the Arts and Humanities community, RSE community, and funding bodies, as they each need to inform their practices accordingly.

5.4.2 Costs on a Proposal

Survey participants were asked "Have you ever included costs for software development in a funding proposal?' (mandatory). With this question we attempted to assess the provision of software development work as part of a funding proposal; in other words, whether software development is recognised as a distinct research output with a dedicated budget line.

'Have you ever included costs for software development in a funding proposal?'	Responses
Yes	64
No (I'm not involved in writing funding proposals)	21
No (but we DID expect to write software as part of the project)	9
No (we DID NOT expect to write software as part of the project)	26

Have you ever included costs for software development in a funding proposal?



More than half of the participants (64) answered that they have included costs for software development in a funding proposal, although 21 of the participants answered that they are not involved in funding proposal writing, and 26 participants answered that they are not expecting to write software as part of the project. There were 9 participants who admitted that they have not included a distinct cost for software development in a funding proposal, while expecting to write software as part of the project. This last category of approach corresponds with a culture of 'hidden labour' for software development in the Arts and Humanities and DH in general, resulting on the one hand in a lack of (research) credit and recognition for software development as shown by initiatives such as The Hidden Ref,[23] and on the other in a limited and structured training provision for software development in Arts and Humanities.

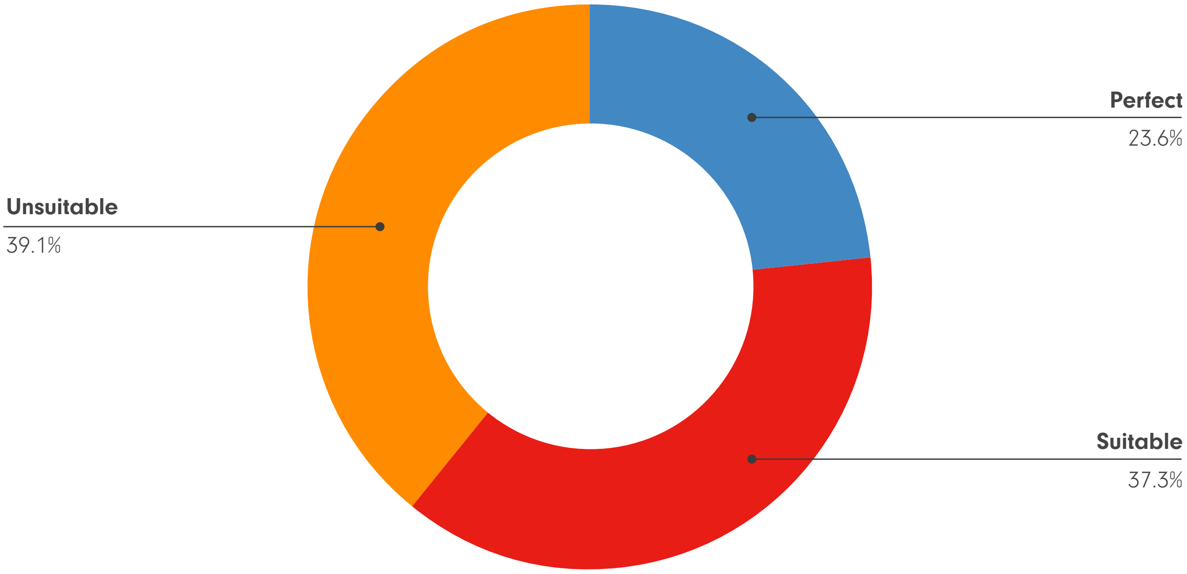
[23] The Hidden REF.

5.4.3 Staffing Models

Survey participants were asked, 'How suitable would the following models be for your software development needs?'. Through this question we seek to understand and map the current staffing models for software development in Arts and Humanities, mainly to complement existing generic RSE career paths and roles as discussed by the RSE society[24] or other RSE-based teams (Smithies, 2019).

For each of the staffing models, we asked participants to choose whether they were 'Unsuitable', 'Suitable' or 'Perfect' for their own group or project.

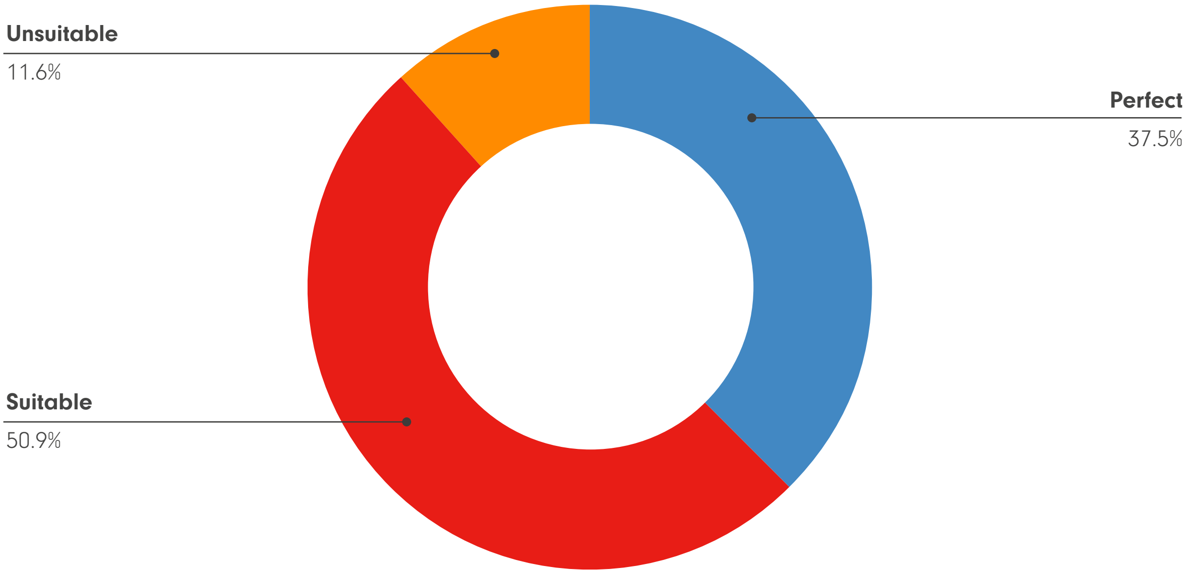
Hire a full-time software developer



The majority of the respondents (39.1%) felt that the model of hiring a full time software developer to cover the software development needs of the project is unsuitable, where the 37.3% answered that is suitable and only 23.6% that is perfect. The Arts and Humanities community still seems reluctant to have a FT role for software development for their projects, resulting, thus, in a constant downgrading of research software as an output of high-quality produced by trained professionals, or in a limited level of innovation in terms of software outputs due to lack of training and resources.

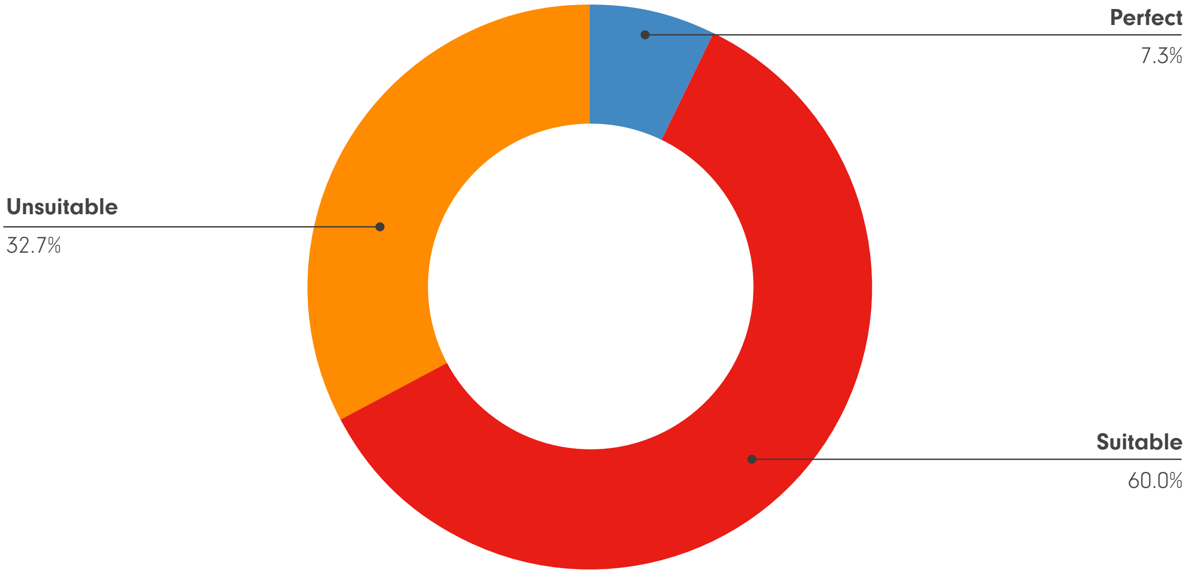
[24] Society of Research Software Engineering.

Recruit developer time from an institutional Research Software Engineering pool



As for the recruitment of developer time from an institutional Research Software Engineering pool, 37.5% of the respondents think that this would be perfect, 50.9% believe that this would be suitable, and 11.6% consider this an unsuitable solution. It has to be noted that an institutional (or even national) RSE pool aims to provide a technical software development and support service to academics from a wide range of disciplines; however, such a provision is still a desideratum in many UK institutions.

Outsource software development



60% of respondents indicated that the outsourcing of software development tasks is suitable, while only 7.3% consider outsourcing as a perfect solution, with 32.7% marking it as an unsuitable solution. What these findings show is a systemic reluctance of embedding software development as part of Arts and Humanities projects rather than outsourcing these tasks, with a significant impact on software training provision, institutional support and sustainability of the digital outputs in Arts and Humanities.

6 RESULTS: OPEN-TEXT QUESTIONS

6.1 WHAT DO YOU USE RESEARCH SOFTWARE FOR?

We asked what people use research software/digital tools for. These factors can vary depending on domain and career stage.

6.1.1 Career Stage

For a general description of the career stage of respondents, see section 5.1.2 Career Stage. In the table below, we look at the number of respondents per career stage, what their response rate was to the question 'What do you use research software for?', the average response length, and the standard deviation in response length (the greater the standard deviation, the more variable the response length was).

Career Stage	Number of respondents at career stage	Number of non-empty responses to question	Question response rate (%)	Average response length in characters	Standard deviation of response lengths
Not stated	2	2	100	48	20
Phase 1	8	5	63	76	93
Phase 2	40	35	88	79	54
Phase 3	42	39	93	142	200
Phase 4	26	23	88	136	167

Those in later career stages (e.g. Phases 3 and 4) tended to give longer answers and have greater variability in response length – some of them had much to say. This is a reflection of their vantage point and experience.

We now look at each career stage in turn, and turn our attention to some of the tasks and subtasks that software is used for, techniques or purposes of software use, and some details of application. It is worth noting that tasks and subtasks can be the purpose of someone's work (e.g. a Research Software Engineer), or a technique to be used along the way to get a research result (e.g. a researcher).

We will not cover 'Not stated' due to the low number of respondents in this category – those who did respond focused on high-level use cases such as searching, collaboration, model development and publication.

6.1.2 Phase 1

The answer to the question 'What do you use research software for?' for this career phase amounted to five responses only.

In terms of technologies, there is clear use of popular programming languages such as R and Python. There is use of GitHub as a code repository and, in some use cases (creative installations), mentions of easily deployable hardware (such as the Raspberry Pi).

In terms of techniques, these cover collecting data via surveys (e.g Open Data Kit), different types of data analysis (e.g. risk modelling and predictive modelling), and visualisation.

The overall purpose falls into a number of areas such as monitoring, prediction, creative installations and general reporting.

6.1.3 Phase 2

There were 35 responses for this career stage. There was only one person who stated that they did not use software. With an increase in career stage, there was a corresponding increase in the type of work being done with software (though there were also more respondents, which may skew the results).

The technologies mentioned are similar to Phase 1, with Python and R being stated. In addition, there is also Excel for those using spreadsheets to help in their research and, further demonstrating the broader work being done at this later stage, bibliographic software, survey platforms, project management and video conferencing platforms. These latter tools do not strictly fall into our category of research software as mentioned above, and so have been excluded from the table below. It is clear, though, that more and more software is enabling research, even if it is not directly focused on addressing the research problem. Research software is also being used as a starting point to build upon.

The types of data being processed are mainly textual data (e.g. books, newspapers, encyclopaedias, archival documents and other historical sources), survey data (qualitative data) and images (maps and multispectral). The responses, again, showed a broad range. This reflects the broad range of researchers who fall under the AHRC remit, and the different sources of data they deal with, which presents challenges when offering training across disciplines, which should be broadly relevant to different groups, but ensuring the content of any training or support is specific to a discipline or domain. It also shows the applicability of techniques from other disciplines, and that shared training with social scientists or physical sciences might be beneficial if advertised and targeted in ways that highlighted the applicability for the humanities.

By analysing the responses, the following technique and/or purposes were identified, along with occurrences and examples of applications.

Task	Subtask
Data Analysis (38)	3D data processing 3D reconstruction workflows Archival catalogues Data digestion Data coding Descriptive statistics Interpretations Image processing (e.g. reflectance transformation imaging, photogrammetry, general 3D imaging) Mapping & Spatial analysis Modelling Qualitative data Quantitative analysis Social Network Analysis Synthesis & Meta-analysis Survey data Text mining Textual analysis (large amounts) Thematic analysis
Data Collection (12)	Database design Digitisation Qualitative data Survey platforms Transcription of sources Web scraping
Data processing & Preparation (14)	Cleaning Data coding Filtering Literature review Textual encoding Wrangling
Publications (8)	Articles Data Editing critical editions Online collections Visualisations Websites (results via)

Task	Subtask
Data management (8)	Collating bibliographies Curation Storage Organisation Managing archival sources
Visualisation (5)	
Data Creation (2)	
Data Access (2)	
Project Management (2)	
Enabling collaboration	
Software development for research	

Although the question did not directly ask about the domains of use, a number of domains were mentioned, including Corpus Linguistics, Heritage and Archaeology.

It is clear that the majority of people at Phase 2 were focused on using software for data analysis, data collection, data processing and preparation, publication, data management and visualisation. These are the traditional steps for the use of data in analytical processes and in research-intensive environments, and it is therefore natural that they are reflected in responses.

In Phase 2 there is more detail and mention of more advanced techniques, and a wider range of uses and technologies and reasons. This matches the increased number of respondents at this stage, as well as the needs and maturity needed in the career stage around research software, especially as it is likely for people in this stage to have access to more funding, a larger team, and a longer project, which makes it easier to execute complex projects which have increased software needs. It is debatable whether the techniques and purposes are intrinsically more difficult or complex, or if this is just due to the time and exposure needed to learn. It would be worth offering specific training to those in Phase 1, i.e. those who may still be students, to help accelerate their progress and journey on the computational path, and equip them to lead projects and engage in collaborative work with peers at other career stages. This can only be better for research.

6.1.4 Phase 3

There were 39 responses from those at Career Phase 3.

More technologies were mentioned than in Phase 2. R and R packages were mentioned. It was interesting that Python was not mentioned, particularly as this was the group from which we received the greatest number of responses; perhaps this relates to the more statistical computing background of those at this career stage, in light of the way humanistic computing has evolved. Nvivo, a CAQDAS package for analysing qualitative data, was mentioned. Other software such as Qualtrics (survey and data collection) was mentioned, even though this does not fall in our definition of research software.

A number of responses focused on the role of the respondent in creating or enhancing software for others. Therefore, in Phase 3, there was a greater emphasis on this support function as a specialisation.

It was interesting that there was more mention of research topics and topics of application in the list of answers (9 responses). These included metadata about research outputs, web based experiments, computational analysis of text, images, documents and metadata, management of museum processes, digital history, data science, enhancing digitised collection records, and a pedagogical focus on exploring new tools so that they can explain them to others. This reflects more clarity and knowledge of computational application areas than at Phase 2.

Task	Subtask
Data Analysis (37)	Data workflows Machine learning Natural Language Processing (NLP) Network analysis Pattern extraction & analysis Quantitative analysis Simulation modelling Social network analysis Spatial analysis Statistical analysis Term identification Text processing
Data Processing & Preparation (30)	Aggregating Coding Creating textual corpora Data cleaning ETL (Extraction, Transformation & Load) Import Integration Linking Metadata creation Structuring
Data Collection (14)	Automated extraction Graphical representation Interviews Named Entity Recognition (NER) Surveys Templates
Visualisation (13)	Graphical representation
Data Management (7)	Curation Harmonisation Retrieval Source management Storage Versioning
Data Quality (3)	Data model testing Validation
Publication (7)	Articles Bibliography management Publishing of research data
Automation (2)	Analysis

Task	Subtask
Development of Software (5)	Application Programming Interfaces (API's) for data access Web applications
Data Exploration (1)	Querying
Data Presentation (2)	Image searching by features Reporting

It is notable that Phase 3 also covers the areas and themes of Phase 2 in terms of what research software is used for. However, there are some new themes (i.e. techniques and purposes) that are highlighted, such as data quality, automation, and data exploration.

This shows the greater maturity of practice at this stage, and is a further reason that those at Phase 3 are best placed to actively promote and contribute to software training planning and delivery. It should be noted that given the big differentiation in terms of software skills and relevant training in different career stages, AHRC and other funders should include such considerations in their various funding schemes as well as asking peer reviewers to comment on what training they think applicants might need, especially for early and mid-career fellowships.

6.1.5 Phase 4

There were 23 responses for those at Phase 4.

There was far more specificity of technology (i.e. software/digital tools) amongst Phase 4 compared to Phase 3 or any earlier phase. There were 17 different mentions of technologies. Some of these fell within our definition of research software, such as the use of programming languages including R and Python, markup via XML, CAQDAS packages such as Nvivo, RQDA and MaxQDA, textual processing tools such as CollateX, transcription editors and file format converters (e.g. pandoc), corpus analysis tools such as AntConc, and network visualisation via Gephi and Palladio. Other non-research software (i.e. software tools) was also mentioned, such as RedMine for project management, and bibliography software such as BibDesk. The fact that software was used to aid the management of research projects rather than conducting the research itself highlights also the collaborative and highly interdisciplinary nature of the projects run by more at more senior career stages: therefore management tools proving more useful for multi-person, interdisciplinary teams, where ECRs and even those at the start of their 'mid' career might be more likely to work (more) independently, and simply not need these.

Available tools for aiding aspects of work are used, and those who use them gain efficiency. It is interesting that those at the more senior career stage are aware of this; again, this points to those at earlier stages needing training and knowledge sharing to be more aware of how software can help them both with research and also with research management. As more senior researchers are often moving to management roles, it becomes obvious that their software needs also change, moving more to management and sustainability work around software developed by earlier career stage researchers to do direct analysis. This difference in software tools employed by people in different career stages should also inform the AHRC’s, or other funders', training provision, as well as shaping relevant funding calls and specific requirements.

The domains mentioned included Heritage Science, Conservation and Art History. A number of research topics were mentioned, such as hosting editions and commentaries from Middle East ancient written heritage, the dynamics of transformation in historic urban areas, corpus linguistics, modelling the semantic complexity of research work in GLAM organisations, and building global networks based on collections. One large-scale, ongoing project was mentioned in the form of the Openly Richly Annotated Cuneiform Corpus (Oracc)[25].

One of the respondents referred to themselves as a research software user rather than a developer. There was also the first mention of teaching as a use of research software, again suggesting a focus on software to facilitate others, as with the management software, rather than direct creation.

[25] Oracc.

Task	Subtask
Data Analysis (25)	Textual analysis Network analysis Qualitative data analysis Language concordancing Image processing Modelling Thematic analysis Quantitative analysis Corpora analysis Geochemical analysis Geographic analysis Statistical analysis Large Scale Text Mining
Technology (17)	Transcription editor Collatex Latex Redmine RQDA Nvivo MaxQDA R AntConc Gephi BidDesk Pandoc XML Python R Palladio
Data Processing & Preparation (8)	Warehousing Processing Images Manipulation Metadata processing Image processing Cleaning Transcription and translation
Data Management (6)	Managing reference material Bibliographical information Store & retrieve specimen metadata Link to digital proxies Storage Supporting reuse

Task	Subtask
Development of Software (5)	Building software for lemmatising cuneiform texts and translating them into modern language Improve performance and usability of websites Speeding up and automatic complex tasks Bespoke software development
Visualisation (4)	Graphing Metadata visualisation
Data Collection	Acquisition Photography Research publication content
Publication (2)	Articles Print images and 3D models
Data Presentation (2)	
Teaching (2)	
Accessibility (1)	Improving accessibility
Usability (1)	
Interoperability & Reuse (1)	International Image Interoperability Framework (IIIF)

There was much more of a focus on specific technology in Phase 4 responses, with clearer understanding of which software packages were useful for various data processing, analysis and visualisation steps. Again, knowledge from Phase 4 could be taught to those in earlier phases (e.g. Phase 3, 2 and even 1) to give them greater confidence in their technology choices and applicability of techniques. As section 5.3.3 Learning demonstrated that people are largely acquiring their training on the job and through self-study, a longer career is bound to mean more knowledge in this area, but a shift towards direct training would go some way towards bridging this skills gap. People at different career stages are using different technology – or the same technology but for different purposes. This strategy should inform both expectations and support provision from funders such as AHRC.

6.2 WHY DO YOU USE THE RESEARCH SOFTWARE THAT YOU DO?

Beyond understanding what people use software for, we aimed to explore why they choose to use particular software systems and digital tools. This is, primarily, to understand what the influences are on such choices.

6.2.1 Career Stage

For a general description of the career stage of respondents see section 5.1.2 Career Stage. In the table below, we look at the number of respondents per career stage, what their response rate was to the question 'Why do you use the research software that you do?', the average response length and the standard deviation in response length (the greater the standard deviation, the more variable the response lengths were).

Career Stage	Number of respondents at career stage	Number of non-empty responses to question	Question response rate (%)	Average response length in characters	Standard deviation of response lengths
Not stated	2	1	50	71	N/A
Phase 1	8	5	63	66	18
Phase 2	40	33	83	127	160
Phase 3	42	35	83	172	194
Phase 4	26	23	88	108	115

6.2.2 Phase 1

Those in Phase 1 mainly stated reasons around functionality, community, ease of use, the matching of needs, and the open source nature of the software they used. Some of them stated that it was a requirement of the role to use certain software, but largely the responses suggested they were able to choose what software they used.

6.2.3 Phase 2

There were 33 responses for this career stage. The number of times the reason is mentioned is noted in brackets after the reasons.

Category	Reason
Reasons more under the recipients' control	Open source (8) Fit for purpose (7) Ease of use (5) Experience with (3) Project or task specific (3) Community (2) Training available (1) Examples of use (1) Accessibility (1) Necessary automation tool (1) Features and flexibility (1) Handles large data (1) Interoperability (1) Enables quality work (1) What we built (1) Too vested to change (1) Ability to export (1) Documentation available (1)
Reasons more related to the role	Free to use (8) Used by others (4) Licence available via institution (4) Collaborators (3) Field of study (2) Commercial package is best available (1) Instrument manufacturer specific (1)

The Phase 2 responses show that sustainability of practice is a general concern, especially given the mentions of Free and Open Source software, fitness for purpose, ease of use, experience, and how it was related to project tasks. The reasons related to the role feature use by others and collaborators, and institutional support, showing the complex set of internal, team-based and institutional reasons that certain software is being chosen. It is interesting that, as with the previous question, the increase in complexity of reasoning matches the increase in maturity within their career, and the career stage-specific needs for longevity of solutions, certain features, and teamwork.

6.2.4 Phase 3

There were 35 responses from those at Phase 3.

Category	Reason
Reasons under the recipients' control	Open source (14)
	Project, domain or task specific (7)
	Fit for purpose (6)
	Community (5)
	Free (5)
	Widely used (5)
	Experience with (4)
	Reputation (4)
	Adaptable to other uses (3)
	Documentation (3)
	Efficient (3)
	Standards (3)
	To aid transparency and reproducibility (3)
	Ease of use (2)
	Features and flexibility (2)
	Single flexible language (2)
	Actively developed (1)
	Allow fast development (1)
	Avoid costly complexity (1)
	Better supports prototyping (1)
	Cross-platform (1)
	Diversity, Equity & Inclusion (1)
	Handles large data (1)
	Integratable (1)
	Interoperability (1)
	Make a good user experience (1)
	Market leader (1)
	Performance (1)
	Problem (1)
	Stable and mature (1)
	Sustainability (1)
	Too vested to change (1)
	Transferable skill (1)
	Transparency and reproducibility (1)
	Well engineered (1)
	Wide range of uses (1)
Reasons related to the role	Only way to get task done (5)
	Collaborators (4)
	Existing expertise in group (2)
	Existing stack (2)
	Licence available via institution (2)
	Institutional support available (1)
	Institutional systems (1)

In Phase 3, there are many more reasons why particular pieces of software are being used (in Phase 2 there were 19 distinct reasons, and in this phase there are nearly double, at 36); this shows the added complexity that comes into decision-making due to the added complexity of the type of work being undertaken.

It is interesting to note the shared concerns with Phase 2 around open source software, being specific to a particular domain, being fit for purpose, having a good associated community, being free, the ability to handle large data, documentation, and interoperability. It is also notable that the notion of 'being too invested to change' cropped up, as they had learnt what they had learnt or the project had built on a particular technology.

The Phase 3 concerns which were not in the earlier phase focused on factors which indicated more experience with software – such adaptability for other use, allowing quicker development, the ability to avoid complexity, supporting prototyping, supporting different systems, and being efficient. The user experience was also important, as was stability, maturity, the ability to aid reproducibility, the building up of transferable skills, flexibility for the technology to apply to a wide range of uses and being widely used. The answers reflected the concerns of much larger, collaborative pieces of software which were being used to build robust and/or production systems. The wide use and focus on transferable skills are indicative of the fact that this is becoming an important part of their skill set and those characteristics of being continually used are important for those in this phase.

Topics present in Phase 2 but not at Phase 3 are also significant: the main concerns not mentioned in Phase 3 but mentioned earlier are around accessibility, examples, tools, the ability to export and having training available, very much matching the concerns of those starting on their research software journey.

6.2.5 Phase 4

There were 23 responses from those at Phase 4.

Category	Reason
Reasons under the recipients' control	Community (4) Only way to get task done (4) Ease of use (4) Open source (3) Collaboration (3) Available (3) Built as solution did not exist (2) Free (2) Documentation (2) Experience with (1) Affordable (1) Interoperability (1) Problem (1) Adaptable to other uses (1) Ease of maintenance (1) Mature (1) Features and flexibility (1) Best choice for the job (1) Reputation (1) Recommendation (1) Fit for purpose (1) FAIR (1) Transparency and reproducibility (1) Standards (1) Examples of use (1)
Reasons related to the role	Available via institution (5) Project, domain or task specific (4) Instrument manufacturer specific (1) Institutional support available (1) To make it easier for others (1)

The difference between Phase 4 and Phase 3 reasons for using particular research software are notable: there is a focus in Phase 4 on collaboration, confirming the larger project context in which those in this stage operate. Also maintenance, and being the only solution, are highlighted, as well as policy level concerns such as the FAIR principles (Findable, Accessible, Interoperable and Reusable).

In terms of overlap, documentation, community, open source software, standards and reproducibility are shared areas – again reflecting the larger project context within which researchers in this phase operate.

It is interesting to note those things in Phase 3 which are not in Phase 4: here there is a larger concern on complexity, data size, transferable skills and software that allows rapid development. This speaks to the fact that those in Phase 3 are more likely to still be very much involved in the engineering and usage of the software, whereas those in Phase 4 understand what good software looks like but are potentially more involved with management and securing funds. Phase 3 researchers may also be leading research projects, but are more likely to be newer to doing so.

6.3 HOW COULD BETTER SOFTWARE PRACTICES AND SOLUTIONS IMPROVE YOUR AREA OR FIELD OF WORK?

Beyond understanding what people use software for and why they choose particular software and digital tools, we aimed to understand how those funded by the AHRC see better software practices and solutions improving their field or area of work. This is to gain a greater understanding of where improvements are likely to match expectations.

6.3.1 Career Stage

For a general description of the career stage of respondents see section 5.1.2 Career Stage. In the table below, we look at the number of respondents per career stage, what their response rate was to the question 'How could better software practices and solutions improve your area or field of work?', the average response length and the standard deviation in response length (the greater the standard deviation, the more variable the response lengths were).

Career Stage	Number of respondents at career stage	Number of non-empty responses to question	Question response rate (%)	Average response length in characters	Standard deviation of response lengths
Not stated	2	1	50	99	N/A
Phase 1	8	5	63	75	18
Phase 2	40	29	73	129	160
Phase 3	42	36	86	258	194
Phase 4	26	19	73	143	64

Analysing the wording used in many of the responses showed that respondents were thinking of improvements in terms of the following adjectives: more (e.g. training), better (e.g. documentation), easier (e.g. tool selection), availability (e.g. standardised digital methods) and 'less' (e.g. barriers to entry).

6.3.2 Phase 1

There were 5 responses from this career stage, hence we will summarise by way of text rather than tabulating the analysis.

In Phase 1, there was a general focus on the need for training (e.g. more entry-level provision, as part of earlier training – such as degree programmes – and therefore brought into normalised practice), community, replication, easier access to data, and greater accessibility of software (e.g. via a screen reader).

6.3.3 Phase 2

There were 29 responses for this career stage.

Adjective	Category
More	Training (4) Resilient software (2) Software sharing (2) Accessibility (2) Efficient solutions (1) Collaboration (1) Good open source choice (1) Research Software Engineering teams (1) Open Systems (1) Integrated systems (1) Visual software (1) Accuracy (1) Data literacy (1) Sustainable methods (1) Widely applicable standard techniques (1) knowledge of digital skills (1) (Re)use (1)
Better	Techniques (4) Interoperability support (2) Adaptation to domain uses (1) Collaboration support (1) Documented software (1) Maintained software (1) Sustained software (1)
Easier	Digital conversations (1) Development (1) Deployment (1) To find out what one should be using (1)
Availability of	Standardised digital methods (1) Alternative research practices (1) Common software writing rules (1)
Less	Barriers to entry of digital techniques (4) Bespoke code (1)

The nature of Phase 2 roles means that sustainability of practice is more of a concern than for Phase 1; though Phase 1 researchers may be concerned about losing access to data and software at the end of their projects, a high percentage will not continue with a research career. This is evident in mentions of free and open source software, fitness for purpose, and ease of use. An increased interest in collaboration is also clear in mentions of interoperability, adapting to different domains, support for collaboration, and software sharing.

We do not have many responses at Phase 1 to compare, but the areas of overlap stop at accessibility and training. The things mentioned in Phase 1 but not in Phase 2 are easier data handling and replication.

In Phase 2, there is a much broader area of concern around how more, better, easier, availability and less of certain things can improve the domain of respondents' work. More resilience, efficiency, integration, reuse and standards point to a more developed understanding of how these can make the day to day work of those in this career phase better, and also hints at the more developed and complex, multi-person and multi-tool setup that is operated in by those at this career stage. Sustainable practice starts to manifest itself at this stage.

6.3.4 Phase 3

There were 36 responses from those at Phase 3.

Adjective	Category
More	Transparency and reproducibility (6) Time to actually learn and practice (3) Reusable software engineering (3) Research Software Engineering (2) Automation (2) Case studies (1) Use of advanced tools (1) Awareness of related work (1) Timely collaboration opportunities (1) Graduated training (1) Confidence is using tools & techniques (1) Interoperability (1) Standardised techniques and workflows (1) Productivity (1) Opportunities to learn on the job (1) Mentorship (1) Resilient software (1) Open Systems (1) Efficiency (1) Training earlier in career (1) Code sharing (1) New technology in teaching and research (1) Compute resources (1) Use of standards (1) Open Data use (1) Digitisation of resources (1) Training in newer techniques (1) Training to help people new to an area (1) Open source (1) Libraries (1) Development and compliance with new standards (1) Taking the learning back from project to update core data systems and models (1)

Adjective	Category
Better	Long-term sustainability (3) Computational understanding (3) Customisation (1) Problem/solution resources (1) Long-term communities (1) Development processes (1) Documentation (1) UX of tools (1) Practices and solutions (1) Discoverability (1)
Easier	To keep up with new circumstances (1)
Availability of	Earlier computational training (3)

It is interesting to note that there was overlap between Phase 2 and Phase 3 in the area of resilience. Whether the changes we saw highlighted were more due to terminology or to do with nuance remains to be seen, but we posit that this is due to a richer understanding of the software/digital space.

In Phase 3 there was a focus on long term sustainability (vs sustainable methods/software in Phase 2) showing a better understanding of the long term importance of sustaining software. There was also a focus on computational understanding (compared to knowledge of skills, techniques, and literacy in Phase 2) showing an appreciation for a deeper understanding of what is happening at the software/digital level. There was mention of standards, libraries and interoperability (vs. a focus on collaboration and conversations in Phase 2). Improvements in training were mentioned in various ways with a focus on earlier training, graduated training and opportunities to learn on the job (whereas at Phase 2 training in general was mentioned and knowledge of digital skills).

There is a strong thematic overlap in the improvement suggested at Phase 3 and Phase 2; however, Phase 3 responses suggest a deeper understanding and more nuance in matters pertaining to the long view around sustainability, community, code integration and the need for appropriate early and ongoing training.

Rather than just using a process of osmosis and rediscovery, this points to the fact that those at Phase 3 have excellent ideas that could accelerate the outlook and direct the efforts of those at Phase 2, and even Phase 1, if the mechanisms were in place. It also suggests that the practical 'do-er' concerns still exist in Phase 3, albeit at a more mature level, and these make them a rich source of information for Phase 4. Phase 3’s broad view while being hands-on makes them particularly useful for providing input and feedback for initiatives being planned by AHRC.

6.3.5 Phase 4

There were 19 responses from those at Phase 4.

Adjective	Category
More	Interoperability (3) Consistency of approach (2) Use of version control (2) Digital Humanities experts (1) Use of tools that combine data, code and presentation (1) Computational awareness (1) Composability of tools (1) Transparency and reproducibility (1) Publishers require reproducibly packaged code (1) Free software (1) Standardisation (1) Case studies (1) Sharing (1) Guidance on consistent production of software (1) Communities of Practice (1) Integrated systems (1)
Better	Documentation (2) Long-term sustainability & availability (2) Tutorials (1) Signalling of cost/benefit of learning something new (1) Signposting of services for the GLAM sector (1) Record linkage from many sources (1)
Easier	To collaborate (1) To pursue the creation of ideas into software (1)
Availability of	A consistent way to access data across the GLAM sector (1)

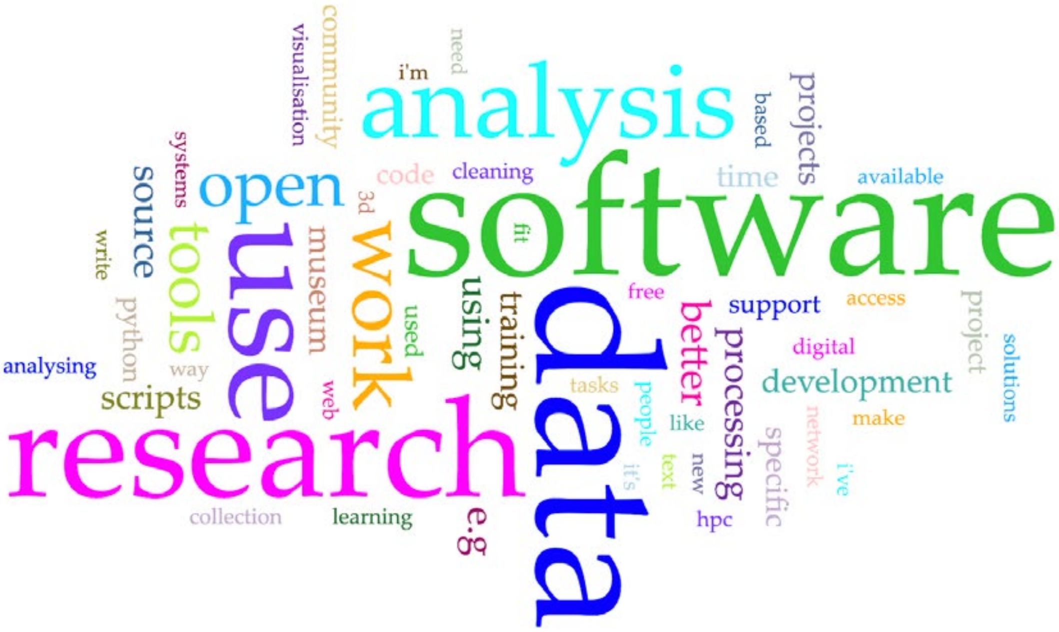
The areas of overlap between Phase 4 and Phase 3 include interoperability, documentation, transparency/reproducibility, code sharing and an increase in computational awareness/understanding.

There was a much more outward-looking view given by those in Phase 4 versus Phase 3. Phase 4 responses included suggestions around communities of practice, consistent production standards, more Digital Humanities experts, publishers requiring reproducibly-packaged code and signposting of services for the GLAM sector, easier ways to turn ideas into code, and the cost/benefit analysis of learning new skills. These areas highlighted a much more strategic and policy-driven understanding of what digital/software aspects could improve the domain.

Phase 3 concerns are around more advanced production concerns, including provision of computing computing resources, development standards, open source and systems, reusability, resilience, more Research Software Engineering provision, better mentorship, and ongoing training.

7 RESULTS: THEMATIC UNDERSTANDING OF RESPONSES

The total number of respondents was 118.



Analysis of the free text answers together shows some common themes: software for the arts and humanities and software appears to be more bound up with 'use' than 'development', with the former appearing in more free text answers. Respondents to the survey highlighted sustainability, especially in later career phases (evident here in words such as 'open', 'source', 'access', 'available'). The appearance of 'training', 'time', 'community', 'better' and 'learning' show a concern with the infrastructure that supports software use (here focused on 'data', 'analysis' and 'tools') and development (though less commonly discussed, 'Python', 'write' and 'scripts' show development is taking place).

Analysis of the thirty most common collocations, i.e. words which appear together, reinforces this focus on data analysis and software use above developing software, though responses may have drawn on the wording of the open text questions (see section 6 Results: Open-Text Questions). It highlights that data analysis, cleaning and processing were most often mentioned when it came to specific use of software, with data collection also appearing in the top 10 collocations and data management also mentioned. Sustainability emerged as a concern through 'open source', though it was not commonly used in its own right.

Term	Collocate	Count
research	software	46
data	analysis	41
analysis	data	34
use	software	28
open	source	28
software	use	26
data	processing	21
data	cleaning	19
use	research	18
data	collection	14
software	research	14

Term	Collocate	Count
software	work	13
research	use	13
work	software	13
software	data	12
better	software	12
data	visualisation	11
software	better	11
research	work	11
research	data	11
data	use	10
data	software	10
data	management	10
use	tools	10
use	data	10
analysis	visualisation	10
work	research	10
tools	use	10
data	web	9
data	research	9

8 RECOMMENDATIONS & FUTURE STEPS

As the results show, there is an undeniable need to tailor support to career stages. What becomes clear is that mid-career users of software, unsurprisingly, often engage in more advanced computational work than more junior colleagues, reflected in their concerns about data quality, automation, and data exploration; meanwhile, those in the most senior positions are using more software to aid in the management of larger activities. A through-line is visible that moves from individual work, to individual work done at a higher level, to team-based work, and finally to managerial oversight of such work. With this knowledge in hand, it becomes clearer what training is required to enable people to be able to take the next step on their journey, and indeed how knowledge can be shared by more junior colleagues who may be receiving training in, or working directly with, newer software. This needs to be supported by funders and institutions, rather than being effectively 'rediscovered' each time.

The benefits of all software skills should be highlighted in funding calls, institutional standards and communities of practice to highlight the benefits for attracting funding, career growth and of good citizenry. Benefits can act as a powerful motivator to adoption. In the development and delivery of training offerings, exposing what other courses and resources are available will help participants journey from novice to expert and allow them to get a better understanding of a path to follow to improve, as well as enabling more established researchers to stay up to date in their knowledge.

This might be reflected in different funding calls' assessment criteria, for example. In Phase 1, there should be more of a focus on career preparation, i.e. training in software for project management and teaching, as well as a focus on supporting the creation of sustainable software and digital projects, both for ECRs and PhD students. Those in Phase 1 need such training to help with the problem of finding out what they do not yet know, to equip them to further their research career and make the connection that will facilitate more ambitious software development and research projects. Opportunities for networking across disciplines, and with other areas such as social sciences, should also be created beyond those within the cohort itself, offering the opportunity to learn from cognate disciplines. In Phase 2 and Phase 3, as well as training on specific techniques, support might focus on how to find, employ and work with RSEs, as well as management software and software for teaching. As time becomes a factor in keeping up to date with what is going on in the field of DH, periodic check-ins or showcases may be beneficial, particularly for Phase 4.

In the following sections, we will focus our recommendations on three groups: funders, institutions, and communities of practice. There is necessarily some overlap between these stakeholders. We will also consider not only specific, concrete actions which can be taken, but also the barriers and attitudes which need to be addressed to facilitate change. These are primarily the view that research software is not a high-value scholarly output, and that A&H researchers are not significant creators and users of software.

8.1 FUNDERS

One of the central problems with current funder support of software development is the lack of recognition of software and digital innovation as a high-value output; this is also a consequence of the REF, as discussed in section 3 Review of Research on Digital/Software Literacy and Use in the Arts and Humanities. Changing this attitude will enable a more creative portfolio of outputs, but will require support in terms of sustainability and training. This lack of recognition also contributes to time pressure, identified by participants as a barrier to their software development and engagement with new tools. With so many demands on their time, researchers are not always able to prioritise software when it is not viewed in as prestigious a light as other kinds of output. Further, though the results here come from a subset of the Arts and Humanities community, it should be recognised that there are still a vocal group of researchers using and developing software. Around 85% of our respondents indicated they use and/or develop software; this should drive a change in how software/digital skills are viewed by funders, from optional/supplementary to essential. Many social science-based computational training offerings would be relevant for those in the humanities; however the relevance for those in the humanities needs to be illustrated, and there are certainly grounds for more knowledge-sharing and mutual influence between these domains in particular. Taking these two facets together, it is clear that the difference in software use versus its perceived importance to careers and as an output highlight the fact that research software needs to be more valued as a research output by both funders of humanities research and publishers. Additional support for software creation and use, as well as a commitment to sustainability, will help transform the landscape.

The AHRC and other funders thus need to facilitate the inclusion of training provision for research software skills development, and ongoing support, as a priority to overcome institutional differences (under a third of respondents felt they had sufficient training). For example, a commitment to sustaining software for a set period, or archiving digital projects, could form part of the AHRC's support of funded PhDs and other ECRs. Some UK institutions have well developed DH centres, or consortia support for DH and digital training, but others do not, leading to an imbalance in support for researchers at all career stages. In light of uneven provision and the ability to cross-fertilise across disciplines, there are opportunities here for the AHRC to work strategically with the British Academy, Leverhulme Trust and ESRC on joint initiatives to support research software and research software infrastructure, as they serve overlapping communities. This training might take the form of introductions to areas of high use (e.g. data cleaning, text/data mining, visualisation) and more advanced offerings such as probabilistic linking, topic modelling, natural language processing, machine learning and computer vision. Production skills show a similar pattern, which reflects a baseline of skills which could be targeted at all researchers in this space (such as version control, web application development and API generation) and those topics which will be useful at later stages, such as software/data ethics, working with developers and user experience.

There is an assumption among some researchers that the AHRC community deals mainly with text and image data. However, the wide range of data types mentioned here (including not only text but numerical data, video, audio and spatial data), and the fact that many people work with multiple data types, would suggest that there is untapped possibility in cross-disciplinary training beyond the humanities remit. Another pervasive myth is that A&H researchers do not make use of HPC/Cloud/GPU resources, but over 50% of respondents indicated that they were either using, or planning to use, these resources to meet their computational needs. As such, there needs to be a greater provision both institutionally and at a funder level.

There is a clear emphasis in the results on better software engineering, reproducibility, scaling and investigating different computing models; the evidence shows these areas need support to allow them to be realised, and to benefit the research being undertaken. Those in later career stages (e.g. Phase 3), who are still creating software as well as taking on more managerial roles, have a depth of understanding that should be utilised to help shape calls for future funding. Including in the AHRC's peer review form a section on software needs and training would allow those reviewing applications to elaborate on what they think those applying might need or benefit from, rather than placing the onus entirely on the applicant to foresee their future software needs. Similarly, those at Phase 4 may do less day-to-day digital work but bring in funding and support teams involved in creating and using software, making them an underutilised resource which could be consulted about policy, standards, application questions, and other sustainability issues which should be more prominent in funder calls, resources made available to prospective/current AHRC project teams, and any supported initiatives.

Among the mid-career phases evaluated in this survey, we noted a lack of confidence in those developing software when it comes to self-reported expertise. This may benefit from further investigation, to see if lack of confidence has an impact on practice, but one way of overcoming this would be to place greater emphasis on interdisciplinary working with RSEs as an option, providing opportunities to network and also practical advice on how to engage in this kind of collaboration. This lack of confidence may also be harming the promotion of creative, innovative work in the humanities, rendering invisible work with the potential to be transformative for other researchers. A commitment to sustaining projects, rewarding and recognising software-based outputs, providing training and opportunities to share learning and network with others to devise new, exciting projects would support the cross-pollination of ideas, reuse of software, and the sharing of expertise.

Networking is an important facet that needs explicit support; the AHRC networking grants are a good way to facilitate this, but ideally this focus on facilitating the creation of new networks and collaborations needs to begin before the point of application, in order to encourage proposals from those who do not already have these existing networks. Late-career researchers have learned their lessons the hard way in terms of what they need to know to do their work, including programming languages/tools, techniques, and infrastructure. Our results show that the majority of those who learned software skills either did so on the job or through personal study. In order to prevent this very individualistic model of DH learning, there should be materials available to those at earlier career stages, support for curriculum development to embed digital skills earlier in the educational journey, and also checkpoints and upskilling for those at different stages of their career. This would increase confidence around skills, offer opportunities for networking, and help share hard-earned expertise, allowing trail-blazers in this area to gain further recognition for their invested time.

The 'Innovation Scholars' programme[26] run by the BBSRC, ESRC and MRC would be an excellent model for AHRC to learn from; the first tranche of projects[27] have already learnt lessons on how to successfully proceed and taking these into a call tailored for the AHRC research area would support innovative upskilling, resource creation and seedcorn Fellowships to aid networking and outreach. This is an opportunity for AHRC to partner with other funders and it would allow the more efficient transmission of knowledge and practice between latter and earlier career phases.

All of these elements need to be made more visible in funders' documentation: although software costs are being added to grant proposals, as the survey results show, there is still an element of 'hidden scholarship' around the development of software. It needs to be more clearly stated that applying for software-related funds is not only legitimate, but encouraged, and exactly what kinds of costs are supported (such as sustainability beyond the funded period, for example). Thinking about academic impact as including dissemination and sharing, not only of software but of best practice with regard to software development and use, would be another way to incorporate best practice thinking into the forms and requirements of funders. A technical peer review college to assess DH bids, and a pool of peer reviewers with specific knowledge in this area, may also be necessary.

[26] Innovation Scholars: Data Science Training in Health and Bioscience

[27] Initiatives boost health and bioscience skills and industry

8.2 INSTITUTIONS

Research-intensive institutions such as universities, Higher Education institutions, and GLAM entities in the UK should enhance and target their provision and training for projects and digital environments, and their support for research software development. As shown in section 5.3.10 Institutional Support for Software, currently there is limited provision for institutional software support in the Arts and Humanities, and this contributes to the misconception that software is not an integral part of the A&H research agenda. Often the provision is focused on website building with a marketing focus. Furthermore, limited investment and support for software development suggests that software is still far from being considered as a valid research output, with academic credit similar to that of a scholarly publication. Institutions thus need to develop and document clear mechanisms for promoting and establishing software as an academic research output of recognised value. This strategy might contain producing criteria for assessing software quality, internal peer review of funding applications that gives cross-departmental opportunities for feedback, developing venues for publishing software reviews as well as support and advocacy networks towards this end.

As noted in the section above, institutional support for AHRC researchers is very uneven across Higher Education in the UK. There are clear and significant gaps. A more generous investment, with strategic planning, is needed to enhance the digital research infrastructure at an institutional level such that it will better support and enhance software use at scale in A&H. What is needed is a robust ecosystem of cloud services, analytics and systems that will enable the A&H to develop and apply research software at scale, in a secure and sustainable way.

Institutional support should focus on training that is advertised in a way that highlights its relevance to humanistic research questions, providing introductory training in core areas as well as opportunities for ongoing upskilling. In order to expand the number of researchers who engage with the training, it is vital to ensure it is made clear how particular techniques can apply to humanities research, as this is often left implicit rather than made explicit in promotion and execution of sessions. This training should start at an earlier stage, i.e. become embedded in the curriculum, and by doing so, institutions would also be providing researchers with networking opportunities that would facilitate more ambitious interdisciplinary, team-based projects as the researchers' careers progress. There is also an opportunity for institutions to do their own mapping of existing software expertise and needs across the board, to facilitate knowledge sharing within and across the institution in a centralised way. A 'train-the-trainer' approach must be adopted, aiming to enhance the existing expertise of more experienced researchers as they communicate it to mid- and early-career stage practitioners. As shown in section 5.3.6 Computing Techniques, software needs highly vary among different career levels and this should be mirrored also in training design and prioritisation. The barriers to this identified by survey participants include time, and that also means time to attend (or deliver) such training; as noted above, this is partly due to the low value placed on software development meaning that it is not seen as a good use of time that is already pressured, but also that it is not often made clear enough how the specific training offered can be adapted to A&H research questions, so it is not clear that such training is a good use of time.

Institutions should also provide in-house research software consultancy, or the opportunity to buy time from a pool of RSEs. This would enable software to be supported throughout its lifecycle, rather than relying on funding to buy upfront developer time to solve a problem without the ability to adapt, reuse, and refocus the software later on. Institutional RSE pools are overwhelmingly seen as the correct model for software support, and should be supported and nurtured; part of this is enabling research software developers, including those in the humanities, to forge a career path around software creation. In this area, AHRC researchers are more neglected than in other domains, and it is crucial to consider how to invest in RSEs who can support humanities research, e.g. by clarifying career progression and developing the confidence of A&H researchers who create software, so that this becomes a viable and attractive career progression, and it thus becomes possible to seek hires from existing communities of practice. There may be a bias among RSEs, since few have a humanities background and may therefore struggle to relate to the goals of humanities partners.

Institutions need to be part of this culture shift in how software is viewed as an output. Increased financial and infrastructural support, a commitment to sustaining software and digital projects, and investment in high-skilled staff and RSEs, would be the first steps to changing the landscape.

8.3 COMMUNITIES OF PRACTICE

The most necessary and significant changes to the A&H landscape in terms of support of software development and use must be led by funders and institutions. However, there are also opportunities for communities of practice to make improvements. By communities of practice, we are referring to research groups (whether formalised/funded or casual), academic societies, and other networks and connections of researchers both within, outside and across institutions. These groups should establish learning opportunities among their community that facilitate a 'trickle down' approach to knowledge sharing; currently, there is a lot of pressure on senior researchers in this space, when our survey results have shown that each career phase can usefully direct and advise the one before, and more junior colleagues may have insights to share into new methods and approaches to those in latter stages. To avoid this additional labour, facilitating networking and sharing of ideas between phases is important, as is peer review of funding bids and research plans.

Those in mid- and senior-phase career stages highlighted an inability to keep up with developments in software, digital projects, and tools. Regular events, both informal showcases and formal workshops targeted to the specific community of practice (whether focused on a historical time period, an author, a method, etc.) could enable more researchers to see the value in the digital. As networks, teams and diverse expertise become more important to later career phases, the most vital work communities of practice can engage in is facilitating networking that can lead to fruitful collaboration, geared towards putting together funding bids that enable collaboration and sharing of knowledge. It would be beneficial for communities of practice to think more broadly than the specific technical skills, encouraging learning on topics including collaboration, project management, and so on.

Where possible, communities of practice should encourage an interest in sustainability, especially among more junior researchers. Where there is financial capacity, offering funding to support and provide a home for software projects which may not have this kind of institutional support, or advertising/creating a directory of relevant projects to facilitate reuse, would be beneficial.

Some of the most important work communities of practice can do is calling on institutions, funders, publishers and the wider community, providing feedback and creating pressure to make it clear what infrastructure and financial support is needed. In this, communities of practice would benefit from making connections with other communities for interdisciplinary knowledge sharing, including with the social sciences, to overcome this barrier of finding out what you do not know. The SSI's fellowship scheme offers a way for researchers to use funding to build and develop these communities, as well as providing access to interdisciplinary networks that can inspire other communities of practice in their relationship to software.

9 BIBLIOGRAPHY

Anderson, K. et al. (2016) 'Student Labour and Training in Digital Humanities', *Digital Humanities Quarterly*, 010(1).

Brown, A. et al. (2020) 'Survey of software use at the University of Southampton'. Research Software Group - University of Southampton. Available at: https://github.com/Southampton-RSG/soton_software_survey_analysis_2019/blob/95263f74ae9cf7a004ee0e5f17c8d58ad86331e8/report/Research%20software%20at%20the%20University%20of%20Southampton.pdf (Accessed: 15 July 2022).

Crouch, S. et al. (2013) 'The Software Sustainability Institute: Changing Research Software Attitudes and Practices', *Computing in Science & Engineering*, 15(6), pp. 74–80. Available at: https://doi.org/10.1109/MCSE.2013.133.

Crymble, A. (2021) *Technology and the Historian: Transformations in the Digital Age*. University of Illinois Press. Available at: https://www.press.uillinois.edu/books/?id=p085697 (Accessed: 28 July 2022).

Duca, D. and Metzler, K. (2019) *The Ecosystem of Technologies for Social Science Research*. SAGE Publishing. Available at: https://doi.org/10.4135/wp191101.

Duca, D. and Metzler, K. (no date) 'The Ecosystem of Technologies for Social Science Research', p. 26.

Gibbs, F. and Owens, T. (2012) 'Building Better Digital Humanities Tools: Toward broader audiences and user-centered designs', *Digital Humanities Quarterly*, 006(2).

Giles Bergel et al. (2020) *Sustaining Digital Humanities in the UK*. Zenodo. Available at: https://doi.org/10.5281/zenodo.4046267.

McGillivray, B. et al. (2020) 'The challenges and prospects of the intersection of humanities and data science: A White Paper from The Alan Turing Institute', p. 638792 Bytes. Available at: https://doi.org/10.6084/M9.FIGSHARE.12732164.

Münster, S. and Terras, M. (2020) 'The visual side of digital humanities: a survey on topics, researchers, and epistemic cultures', *Digital Scholarship in the Humanities*, 35(2), pp. 366–389. Available at: https://doi.org/10.1093/llc/fqz022.

Rampin, R. and Rampin, V. (2021) 'Taguette: open-source qualitative data analysis', *Journal of Open Source Software*, 6(68), p. 3522. Available at: https://doi.org/10.21105/joss.03522.

Schindler, D. et al. (2022) 'The role of software in science: a knowledge graph-based analysis of software mentions in PubMed Central', *PeerJ Computer Science*, 8, p. e835. Available at: https://doi.org/10.7717/peerj-cs.835.

Schreibman, S. and Hanlon, A.M. (2010) 'Determining Value for Digital Humanities Tools: Report on a Survey of Tool Developers', *Digital Humanities Quarterly*, 004(2).

Smithies, J. (2019) 'Research Software (RS) Careers: Generic Learnings from King's Digital Lab, King's College London'. Available at: https://doi.org/10.5281/zenodo.2564790.

Software use in Southampton (2019) *Slides*. Available at: https://slides.com/simonhettrick/software-in-southampton (Accessed: 13 April 2022).

Summit on Digital Tools for the Humanities: Report on Summit Accomplishments (2006). Virginia: University of Virginia. Available at: http://www.iath.virginia.edu/dtsummit/SummitText.pdf (Accessed: 13 April 2022).

Warwick, C. et al. (2006) *The LAIRAH project : log analysis of digital resources in the arts and humanities. Final report to the Arts and Humanities Research Council., Project Report. Arts and Humanities Research Council, Swindon, UK*. Swindon, UK: Arts and Humanities Research Council. Available at: http://www.ahrcict.rdg.ac.uk/activities/strategy_projects/reports/ (Accessed: 13 April 2022).

10 APPENDIX A: SURVEY

This is a document view of the original survey including preamble, questions, hint text and any fixed-choice responses: (the section number and section title have been merged to decrease the level of nesting of headings).

The title of the survey was, 'AHRC Research Community; survey of digital/software requirements'.

10.1 SECTION 1 - PREAMBLE

This survey is being run by the UK Software Sustainability Institute (SSI - https://www.software.ac.uk). SSI champions and support the cause of digital tools and computer-aided methodologies (aka software) in the research process - Better Software, Better Research.

We are working to improve our understanding of the digital practices people undertake or wish they could undertake in the arts, humanities and GLAM (galleries, libraries, archives and libraries) communities so we can tailor our approach to supporting this community, and make recommendations to others working in the field.

The survey asks about your views on digital-tools/software, your experience of developing digital-tools/softwares , and your practices and preferences for recruiting help with digital tool/software development.

This survey's focus is the AHRC Research Community and therefore it has a UK focus; participants should be connected with a UK based institution. We will share the findings with AHRC to help them direct their digital infrastructure funding to better align with the communities' needs. Note, we will not share any Personally Identifiable Information (PII) as defined by GDPR with the AHRC.

We are seeking input from all roles, including senior decision makers, researchers, curators, librarians and software developers in the AHRC remit. We are actively seeking views from people who are involved in software development and those who do not develop digital tools / software alike.

We estimate that the questionnaire will take 15 minutes to complete.

We ask for your email address to allow us to send you a copy of your responses. Your email address is also needed if you agree for us to contact you for a follow up conversation or if you would like to participate in the prize draw (for a chance to win one of four £50 shopping vouchers).

The data collected by this form is covered by the Software Sustainability Institute's Survey's privacy policy - https://www.software.ac.uk/privacy-surveys. This policy is fully compliant with GDPR.

If you have any questions about the survey please contact info@software.ac.uk.

We would like to thank our reviewers who contributed towards improving this survey: Adam Crymble (UCL), Ross Wilson (University of Cambridge), Melodee Beals (Loughborough University), Lisa Otty (University of Edinburgh), Pip Willcox (The National Archives), Jane Winters (University of London), Nicola Osborne (University of Edinburgh).
We would also like to acknowledge Anna-Maria Sichani (University of Sussex) for her useful suggestions. Mentioning them does not necessarily mean an endorsement by them.

Thank you for your help!

10.2 SECTION 2 - ABOUT YOU

10.2.1 Email (mandatory)

10.2.2 confirm that I am 18 or over and I am a researcher who self identifies as part of the broad AHRC research community (mandatory)

10.2.3 Institutional Affiliation (if applicable) (mandatory)

This could be the organisation/company that directly pays your salary or one or more you are affiliated with in some other way.

10.2.4 Research Discipline (mandatory)

Level one codes from the Primary Research Areas covered by AHRC discipline funding remit - https://ahrc.ukri.org/funding/research/subjectcoverage/ahrc-disciplines/. Use the 'Other' option if your discipline is not listed. Please tick all that apply.

N/A - I do not do research

Archaeology

Classics

Cultural and Museum Studies

Dance

Design

Development Studies

Digital Humanities

Divinity and Religion

Drama and Theatre Studies

History

Information and Communication Technologies

Languages and Literature

Law and Legal Studies

Library and Information Studies

Linguistics

Media

Music and Visual Arts

Philosophy

Political Science and International Studies

Theology

Other:

10.2.5 Which of the following organisations have you sought funds for your research from?

Please tick all that apply.

AHRC

BBSRC

EPSRC

ESRC

MRC

NERC

STFC

Innovate UK

Andrew Mellon Foundation

Arts Council England

British Academy

Creative Scotland

European Research Council

Heritage Lottery Fund

Humanities in the European Research Area

Leverhulme Foundation

My institution

Charitable/Philanthropic society

Scholarly Societies (e.g. Royal Historical Society, Past & Present Society)

Wellcome Trust+

Wolfson Foundation

My research/work does not normally seek external funding

Other:

10.2.6 What is your job/role title?

10.2.7 What is your career stage?

Please take a look at https://www.leru.org/files/UK-England.pdf and (by analogy) choose the appropriate phase that matches your career stage if the choices below are not clear.

Phase 1 - Junior (e.g. PhD candidate, Junior Research Software Engineer)

Phase 2 - Early (e.g. Research Assistant/Associate, first grant holder, Lecturer, Research Software Engineer)

Phase 3 - Mid / Recognised (e.g. Senior Lecturer, Reader, Senior Researcher, Senior Research Software Engineer, Research Software Group Leader)

Phase 4 - Established / Experienced / Senior (e.g. Professor, Director of Research Computing, Distinguished Engineer, Chief Data Scientist)

10.3 Section 3 - Research software and you

10.3.1 Do you use research software? (mandatory)

'Research software' is any software or digital tool that you have used in the course of your research that has helped you produce a research output (e.g. a publication). This might be anything from a short script, such as one written in Python or R, to help you clean your data, web/mobile apps, to a fully-fledged software suite or specialised toolset, whether you access this online or run it on your own computer. It includes code that you have written yourself and code written by someone else, either specifically for your project or a general tool for data, text or statistical analysis. It does not include common software applications used to prepare research publications, such as word processors (e.g. Word, Pages, LibreOffice). It also does not include online databases (e.g. Literature Online and Eighteenth-Century Collections Online). However, the use and/or construction of spreadsheets that perform calculations or transformation automatically according to a set of pre-programmed rules, are considered to be software.

Yes

No

10.3.2 How important is research software to your work? (mandatory)

Not at all

1

2

3

4

5

Vital

10.3.3 What do you use research software for?

Please provide details of what research-related task you use software to help you with.

10.3.4 Why do you use the research software that you do?

This may be due, for example, to a collaboration you are a part of, or because you or your collaborators wrote it or because of the software's reputation, availability (free and/or open source), support community and likelihood of being fit for your purpose.

10.3.5 How could better software practices and solutions improve your area or field of work?

10.3.6 Have you developed your own research software?

This includes anything from a script to automate some tasks (e.g. data cleaning) to writing bespoke tools to contributing to larger pieces of software.

Yes if this is chosen then go to Section 4

No if this is chosen then go to Section 5 (i.e. skip Section 4)

10.3.6.1 What research software have you developed?

If you answered 'Yes' to the previous question - you can provide further details here (we are interested in the top one to three pieces of software that you have developed).

10.4 Section 4 - Your software development

10.4.1 How do you rate your software development expertise?

Beginner

1

2

3

4

5

Professional

10.4.2 Where have you learnt your software skills

Tick all that apply

On the job (i.e. time allocated in the role to learn)

In my own time (e.g. books, YouTube)

Part time short courses (e.g. Software Carpentry, Programming Historian, MOOCS)

Part time long courses (e.g. part time degree or masters)

Other:

10.4.3 Do you feel that you have received sufficient training to develop reliable software? (mandatory)

If this question is not relevant to you personally (e.g. you are a project leader who does not write software) then please use the 'Other' option to let us know your reason.

Yes

No

Other:

10.4.4 What types of information/data do you or your team work with?

Tick all that apply

Textual

Numerical

Image

Video

Audio

Other:

10.4.5 What computing techniques are you interested in exploring for your research? (mandatory)

Computer Vision - (choose one from: Currently using, Interested in using, Interested in learning more, No plans to use or learn, Not aware of this)

Data Cleaning - (choose one from: Currently using, Interested in using, Interested in learning more, No plans to use or learn, Not aware of this)

Data Visualisation (including 3D) - (choose one from: Currently using, Interested in using, Interested in learning more, No plans to use or learn, Not aware of this)

Database Design - (choose one from: Currently using, Interested in using, Interested in learning more, No plans to use or learn, Not aware of this)

Immersive technologies - Virtual Reality (VR) / Augmented Reality (AR) / Mixed Reality (MR) - (choose one from: Currently using, Interested in using, Interested in learning more, No plans to use or learn, Not aware of this)

Machine Learning / Artificial Intelligence / Generative Adversarial Networks - (choose one from: Currently using, Interested in using, Interested in learning more, No plans to use or learn, Not aware of this)

Natural Language Processing (NLP)/ Textual Analysis / Sentiment Analysis - (choose one from: Currently using, Interested in using, Interested in learning more, No plans to use or learn, Not aware of this)

Network Analysis - (choose one from: Currently using, Interested in using, Interested in learning more, No plans to use or learn, Not aware of this)

Probabilistic Linking - (choose one from: Currently using, Interested in using, Interested in learning more, No plans to use or learn, Not aware of this)

Statistics - (choose one from: Currently using, Interested in using, Interested in learning more, No plans to use or learn, Not aware of this)

Text/Data Mining - (choose one from: Currently using, Interested in using, Interested in learning more, No plans to use or learn, Not aware of this)

Topic Modelling - (choose one from: Currently using, Interested in using, Interested in learning more, No plans to use or learn, Not aware of this)

Working with Geospatial data - (choose one from: Currently using, Interested in using, Interested in learning more, No plans to use or learn, Not aware of this)

10.4.6 What computing production skills are you interested in exploring for your research? (mandatory)

Agile development - (choose one from: Currently using, Interested in using, Interested in learning more, No plans to use or learn, Not aware of this)

Application Programming Interfaces - (choose one from: Currently using, Interested in using, Interested in learning more, No plans to use or learn, Not aware of this)

Continuous integration - (choose one from: Currently using, Interested in using, Interested in learning more, No plans to use or learn, Not aware of this)

Cross Platform development - (choose one from: Currently using, Interested in using, Interested in learning more, No plans to use or learn, Not aware of this)

Effective skills for working with software developers (e.g. specifying/managing the brief) - (choose one from: Currently using, Interested in using, Interested in learning more, No plans to use or learn, Not aware of this)

Information Architecture - (choose one from: Currently using, Interested in using, Interested in learning more, No plans to use or learn, Not aware of this)

Middleware Services / Microservices - (choose one from: Currently using, Interested in using, Interested in learning more, No plans to use or learn, Not aware of this)

Mobile apps development - (choose one from: Currently using, Interested in using, Interested in learning more, No plans to use or learn, Not aware of this)

Software & data ethics (e.g. bias and implications) - (choose one from: Currently using, Interested in using, Interested in learning more, No plans to use or learn, Not aware of this)

Unit testing - (choose one from: Currently using, Interested in using, Interested in learning more, No plans to use or learn, Not aware of this)

User Experience - (choose one from: Currently using, Interested in using, Interested in learning more, No plans to use or learn, Not aware of this)

User Testing - (choose one from: Currently using, Interested in using, Interested in learning more, No plans to use or learn, Not aware of this)

Version control - (choose one from: Currently using, Interested in using, Interested in learning more, No plans to use or learn, Not aware of this)

Web application development - (choose one from: Currently using, Interested in using, Interested in learning more, No plans to use or learn, Not aware of this)

10.4.7 Are there any other computing techniques or production skills that you are interested in exploring for your research?

10.4.8 Which statement below best matches the scale of your computing needs?

They are served by my desktop/laptop

I use High Performance Computing (HPC) / Cloud Computing

I am transitioning from my desktop/laptop to HPC / Cloud Computing

Other:

10.4.9 How would you rate the current level of support for your software-development needs from your institution?

SSI recommends institutions provide training, research software consultancy and the ability to buy time from a pool of Research Software Engineers to ensure research software is supported throughout its lifecycle and the people who develop it are able to forge career paths.

Poor

1

2

3

4

5

Excellent

10.5 Section 5 - Recruitment of software developers

10.5.0.1 Have you or someone in your group ever hired someone specifically to develop software? (mandatory)

Yes

No

10.5.0.2 Have you ever included costs for software development in a funding proposal? (mandatory)

Please tick all that apply. This includes the cost of hiring a person or paying for someone's time to spend some or all of their time on developing research software.

Yes

No (I'm not involved in writing funding proposals)

No (but we DID expect to write software as part of the project)

No (we DID NOT expect to write software as part of the project)

10.5.1 How suitable would the following models be for your software development needs?

Research Software Engineering pools of staff are a relatively new phenomenon at institutions, but not all institutions have them yet. To find out more about Research Software Engineering visit - https://society-rse.org.

Hire a full-time software developer (Choose one: Unsuitable, Suitable, Perfect)

Recruit developer time from an institutional Research Software Engineering pool (Choose one: Unsuitable, Suitable, Perfect)

Outsource software development (Choose one: Unsuitable, Suitable, Perfect)

10.6 Section 6 - Follow up and prize draw

10.6.1 Can we contact you for a follow up conversation?

For example, we are interested in running focus groups around data and software loss in AHRC based research areas and may contact you to take part.

Yes

No

10.6.2 Name

If you answered 'Yes' to the previous question, providing your name would make it easier for us when we contact you.

10.6.3 Do you wish to enter the prize draw for one of four £50 shopping vouchers?

Yes

No

END OF SURVEY

A copy of the data collected can be found at 10.5281/zenodo.7594343. Personally identifiable information (e.g. name, email address) and other information (e.g. job role, institution) which could easily lead to deductive disclosure have been removed.

Software
Sustainability
Institute

www.software.ac.uk