FAIR^2 for research software:
Developing a FAIR and Reproducible research software training programme

Tamora James

Research Software Engineering, University of Sheffield

ACIT Hub Co-design workshop, 2025-07-03

University of Sheffield

FAIR Principles for Research Software

- **Findable** Software, and its associated metadata, is easy for both humans and machines to find
- **Accessible** Software, and its metadata, is retrievable via standardised protocols
- **Interoperable** Software interoperates with other software by exchanging data and/or metadata, and/or through interaction via application programming interfaces (APIs), described through standards
- **Reusable** Software is both usable (can be executed) and reusable (can be understood, modified, built upon, or incorporated into other software)

FAIR Principles for Research Software

*"The FAIR4RS Principles are relevant to any stakeholder in the research community seeking to increase **transparency**, **reproducibility**, and **reusability** of research."*

*"open source software ideally should start working toward satisfying the FAIR4RS Principles **when it is initially being developed**"*

Introducing the FAIR Principles for research software, Barker et al. (2022)

https://doi.org/10.1038/s41597-022-01710-x

Towards adoption of FAIR for Research Software

Training is a key component of successful adoption of FAIR4RS:

- *Development of programmes offering FAIR4RS skills training*
- *Tools for assessing FAIRness of research software*
- *Examples of implementation of the FAIR4RS principles*
- *Strategic approaches for supporting FAIR research software*

The FAIR for Research Software Principles after two years: An adoption update, Barker et al. (2024)

https://doi.org/10.5281/zenodo.10816032

FAIR and reproducible research software

- Reproducibility is not part of the FAIR4RS principles, but it is an important tenet of open science
- Reproducibility of research software supports transparency and robustness of the research process

UoS statement on Open Research

“We aspire to open research culture that values a diverse range of contributions and ***adheres to the FAIR principles*** *to enable the results of our research to be of maximum benefit to society (findable, accessible, interoperable and reusable), whilst also respecting circumstances that limit data sharing (for example, due to issues of privacy, non-consent, contractual agreements, legislation or practicality).”*

University of Sheffield, Statement on Open Research

https://www.sheffield.ac.uk/openresearch/university-statement-open-research

RSE team goals

"Enhance the University's capacity for producing ***high quality, efficient and sustainable research software****"*

"Act as the hub of a university community that supports ***good practice in research software****"*

Development of FAIR^{2}4RS skills training

Aim

Develop a training programme to provide the University of Sheffield research community with the knowledge and tools they need to create **FAIR** and **R**eproducible research software

Target audience

Researchers across the university, including postgraduate research students, who create code as part of their research

Who was involved?

Research Software Engineering

The Research Software Engineering team is composed of 11 engineers and supports research across the University in building research software. Areas of expertise within the group include: code optimisation and performance, reproducibility and embedding good software engineering practice with a project or team, developing computational workflows, GPU computing and Deep Learning, data science/visualisation, High Performance Computing, general software development, development of impact case studies, consultancy, training and education delivery and support.

Data Analytics Service

The Data Analytics Service (IT Services) supports research excellence at the University of Sheffield by bridging technical and analytical gaps through consultation, delivering training, and long-term collaboration with research teams. We support researchers with reproducible data analysis, data visualisation, data engineering, machine learning, statistics, big data, research software, web design, and much more.

University of Sheffield Library Team

The Library's Scholarly Communications Team provides specialist services to support researchers at the University of Sheffield. We offer guidance on making research outputs open access, and give support on good practice in research data management, copyright and licensing as well as open research more broadly.

Programme design

- Identify key topics
- Review existing materials
- Design modular programme to meet individual needs
- Aim for a consistent look and feel
- Different components may use different delivery formats

Programme design

- **Overview**
  - Motivation
- **Software lifecycle planning**
  - Scope
  - Lifespan
  - Licensing
  - DOIs / metadata
- **Coding best practice**
  - Code design
  - Version control
  - Testing / CI
  - Documentation
- **Tools for reproducibility**
  - Reproducible environments
  - Packaging
- **Publishing/dissemination**
  - Software papers
- **Summary**
  - End-to-end demo

FAIR² FOR RESEARCH SOFTWARE

University of Sheffield

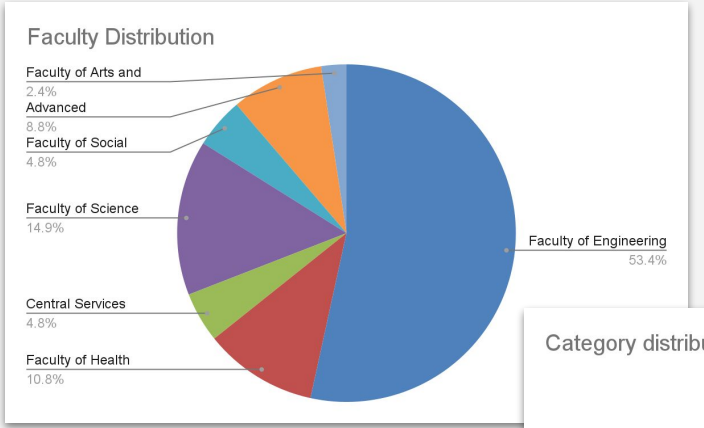
Implementation and delivery

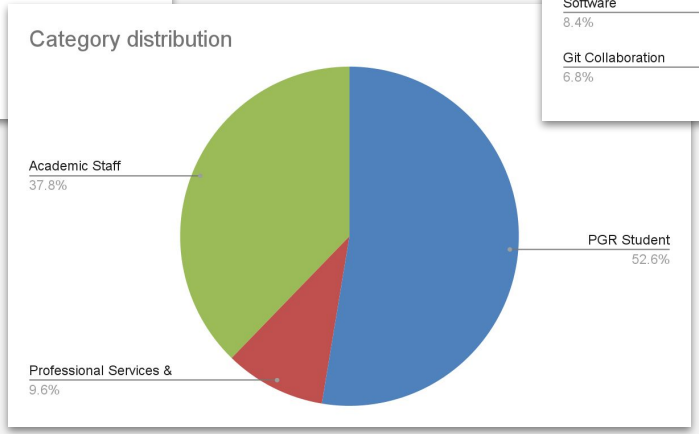
- Use of lesson template for consistent look and feel
  - Based on The Carpentries lesson template
  - Modified to apply UoS branding and ensure consistency
    https://github.com/RSE-Sheffield/fair4rs-lesson-setup
- Programme schedule designed to account for prerequisites
- Format for individual components varied
  - Introductory talk
  - Hands on skills training
  - Walkthrough / live demonstration
- Course materials published under creative commons license
  - https://github.com/FAIR2-for-research-software

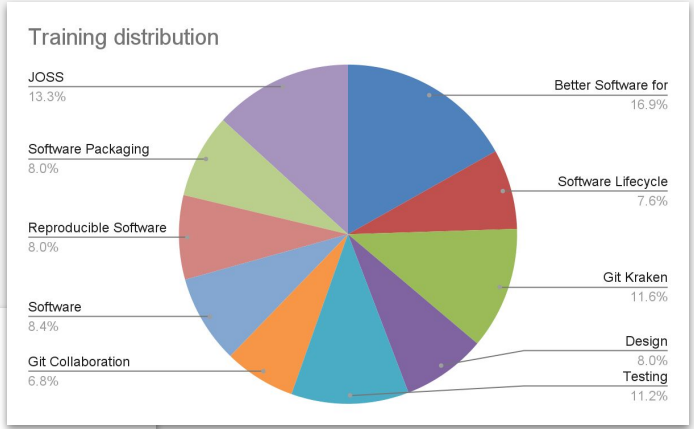
Impact

- Over **250 registrations** and **190 attendees** across the University of Sheffield
- Over **45 hours of training** provided
- Course materials developed as **open educational resources** under Creative Commons licensing for ongoing use
- Feedback indicates that training is **timely** and **relevant to researchers' needs**
    - *"[The instructor] delivered the content in an easy-to-digest manner, allowing for time to put concepts into practice. In addition, it is clear that the instructor knew where to draw the line in terms of not diving into unnecessary detail."*
    - *"The willingness of the course leaders to help me and to make sure I understood everything as we progressed through the training."*
    - *"I appreciated that time was set aside to put the learning into practice."*

Impact


Faculty Distribution
Faculty of Arts and
2.4%
Advanced
8.8%
Faculty of Social
4.8%
Faculty of Science
14.9%
Central Services
4.8%
Faculty of Health
10.8%
Faculty of Engineering
53.4%


Category distribution
Academic Staff
37.8%
Professional Services &
9.6%
PGR Student
52.6%


Training distribution
JOSS
13.3%
Software Packaging
8.0%
Reproducible Software
8.0%
Software
8.4%
Git Collaboration
6.8%
Better Software for
16.9%
Software Lifecycle
7.6%
Git Kraken
11.6%
Design
8.0%
Testing
11.2%

Challenges

- In-person vs hybrid delivery
  - In-person delivery chosen for enhanced interaction between trainer and trainee and to reduce the overhead for the instructors of providing hybrid delivery
  - Excluded some participants who wanted to join online
- Use of Python for example code
  - Python chosen for hands-on example code since familiar to the scientific research community
  - Using Python may have disadvantaged participants who do not know this language
- Time constraints
  - Challenge of covering material in sufficient depth in the time available
- Reaching target audiences
  - No single channel of communication; aimed to reach PGRs, ECRs, Open Research community
  - Registration statistics showed good uptake across the University

Learning

- Collaborating across different teams strengthened the quality of the offering
  - Feedback highlighted knowledgeability of instructors and clarity of course materials and delivery
- Coordination and communication amongst the team members helped to ensure smooth running of the programme
  - Facilitated by a dedicated programme manager and regular meetings during design phase
- Not everything went to plan and instructors had to be flexible
  - Room booking challenges, changes to software packages, participants requiring extra support
- Feedback from both participants and instructors helpful to reflect on and refine the programme for subsequent delivery
  - Need to adjust timings and content covered for some components

Next steps

- Preparing for second delivery in 2025/26 academic year
- Coordinating with AMRC Centre for Doctoral Training for combined delivery of core modules
- Assigning DOIs and publishing via Journal of Open Source Education

University of Sheffield | Research Software Engineering

FAIR² FOR RESEARCH SOFTWARE

New training programme starting **October 2024**

Learn to make your research code **FAIR** and **Reproducible**

- Software lifecycle planning
- Coding best practices
- Tools for reproducibility
- Publishing research software

SCAN FOR DETAILS