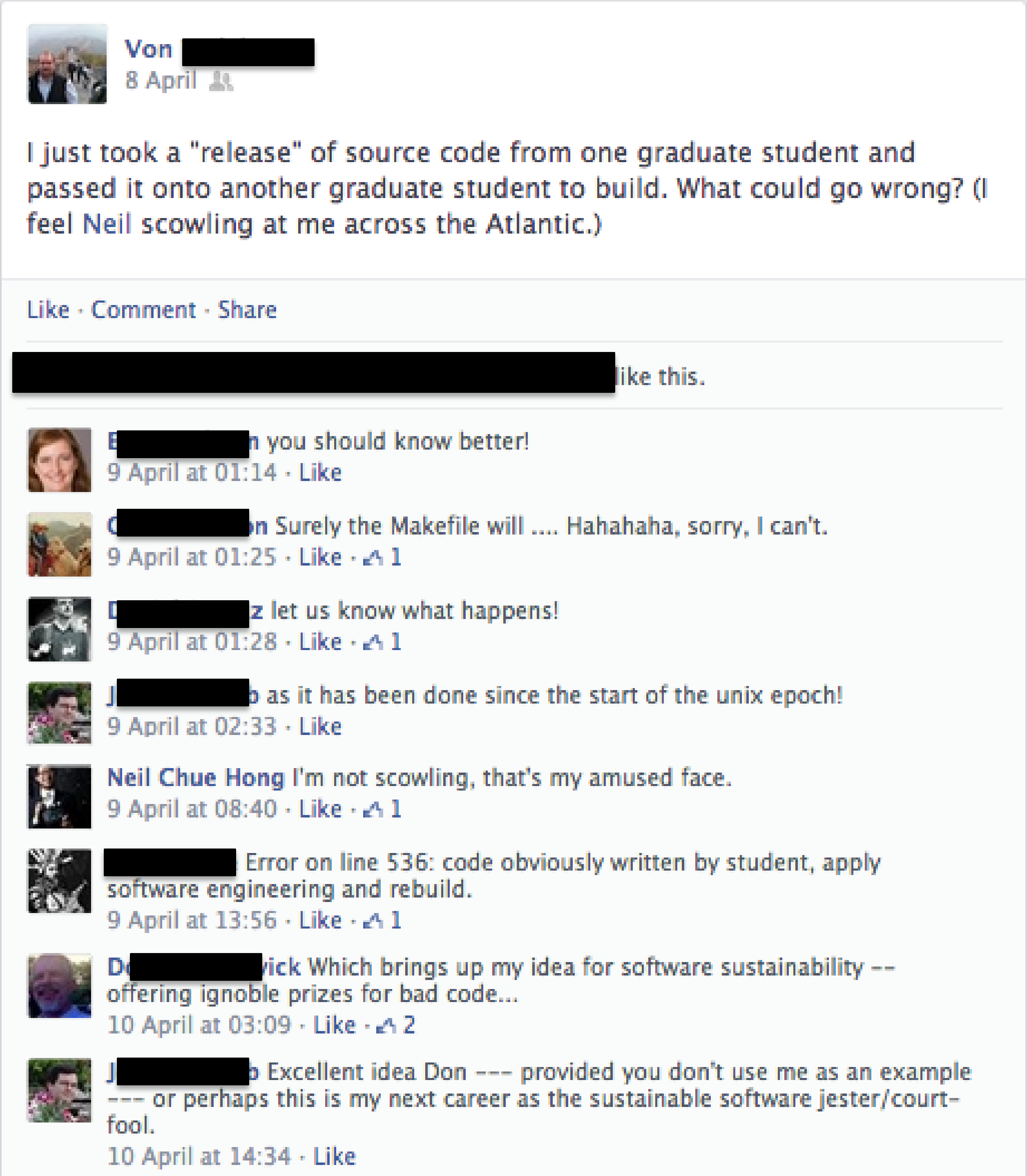
www.software.ac.uk

The Software Sustainability Institute

20 January 2015, HEP Software Foundation workshop
Neil Chue Hong (@npch), Software Sustainability Institute
ORCID: 0000-0002-8876-7606 | N.ChueHong@software.ac.uk

Supported by EPSRC

Project funding from Jisc BBSRC NERC



Von [redacted]
8 April

I just took a "release" of source code from one graduate student and passed it onto another graduate student to build. What could go wrong? (I feel Neil scowling at me across the Atlantic.)

[redacted] like this.

[redacted] you should know better!
9 April at 01:14 · Like

[redacted] Surely the Makefile will .... Hahahaha, sorry, I can't.
9 April at 01:25 · Like · 👍 1

[redacted] let us know what happens!
9 April at 01:28 · Like · 👍 1

[redacted] as it has been done since the start of the unix epoch!
9 April at 02:33 · Like

Neil Chue Hong I'm not scowling, that's my amused face.
9 April at 08:40 · Like · 👍 1

[redacted] Error on line 536: code obviously written by student, apply software engineering and rebuild.
9 April at 13:56 · Like · 👍 1

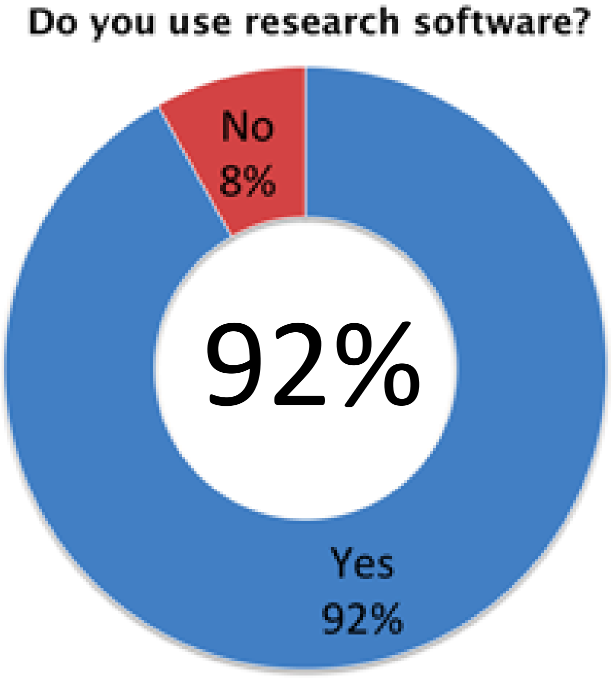
[redacted] Which brings up my idea for software sustainability -- offering ignoble prizes for bad code...
10 April at 03:09 · Like · 👍 2

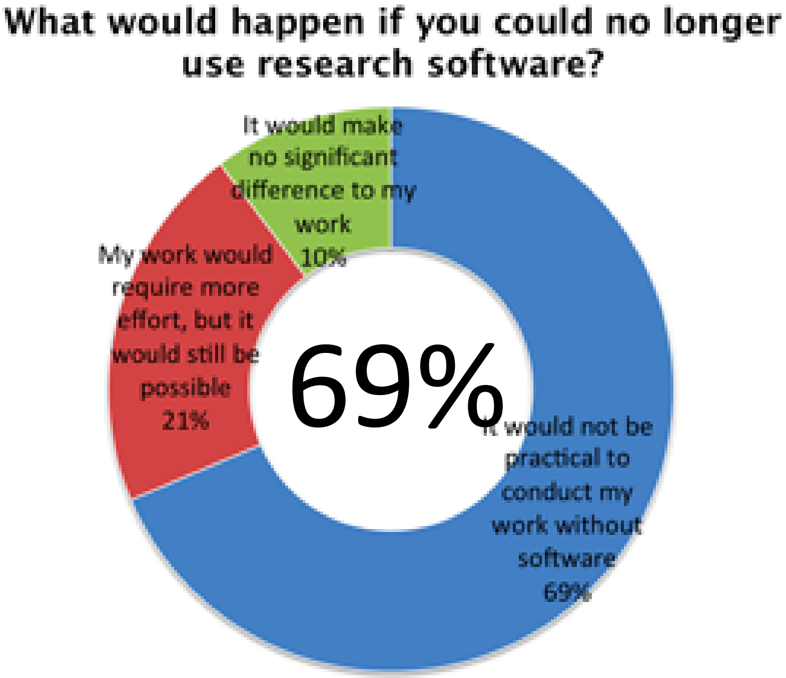
[redacted] Excellent idea Don --- provided you don't use me as an example --- or perhaps this is my next career as the sustainable software jester/court-fool.
10 April at 14:34 · Like

- Of course, we don't deliberately set out to create code that can't be used by others

  Right?

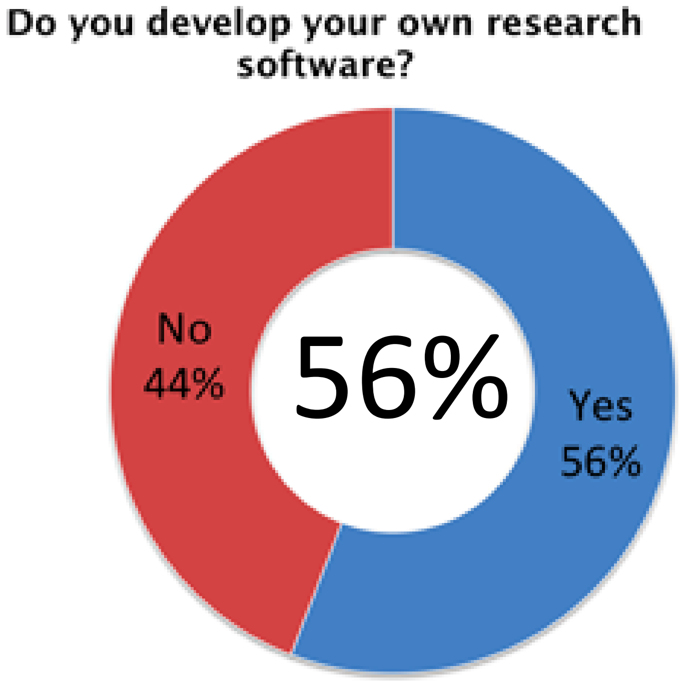
Software isn't special, it's mainstream

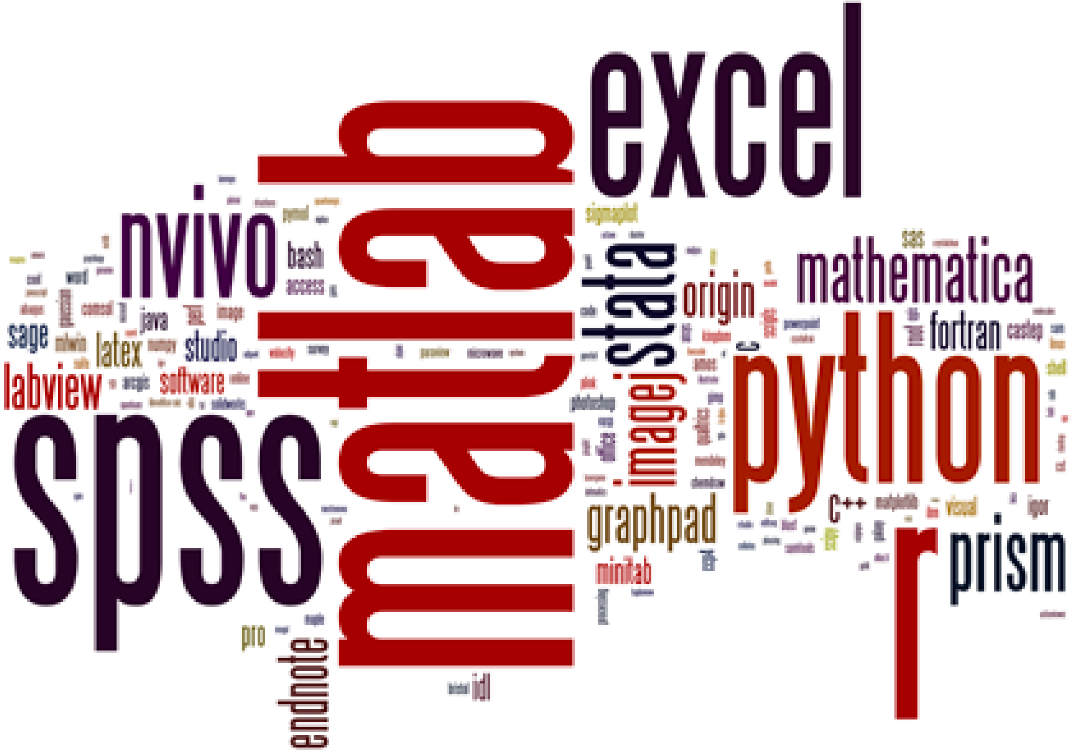




Survey of researchers from 15 Russell Group unis conducted by SSI between Aug- Oct 2014.
406 respondents covering representative range of funders, discipline and seniority.

And it isn't just using software, it's researchers developing software too





Survey of researchers from 15 Russell Group unis conducted by SSI between Aug- Oct 2014. 406 respondents covering representative range of funders, discipline and seniority.

So what’s the issue?

30%	Of UK research investment has been spent on research which relies on software
71%	Of UK researchers have had no formal software development training
77%	Of PIs had not included costs for software development in bids
4%	Of jobs advertised in UK universities were software related

… and then there are gender related issues

Survey of researchers from 15 Russell Group unis conducted by SSI between Aug - Oct 2014. 406 respondents covering representative range of funders, discipline and seniority. Analysis of data from 49,650 grant titles and abstracts published on Gateway to Research covering 2010-2014. Analysis of job adverts posted to jobs.ac.uk in 1H2014.

Just the Nature of the problem?

are?

```
C:\lab>
f77 -o
data.exe
>
>
...ERROR

...why scientific programming does not
compute

>
```

Published online 13 October 2010 | Nature 467, 775-777 (2010)
doi:10.1038/467775a

The modern researcher…

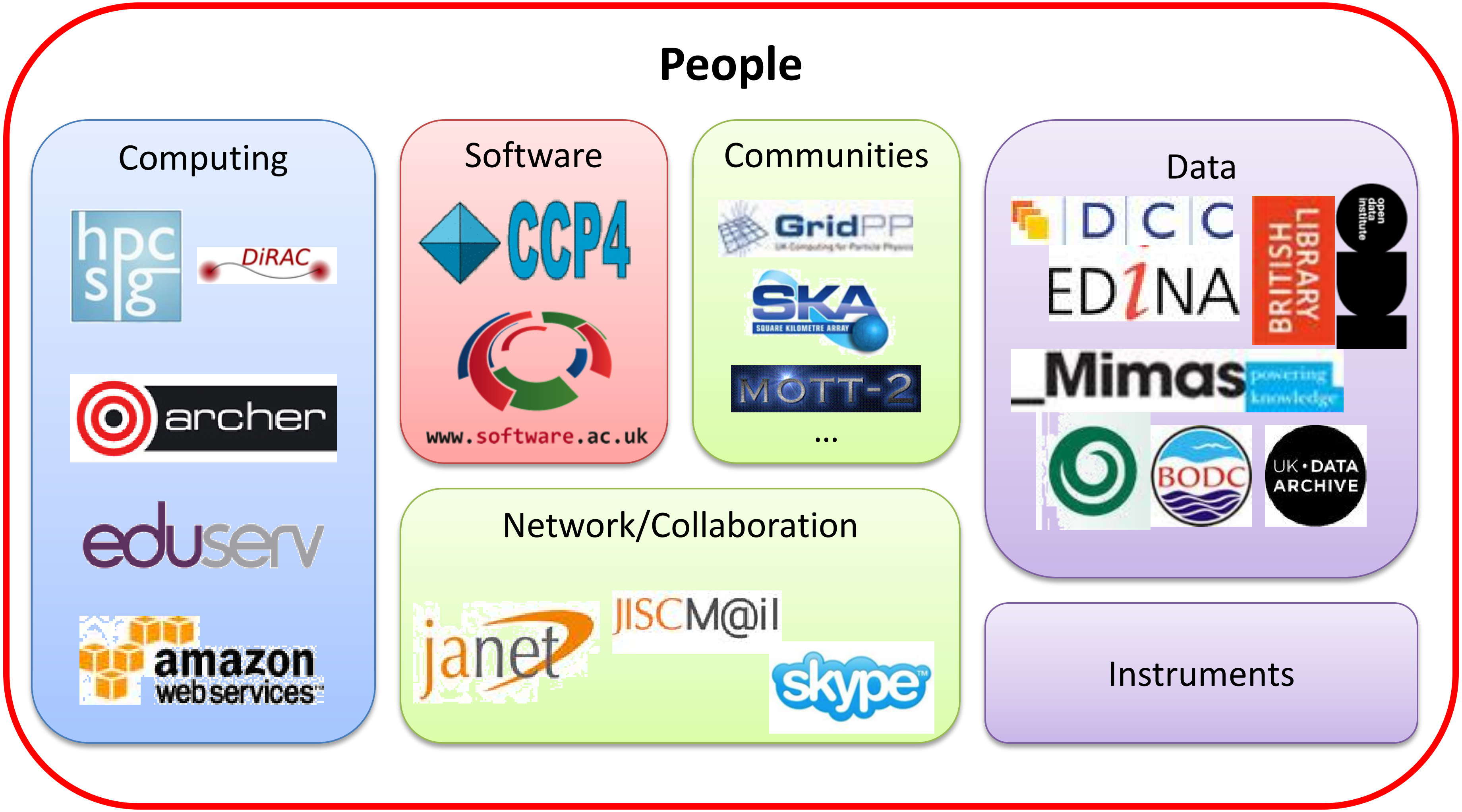
Where do they learn how to do this?

Picture of Otto Stern courtesy of Emilio Segre Visual Archives

- … worries about:
  - Data management and analysis
  - Reproducible research
  - Scalable simulations
  - Integration of models and workflows
  - Collaboration

UK Research Computing Ecosystem

www.software.ac.uk



People

Computing

Software

Communities

...

Data

Network/Collaboration

Instruments

SSI Drivers and Themes

- Two key drivers which cause people to seek the SSI’s advice:
  - They want to be more productive in their research
  - They don’t want to be embarrassed by appearing worse than their peers
- Broadly, our work falls into a few key themes:
  - Developing the scientific computing / software development skill base
  - The role and reward of software in research
  - Recognition of software career paths
  - Re[peatable|producible|computable] research

The Software Sustainability Institute

www.software.ac.uk

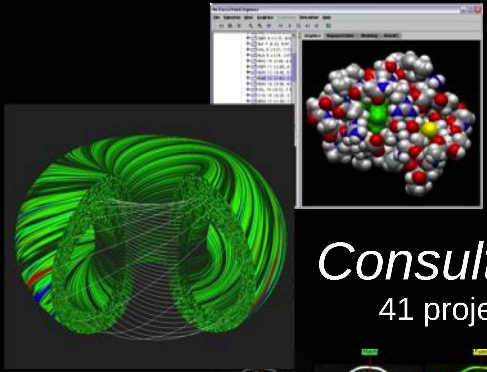
EPSRC
Pioneering research and skills

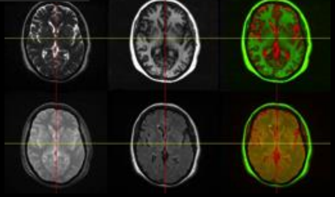
A national facility for cultivating world-class research through software

- Better software enables better research
- Software reaches boundaries in its development cycle that prevent improvement, growth and adoption
- Providing the expertise and services needed to negotiate to the next stage
- Developing the policy and tools to support the community developing and using research software

Supported by EPSRC
Grant EP/H043160/1

Software





Consultancy

41 projects

Advice

92 evaluations
4 surgeries

Training

Courses

33 UK SWC
workshops
1000+ learners

Guides

50,000 readers

Congratulations! You've inherited some code

Digital preservation and curation

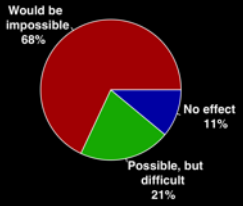
Communication

Website & blog

150+ contributed articles
19,000 unique visitors per month

Research

740 researchers
50,000 grants
analysed



Campaigns

You would do well in the field of computer technology.

272 RSEs engaged

BETTER SOFTWARE BETTER RESEARCH

1700 signatures

13 issues highlighted

Policy

Workshops

20+ workshops organised

Fellowship

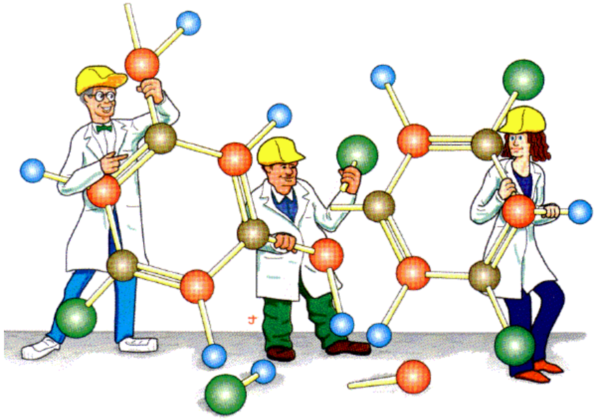
41 domain
ambassadors

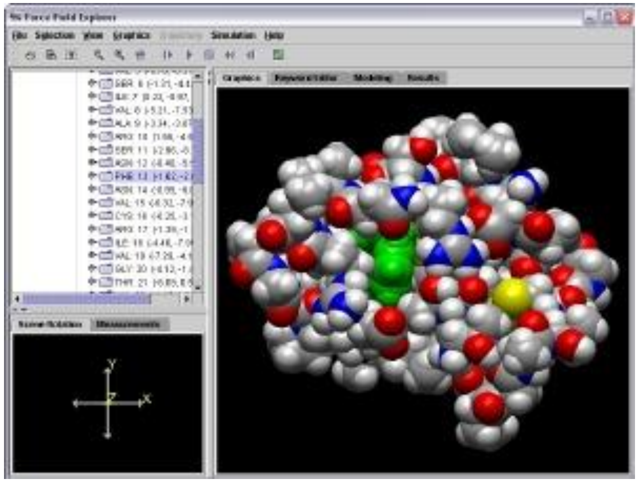
Community

Case Study: Tinkering with APES

www.software.ac.uk

- NSF/EPSRC Funded
  - Polarisable empirical force fields (AMOEBA)
  - AMBER, Tinker, DL_POLY, ONETEP, Q-Chem
- SSI’s work is helping to coordinate development across many packages and many people
  - Different languages, licenses, styles, teams




Force Field Explorer
File Selection View Graphics Simulation Help
Graphics Keyword Editor Modeling Results
Y
X
Z

- http://apes-soft.github.io/

Farah Ahmed

Mark Basham

Jane Charlesworth

Tom Crick

Stuart Dunn

Stephen Eglen

Michael Fischer

Liberty Foreman

Philip Fowler

Laurent Gatto

Robyn Grant

Derek Groen

Alexander Konovalov

Alexandra Simperler

Leanne Mary Wake

Yannick Wurm

www.software.ac.uk

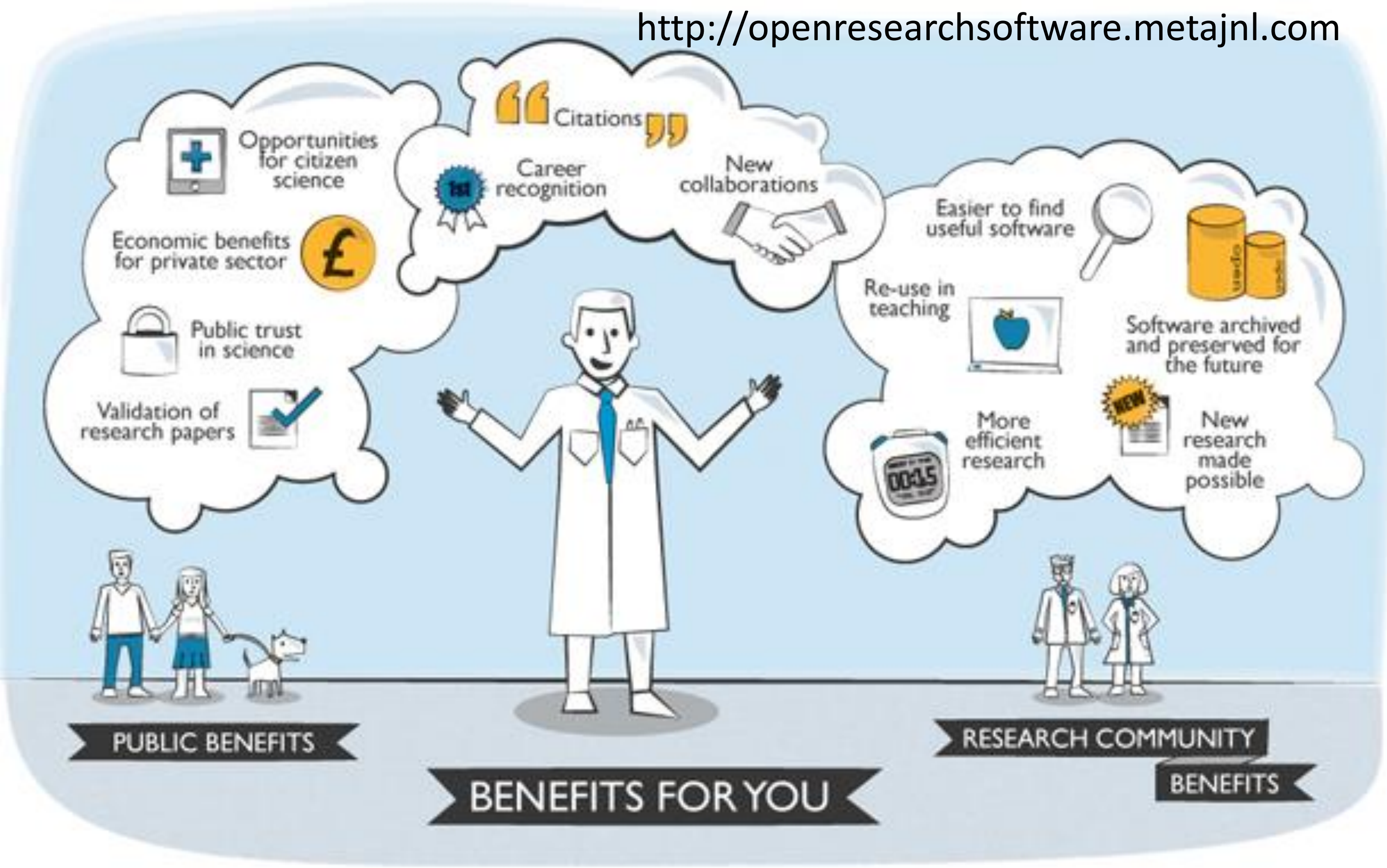
- 2014: 16 fellows
- 2013: 15 fellows
- 2012: 10 fellows
- Range of subjects, career stages

software.ac.uk/fellows

Journal of Open Research Software

www.software.ac.uk

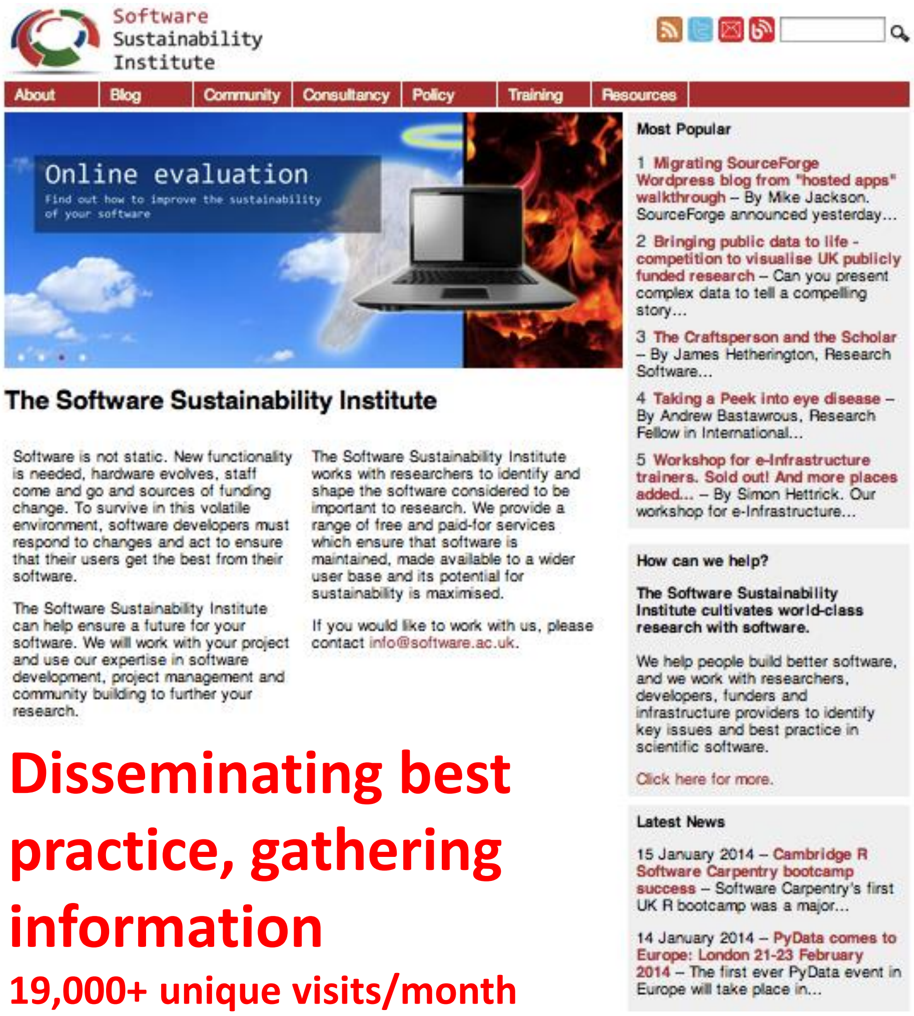
http://openresearchsoftware.metajnl.com


Opportunities for citizen science
Economic benefits for private sector
£
Public trust in science
Validation of research papers
Citations
1st
Career recognition
New collaborations
Easier to find useful software
Re-use in teaching
Software archived and preserved for the future
More efficient research
00:15
NEW
New research made possible
PUBLIC BENEFITS
BENEFITS FOR YOU
RESEARCH COMMUNITY BENEFITS

Software Sustainability Institute

SSI Website

www.software.ac.uk



Software Sustainability Institute

About | Blog | Community | Consultancy | Policy | Training | Resources

Online evaluation

Find out how to improve the sustainability of your software

The Software Sustainability Institute

Software is not static. New functionality is needed, hardware evolves, staff come and go and sources of funding change. To survive in this volatile environment, software developers must respond to changes and act to ensure that their users get the best from their software.

The Software Sustainability Institute can help ensure a future for your software. We will work with your project and use our expertise in software development, project management and community building to further your research.

The Software Sustainability Institute works with researchers to identify and shape the software considered to be important to research. We provide a range of free and paid-for services which ensure that software is maintained, made available to a wider user base and its potential for sustainability is maximised.

If you would like to work with us, please contact info@software.ac.uk.

Most Popular

1 Migrating SourceForge Wordpress blog from "hosted apps" walkthrough – By Mike Jackson. SourceForge announced yesterday…

2 Bringing public data to life - competition to visualise UK publicly funded research – Can you present complex data to tell a compelling story…

3 The Craftsperson and the Scholar – By James Hetherington, Research Software…

4 Taking a Peek into eye disease – By Andrew Bastawrous, Research Fellow in International…

5 Workshop for e-Infrastructure trainers. Sold out! And more places added… – By Simon Hettrick. Our workshop for e-Infrastructure…

How can we help?

The Software Sustainability Institute cultivates world-class research with software.

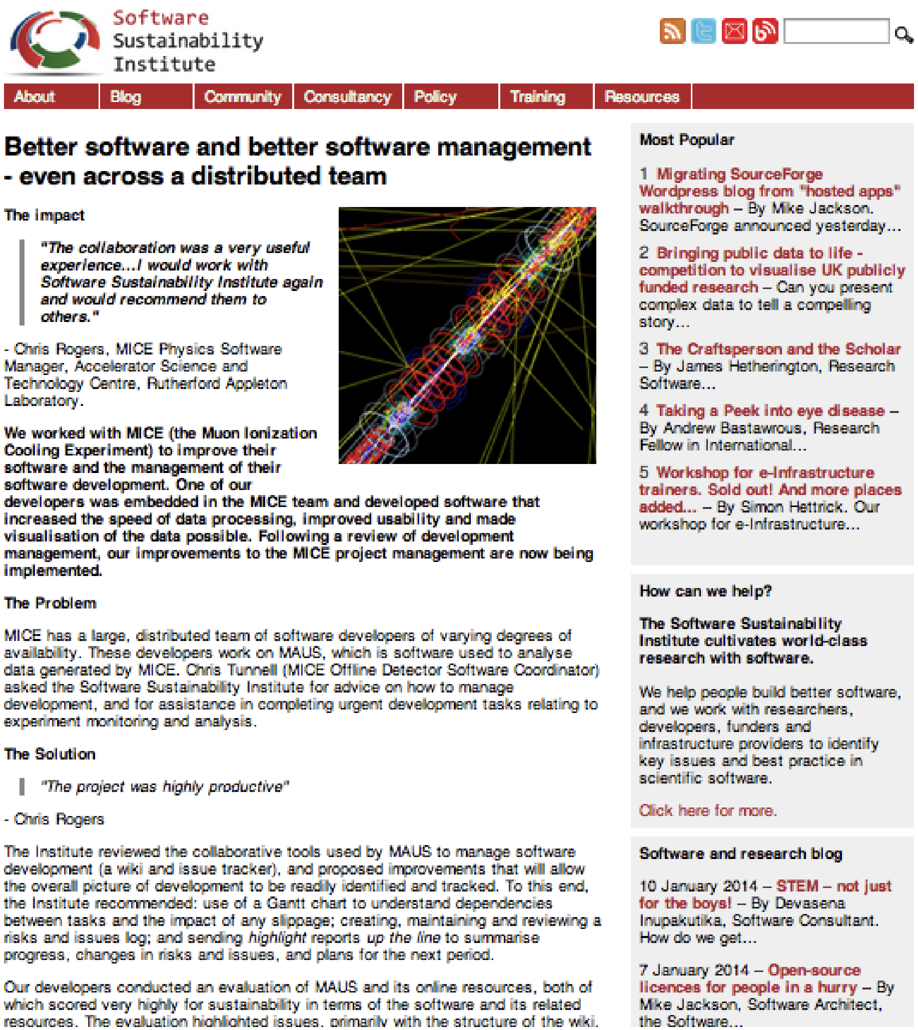
We help people build better software, and we work with researchers, developers, funders and infrastructure providers to identify key issues and best practice in scientific software.

Click here for more.

Latest News

15 January 2014 – Cambridge R Software Carpentry bootcamp success – Software Carpentry's first UK R bootcamp was a major…

14 January 2014 – PyData comes to Europe: London 21-23 February 2014 – The first ever PyData event in Europe will take place in…



Software Sustainability Institute

About | Blog | Community | Consultancy | Policy | Training | Resources

Better software and better software management - even across a distributed team

The impact

> *"The collaboration was a very useful experience…I would work with Software Sustainability Institute again and would recommend them to others."*

- Chris Rogers, MICE Physics Software Manager, Accelerator Science and Technology Centre, Rutherford Appleton Laboratory.

We worked with MICE (the Muon Ionization Cooling Experiment) to improve their software and the management of their software development. One of our developers was embedded in the MICE team and developed software that increased the speed of data processing, improved usability and made visualisation of the data possible. Following a review of development management, our improvements to the MICE project management are now being implemented.

The Problem

MICE has a large, distributed team of software developers of varying degrees of availability. These developers work on MAUS, which is software used to analyse data generated by MICE. Chris Tunnell (MICE Offline Detector Software Coordinator) asked the Software Sustainability Institute for advice on how to manage development, and for assistance in completing urgent development tasks relating to experiment monitoring and analysis.

The Solution

> *"The project was highly productive"*

- Chris Rogers

The Institute reviewed the collaborative tools used by MAUS to manage software development (a wiki and issue tracker), and proposed improvements that will allow the overall picture of development to be readily identified and tracked. To this end, the Institute recommended: use of a Gantt chart to understand dependencies between tasks and the impact of any slippage; creating, maintaining and reviewing a risks and issues log; and sending *highlight* reports *up the line* to summarise progress, changes in risks and issues, and plans for the next period.

Our developers conducted an evaluation of MAUS and its online resources, both of which scored very highly for sustainability in terms of the software and its related resources. The evaluation highlighted issues, primarily with the structure of the wiki.

Most Popular

1 Migrating SourceForge Wordpress blog from "hosted apps" walkthrough – By Mike Jackson. SourceForge announced yesterday…

2 Bringing public data to life - competition to visualise UK publicly funded research – Can you present complex data to tell a compelling story…

3 The Craftsperson and the Scholar – By James Hetherington, Research Software…

4 Taking a Peek into eye disease – By Andrew Bastawrous, Research Fellow in International…

5 Workshop for e-Infrastructure trainers. Sold out! And more places added… – By Simon Hettrick. Our workshop for e-Infrastructure…

How can we help?

The Software Sustainability Institute cultivates world-class research with software.

We help people build better software, and we work with researchers, developers, funders and infrastructure providers to identify key issues and best practice in scientific software.

Click here for more.

Software and research blog

10 January 2014 – STEM – not just for the boys! – By Devasena Inupakutika, Software Consultant. How do we get…

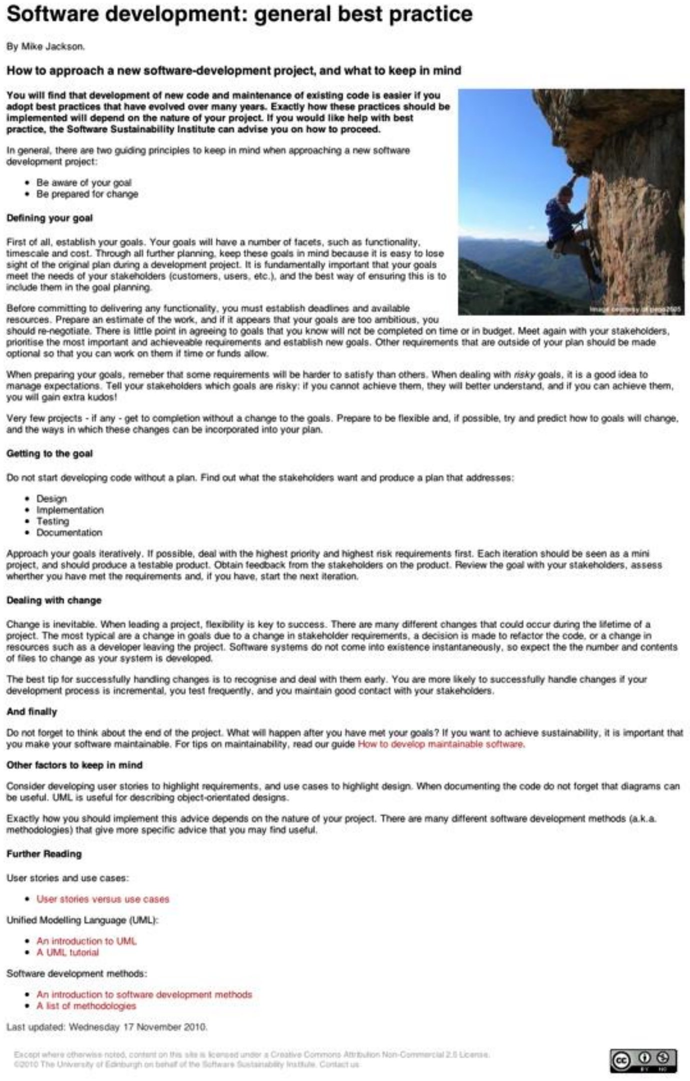
7 January 2014 – Open-source licences for people in a hurry – By Mike Jackson, Software Architect, the Software…

Disseminating best practice, gathering information

19,000+ unique visits/month

SSI Guides and Top Tips

- Guides provide in depth information
  - Licences
  - Software development
  - Project management
  - Repositories and project infrastructure
  - Open source
  - Community building
  - Publicising software
  - Policy
- Top Tips provide quick overviews
  - Software development
  - Repositories and project infrastructure
  - Software carpentry
  - Citing software
  - Data handling
  - Promoting and communicating your project
  - Community building and project management



Software development: general best practice

By Mike Jackson.

How to approach a new software-development project, and what to keep in mind

You will find that development of new code and maintenance of existing code is easier if you adopt best practices that have evolved over many years. Exactly how these practices should be implemented will depend on the nature of your project. If you would like help with best practice, the Software Sustainability Institute can advise you on how to proceed.

In general, there are two guiding principles to keep in mind when approaching a new software development project:

- Be aware of your goal
- Be prepared for change

Defining your goal

First of all, establish your goals. Your goals will have a number of facets, such as functionality, timescale and cost. Through all further planning, keep these goals in mind because it is easy to lose sight of the original plan during a development project. It is fundamentally important that your goals meet the needs of your stakeholders (customers, users, etc.), and the best way of ensuring this is to include them in the goal planning.

Before committing to delivering any functionality, you must establish deadlines and available resources. Prepare an estimate of the work, and if it appears that your goals are too ambitious, you should re-negotiate. There is little point in agreeing to goals that you know will not be completed on time or in budget. Meet again with your stakeholders, prioritise the most important and achieveable requirements and establish new goals. Other requirements that are outside of your plan should be made optional so that you can work on them if time or funds allow.

When preparing your goals, remeber that some requirements will be harder to satisfy than others. When dealing with *risky* goals, it is a good idea to manage expectations. Tell your stakeholders which goals are risky: if you cannot achieve them, they will better understand, and if you can achieve them, you will gain extra kudos!

Very few projects - if any - get to completion without a change to the goals. Prepare to be flexible and, if possible, try and predict how to goals will change, and the ways in which these changes can be incorporated into your plan.

Getting to the goal

Do not start developing code without a plan. Find out what the stakeholders want and produce a plan that addresses:

- Design
- Implementation
- Testing
- Documentation

Approach your goals iteratively. If possible, deal with the highest priority and highest risk requirements first. Each iteration should be seen as a mini project, and should produce a testable product. Obtain feedback from the stakeholders on the product. Review the goal with your stakeholders, assess wherther you have met the requirements and, if you have, start the next iteration.

Dealing with change

Change is inevitable. When leading a project, flexibility is key to success. There are many different changes that could occur during the lifetime of a project. The most typical are a change in goals due to a change in stakeholder requirements, a decision is made to refactor the code, or a change in resources such as a developer leaving the project. Software systems do not come into existence instantaneously, so expect the the number and contents of files to change as your system is developed.

The best tip for successfully handing changes is to recognise and deal with them early. You are more likely to successfully handle changes if your development process is incremental, you test frequently, and you maintain good contact with your stakeholders.

And finally

Do not forget to think about the end of the project. What will happen after you have met your goals? If you want to achieve sustainability, it is important that you make your software maintainable. For tips on maintainability, read our guide How to develop maintainable software.

Other factors to keep in mind

Consider developing user stories to highlight requirements, and use cases to highlight design. When documenting the code do not forget that diagrams can be useful. UML is useful for describing object-orientated designs.

Exactly how you should implement this advice depends on the nature of your project. There are many different software development methods (a.k.a. methodologies) that give more specific advice that you may find useful.

Further Reading

User stories and use cases:

- User stories versus use cases

Unified Modelling Language (UML):

- An introduction to UML
- A UML tutorial

Software development methods:

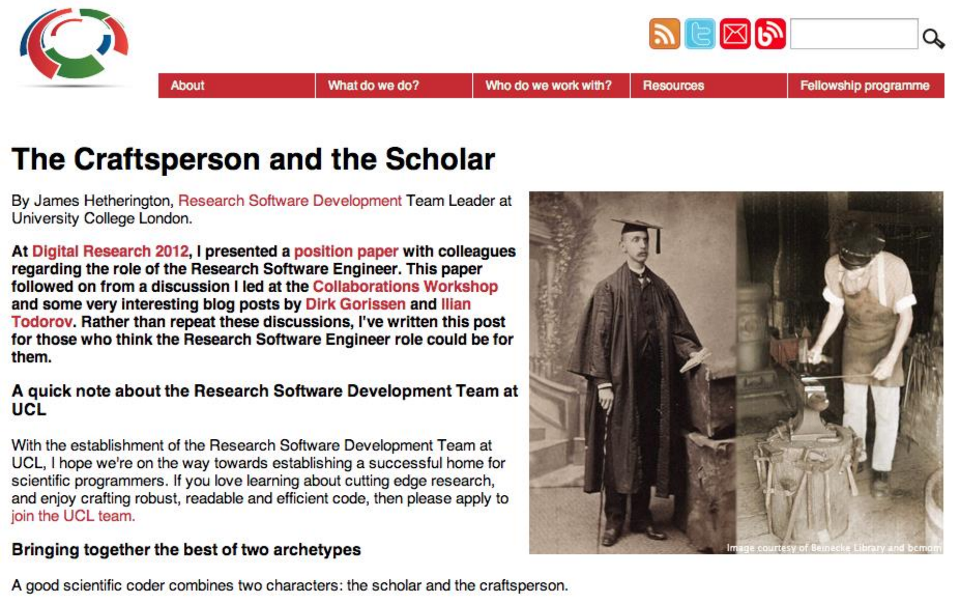
- An introduction to software development methods
- A list of methodologies

Last updated: Wednesday 17 November 2010.

SSI Blog

www.software.ac.uk

- Articles on research software and related issues
  - Ask the Institute
  - A Day in the Software Life
  - Heroes of Software Engineering
  - Top Tips
  - Women in Software
- 150+ posts written by external contributors
  - 10,000+ unique pageviews/month



The Craftsperson and the Scholar

By James Hetherington, Research Software Development Team Leader at University College London.

At Digital Research 2012, I presented a position paper with colleagues regarding the role of the Research Software Engineer. This paper followed on from a discussion I led at the Collaborations Workshop and some very interesting blog posts by Dirk Gorissen and Ilian Todorov. Rather than repeat these discussions, I've written this post for those who think the Research Software Engineer role could be for them.

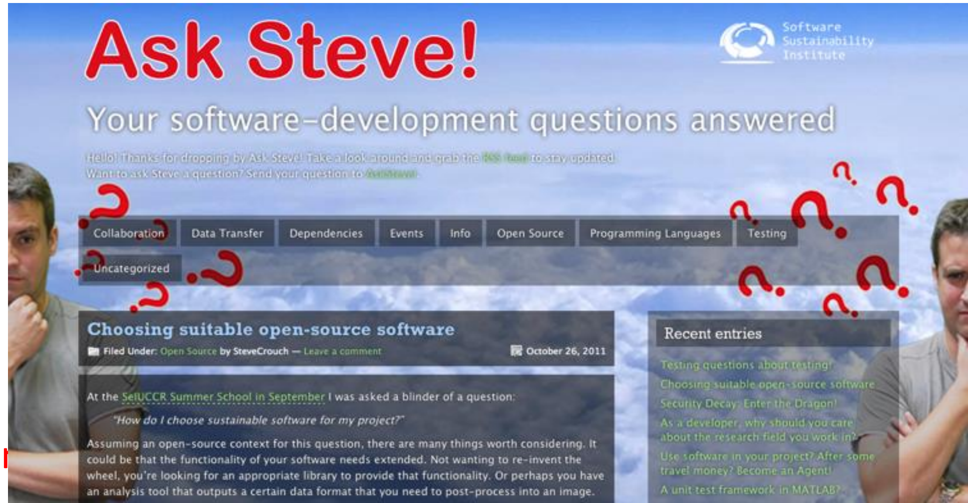
A quick note about the Research Software Development Team at UCL

With the establishment of the Research Software Development Team at UCL, I hope we're on the way towards establishing a successful home for scientific programmers. If you love learning about cutting edge research, and enjoy crafting robust, readable and efficient code, then please apply to join the UCL team.

Bringing together the best of two archetypes

A good scientific coder combines two characters: the scholar and the craftsperson.

Image courtesy of Beinecke Library and bcmom



Ask Steve!

Your software–development questions answered

Hello! Thanks for dropping by Ask Steve! Take a look around and grab the RSS feed to stay updated.
Want to ask Steve a question? Send your question to AskSteve!

Choosing suitable open-source software

Filed Under: Open Source by SteveCrouch — Leave a comment

October 26, 2011

At the SelUCCR Summer School in September I was asked a blinder of a question:

> *"How do I choose sustainable software for my project?"*

Assuming an open-source context for this question, there are many things worth considering. It could be that the functionality of your software needs extended. Not wanting to re-invent the wheel, you're looking for an appropriate library to provide that functionality. Or perhaps you have an analysis tool that outputs a certain data format that you need to post-process into an image.

Recent entries

- Testing questions about testing!
- Choosing suitable open-source software
- Security Decay: Enter the Dragon!
- As a developer, why should you care about the research field you work in?
- Use software in your project? After some travel money? Become an Agent!
- A unit test framework in MATLAB?

SSI Training

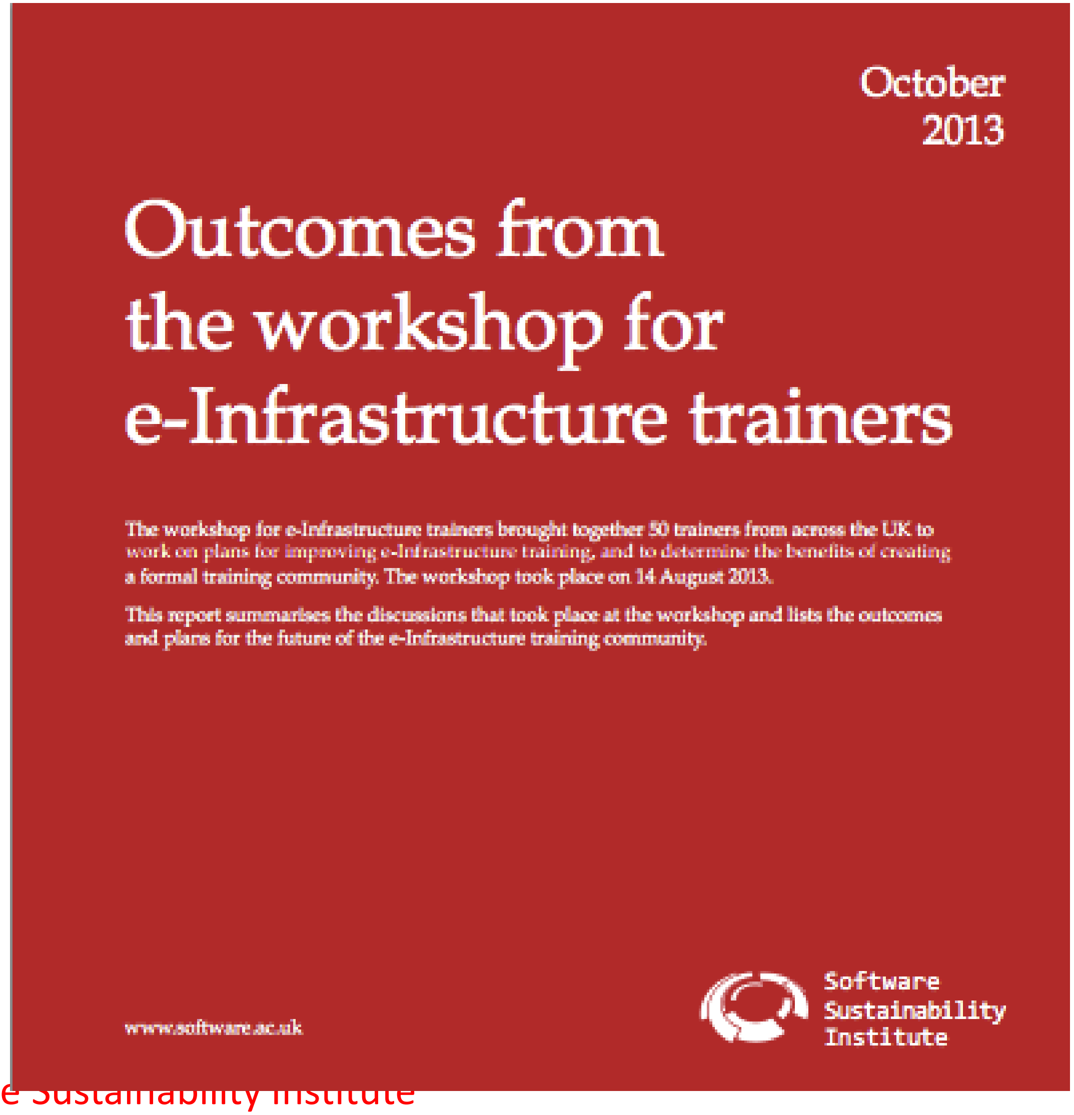
- Software Carpentry
  - http://software-carpentry.org
  - International initiative to teach basics of software engineering to computational researchers
    - The "why" more than the "how"
  - Phenomenally successful – 2x oversubscription
  - Cheap to run but budget for 3x the coffee!
  - SSI are UK Coordinators for SWC
  - We ran 13 workshops in 2013 to 600+ learners

- Software Sustainability Surgeries
  - "Bring your own code"
  - "What makes Good Code good?"
  - Run at existing conferences, and for software funding programmes
  - Offering bespoke advice as well as training

Creating a training community

- Bringing together 39+ organisations with interest in e-Infrastructure training
- Raising issues and enablers with RCUK, BIS

software.ac.uk/policy



October 2013

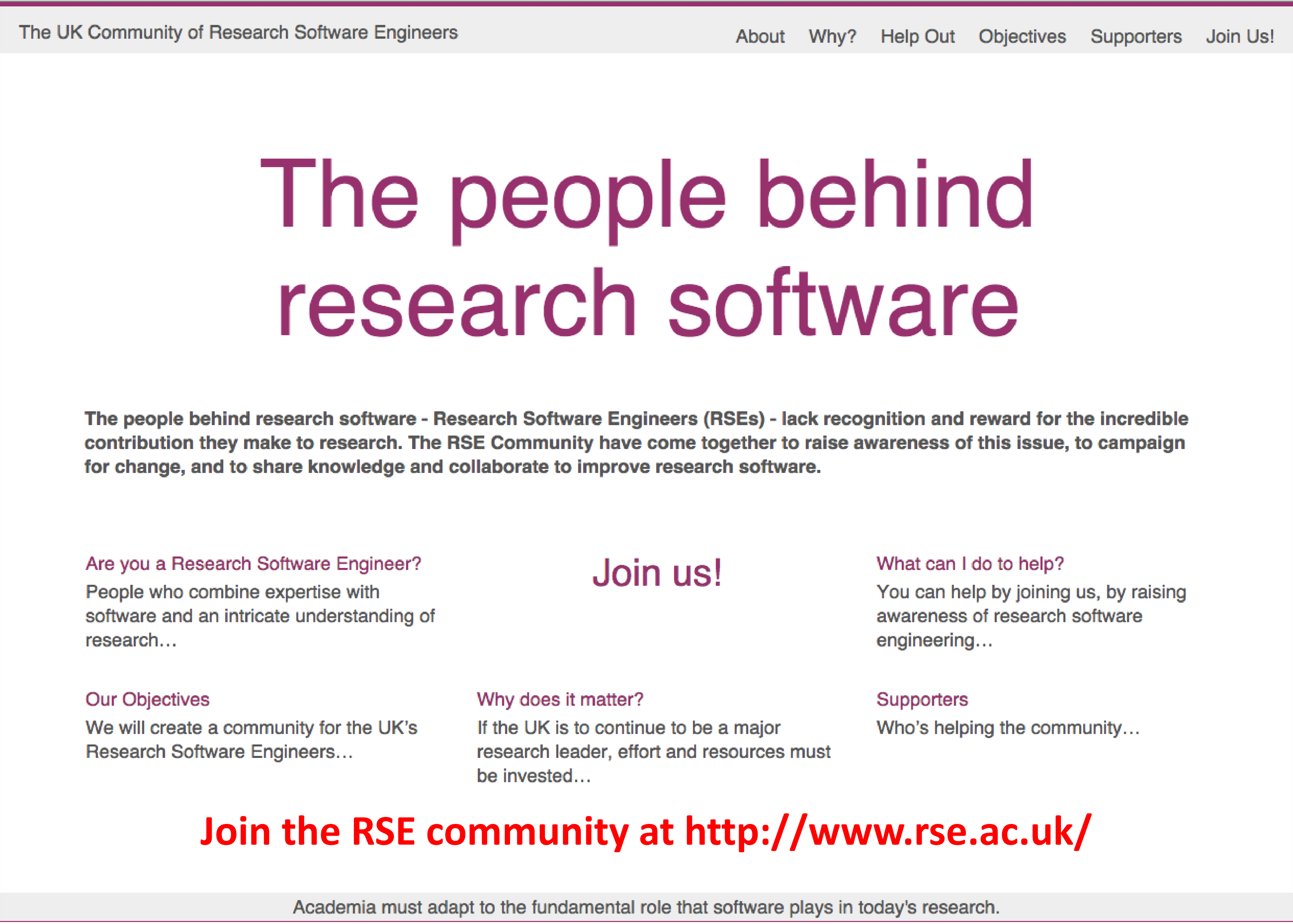
Outcomes from the workshop for e-Infrastructure trainers

The workshop for e-Infrastructure trainers brought together 50 trainers from across the UK to work on plans for improving e-Infrastructure training, and to determine the benefits of creating a formal training community. The workshop took place on 14 August 2013.

This report summarises the discussions that took place at the workshop and lists the outcomes and plans for the future of the e-Infrastructure training community.

www.software.ac.uk

Software Sustainability Institute



The people behind research software

The people behind research software - Research Software Engineers (RSEs) - lack recognition and reward for the incredible contribution they make to research. The RSE Community have come together to raise awareness of this issue, to campaign for change, and to share knowledge and collaborate to improve research software.

Are you a Research Software Engineer?

People who combine expertise with software and an intricate understanding of research…

Join us!

What can I do to help?

You can help by joining us, by raising awareness of research software engineering…

Our Objectives

We will create a community for the UK's Research Software Engineers…

Why does it matter?

If the UK is to continue to be a major research leader, effort and resources must be invested…

Supporters

Who's helping the community…

Join the RSE community at http://www.rse.ac.uk/

Academia must adapt to the fundamental role that software plays in today's research.

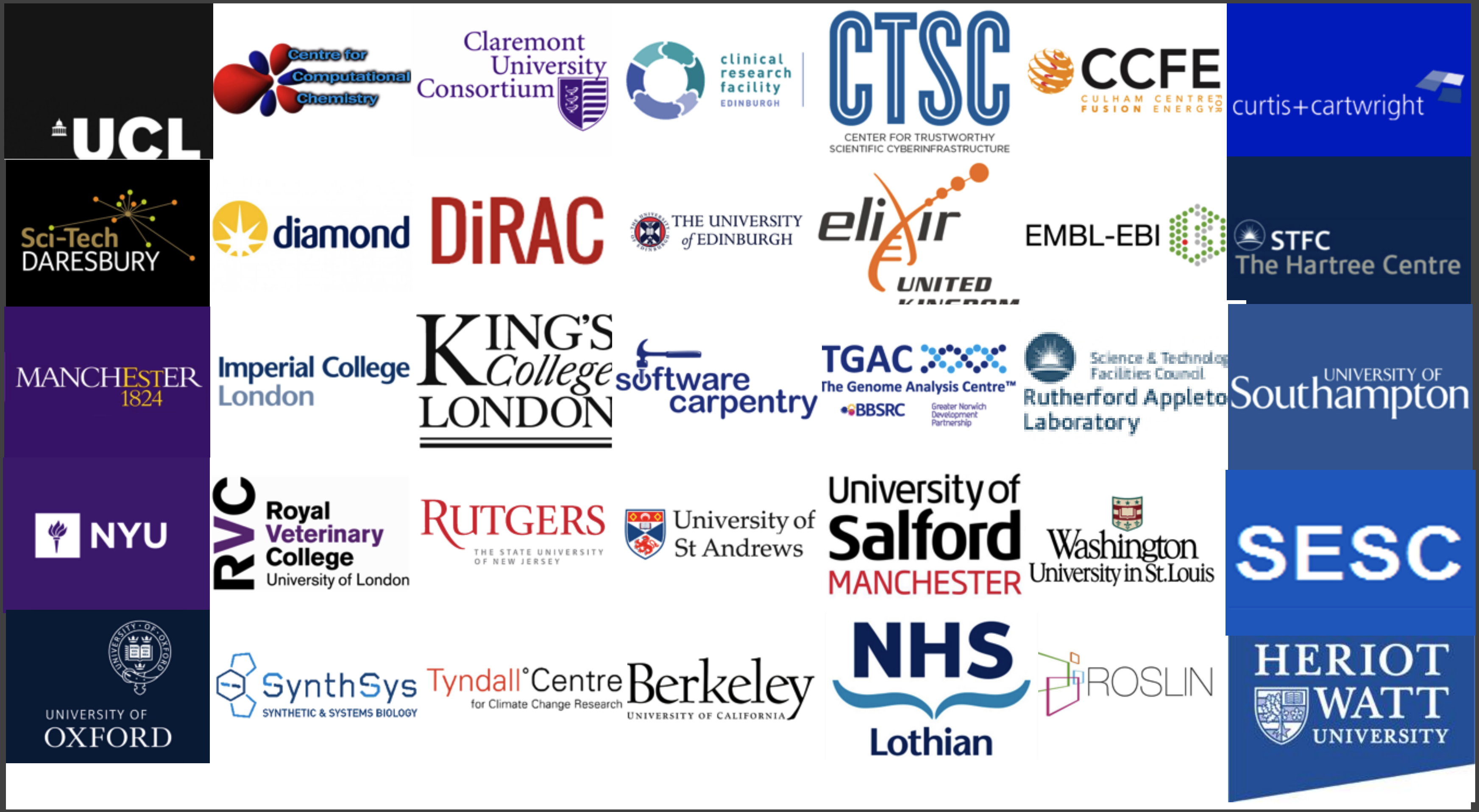
Software Sustainability Institute

No-one sets out to write unsustainable software

People need the skills and incentives to maintain software through its lifetime

The Institute helps UK researchers benefit from more sustainable software

Work with us – www.software.ac.uk


UCL
Centre for Computational Chemistry
Claremont University Consortium
clinical research facility EDINBURGH
CTSC CENTER FOR TRUSTWORTHY SCIENTIFIC CYBERINFRASTRUCTURE
CCFE CULHAM CENTRE FOR FUSION ENERGY
curtis+cartwright
Sci-Tech DARESBURY
diamond
DiRAC
THE UNIVERSITY of EDINBURGH
elixir UNITED KINGDOM
EMBL-EBI
STFC The Hartree Centre
MANCHESTER 1824
Imperial College London
KING'S College LONDON
software carpentry
TGAC The Genome Analysis Centre™ BBSRC Greater Norwich Development Partnership
Science & Technology Facilities Council Rutherford Appleton Laboratory
UNIVERSITY OF Southampton
NYU
RVC Royal Veterinary College University of London
RUTGERS THE STATE UNIVERSITY OF NEW JERSEY
University of St Andrews
University of Salford MANCHESTER
Washington University in St.Louis
SESC
UNIVERSITY OF OXFORD
SynthSys SYNTHETIC & SYSTEMS BIOLOGY
Tyndall°Centre for Climate Change Research
Berkeley UNIVERSITY OF CALIFORNIA
NHS Lothian
ROSLIN
HERIOT WATT UNIVERSITY