www.software.ac.uk

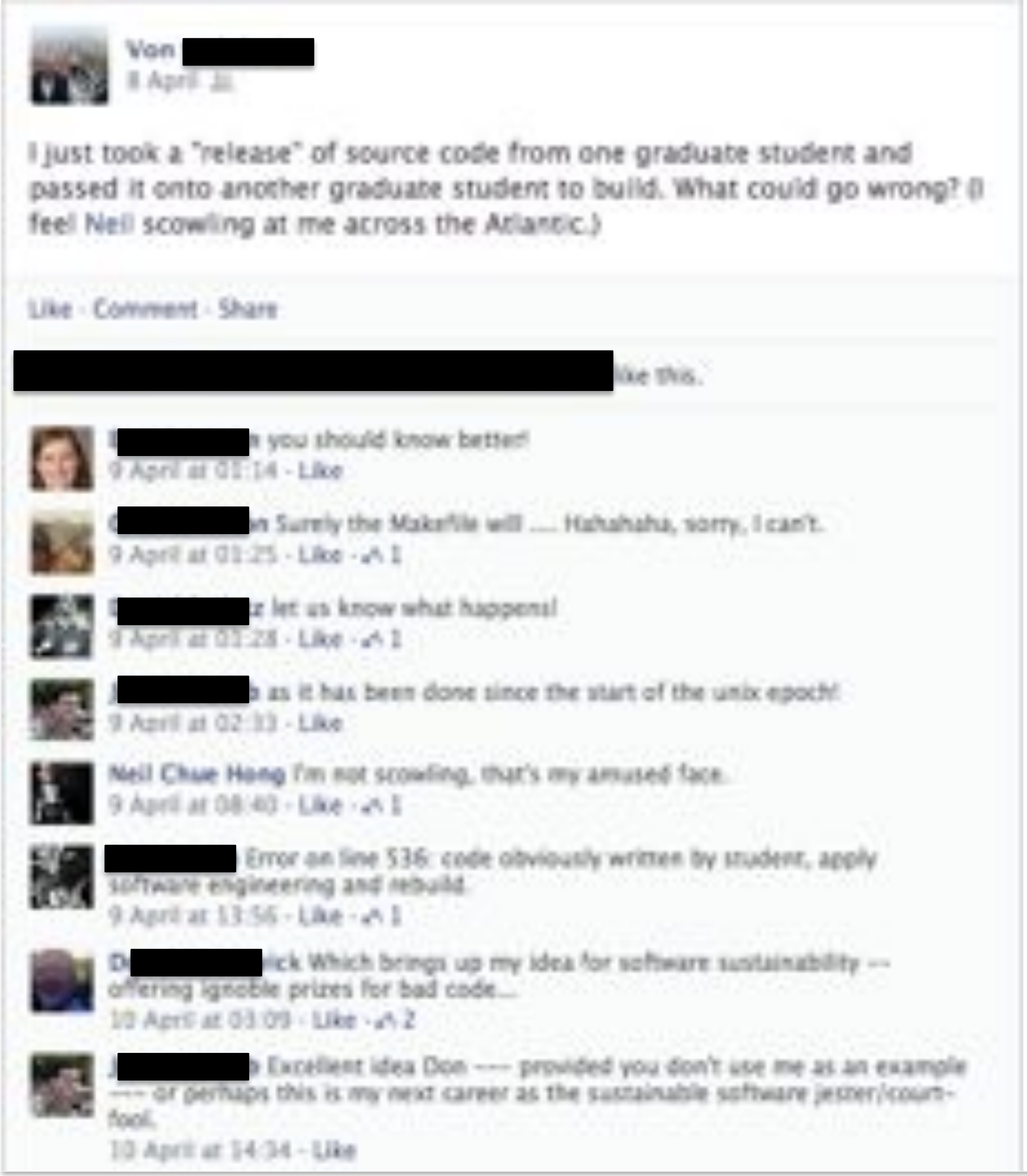
Introduction to the Software Sustainability Institute

13 August 2014, CCP-SAS webinar

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

Supported by

Project funding from



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.)

Like · Comment · Share

[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

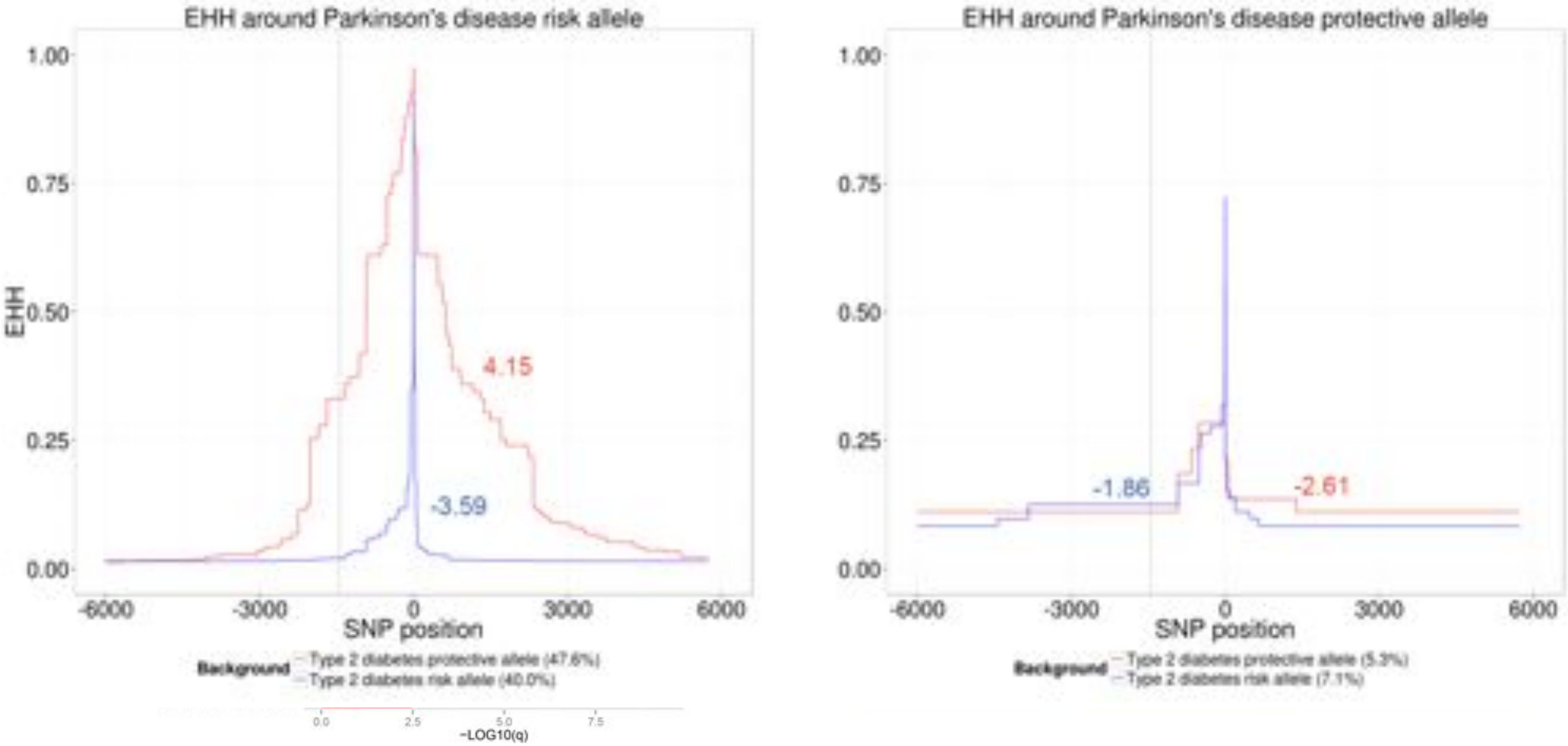
[redacted] Excellent idea Don ---- provided you don't use me as an example ---- or perhaps this is my next career as the sustainable software jester/court-fool.
10 April at 14:34 · Like

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

  Right?

Pleiotropic loci – diabetes and Parkinson’s disease

www.software.ac.uk



ROSLIN

Software Sustainability Institute

Selection at pleiotropic loci underlies disease co-occurrence in human populations.
Navarro, Haley, Karosas et al.
Submitted to Nature Genetics

Behind every great piece of science…

www.software.ac.uk

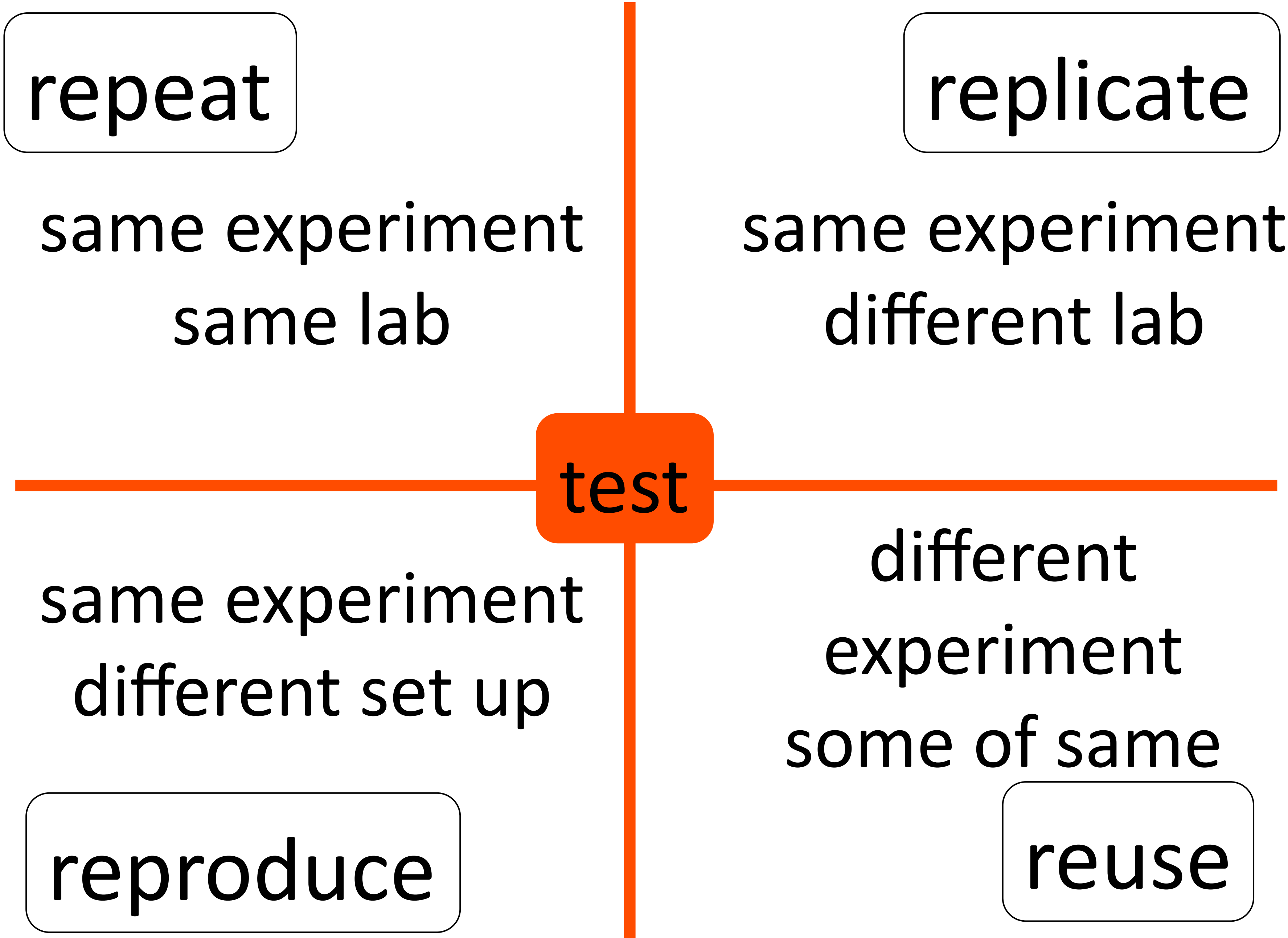
```
#go through each SNP of interest
for(my $x = 0; $x < scalar @pos; $x++)
{
    #and then each downstream SNP of interest
    for(my $y = $x+1; $y < scalar @pos; $y++)
    {
        #if SNPs within our chosen distance (500kb) and both present in the haplotypes file
        if((!($trait[$x] eq $trait[$y])) && (abs($pos[$x] - $pos[$y]) <= 500000) && (exists($legArrayPos{$po
        {
            my $snp1ArrayPos = "”;
            my $snp2ArrayPos = "”;
            my $snp1All = "”;
            my $snp2All = "”;

            #create output file for this SNP pair
            my $filename = "ConditionedResults2/$chr[$x].$pos[$x]-$pos[$y].EHH.GBR.2.txt”;
            print "$filename\n”;
            unless (-e $filename) {
                open(OUT, ">$filename");

                #####################CHANGE THESE IF NOT FOCUSING ON SECOND SNP########################
                my $start = $pos[$y]-500000;
                if ($start < 1) {
                    $start = 1;
                }
                my $end = $pos[$y]+500000;
                if ($end > $chrLengths{$chr[$x]}) {
                    $end = $chrLengths{$chr[$x]};
                }
```

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



repeat

same experiment
same lab

replicate

same experiment
different lab

test

same experiment
different set up

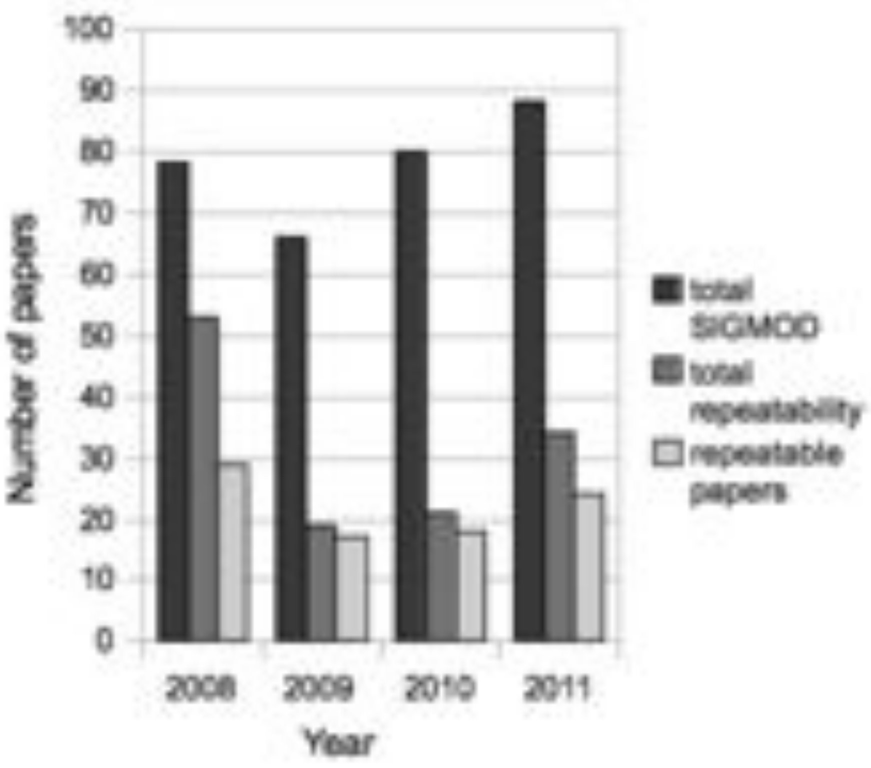
reproduce

different
experiment
some of same

reuse

Figure by Carole Goble adapted from Drummond C, Replicability is not Reproducibility: Nor is it Good Science, online and Peng RD, Reproducible Research in Computational Science *Science 2 Dec 2011: 1226-1227.*

SIGMOD Reproducibility



Bonnet et al, SIGMOD Record,
June 2011 (Vol. 40, No. 2)
doi: 10.1145/2034863.2034873

- SIGMOD conference offered to attempt to repeat/reproduce papers accepted at conference
  - 2008-2012
- "High burden on reviewers when setting up experiments"
  - Use of VMs advocated

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

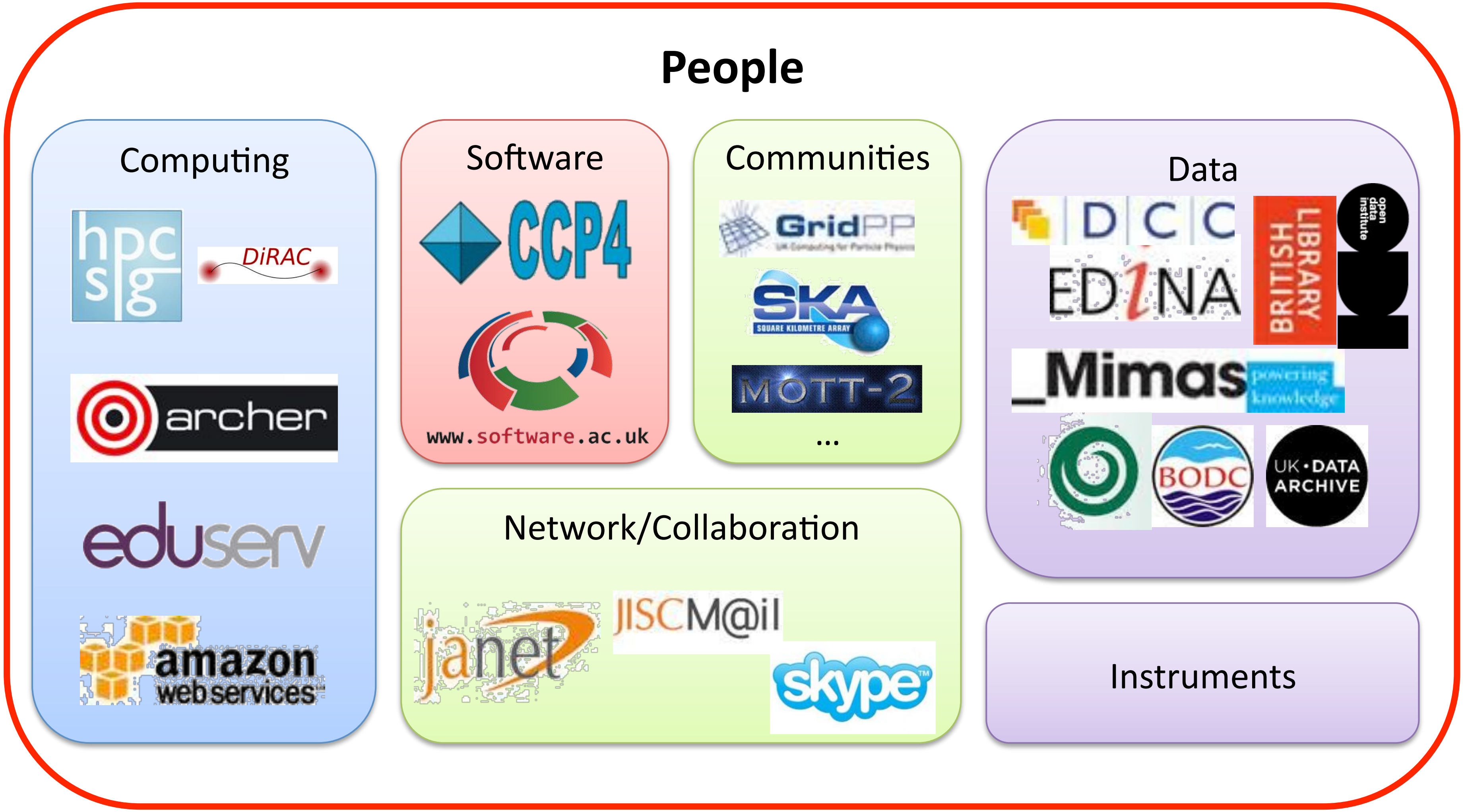
A national facility for cultivating world-class research through software

- 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

UK Research Computing Ecosystem

www.software.ac.uk



People

Computing

Software

Communities

Data

Network/Collaboration

Instruments

SSI Objectives

1. Getting software on the research agenda
2. Supporting communities that want change
3. Increasing skills
4. Improving software

- Building a platform
- Engaging the community
- Being an authoritative voice

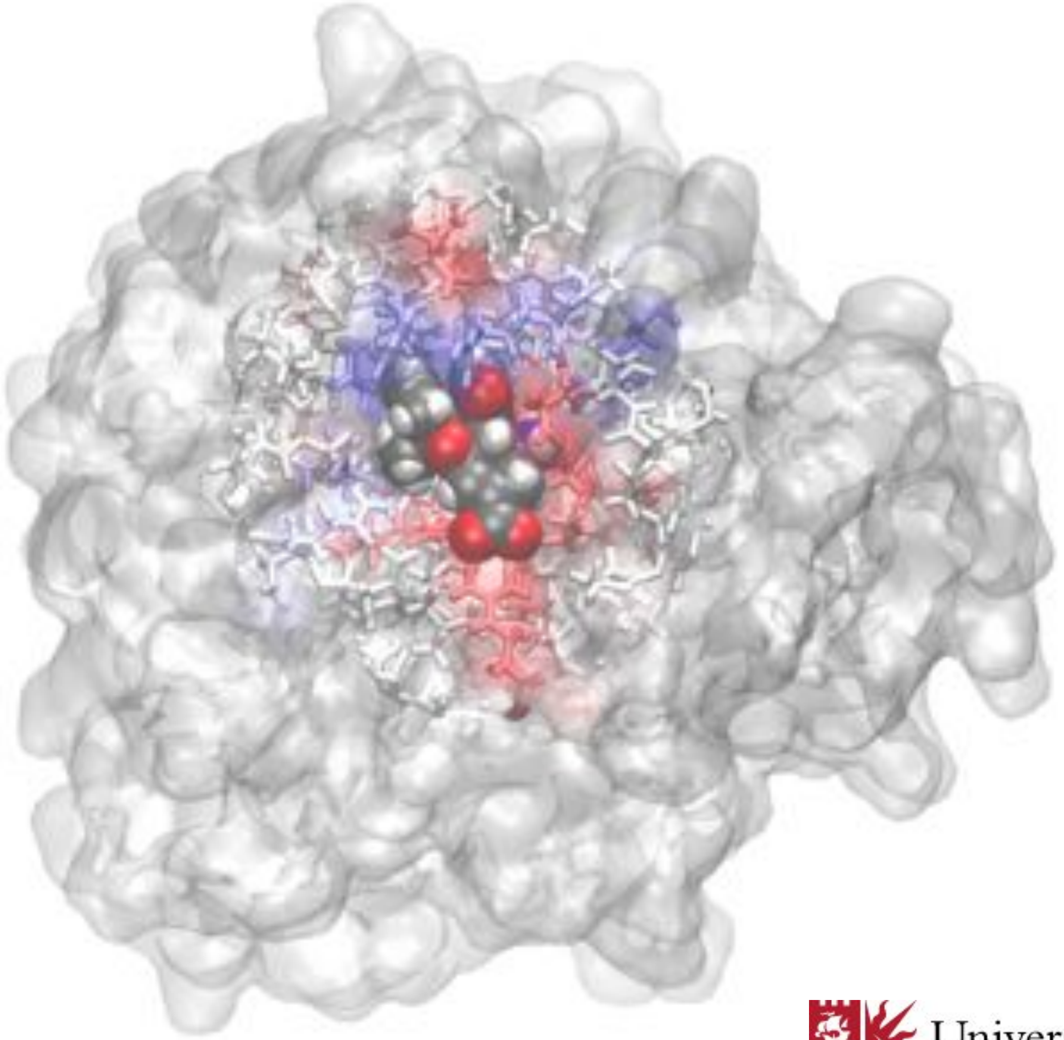
SSI Organisation

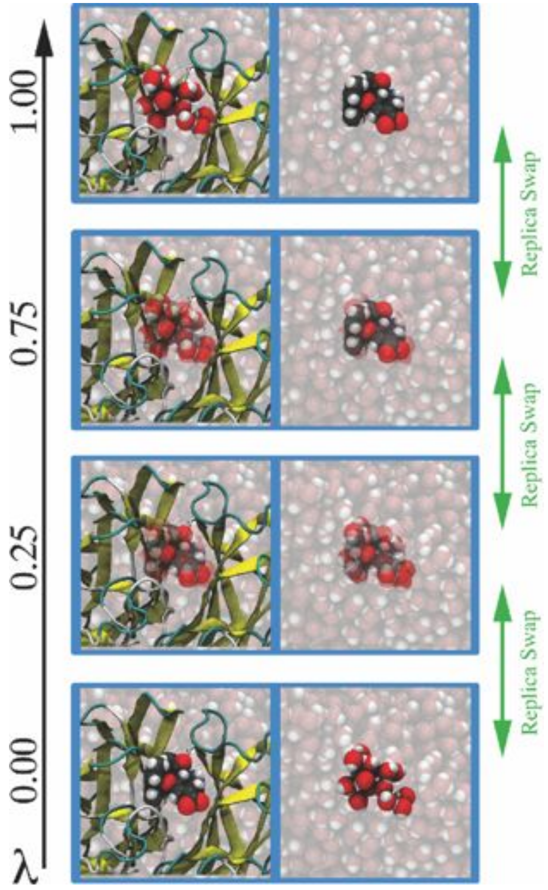
- Community Engagement (Lead: Shoaib Sufi)
  - Fellowship Programme
  - Events and Roadshows
- Research Software (Lead: Steve Crouch)
  - Open Call for Consultancy Projects / Funded Collaborations
  - Software Evaluation
- Policy (Lead: Simon Hettrick)
  - Guides and Case Studies
  - Best Practice and Policy
- Training (Lead: Aleksandra Pawlik)
  - Software Carpentry
  - Software Surgeries

- Collaboration between universities of Edinburgh, Manchester, Oxford and Southampton.

Water Swap
Reaction Coordinate

www.software.ac.uk





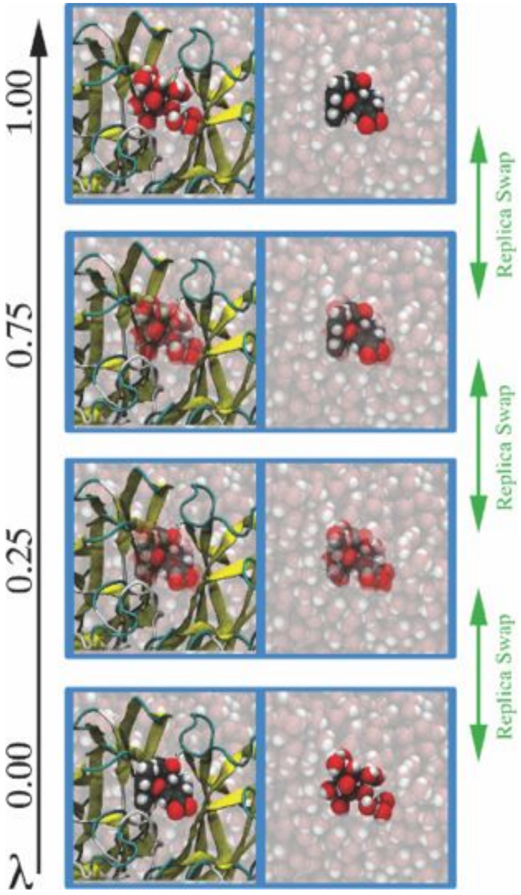
University of BRISTOL

A water-swap reaction coordinate for the calculation of absolute protein-ligand binding free energies
Woods CJ, Malaisree M, Hannongbua S, Mulholland AJ
J. Chem. Phys. (2011) vol. 134, pp. 054114
http://dx.doi.org/10.1063/1.3519057

Case Study: Ligand Binding

www.software.ac.uk

- Centre for Computational Chemistry, Bristol
  - New methods for rapid MC sampling of biomolecular systems modelled using QM/MM
  - Developed two codes ProtoMS (F77) + Sire (C++)
  - Water-Swap Reaction Coordinate method to calculate absolute protein-ligand binding free energies
- SSI's work is helping to scale development
  - ProtoMS and Sire both single developer codes
  - ASPIRE/ACQUIRE framework has multiple devs
    - Split architecture between ASPIRE (adaptive multiresolution hybrid MD simulation) and ACQUIRE (WorkPacket scheduling system with optimisation for time to result vs "green-ness"



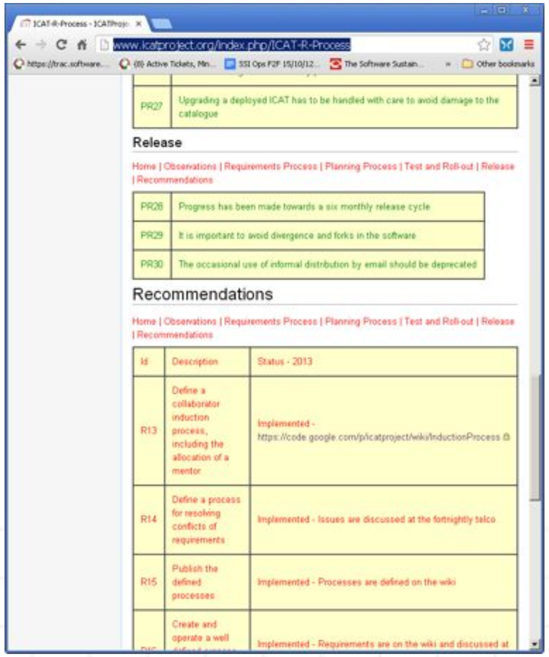
- http://www.siremol.org/adaptive_dynamics

Case Study: ICAT

www.software.ac.uk

- ICAT metadata catalogue, STFC
  - Support experimental data linking at Large Facilities
  - RAL UK (ISIS, DIAMOND, CLF), SNS US, ELLETRA Italy
  - ICAT operationally critical at sites, other projects looking to use
  - Undertook interview-based organisational review
    - 92 observations, 32 recommendations
- SSI's work means the project has move forward
  - Taking forward: steering group, induction process, expanded roadmapping activity, technical workshops via Skype, involved with deployment
  - 1 year post-project
    - 25 (75%) recommendations implemented
    - Used by other major projects (PandataODI, CRISP)

- http://www.software.ac.uk/blog/2012-02-17-evaluating-software-behind-some-worlds-large-experimental-facilities
- http://icatproject.org/


ICAT-R-Process - ICATProje
www.icatproject.org/index.php/ICAT-R-Process
https://trac.software...
(8) Active Tickets, Min...
SSI Ops F2F 15/10/12...
The Software Sustain...
Other bookmarks

PR27	Upgrading a deployed ICAT has to be handled with care to avoid damage to the catalogue

Release

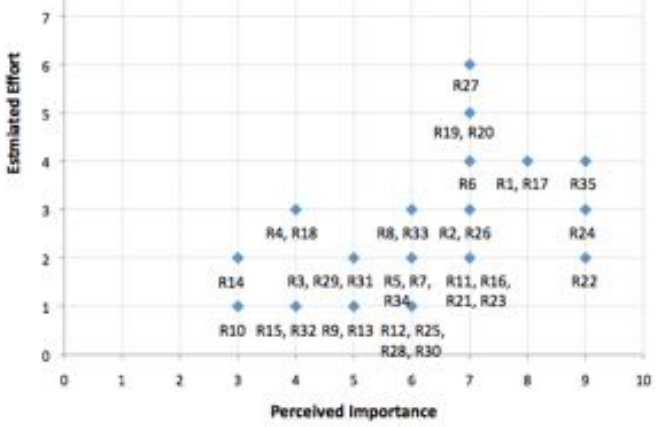
Home | Observations | Requirements Process | Planning Process | Test and Roll-out | Release | Recommendations

PR28	Progress has been made towards a six monthly release cycle
PR29	It is important to avoid divergence and forks in the software
PR30	The occasional use of informal distribution by email should be deprecated

Recommendations

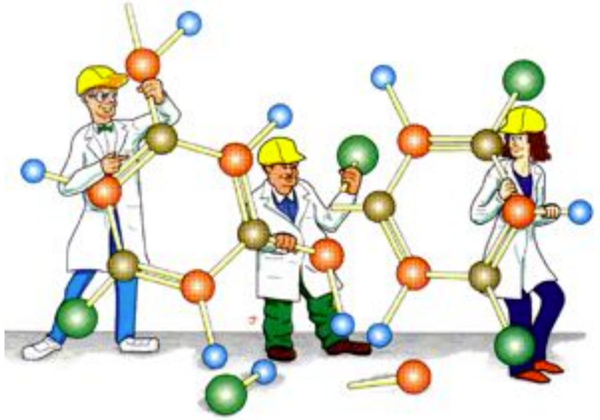
Home | Observations | Requirements Process | Planning Process | Test and Roll-out | Release | Recommendations

Id	Description	Status - 2013
R13	Define a collaborator induction process, including the allocation of a mentor	Implemented - https://code.google.com/p/icatproject/wiki/InductionProcess
R14	Define a process for resolving conflicts of requirements	Implemented - Issues are discussed at the fortnightly telco
R15	Publish the defined processes	Implemented - Processes are defined on the wiki
R16	Create and operate a well	Implemented - Requirements are on the wiki and discussed at

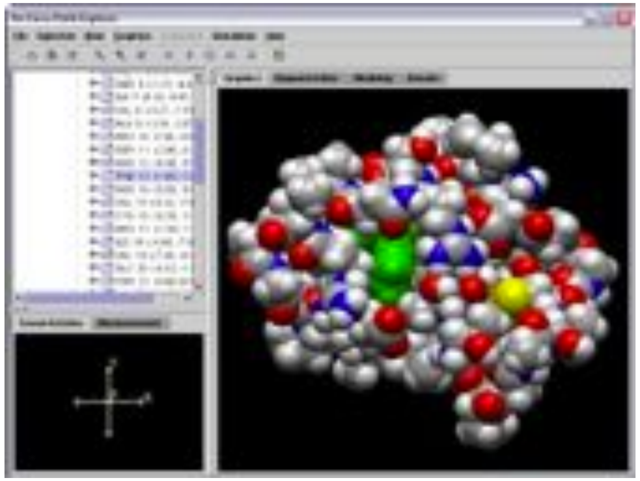


Software Sustainability Institute

Case Study: Tinkering with APES



- 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



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

Current Consultancy (1)

BoneJ

RVC Royal Veterinary College University of London

- *Trabecular geometry & whole bone shape analysis*
- Development review, community survey, CMS
- Assist future BBSRC BBR funding
  - ‘SSI’s involvement noted in a very positive light’

Lower Limb Model

Imperial College London

- *Musculoskeletal model for lower limbs*
- Wellcome Trust, EPSRC funded
- Development review, moving to Assembla
  - ‘Institute’s team were invaluable’

ForestGrowth-SRC

UNIVERSITY OF Southampton

- *Process-based tree growth model*
- NERC, Forestry Commission funded
- Run on IRIDIS cluster – take hours, not days
- Joint IfLS funding

Current Consultancy (2)

LabBook

Imperial College London

- *Fast, accurate, secure experiment recording/sharing*
- Wide domain potential
- Seeking funded development
- Infrastructure, outreach, roadmapping & feedback

DawnScience

Science & Technology Facilities Council Rutherford Appleton Laboratory

- *Eclipse-based workbench for scientific data analysis*
- STFC funded
- Collaboration review, community marketing
  - Uptake key measure of project success

BASIL/FABBER

UNIVERSITY OF OXFORD

- *Analysis tool library for FMRI, MRI, DTI brain imaging*
- Industry / EPSRC funded
- Architectural review, apply to new problems
- 10 groups outside Oxford using it

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

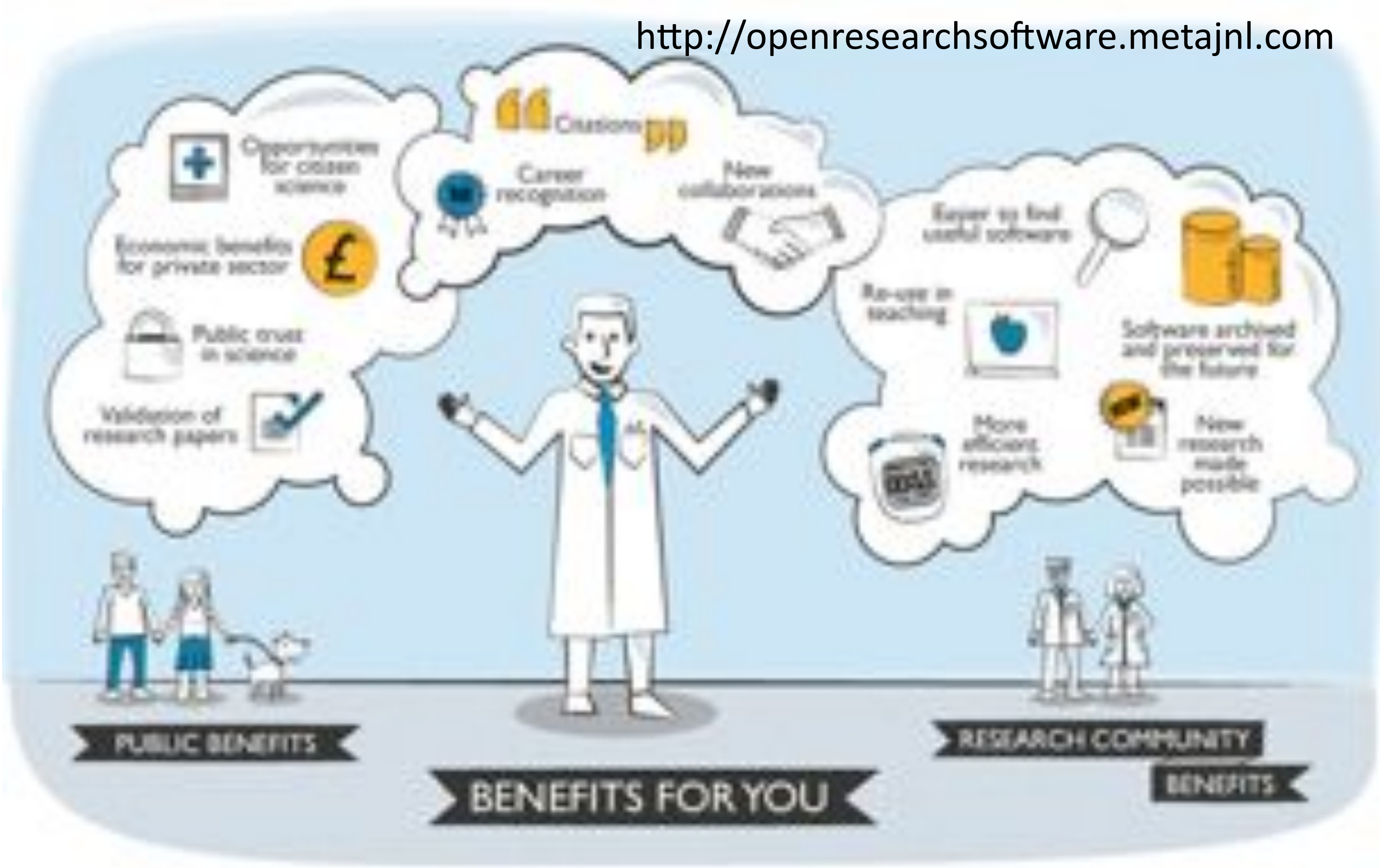
- 2014: 16 fellows
- 2013: 15 fellows
- 2012: 10 fellows
- Range of subjects, career stages

software.ac.uk/fellows

Journal of Open Research Software

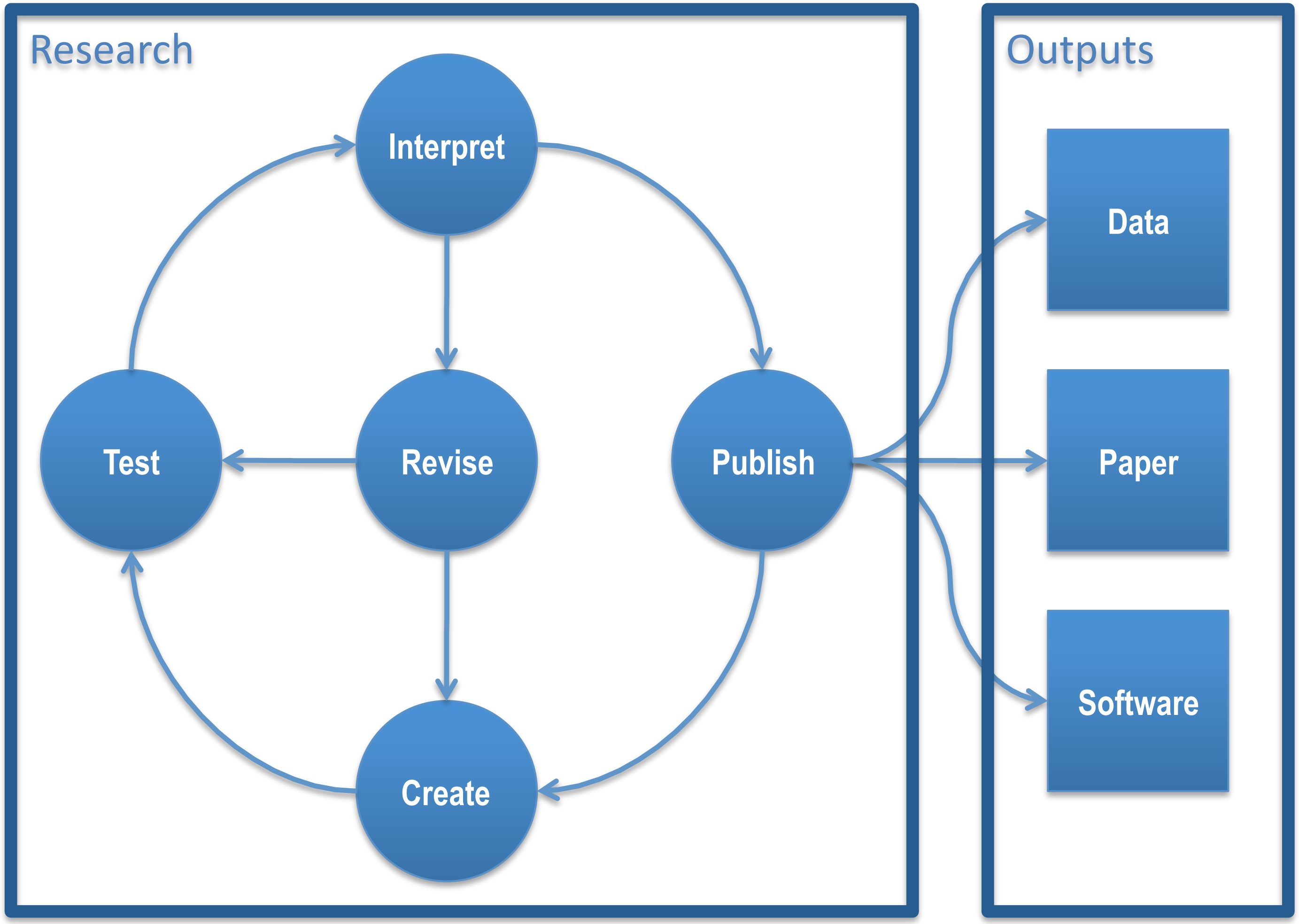
www.software.ac.uk

http://openresearchsoftware.metajnl.com


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

Software Sustainability Institute

The Research Cycle

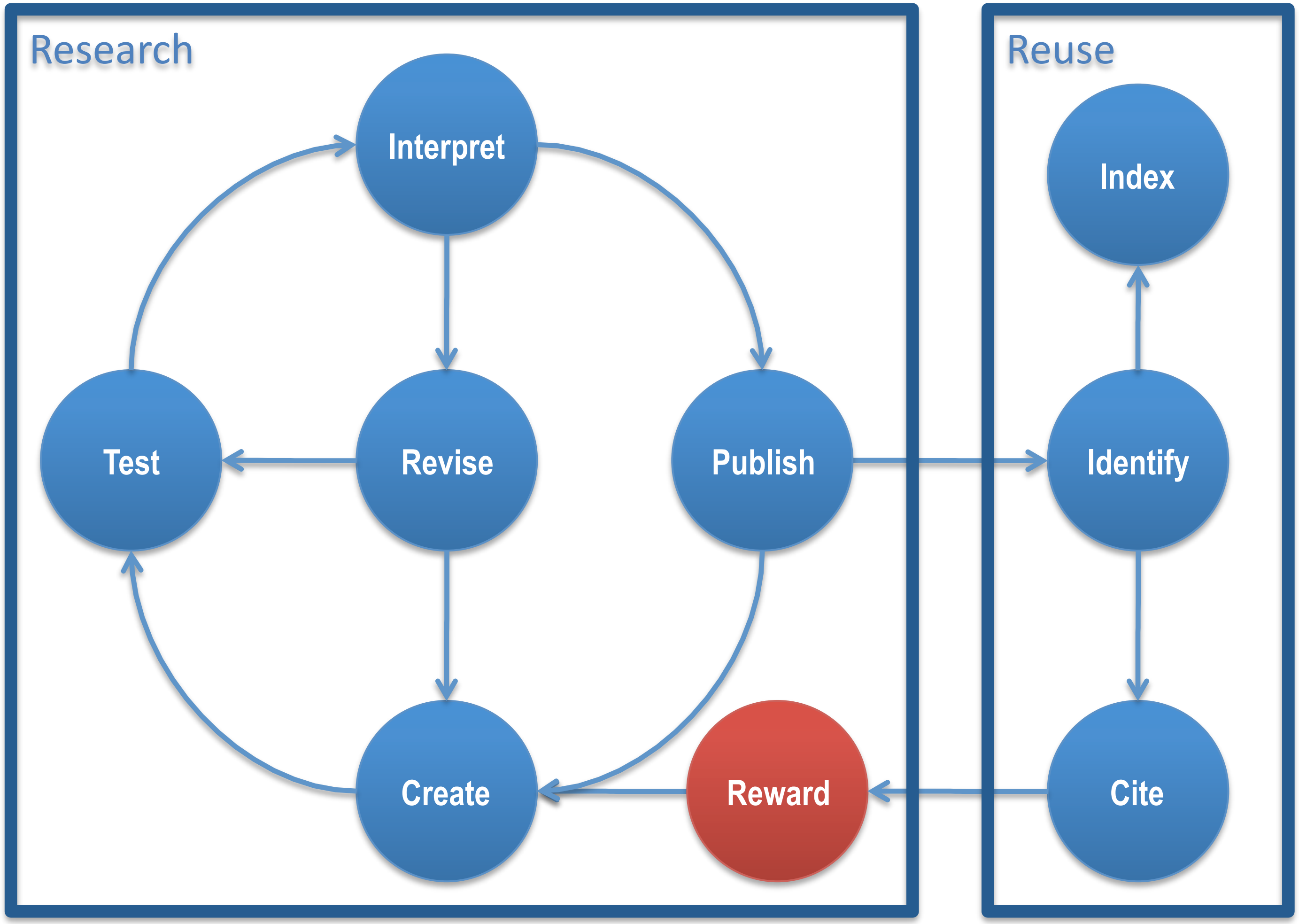


Research is a *continuous* cycle.

When we publish we are contributing to the body of knowledge.

Research/Reuse/Reward Cycle

www.software.ac.uk



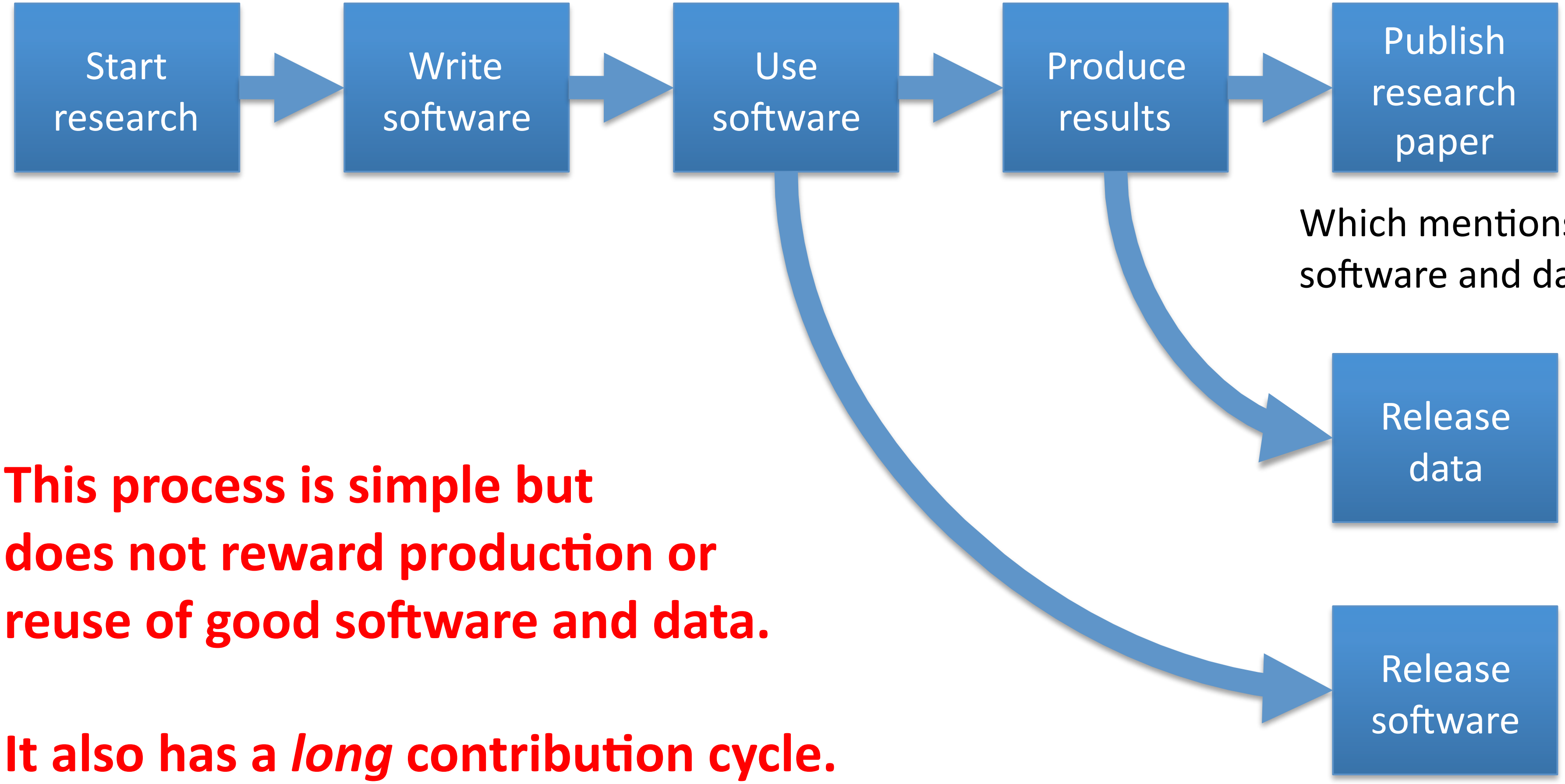
Reuse is also a cycle. We build our research on the work of others.

Reward mechanisms *should* encourage reuse.

Software Sustainability Institute

The current process

www.software.ac.uk

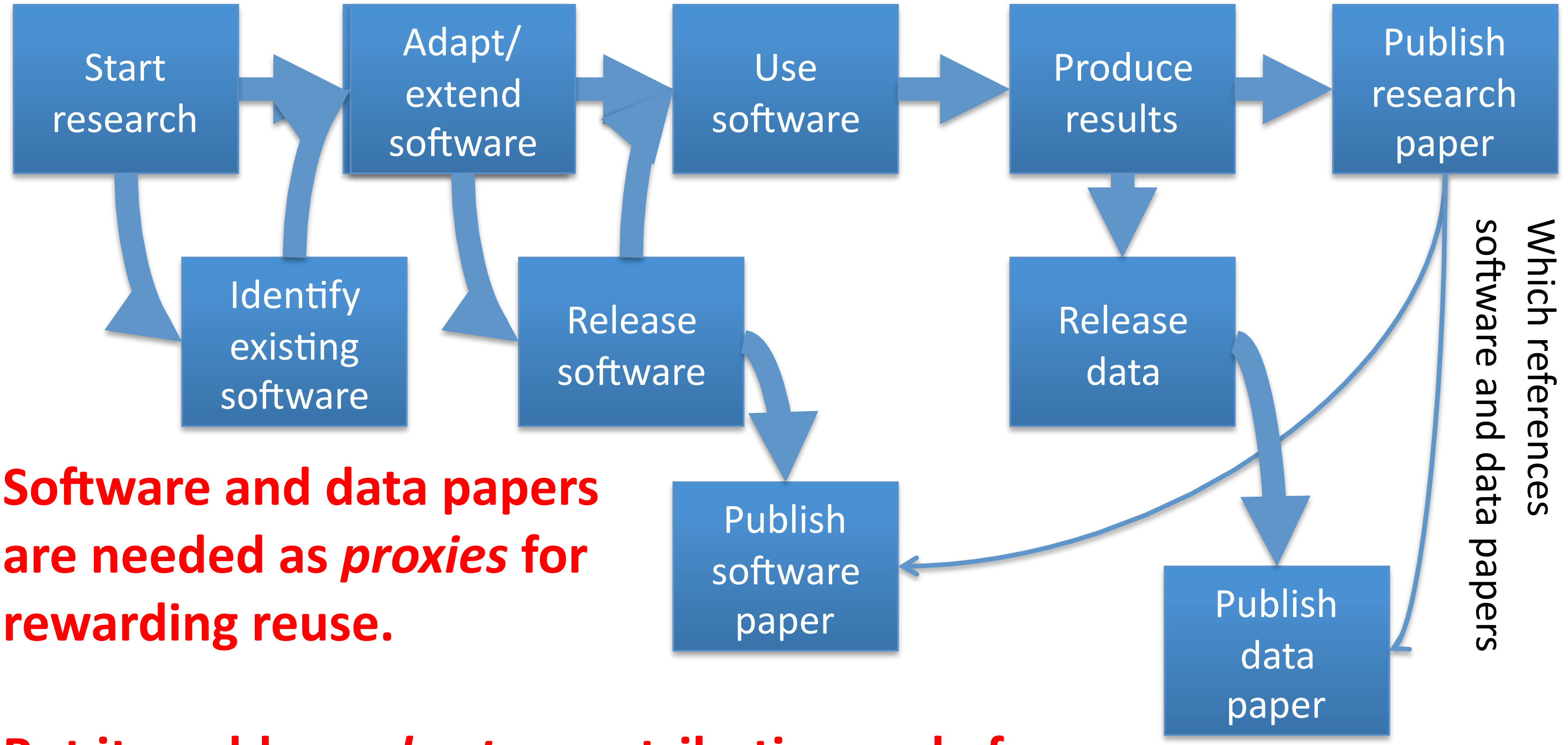


This process is simple but does not reward production or reuse of good software and data.

It also has a *long* contribution cycle.

A better process?

www.software.ac.uk

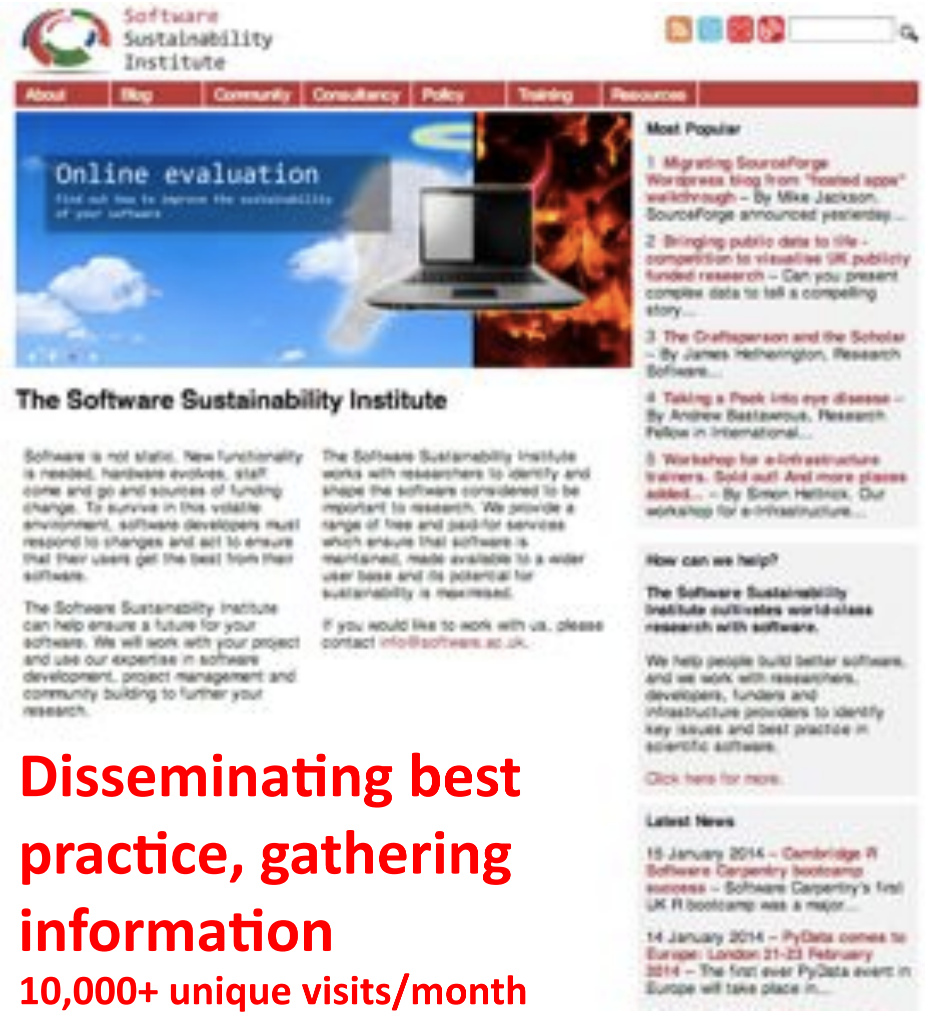


Software and data papers are needed as *proxies* for rewarding reuse.

But it enables a *shorter* contribution cycle for data and software.

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.

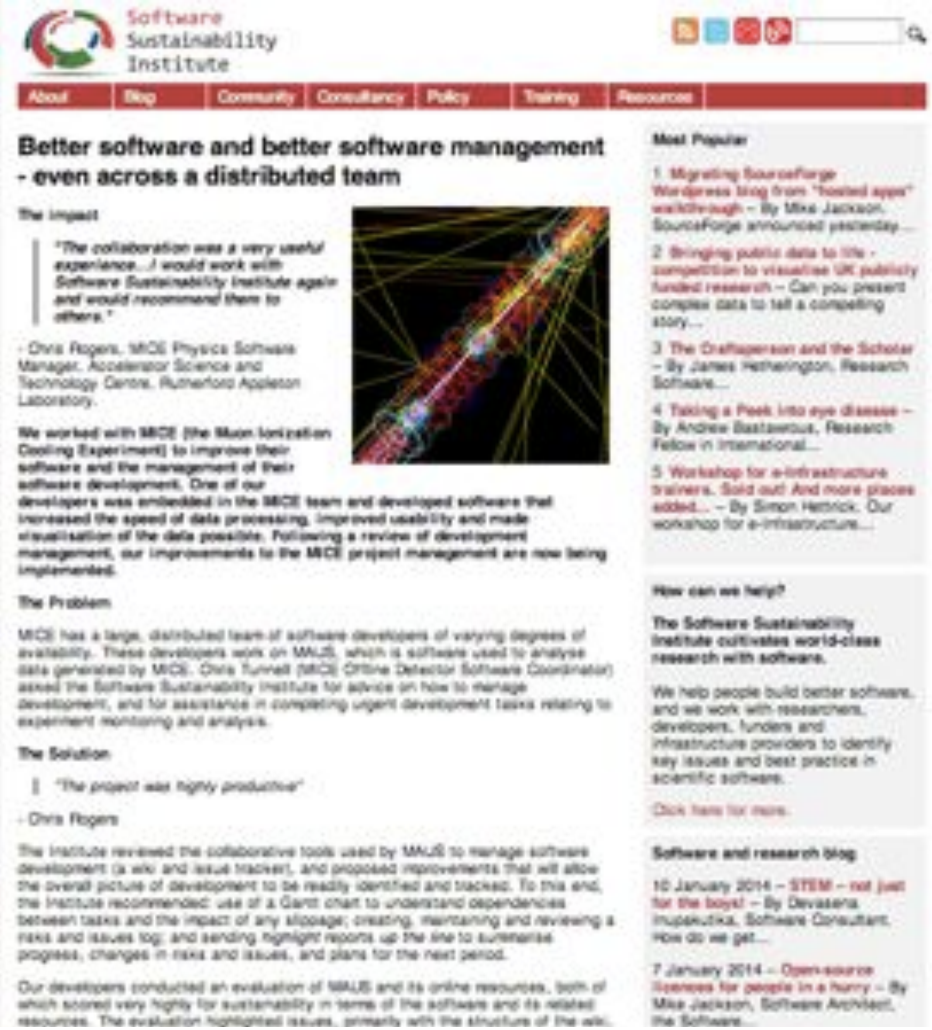
We help people build better software, and we work with researchers, developers, funders and infrastructure providers to identify key issues and best practice in scientific software.

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...



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...

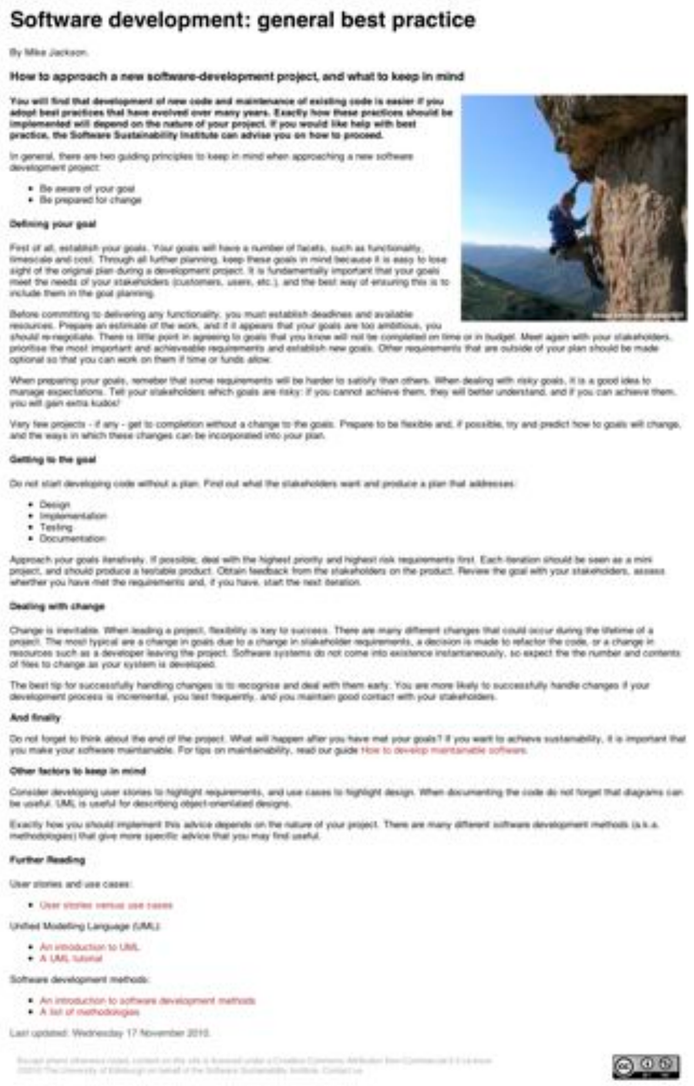
Disseminating best practice, gathering information

10,000+ unique visits/month

SSI Guides and Top Tips

www.software.ac.uk

- 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 whether 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 handling 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:

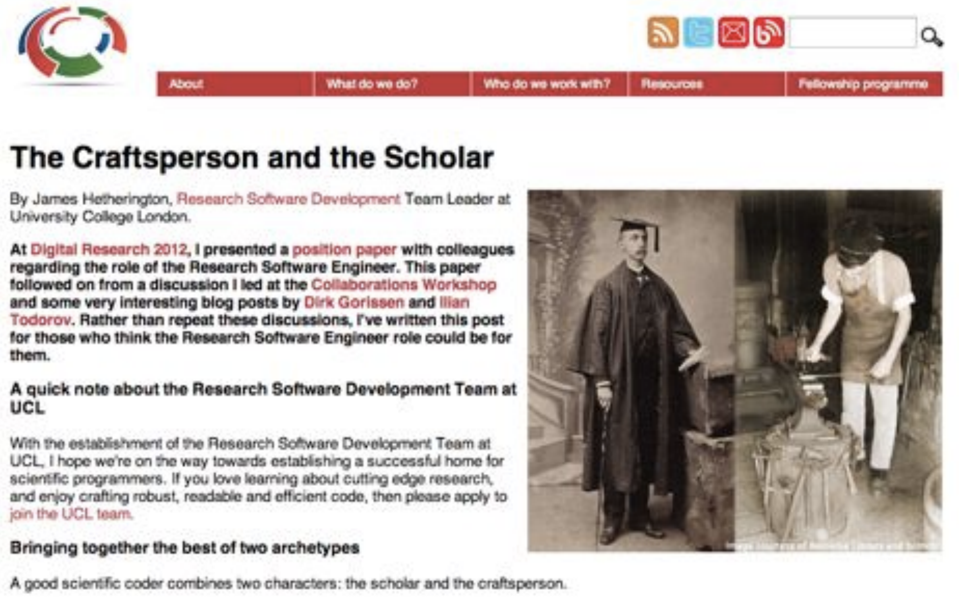
- An introduction to software development methods
- A list of methodologies

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
  - 6,000+ unique pageviews/ month



The Craftsperson and the Scholar

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

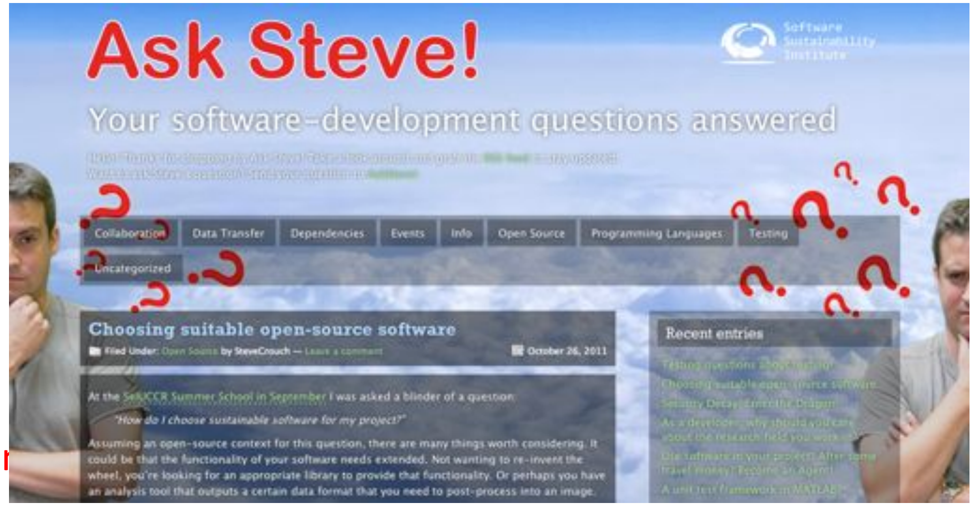
At Digital Research 2012, I presented a position paper with colleagues regarding the role of the Research Software Engineer. This paper followed on from a discussion I led at the Collaborations Workshop and some very interesting blog posts by Dirk Gorissen and Ilian Todorov. Rather than repeat these 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.



Choosing suitable open-source software

Filed Under: Open Source by SteveCrouch — Leave a comment October 26, 2011

At the SeUCCR 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

October 2013

Outcomes from the workshop for e-Infrastructure trainers

The workshop for e-Infrastructure trainers brought together 30 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.

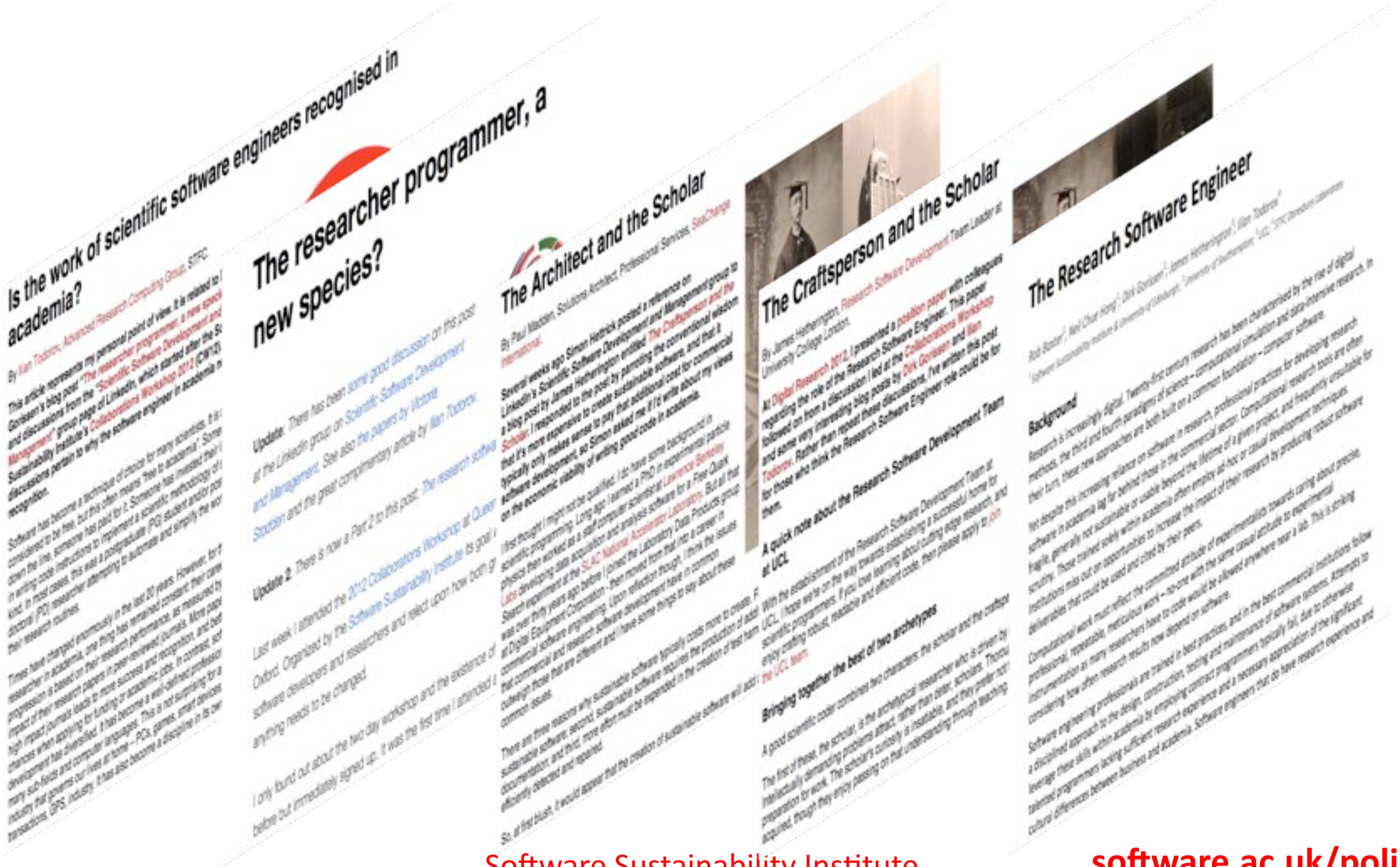
www.software.ac.uk

Software Sustainability Institute

software.ac.uk/policy

Campaigning for careers

www.software.ac.uk



Is the work of scientific software engineers recognised in academia?

By Ilian Todorov, Advanced Research Computing Group, STFC.

This article represents my personal point of view. It is related to … Gorissen's blog post "The researcher programmer, a new species" … and discussions from the "Scientific Software Development and Management" group page of LinkedIn, which started after the … Sustainability Institute's Collaborations Workshop 2012 (CW12) … discussions pertain to why the software engineer in academia … recognition.

The researcher programmer, a new species?

Update: There has been some good discussion on this post at the LinkedIn group on Scientific Software Development and Management. See also the papers by Victoria Stodden and the great complimentary article by Ilian Todorov.

Update 2: There is now a Part 2 to this post: The research softwa…

The Architect and the Scholar

By Paul Madden, Solutions Architect, Professional Services, SeaChange International.

Several weeks ago Simon Hettrick posted a reference on LinkedIn's Scientific Software Development and Management group to a blog post by James Hetherington entitled The Craftsperson and the Scholar. I responded to the post by parroting the conventional wisdom that it's more expensive to create sustainable software, and that it typically only makes sense to pay that additional cost for commercial software development, so Simon asked me if I'd write about my views on the economic viability of writing good code in academia.

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

The Research Software Engineer

Rob Baxter[1], Neil Chue Hong[1], Dirk Gorissen[2], James Hetherington[3], Ilian Todorov[4]

[1] Software Sustainability Institute & University of Edinburgh; [2] University of Southampton; [3] UCL; [4] STFC Daresbury Laboratory

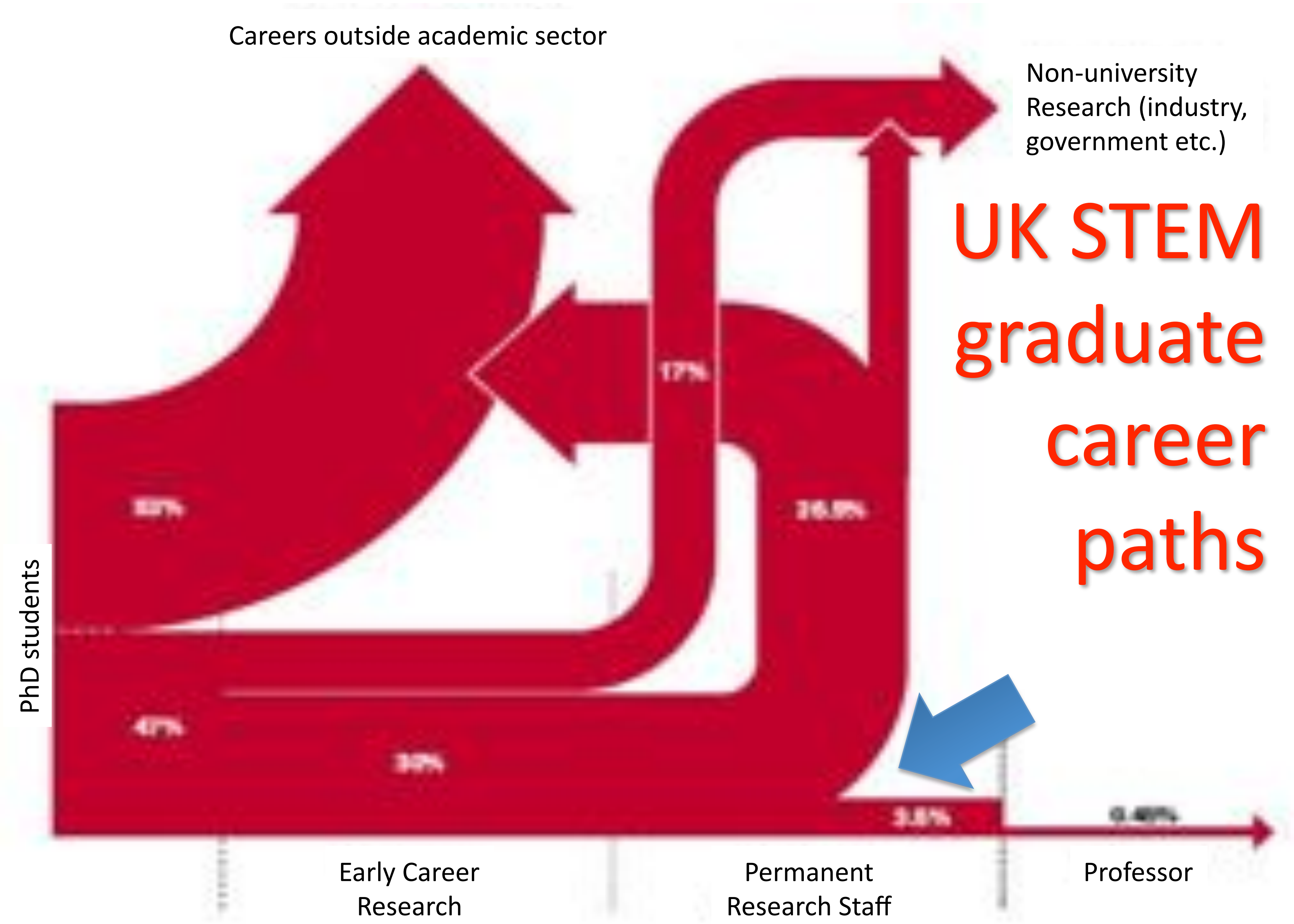
Background

Research is increasingly digital. Twenty-first century research has been characterised by the rise of digital methods, the third and fourth paradigms of science – computational simulation and data-intensive research. In their turn, these new approaches are both built on a common foundation – computer software.

Software Sustainability Institute

software.ac.uk/policy

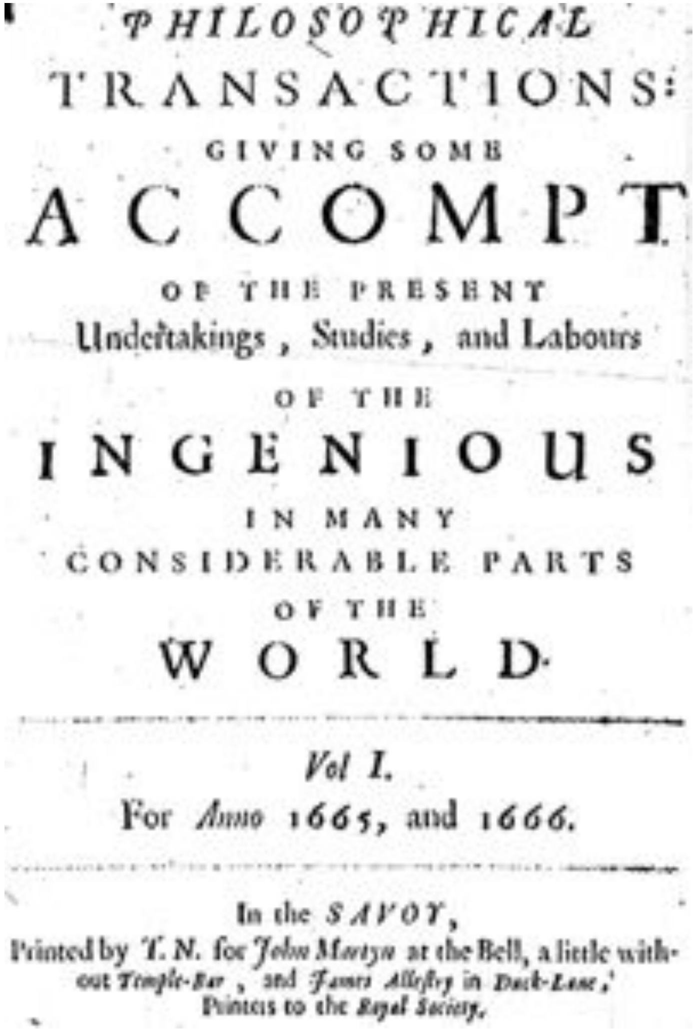
UK STEM graduate career paths



Source: The Scientific Century, Royal Society, 2010 (revised to reflect first stage clarification from “What Do PhD’s Do?” study)

Shake up the system

- *“Swim or drown”* is not an efficient learning method
- *“Publish or perish”* is not an effective reward mechanism
- *“Becoming a Professor”* is not a scalable career path
- *“I’ll just have to do it myself”* is not a modern way of doing science



PHILOSOPHICAL

TRANSACTIONS:

GIVING SOME

ACCOMPT

OF THE PRESENT

Undertakings, Studies, and Labours

OF THE

INGENIOUS

IN MANY

CONSIDERABLE PARTS

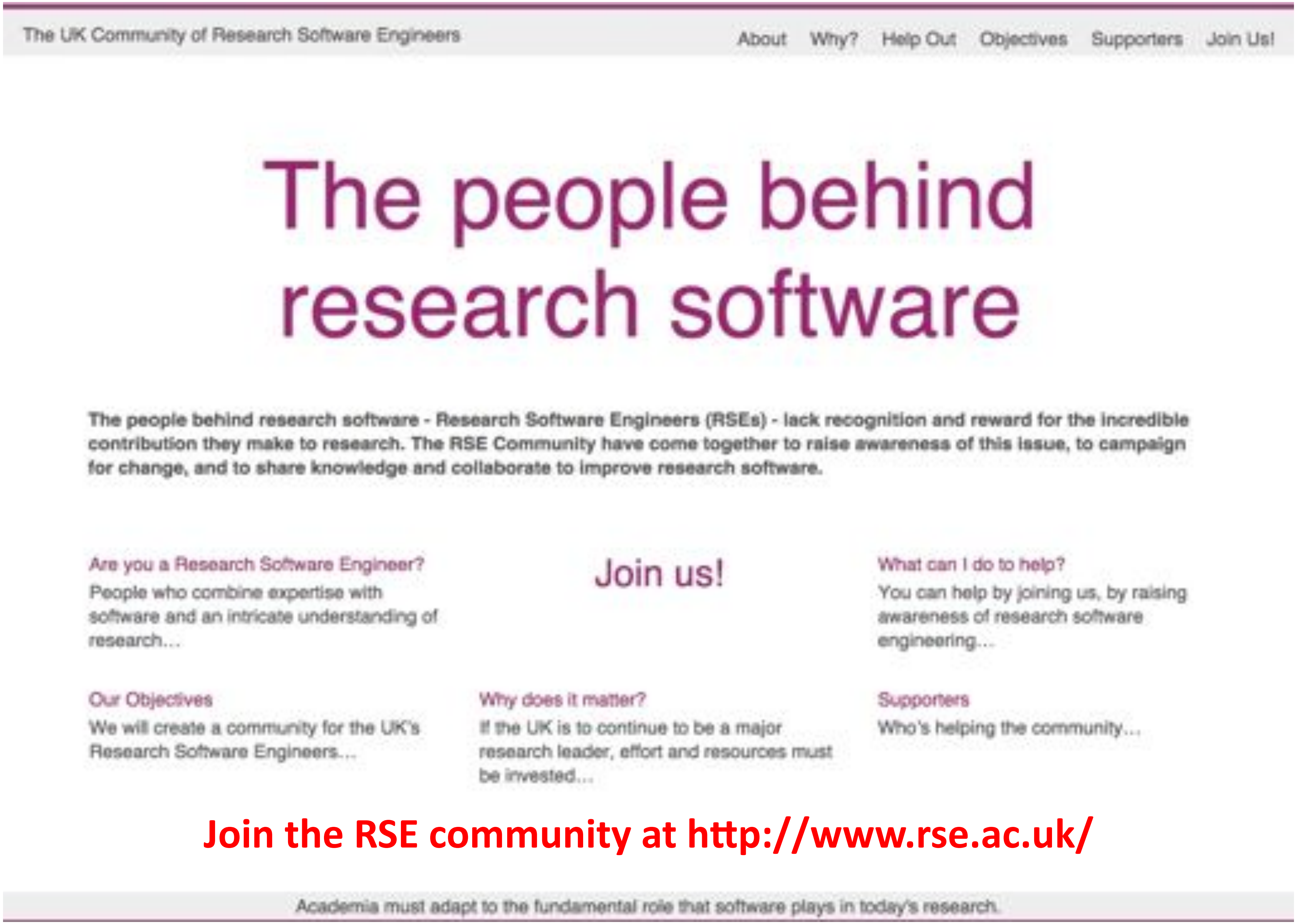
OF THE

WORLD.

Vol I.

For *Anno* 1665, and 1666.

In the *SAVOY*,
Printed by *T. N.* for *John Martyn* at the Bell, a little without *Temple-Bar*, and *James Alleſtry* in *Duck-Lane*,
Printers to the *Royal Society*.



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…

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

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

A national facility for cultivating world-class research through software

www.software.ac.uk

Some of our collaborations

HERIOT WATT UNIVERSITY

THE UNIVERSITY OF EDINBURGH

CCFE CULHAM CENTRE FOR FUSION ENERGY

WWF®

UEA University of East Anglia

clinical research facility EDINBURGH

CeRch Centre for e-Research

MANCHESTER 1824

software carpentry

Imperial College London

Science & Technology Facilities Council

UNIVERSITY OF OXFORD

University of BRISTOL

NGS

UCL

MICE

UNIVERSITY OF Southampton

Become our next collaborators!

Website: www.software.ac.uk

Email: info@software.ac.uk

Twitter: twitter.com/SoftwareSaved

EPSRC Pioneering research and skills

JISC

DEATH TO DECAY!
WWW.SOFTWARE.AC.UK