

UCL Research Software Development Team, Call for Proposals

Submissions are invited to a UCL internal call for proposals for research software development.

The UCL Research Software Development Team (RSDT) was launched in Autumn 2012 as part of the Research IT Services Department within the Information Services Division. This is a central team of professional software developers with particular expertise in designing, constructing and maintaining software for academic research. Our goal is to enhance UCL's capacity to produce high quality scientific software, from the simplest scripts to complex simulations running on state of the art high performance computers. We do this by collaborating with researchers who are creating their own software.

Research teams preparing grants involving software development should consider involving the team as collaborators, costing programming effort from our developers into the grant – we can provide fractional or full time effort as needed, throughout the project or at critical intervals. Please get in touch to discuss details of costs and processes, by emailing rc-softdev@ucl.ac.uk.

In addition, we will be providing a termly call for free software development effort in line with UCL's research priorities, and the first such call is now open. Proposals will be reviewed by our academic governing bodies, which are staffed by leading UCL computational academics. (See below). Proposals should be submitted **by 14th February**, emailed to rc-softdev@ucl.ac.uk following the guidance below. Each supported project will be provided with 50% FTE for three months from April to June 2013. Key staff from submitting groups should be available in the period 21 Feb-7 March for discussions. A similar call will be issued in April for effort in the third quarter.

The team are always happy, where time permits, to consult, at no cost, with researchers on issues relating to computational research, including support for computational grant submissions, help with recruitment of research programmers, and to provide advice on thorny software design or implementation questions.

We will be making further announcements as we develop our services with regard to software development infrastructure and teaching and training.

Dr James Hetherington -- Team Leader, UCL Research Software Development
Clare Gryce – Head, UCL Research Computing and Facilitating Services
Andrew Smith – Chair, UCL Research Software Development Executive
Prof. Anthony Finkelstein – Chair, UCL Research Software Development Board

Guidance on Submissions

Submissions should be made as follows:

- PDF document
- Submission by 14th February
- Email to rc-softdev@ucl.ac.uk
- Open format
 - Maximum 4 pages, 12 point text, including figures and references
 - addressing the points following
- Project Title
- Authors
 - status and departments
 - including all to be involved in the project, including:
 - research staff
 - students
 - academic staff
 - indicating author status and title
 - indicate lead author (PI)
 - indicate corresponding author(s) with UCL email address and preferred contact telephone number.
- Abstract of the points you consider most important in judging your proposal
 - This should be included **in the first page**
 - In the event of a very large number of applications, we do not guarantee that subsequent content will be read by all reviewers
- Introduction to research area
 - Assume a basic knowledge of the subject
 - As to a starting graduate student
 - Include key publications which can build a grounding in the field
 - Review the status of computational research in the area
 - References to any existing codes which complement or compete with the code being proposed for effort
- Introduction to research group
 - Authors' track record in the field
 - Recent publications
 - Computational experience of group
 - Software training levels, languages, and competencies of staff
 - Existing processes and tools used for organizing software development effort
- Introduction to code to be worked on
 - If a brand new code, envisaged answers can be given
 - Overview of code purpose and use
 - References to documentation or papers if available
 - High level description of code structure and design approaches
 - Suitable to help someone first looking at the code base
 - Elements used, with references to web or research literature where appropriate:
 - Languages
 - Libraries
 - Techniques and methods
 - Algorithms
 - Build tools
 - Testing and deployment tools
 - Provide URL links to existing code or email as attachment

- Engineering status of existing code, with impact on research
 - Degree of testing
 - Incidence of crash or wrong-answer bugs
 - Performance issues
 - Readability and structural soundness
- Suggested objectives for the project
 - Organised by priority
 - All must be justified in terms of research needs and impact
 - Why is this feature or improvement needed?
 - What will be the outcomes for the research group, for UCL, and for the field if this is implemented
 - Final objectives will be agreed in collaboration with the team if selected
 - Support for further requirements analysis will be provided as part of the project
 - Effort is provided on a defined-effort not defined-outcome basis
 - See example suggested objectives below
- Impact of the project
 - Potential for use of software beyond originating group
 - Development of software components, tools, insight or methods which could be of benefit to other research projects
 - Potential for receipt of otherwise unavailable research funding
 - Prevention of software falling into disuse or being forgotten
 - Development of skills within research group
 - Transfer of software engineering knowledge to active computationally-focused PhD students and postdocs
 - Unlocking potential for further development of code beyond engagement
- Justification for application
 - Justification for use of RSDT staff
 - See appropriateness section in selection criteria
 - Why are research software developers required as opposed to general programmers?
 - Why can't existing research staff do this?
 - Justification for use of free project
 - Existing chances of receiving funding for software development from other sources
 - Impact of project on those chances

Examples of suggested objectives:

There are many potential projects with which the team could be involved. To assist applying groups, we suggest here a few examples of objectives that might be included, though many additional possibilities beyond these are possible.

- New algorithms or methods
- Addition of new components to existing models
- Import/export tools for linking with other codes or loading new data formats (e.g. for multiscale modeling, data import/export)
- Performance improvements:
 - Performance and/or scaling objectives for the project in terms of time, memory or problem size
 - Parallelisation of existing serial code
 - Use of accelerators/GPGPUs
 - Vectorisation or replacement of inefficient high level code (e.g. Matlab, python, R...)
- Design and architectural effort
 - Redesign, tidying, or refactoring objectives
 - Replacement of poorly-understood legacy code with new, clear, tested implementations
 - Investigation of replacement of parts of the code with appropriate library use
- Testing, reliability and correctness improvements
 - Creation of automated test harness
- Usability improvements
 - New easy to use command line interface
 - Graphical interface
 - New user interface features
 - Documentation effort
- Deployability improvements
 - Availability of code on additional platforms or environments
 - Deployment and packaging of code base
 - Deployment to standard package servers

Selection Criteria

To assist in preparation of proposals, and for reasons of transparency, we present here the criteria which the UCL Research Software Development Executive will use in the selection of proposals.

Outcomes

The selected projects will likely result in:

- Excellence in research outcomes
 - Contributing to work leading to publication in leading journals
 - High-profile research leading to positive public, academic, and research council attention
 - Prevention of software falling into disuse or being forgotten
- Use of software beyond originating group
- Development of software components, tools, insight or methods which could be of benefit to other research projects
 - Particularly where these might lead to publishable research papers with RSDT involvement
- Receipt of otherwise unavailable research funding
 - Particularly where subsequent paid involvement of RSDT is likely
- Development of skills within research group
 - Transfer of software engineering knowledge to active computationally-focused PhD students and postdocs
- Unlocking potential for further development of code beyond engagement
 - E.g. where adding new features to code is currently made difficult by poorly understood and structured legacy code without tests

Appropriateness

Projects will be more likely to be selected which require:

- specialist developers with a background in research, with:
 - sophisticated understanding of algorithms, methods, or tools typical of research software e.g.:
 - Parallel programming
 - Numerical methods and libraries
 - Research information management expertise
 - Semantic web
 - Information visualization
 - Statistical analysis
 - an ability to understand advanced scientific or mathematical topics in order to properly understand the project requirements
- specialist developers with an understanding of professional software engineering with:
 - a commitment to engineering best practice, capable of managing complex interacting requirements to produce sustainable, clear, correct code
 - the freedom to focus on programming without distraction by publication pressure
- Cost-free software development
 - where funding for development is unlikely to be obtained from other sources
 - Groups with access to development funds should consider engaging RSDT developers through paid projects
- Demonstrably correct, auditable code

Applicants should note that this call should not be seen as a way of sourcing basic programming or systems administration skills, or extra research staff.

Governance structures

The UCL Research Software Development Executive will assess applications to this call. This body meets monthly to allocate the resources of the RSDT. The membership of the Executive is:

Andrew Smith (UCL Energy Institute) (Chair)
Filipe Abdalla (Astrophysics)
Ben Waugh (Physics)
Michail Stamatakis (Chemical Engineering)
Miguel Bernabeu Llinares (CoMPLEx and Centre for Computational Science)
Andrew Martin (Structural and Molecular Biology)
Timo Betcke (Mathematics)
Matt Clarkson (Centre for Medical Image Computing)
Earl Barr (Computer Science)

The Executive reports to a Board, which meets termly to set policy and provide strategic guidance for this initiative. The membership of the Board is:

Anthony Finkelstein (Dean, Engineering Sciences) (Chair)
Richard Catlow (Dean, Maths and Physical Sciences)
Peter Coveney (Director, Centre for Computational Science)
Simon Arridge (Centre for Medical Image Computing)
Erik Burman (Mathematics)
John Shawe-Taylor (Computer Science)

The Board reports in turn to the Research Information and IT Services Group (RIISG), UCL's senior governance group for research IT, which is chaired by the Vice-Provost for Research, David Price. See http://www.ucl.ac.uk/isd/staff/research_services/governance/riisg