Job Description – JD178 v1.3

Role Title	Senior Research Software Developer
Department/section	• Research Software Development Team • Research Computing and Facilitating Services
Base location	UCL Bloomsbury Campus, London
Grade	8
JDO Reference	Technical Specialist / Technical Operations NJ11-70 (grading ref NJ14-233)
Reporting to	Team Leader, Research Software Development
Direct reports	0-3 Research Software Developers

Works closely with	• Team and group colleagues
	• Research Software Development Team Leader
	• Head of Research Computing and Facilitating Services
	• Research teams from across UCL:
	  • Graduate students
	  • Research staff
	• UCL academic leaders in computational and data intensive research
	• Members of the following organisational units within Information Services Division
	  • Research Computing Platforms Team
	  • Research IT Facilitating Services team (outreach, training, advisory)
	  • Research Platforms team (production infrastructure)
	  • Research Data and Network Services group

Date updated	28 September 2016

Working context

Information Services Division (ISD) supports and enhances learning, teaching, research and a range of administrative processes across UCL.

ISD provides a wide range of information and technology related services to over 35,000 users: staff and students of UCL and associated institutions. Facilities and services provided by ISD include all core data centres, email, desktop, printing, networking and software/hardware purchasing. It also supports research computing platforms, e-learning and IT security across UCL.

The key administrative systems including HR, Finance and Student Information are also provided by ISD, as are website development and support, and creative design.

With some 300 staff the Division is structured into seven departments - **Technology Services**, **Applications Services**, **IT for Professional Services**, **IT Change and Project Delivery Services**, **IT for SLMS**, **Research IT Services** and **Learning Technology and Media Services**.

- **Technology Services** provides infrastructure and other technology services such as data centres, networks, server and virtualisation support plus the common IT service desk.
- **Application Services** provide application support and enhancement services including database administration.
- **IT for Professional Services** provides the technology and support for core UCL internal services such as Finance.
- **IT Change and Project Delivery Services (ITCPD)** owns the project delivery methodology and is responsible for delivering new and enhanced technology enabled change and services which support UCL's strategic goals. Centralised administration and communication services are also part of ITCPD
- **IT for SLMS** provides local support for users in the School of Life and Medical Sciences (SLMS) and is integrated into the rest of ISD. IT for SLMS includes some shared service groups covering infrastructure, research, learning & teaching and medical Illustration services which link into the corresponding groups within the rest of ISD.
- **Research IT Services** and teams in **Learning and Technology and Media Services** focus on two key missions of the university - research and teaching & learning. They aim to ensure services in these areas are world leading.
- **The Research IT Services Department (RITS)** includes two Groups: Research Computing and Facilitating Services and Research Data and Network Services, as well as having a close relationship with the Research Application Services Group within the Application Services Department.
- The **Research Software Development Team** is one of three teams within the **Research Computing and Facilitating Services** group, the other two being *Research Computing Platforms*, and *Facilitation* services (outreach, training coordination and advice).

Job purpose

- Research Software Developers work with UCL researchers to build and maintain readable, reliable and efficient research software.
- They collaborate with research colleagues from across UCL to construct, improve, and maintain codes used for modelling, analysis and simulation in UCL research.
- They are required to rapidly assimilate research context through publications and conversation with research groups, understand the computational algorithms, requirements and interfaces needed within the research effort, and construct high-quality software for research colleagues which will result in a sustained impact on their research programme.
- They work to improve reliability and performance for parallel codes, while maintaining readability and structure, on UCL's High Performance Computing (HPC) platforms, and other

external facilities to which UCL researchers have access, including national supercomputing facilities such as ARCHER.

- They teach UCL research staff and students the effective use of software for research, through leading hands-on training sessions covering topics in programming and software engineering best practice.
- They provide consulting on software practices, techniques, design, and architecture to research groups, helping to build well-structured and maintainable research software.
- They continually study new and existing technologies, tools and ideas in research computing, maintaining expertise in many areas of computational research.
- They build and maintain relationships within the research and e-Infrastructure communities in UCL and beyond, seeking opportunities to contribute to research, and to generate and prepare opportunities for new research projects and funding.
- They contribute to the wider ecosystem of support for computational research in UCL, working with departmental and group colleagues, departmental IT staff and other ISD colleagues to help build integrated systems and services which meet the needs of researchers.
- They maintain and support the state-of-the-art infrastructure and services needed for effective research software engineering, including continuous integration, version control, and code review.
- They support release and dissemination of UCL research software, through open source, scholarly, and commercial channels, ensuring codes are easy to deploy and install.
- They explain and document the software they help create, contributing to research publications and code documentation, and user support for codes they create and maintain.
- Senior research software developers provide tactical leadership for delivery of research software development services.
- They manage and mentor other team members ensuring the code they produce meets rigorous quality standards.
- They review and revise code and documentation produced by team members, amending as necessary to ensure consistency and quality.
- They monitor the teams' processes, metrics, and dashboards, ensuring that internal and external processes are followed, including coordination of code review and issue tracking.
- They contribute to the design of the team's ways of working, ensuring continuous improvement and adoption of appropriate techniques, technologies and tools.
- They manage external funding opportunities from opportunity through to successful funding, liaising with funding bodies, UCL Professional Services staff and Principal Investigators, preparing proposal costings and contributing to bid documents.
- They contribute to the development and design of teaching and training courses in research programming and related subjects.
- They coordinate the delivery of collaborative research programming projects, allocating team resources efficiently to ensure timely delivery of research outcomes.
- They ensure Research Software Development services are effectively embedded in Divisional processes and may act as Service Operational Manager for a subset of the Research

Software Development services, preparing documentation for Divisional projects, and liaising with Divisional colleagues.

- They represent RITS at University, national and international events, enhancing the recognition and reputation of the Department and UCL, and creating opportunities for collaboration.
- Working with the Team Leader, Head of Group, and academic leaders in the area, they also contribute to strategic leadership, and may deputise for these colleagues as necessary on strategic and governance bodies.

Main accountabilities and tasks

- The table below summarises the main tasks undertaken for this role.

Accountabilities	Key tasks	% time spent
Research programming	• Collaborate with research colleagues from across UCL to construct, improve, and maintain codes used for modelling, analysis and simulation in UCL research. • Rapidly assimilate research context and understand the computational algorithms, requirements and interfaces involved in a research programming project. • Design and construct high-quality software for research colleagues which will result in a sustained impact on their research programme(s). • Improve reliability and performance for parallel codes, while maintaining readability and structure, on UCL-owned and other High Performance Computing platforms. • Assist colleagues in analysis and problem-solving tasks, sharing knowledge and expertise with team members. • Manage and mentor other team members ensuring the code they produce meets rigorous quality standards • Review and revise code and documentation produced by team members, amending as necessary to ensure consistency and quality	25
	Project Leadership • Source and manage external funding opportunities from opportunity generation to successful funding, liaising with funding bodies, UCL professional services staff and principal investigators.	

	• Prepare proposal costings and contribute to bid documents. • Coordinate and lead the delivery of collaborative research programming projects, organising meetings, reviews and preparing interim reports.	
Software development leadership	• Monitor processes, metrics, and dashboards, ensuring that internal and external processes are followed • Coordination of code review and issue tracking. • Contribute to the design of the team's ways of working, ensuring continuous improvement and adoption of appropriate techniques, technologies and tools. • Contribute to strategic leadership, working with senior colleagues • Deputise as necessary on strategic and governance bodies.	10
Provide training relating to research software engineering	• Train UCL research staff and students in the effective use of software for research. • Develop and design teaching and training in research computing, suitable for a range of audiences with a very variable degree of computational experience. • Advise researchers on software practices, techniques, design, and architecture.	15
Maintain and enhance research software expertise	• Maintain expertise in many areas of computational research through both independent study and training courses. • Practice and enhance research programming skills by contributing to relevant open source projects. • Maintain and develop expertise in technical leadership and software development processes	10
Publish, document and support use of research software outputs	• Support release and dissemination of UCL research software, through open source, scholarly, and commercial channels. • Explain and document software created with the team’s involvement, contributing to research papers published in the academic literature, project reports and case studies, and code documentation and manuals.	10

	• Provide online and face-to-face user support for software they create or maintain to both UCL and external users.	
Engage with and contribute to wider research software community	• Build and maintain relationships within the research and e-Infrastructure communities in UCL and beyond, actively seeking opportunities for collaboration with researchers including programming and grant preparation. • Contribute to community activities such as seminars and networking events. • Seek out and develop opportunities for new research projects and funding. • Attend conferences and community events in a variety of software engineering/research computing fields in the UK and abroad. • Represent RITS at University, national and international events, enhancing the recognition and reputation of the Department and the University, and creating opportunities for collaboration. • Contribute ideas, experience and thinking to technical working groups in and beyond UCL.	15
Maintain and support research software development infrastructure and services	• Maintain and support the state-of-the-art infrastructure and services needed for effective research software engineering, in areas such as continuous integration, version control, and code review. • Maintain and manage systems and servers used to deliver software development infrastructure services. • Author and maintain documentation relating to software development infrastructure services. • May act as Service Operational Manager for one or more Research Software Development services.	10
Contribute to departmental and divisional activities	• Assist Research IT Services colleagues in the delivery of other departmental services. • Contribute to the wider ecosystem of support for computational research in UCL, working with departmental IT staff and ISD colleagues to help build and maintain integrated systems and services that meet the needs of researchers. • Contribute to wider departmental and divisional activities through discussions and meetings. • Prepare and maintain project management documentation for divisional projects	5

- Liaise effectively with divisional colleagues on security, platforms, project management and design matters.

Supervision and/or line management of the team.

- Provide supervision and leadership and/or line management to up three Research Software Developers, which may include: performance management and coaching, objective setting, identification and follow-up of training and development needs, and appraisal.
- Coordinating and allocating team resources efficiently to ensure timely delivery of projects and other tasks.

Person Specification

Essential

1. PhD degree in a computationally based field, or equivalent professional experience.
2. Experience as a computational researcher, including authorship on multiple relevant research publications.
3. Significant experience of using and developing scientific applications to produce research outputs.
4. Ability to rapidly acquire fluent knowledge of new programming languages, libraries and platforms.
5. Experience of analysing, researching and solving complex programming problems.
6. Experience using Unix-based operating systems and Unix system tools and utilities.
7. Advanced skills in C++, C#, Java and/or Fortran 2003.
8. Experience with the advanced use of high-level dynamic languages for numerically-intensive research, preferably Python, Julia or R.
9. Advanced knowledge of applied mathematics.
10. Excellent written and verbal communication skills including the ability to effectively present complex or technical information to a range of audiences.
11. Ability to work collaboratively and as part of a team.
12. Ability to work under own initiative.
13. Desire to keep up-to-date and learn about new developments in computational research.
14. Knowledge of and commitment to software development best practise including issue tracking, testing, documentation, version control, and continuous integration.
15. Experience of technologies supporting software re-use and deployment.
16. Knowledge of and experience with object-oriented design, design patterns and refactoring.
17. Experience designing and/or delivering computational teaching or training courses.

18. Expertise in several specialist areas of technical computing from the list below. While no single specialism is essential, candidates must be able to demonstrate completion of significant work using several of these technologies:
    - Parallel programming with OpenMP or MPI
    - Accelerators: CUDA, OpenACC, OpenCL
    - Partitioned global address space: UPC or Co-Array Fortran
    - Build tools: preferably CMake
    - Parallel debuggers and profilers
    - Parallel visualisation, high performance image processing, computational geometry
    - Parallel numerical algorithms and libraries
    - Inter-language binding technologies such as Cython or Swig
    - Parallel programming with high level languages (e.g. MPI4py)
    - Highly scalable databases, both relational and/or NoSQL
    - Cloud computing, containerization and virtualization (e.g. Docker, MS Azure, Vagrant)
    - Map/Reduce, Hadoop, Spark, HDFS
    - Semantic Web, RDF, OWL, SPARQL
    - Devops for management of scalable infrastructure: Puppet, Chef, Ansible…
19. Experience of working in a team using Agile processes, such as SCRUM or XP.
20. Proven ability to forge effective professional relationships at all levels
21. Experience of open source software practices, with at least one accepted contribution to an open source project
22. Proven ability to manage multiple concurrent tasks and activities, working to deadlines and prioritising as appropriate
23. Experience mentoring and leading other research programmers (formal line management experience is not essential, but such candidates should be able to show they have effectively guided the work of more junior colleagues)

Desirable

1. Operational experience maintaining actively used systems and services, including change and problem management
2. Advanced software process and project management expertise, with knowledge of multiple process frameworks and theories, from a variety of cultures (SCRUM, XP, Lean, PRINCE II, ITIL), with an understanding of commonalities and differences.
3. Knowledge of and interest in software leadership, demonstrable through awareness of appropriate books, websites or blogs
4. Experience as a leader/coordinator of agile software management processes, for example as a "scrum master".
5. Understanding of commercial aspects of the software industry, with experience of organisations of multiple maturity levels and sizes
6. Understanding of research and academic funding sources, with authorship on one or more research grants

7. Demonstrable leadership in one or more software communities, such as through membership of standards bodies, organisation of community events, or a lead maintainer role for open source tools or packages.
8. Recognition as a leader or senior practitioner in research programming, with involvement in appropriate community or professional bodies

Special working conditions

- None

Conditions of Service

- The normal hours of work are 36.5 hours per week. However, this is a senior post and flexibility will be expected in response to varying workload. Reasonable notice will be given and where properly authorised such work will be recompensed as stated in the terms and conditions. The annual leave entitlement is 27 days per year, plus 6 College closure days, plus public holidays.

Additional Information

- The job description reflects the present requirements of the post, and as duties and responsibilities change/develop, the job description will be reviewed and be subject to amendment in consultation with the post holder. The post holder will carry out any other duties as are within the scope, spirit and purpose of the job as requested by the line manager or Head of Department/Division.
- The post holder will actively follow UCL policies including Equal Opportunities policies and be expected to give consideration within their role as to how they can actively advance equality of opportunity and good relations between people who share a relevant protected characteristic and people who do not share it.
- The post holder will maintain an awareness and observation of Fire and Health and Safety Regulations.
- The post holder must ensure organisational compliance, and conformance with the Data Protection Principles. All data, whether stored electronically or by other means must be processed in accordance with the Data protection Act 1998.
- The post holder will attend staff meetings and training as required.
- The post holder will maintain their knowledge and skills through professional activities with agreement from their line manager and readership of relevant publications.