Software and research

Collaborate, share and learn how to develop sustainable scientific software

Nowadays, software and research go hand-in-hand. To advance their research, scientists not only use but also develop their own software packages. This, supporting scientific software development is essential and needs to focus on helping scientists produce robust, reliable and maintainable software. As data policies are increasingly considering software produced for research purposes as 'data', (e.g. the AGU's data policy), this is a topic relevant to Earth and space sciences community. The EGU journal Geoscientific Model Development in particular has taken a lead by requiring that authors make code available and accessible to potential users.

Researchers are not left alone with the problems they may face with software in their research. There are organisations specialising in supporting them, such as the UK-based Software Sustainability Institute. The Institute is based at the universities of Edinburgh, Manchester, Oxford and Southampton, and draws on a team of experts with a breadth of experience in software development, project and programme management, research facilitation, publicity and community engagement.

The Institute is a national facility for cultivating world-class research through software. For example, it provides software engineering consultancy for research projects that develop research code. Community outreach activities include the annual Collaborations Workshop and the Fellowship Programme. The Workshop brings together researchers, software developers, managers, funders and more to explore important ideas in software and research and to plant the seeds of interdisciplinary collaboration. The Fellowship Programme funds researchers to organise or attend events or meetings in exchange for their expertise and advice. The Software Sustainability Institute also delivers training in software skills for researchers and runs a variety of campaigns to support software in research, such as the Research Software Engineer campaign.

At the 2013 AGU Fall Meeting, four Institute fellows (Allen Pope, Melody Sandells, Kayla Iacovino, and Robin Wilson) proposed and convened a townhall meeting to discuss topics related to software in research. The session had about 50 attendees ranging from scientists developing their own code to software engineers working alongside researchers. Marco Tedesco, Program Director at the US National Science Foundation's Polar Cyberinfrastructure Program, kicked off the meeting by highlighting the three main topics for the session: collaboration strategies and technologies, software training for researchers, and code reuse, sharing and publishing.

The diverse audience allowed for exploration of the topic of software in research from different perspectives. The outcomes of the discussion further echoed many of the experiences that the Institute has had so far. Undoubtedly, these topics and issues are familiar to a number of researchers in the geophysical sciences.

Collaboration strategies and technologies

A wide variety of tools are used for collaboration: Google Drive, Dropbox, Skype and the old-school email. In general, collaboration on the same resources works well, but many kinks are still in the works. Some institutions don't allow using certain popular tools among scientists (e.g. Skype), some tools don't scale up well (e.g. large Google Hangouts), and some researchers don't have the budget in place for video teleconferencing. A common problem shared was transferring terabytes of input data to collaborators: as someone commented: "We mail the hard drives."

Software training for researchers

All agreed that software training is essential for anyone doing research and should begin as early as possible in a researcher's career. Software engineering skills must be taught, otherwise researchers are set up to write code that is unstable, difficult to maintain and has security gaps. Training should not be limited only to a programming language. Students should be taught good practices on structuring, commenting and testing code. There are many training materials available online, but without basic training in software development, students will not be able to make good use of them. It was suggested this was a ripe area for a workshop at future conferences.

Code reuse, sharing and publishing

Version control for managing code turned out to be a familiar solution with the session's attendees. Many used GitHub, but some noted that version control can be a bit of a headache. The time you spend figuring it out is usually precious time taken off research. The attendees also discussed the benefits of code review and different ways it could be done: from pair programming, to group meetings where the code is analysed line by line. Everyone was in favour of source code being published alongside the papers it was used for, following the example of journals such as Computer Physics Communications.

The session showed that exchanging ideas and sharing experiences helps to learn how to address the issues related to software and research. A lot has been done but there is still a lot to do. There are a number of initiatives helping scientists in getting trained up to develop sustainable software (such as Software Carpentry).

To continue and broaden this discussion, the Software Sustainability Institute is conducting a follow-on survey. We encourage all EGU members to please take five minutes to participate at http://tinyurl.com/AGU13software.

Aleksandra Pawlik[1], Allen Pope[2], Mel Sandells[3], Kayla Iacovino[4], Robin Wilson[5]

[1]Software Sustainability Institute, UK

[2]Earth Sciences Department, Dartmouth College, USA and National Snow and Ice Data Center, University of Colorado, Boulder, USA

[3]Department of Meteorology, University of Reading, UK

[4]Geography Department, University of Cambridge, UK

[5]Geography & Environment, University of Southampton, UK