Working towards Sustainable Software for Science: on the creation, maintenance and evaluation of open-source software

blogs.lse.ac.uk/impactofsocialsciences/2014/07/10/working-towards-sustainable-software-for-science/

7/10/2014

Alongside research papers and data, software is a vital research object. As more become confronted with its significance in the future of scientific discovery, a variety of opinions and philosophies are emerging over how to approach sustainable scientific software development. **Matthew Turk** *provides background on his involvement in the Working towards Sustainable Software for Science: Practice and Experiences (WSSSPE) workshops and the launch of the special collection from the Journal of Open Research Software.*

The advances in computer hardware over the last decade have been absolutely staggering – and scientists have been aggressive about applying these advances to the process of discovery. iPads have famously been shown to be more powerful than the supercomputers from only a few hardware generations past. Cloud computing has made available essentially limitless storage and computing power on demand. The connections between individuals that have been enabled by the Internet have accelerated scientific communication, enabled new types of discourse, and provided the opportunity to directly transfer data and technology between researchers from diverse fields and backgrounds. And perhaps most importantly for younger researchers and those inexperienced in computation, high-level languages and packages have enabled more direct development of tools, reducing the hurdles necessary to bring these developments to bear on a scientific problem.

And yet … have these advances been utilized most efficiently? A growing movement has begun to question whether or not the development of software is being properly applied — are students being adequately trained? Are we fostering software that will persist beyond the end of a funding cycle, or is it essentially disposable? Do we know how to develop communities that encourage reuse, that build ties, that improve reproducibility and development of software? The current estimates for the cost of a productivity on a GPU are roughly 10^-17 dollars per FLOP; on the flip side, the cost of a single second of human labor is roughly $0.002. Shepherding investments in software must be addressed from both the perspective of the human costs as well as the hardware costs.

Gift economy. Image credit: Klallam people at Port Townsend (Wikimedia, Public Domain)

Even as we begin this discussion, of “sustainability” of software, it is evident that different people have different opinions of what this means. Does sustainability relate only to bug fixes, or does it include feature enhancements as well? How do we measure the sustainability of a community, where researchers change funding streams and migrate between groups and topics over time?

At Supercomputing in 2013, a meeting was held entitled “Working towards Sustainable Software for Science: Practice and Experiences,” or WSSSPE (pronounced “Whispy”) for short. In advance of the workshop, contributions were solicited that addressed topics in a variety of subject areas:

- the development process that leads to new software
- the support, maintenance and usage of software
- the role of open source communities or industry
- policy issues relating to developing sustainable software
- education and training

The papers were requested to be deposited in open repositories, such as FigShare or the arXiv, and then were sent to program committee members for selection and review. At the workshop itself, keynotes were given and lively discussions were held around the topics in the papers — ranging from the role of research software engineers, to career paths, to the reuse of software between domains, and even the development of stable sources of revenue for projects seeking to grow beyond the end of a grant lifecycle. The participation in the workshop — from the lively discussions, to the flow of tweets, and even the comprehensive collaborative note taking — demonstrated the clear need for even having these conversations. Even though many different opinions and philosophies about how to approach sustainable scientific software development were expressed, the level of participation and excitement clearly showed that people are eager to try to find paths toward sustainability, and sharing those paths with others.

Following the workshop, authors and participants were invited to submit articles to the *Journal of Open Research Software* (JORS), to be included in a special collection of papers from the WSSSPE community. JORS, a relatively new, open access journal dedicated to the publication of research software — articles which may not otherwise have a clear home in other publications — was a natural fit for collecting articles from the workshop. After submission, the articles were subjected to peer review before being selected for inclusion in the collection. I’ve read each of these articles, and I’m pleased with the contribution each has to make — from experiences in running collaborative code, to suggestions for developing sustainable funding models, and even proposals for new frameworks for discussing and measuring the impact of software. The collection also includes a comprehensive overview of the workshop, detailing points of discussion, individual papers, and broader impressions of potential future work.

This collection also launches a new section in the *Journal of Open Research Software* entitled “Issues in Research Software.” These full-length research papers cover different aspects of creating, maintaining and evaluating open source research software. The aim of the section is to promote the dissemination of best practice and experience related to the development and maintenance of reusable, sustainable research software. We will continue to publish Software Metapapers describing open-source software with high reuse potential.

I hope that you find the papers in the collection as interesting, exciting and most importantly useful as we have found the process of reading, discussing and selecting them. On Thursday, July 10, the second WSSSPE workshop (“WSSSPE1.1”) will be held at the SciPy conference in Austin, TX, focusing on efforts by the Scientific Python community to develop sustainable scientific software. And then, this fall, WSSSPE2 will be held at Supercomputing 2014 — papers for this workshop will be solicited, but with a slightly different twist. They will be focused on calls to action, on developing plans to move the discussion forward, and to implement change however possible. At the

WSSSPE website, there is more information, including how to participate, how to submit papers, and to attend the workshops. I hope to see you there!

Note: This article gives the views of the authors, and not the position of the Impact of Social Science blog, nor of the London School of Economics. Please review our Comments Policy if you have any concerns on posting a comment below.

About the Author

Matthew Turk *is a Research Scientist at the National Center for Supercomputing Applications at the University of Illinois. Prior to that, he conducted postdoctoral work at Columbia University and University of California San Diego, and received his PhD in 2009 from Stanford University.*