Submission in Response to NSF CI 2030 Request for Information

DATE AND TIME: 2017-04-05 14:10:55
REFERENCE NO: 250

Author Names & Affiliations

- Frank Löffler - Louisiana State University
- Daniel S. Katz - University of Illinois Urbana-Champaign
- Lucas A. Wilson - University of Texas at Austin
- Sandra Gesing - University of Notre Dame
- Damon McDougall - University of Texas at Austin
- Jeffrey Carver - University of Alabama
- Steven R. Brandt - Louisiana State University

Contact Email Address (for NSF use only)

(Hidden)

Research Domain, discipline, and sub-discipline

Computational and Data Science, as applied to all science and engineering fields

Title of Submission

Research Software Training Initiative: Identifying and addressing challenges in scientific software development

Abstract (maximum ~200 words).

Software is pervasive in research. Software is a research modality, like an instrument. It needs to be done right, or the underlying research would not be possible, or even worse, would produce incorrect results. Research software ranges from use of commercial packages or software associated with instruments, to spreadsheets, scripts and programs that are used in data analysis and modeling, and includes software used for long-running high-performance computing simulations. Proper development and use of software is needed for research integrity, reproducibility and trust in the findings. This is equivalent to proper design and use of an instrument in an experiment. We believe that it is vital to underline this importance as software is used more and more within many scientific domains, and to provide suggestions on how funding agencies can help.

Question 1 Research Challenge(s) (maximum ~1200 words): Describe current or emerging science or engineering research challenge(s), providing context in terms of recent research activities and standing questions in the field.

Many researchers develop or modify software as part of their research. In addition, different tools are often combined to form larger frameworks and workflows. This effectively generates new software even if it is not perceived as such. In all these cases, it is important for researchers to use appropriate software development methods and tools. When researchers do not use such approaches a number of

problems arise, including: (1) less confidence in the correctness and reproducibility of the results of the software; (2) increased difficulty in maintaining the software; and (3) less chance of reusing and extending the software for future research. Using appropriate development methods for software is analogous to the proper manufacture of physical research instruments.

Of all scientists that develop software as part of their research, only few receive formal training in proper development and usage techniques. Yet, more and more researchers are faced with the challenge of having to either develop their own software, or having to re-use poorly documented and tested software that was developed by fellow researchers. Lack of training in handling of essential research tools is slowly being recognized as a problem by academic institutions, but progress so far has been slow and is not widespread.

Reasons for the unsatisfying slow progress include: (1) low visibility and understanding of the risks of poorly used or poorly designed software for science; (2) low recognition of research software contributions (e.g., code, documentation, testing, and training) as valuable products for publication, citation, and career development; (3) misunderstanding that training others in the best practices in sustainable scientific software development would necessarily require new courses that must fit in already-full curricula, and thus compete with existing courses.

Question 2 Cyberinfrastructure Needed to Address the Research Challenge(s) (maximum ~1200 words): Describe any limitations or absence of existing cyberinfrastructure, and/or specific technical advancements in cyberinfrastructure (e.g. advanced computing, data infrastructure, software infrastructure, applications, networking, cybersecurity), that must be addressed to accomplish the identified research challenge(s).

One way to address the challenge described above is to provide sufficient training in software development and use, if possible as integral part of the domain science training itself. A practical way to achieve this is by making changes at individual institutions. Progress in this direction can be seen in a few places, is usually driven by interested individuals, but is far too slow in general. This process must be accelerated and includes underlining the risks of the lack of proper training and raising the perception of the importance of software for research in general. NSF may not be able to provide sufficient direct funding, but it has excellent instruments to incentivize the necessary changes through the science projects it sponsors.

NSF could judge individual projects that include the use or development of software, in part, on how well project participants have been or will be trained in the use and development of software. Language could be added to specific solicitations that typically involve software best practices. For example:

1. How will project participants be trained in the development and/or proper use of software? (e.g., by using best software practices in core science courses)

2. How does the project assure the usability of the software for the targeted community? (e.g., by collecting requirements from the user community and by ensuring that it is actively part of the design process)

3. How will software results be tested and validated, by whom?

4. How does software support research transparency and reproducibility? (e.g., working with software versioning and revision control systems, data repositories)

Asking sponsored research investigators to answer these or similar questions alone raises the awareness that this is indeed a problem and needs to be addressed. Moreover, it retains the freedom of individual investigators to find and propose a solution that works best for their project. For little effort from the NSF and moderate, but acceptable effort from the proposers, the awareness of proper software development and use would be raised. One of the expected effects would be better quality scientific software, resulting often in more sustainable scientific software.

A similar approach was taken in the “data management plan” of NSF proposals, which includes the management of software. However, it falls short when it comes to ensuring proper training of developers and users. It is hard to find a data management plan, or whole proposal, that describes how project participants have been or will be trained in the use of the software that is proposed to be used or developed, since this is not required.

One viable way to make sure project leaders take an active role in developing respective training plans is to make such a plan a necessary part of a proposal that involves software. In addition, this plan should be made public for funded projects. This would enable the science community to examine the sustainability plan and compare with the actual implementation, as well as learn from those projects that already excel in software quality. The latter suggestion of making this document public also applies in the data management plan, and is discussed in another response to this RFI, titled “Sustaining Software as a Key Element of Cyberinfrastructure” and led by Daniel S. Katz.

To summarize, we have highlighted the necessity of software use and development for conducting scientific research. We also identified several concerns that arise when scientific software development or use is not executed in line with software engineering best practices. To address the concerns we identified, we outlined a solution: individual institutions must provide training infrastructure, either dedicated or deeply integrated into their curriculum. At some institutions this solution is already implemented, but we note that progress is slow. Therefore, to promote growth, it is necessary for the NSF to incentivize this change by requiring sponsored researchers to answer questions that lay out how software will be used and developed in a way that produces correct, verifiable, maintainable, and sustainable software.

Question 3 Other considerations (maximum ~1200 words, optional): Any other relevant aspects, such as organization, process, learning and workforce development, access, and sustainability, that need to be addressed; or any other issues that NSF should consider.

To show explicitly that the community is committed to the comments made here, the following community members are signing off on this response:

Ricardo Taborda
University of Memphis
Computational Seismology and Earthquake Engineering

Mark Miller
University of California, San Diego
Systematics and Molecular Biology

Christopher Paciorek
UC Berkeley
Statistics, Environmental Science

Matthew Parno
Cold Regions Research and Engineering Laboratory
Bayesian Statistics, Applied Mathematics

Shawn McKee
University of Michigan
High-energy Physics

Shyue Ping Ong
University of California, San Diego
Materials Science

Carl Boettiger
University of California, Berkeley
Ecology and Environmental Science

Roberto De Pietri
Parma University, Italy
Relativistic Astrophysics, Gravitational Physics

Erik Schnetter
Perimeter Institute for Theoretical Physics, Waterloo, Ontario, Canada
Theoretical Physics, Gravitational Physics

Bruce Berriman
Caltech/Infrared Processing and Analysis Center
Astronomy, Image Processing

Jorge Pullin
Louisiana State University
Gravitational physics

David Hudak
Ohio Supercomputer Center
High Performance Computing and Cyberinfrastructure

Jonah Miller
Perimeter Institute for Theoretical Physics, Waterloo, Ontario, Canada
Theoretical Physics, Gravitational Physics

Katherine Lawrence
University of Michigan
Organizational Behavior

Elbridge Gerry Puckett
University of California, Davis
Computational fluid mechanics

Charles Torre
Utah State University
Gravitational Physics, Mathematical Physics, Symbolic Computation

Richard Furnstahl
The Ohio State University
Theoretical Nuclear Physics

Francisco Guzman
Universidad Michoacana, Mexico
Astrophysics, Computational Physics

Chih-Jen Sung
University of Connecticut
Mechanical Engineering

Magali Billen
University of California, Davis
Earth mantle and lithosphere dynamics