Stewardship and Sustainability of Office of Science Software through a Scientific Software Hub

Anshu Dubey, Mathematics and Computer Science, Argonne National Laboratory
Katherine Riley, Argonne Leadership Computing Facility, Argonne National Laboratory
Nicholas Schwarz, Advanced Photon Source, Argonne National Laboratory
David E. Bernholdt, Computer Science and Mathematics and Oak Ridge Leadership Computing Facility, Oak Ridge National Laboratory
Bronson Messer, Oak Ridge Leadership Computing Facility, Oak Ridge National Laboratory
Mathieu Doucet, Neutron Scattering Division, Oak Ridge National Laboratory
Rama K. Vasudevan, Center for Nanophase Materials Sciences, Oak Ridge National Laboratory
Deborah Agrawal, Computing Research Division, Lawrence Berkeley National Laboratory
Katerina Antypas, National Energy Research Scientific Computing, Lawrence Berkeley National Laboratory
Harinarayan Krishnan, Advanced Light Source/Computing Research Division, Lawrence Berkeley National Laboratory
Edward Balas, Energy Sciences Network, Lawrence Berkeley National Laboratory

Executive Summary

We envision a multi-institutional "hub" to provide focus, coordination, and core capabilities for the stewardship and sustainability of the U.S. Department of Energy Office of Science (SC) software ecosystem. The hub will engage with the full range of stakeholders, including the producers and consumers of research software, facility staff, institutional leadership, and SC sponsors to coordinate the stewardship of the SC software ecosystem.

The hub will start by working with stakeholders to establish a rubric for identifying what software capabilities *need* to be sustained and determining the most effective ways to satisfy those needs. This is timely in view of the large investment in development of software technologies (ST) under the Exascale Computing Project (ECP). ECP ST is a test-bed that can be very effectively used to inform this rubric, which can then be applied to identify specific software capabilities and paths forward, again in conjunction with the full set of stakeholders.

Moreover, we will build on efforts such as the ECP Software Development Kits (SDKs), the Extreme-Scale Scientific Software Stack (E4S), and the Interoperable Design of Extreme-Scale Application Software (IDEAS) subprojects, which have demonstrated success in most areas of the proposed ScSH hub, especially the curation of methodologies, education and outreach around software development practices, and facilitation of software process improvements with development teams.

The remainder of the document is organized in the form of Heilmeier-catechism**,** addressing the questions, What are we trying to do?, How is it done today, and what are the limits of current practice?, What is new in our approach, and why do we think it will be useful?, Who cares? If we are successful, what difference will it make?, What are the risks?, How much will it cost?, How long will it take?, and What are the mid-term and final "exams'' to confirm success?

What are we trying to do?

The goal of the **Scientific Software Hub (ScSH)** is to coordinate the stewardship and promote the sustainability of scientific research software important to the DOE Office of Science (SC) to provide a robust software ecosystem that supports the needs of SC through comprehensive engagement with stakeholders (SC-funded software developers and users; computing, experimental, and observational facilities; and DOE program managers). This goal is aligned with the recommendations from the ASCR/ECP transition report. The ScSH will pursue initiatives in training and education in software best practices, software engineering, and project management; advocate for career paths for software professionals within and beyond the hub; and study the science of research software to identify the most effective techniques for ensuring the quality of this software over time.

How is it done today, and what are the limits of current practice?

- **No mechanism currently exists for stewardship or ensuring the sustainability of the SC software portfolio.** Research software used in SC programs originates from sources both within and outside SC. There is no mechanism to engage the stakeholders to understand what research software is central to addressing the DOE mission or the state of those packages with respect to their quality and sustainability. Exacerbating the situation, software is often produced as a *research* tool but is adopted by others as a *production* resource, with the producers and consumers (and their sponsors) having very different goals and expectations for the software. Bridging such gaps is currently done piecemeal (or not at all) and without a broader understanding of the SC software ecosystem.

  Arguably, selected pockets within SC do exist that give more consideration to the sustainability and distribution of software and provide a degree of stewardship. Examples include the Extreme-scale Scientific Software Stack (E4S) coming from the Exascale Computing Project (ECP), and the Computer Science and Applied Math Institutes of the Scientific Discovery through Advanced Computing Program (SciDAC). Compute, experimental, and other facilities budget some staff and funding for software necessary for operations and users' research.The ORNL Neutron Sciences Directorate (SNS and HFIR facilities) has partnered with the Research Software Engineering Group in the Computer Science and Mathematics Division of ORNL to address stewardship concerns for the SNS data reduction software. The High Energy Physics Center for Computational Excellence (HEP-CCE) is investigating the use of ASCR supercomputing resources for a set of data-intensive computational tasks, where the effort is based around a set of pilot projects and software technologies that emphasize portability across architectures. Most such efforts tend to be localized to particular programs or projects, however; and where the development or use of software crosses those boundaries (which happens often and should be considered positive), the individual teams are left to coordinate as best they can, usually without full awareness of the best practices for sustainable software.

- **The production readiness and sustainability of SC software vary widely.** As pointed out above, a great deal of the software in the SC ecosystem is produced in the context of addressing *research* goals. Different research teams have different perspectives on their software. Some recognize the long-term value of producing a high-quality, sustainable research tool, if only for their own use, and will utilize effective software development practices to achieve those goals. For others, the software is a means to a scientific result and not of high concern in and of itself. From the perspective of sponsoring science programs, both teams may produce laudable results; but from a software perspective, results may vary widely. Moreover, the more the software becomes a "black-box" tool in support of the scientific research of others, the greater the need for higher levels of quality and sustainability.

  In part these outcomes stem from the incentive structures both in individual institutions and in SC funding programs that tend not to place value on software quality and sustainability; and in part the outcomes stem from the lack of education and training of software developers in the best practices to produce higher-quality and more sustainable software.

- **SC software developers are not equipped with the knowledge and support needed to ensure high-quality, sustainable software.** Software design, maintenance, and re-engineering are often overlooked in the formal education of the SC workforce. While many resources are available on software engineering in general, the engineering of scientific research software is often a different experience from that of a consumer software package or an enterprise business system. Efforts are needed to research practices that work effectively for scientific software and to tailor them to the needs of our community. Similarly, stronger engagement of software professionals (e.g., "research software engineers") is needed in order to contribute to the development and sustainment of important software products. Both training and the involvement of software professionals are needed to disseminate the necessary knowledge and guide developers in adopting effective scientific software practices.

  Currently, a small number of projects are aimed at the identification and dissemination of best practices for the development and sustainment of scientific software. Within DOE, the Interoperable Design of Extreme-Scale Application Software (IDEAS) project has been supported variously by ASCR, BER, and ECP in such efforts, including curating methodologies, providing trainings and outreach, and working with particular software teams on software process improvement. Similarly, software professionals, particularly research software engineers (RSEs), are relatively new to the SC software ecosystem. Some laboratories and universities have formal RSE organizational units and recognized professional tracks; but they are far from universal, and where they exist, they are typically new and just beginning to engage with scientific software developers at their institutions. Researchers in the ASCR computer science and applied math programs also need to be engaged to help ensure that key software is using the best available technologies and methods.

What is new in our approach, and why do we think it will be successful?

We envision a multi-institutional "hub" to provide focus, coordination, and core capabilities for the stewardship and sustainability of the SC software ecosystem. The hub will comprise software professionals, researchers with strong backgrounds in software and scientific computing, and key representatives of other stakeholder groups. It will engage with the full range of stakeholders, including

the producers and consumers of research software, facility staff, institutional leadership, and SC sponsors to coordinate the stewardship of the SC software ecosystem.

The hub will start by working with stakeholders to establish a rubric for identifying what software capabilities need to be sustained and determining the most effective ways to satisfy those needs, through the maintenance and evolution of existing software or the development of new. This rubric will then be used to identify what software capabilities need to be sustained and determine the most effective ways to satisfy those needs. The focus of this effort will be on the software technologies (ST) under the Exascale Computing Project (ECP). This is timely in view of the large investment in development of ECP ST.

Our expectation is that the bulk of the effort to maintain and evolve the software will take place *outside of* but in coordination with the software professionals within the hub. The hub's software professionals and software scientists primarily will *facilitate* the improvement of software development practices within the client software projects. The hub will also provide expertise in niche, less-used, and emerging approaches to software development that client projects can tap into when "in-house" expertise is not (yet) available.

Complementing this direct involvement with client projects, the hub will carry out a much wider range of outreach activities with the goals of (a) enhancing the awareness and knowledge of better development practices for scientific software within the SC and broader community, (b) advocating for the establishment of incentives to motivate better quality and sustainability in general, throughout the SC software ecosystem, and (c) advocating for the professionalization of scientific software development and for appropriate recognition of practitioners within institutions and SC itself. The full range of activities of the ScSH will be grounded in a modest program of research in the science of research software through taking a research perspective to understanding what "works" and does not work in the development and maintenance of scientific software, and why.

We believe that this strategy will be successful because it is timely and it builds on a variety of ideas that have already demonstrated their value in the SC software ecosystem. The timeliness of the project is evident from the rapidly growing recognition across the community of both the importance of research software across the broad range of SC programs and the challenges being faced in sustaining and adapting it to new requirements, multiplied by the rapid evolution of computing hardware, both in major facilities and at the "edge," and the challenges of addressing both the pace of change and growing diversity of hardware architectures. Moreover, we will build on efforts such as the ECP Software Development Kits (SDKs), the E4S, and the IDEAS project, which have demonstrated success in most areas of the proposed ScSH, especially the curation of methodologies, education and outreach around software development practices, and facilitation of software process improvements with development teams. The SNS partnership with the Research Software Engineering mentioned earlier is another such example. Furthermore, the Department of Software Engineering and Research, also newly formed within Sandia National Laboratories, provides an example of how the study of the science of research software can add value to efforts aimed at practical improvements in software quality and sustainability.

Who cares? If we are successful, what difference will it make?

Numerous stakeholders will benefit from the ScSH.

- **Software producers.** The developers of software important to SC may reside in research organizations or facilities at national labs, academia, or industry. But the vast majority can benefit from increasing their knowledge and understanding of software development best practices and, where appropriate, more direct facilitation of such improvements. Experience shows that improvements in software practices lead in the longer term to better scientific productivity as well. Further, better software quality facilitates adoption by others.
- **Software consumers.** Users of scientific software within the SC ecosystem will find that better-quality software is more usable, more trustworthy, and, if it is backed by a commitment to sustainment, more easily adopted and relied upon in their own research and development (R&D) activities, thus making the consumers more productive overall.
- **Facilities.** Experimental, observational, and computational facilities rely extensively on software produced within the SC ecosystem, both in their own operations and for their users. The software may be provided by the facility or brought by the users, but no matter the origin, the facility has an interest as part of its own responsibility to steward the resources that the facility itself provides to users.
- **SC program managers.** As sponsors of R&D, program managers are in a challenging situation. They of course want to get the most scientific return for their investment. Reliance on research software is pervasive across not only computational but also experimental and observational research programs, but it is often hidden -- neither well recognized nor well understood. By gaining a more thorough understanding of the software ecosystem, program managers will be able to reason more effectively about how they might improve the return on their investments in R&D. ScSH can also facilitate conversations across different offices and programs, where software is produced by one research program and consumed by another, to ensure that needs are understood and to help maximize the value of the software by ensuring that it embodies the best available computer science and applied math capabilities.
- **Institutional hosts of producers, consumers, and facilities.** The ScSH will advocate for recognition for software-based research contributions as well as recognition and career paths for research software professionals. In the longer term, institutions adopting such approaches will have advantages in being able to attract and retain a workforce with skills and interests that are increasingly important to the overall scientific enterprise.

An important role of the hub, which does not exist today, will be to broker conversations on software sustainability and stewardship that include all relevant stakeholders. Stakeholders will also benefit, in varying but significant degrees, from the activities of the hub, including education and training, direct facilitation of software process improvement, and advocacy.

What are the risks?

- **SC offices and programs don't "buy in" to the ScSH approach.** DOE HQ staff will play an important role in enacting a strong stewardship program. Sustainability of key software can be achieved only by investment by the parties that care about and benefit from it. No matter how that money is accounted for, it ultimately comes from DOE HQ. We anticipate that in many cases interests in particular software capabilities will cross multiple programs and offices. For example, a tool or library developed under ASCR R&D programs might become important to a Basic Energy Sciences (BES) facility or a Nuclear Physics (NP) computationally based research program. We believe that the hub can adapt to such situations by focusing its efforts on the

offices that do buy in and demonstrating benefits that would encourage other offices to participate. There are many opportunities and the need to improve the sustainability of SC research software -- within offices and across offices -- no matter which offices are involved.

- **Stakeholders are unwilling to commit sufficient funding to stewardship and sustainability.** Although one aspect of the ScSH will be to promote general improvements in software development processes, with an eye to overall improvement over time, once a need is identified in initial discussions about the sustainability of particular packages, addressing that need will not be without cost. Various approaches can be used to obtain the financial support needed for such efforts, approaches that, in practice, some might view as "taking away" from or "competing with" R&D activities or other aspects of facility operations. One aspect of addressing this risk is to study the projects the ScSH undertakes and guides from the perspective of cost and benefit, to develop evidence to strengthen the case that improvements in sustainability provide a net benefit in scientific productivity (or other appropriate metrics). A complementary approach might be to change the funding model for stewardship activities, for example, by assessing a small set-aside effort from each software-intensive R&D program to support stewardship efforts on behalf of that program.
- **Contributions to scientific software are not adequately incentivized.** An important element of where we are today is that, typically, neither employers nor funding agencies value contributions to scientific software and its sustainability as they do scientific results and impact. The ScSH will be a strong advocate for changing this situation, but it requires support and the will within institutions and sponsors to change the situation. Fortunately, there is growing recognition in all quarters of our increasing dependence on research software and the looming crisis in software sustainability. At the same time, the reproducibility of scientific research is getting greater and greater scrutiny, which, for software-intensive research, also promotes the value of better software development practices that lead to both more trustworthy and more sustainable software. We believe that advocating for a series of incremental improvements will also help make the needed changes more palatable to all stakeholders.

  Another aspect of this risk relates to the ability of employers to recruit and retain the research software professionals that will be needed to take on many stewardship activities. We believe that institutions seeing others doing a better job of software sustainment by having the right mix of staff for the work will be incentivized to modify their policies and practices to allow them to strengthen their own workforce. ScSH and other stakeholders can also help by guiding the funding to carry out the sustainment activities to those institutions that demonstrate that they can get the job done.

- **ScSH is not viewed as a fair and impartial party in stewardship discussions.** To be effective, the ScSH must have the trust of stakeholders. We need to avoid participants coming out of the process feeling like there are "winners" and "losers," even when hard decisions may need to be made. Transparency in setting the criteria by which sustainability and needs are judged will be important to mitigate this risk. Research activities aimed at more thorough understanding of the costs and benefits of improving the software development process and reworking scientific software will also facilitate informed and transparent decision-making.
- **The work of the ScSH with individual software packages could undercut its work on generally improving awareness, knowledge, and use of better software practices.** The existence of a stewardship initiative that invests resources in increasing the sustainability of particular

software packages might be seen by some as undercutting the hub's complementary efforts to increase knowledge of and attention to software sustainability across all software producers. Why should developers spend the time and effort when someone will step in to "rescue" them down the road? The first and strongest counter to this attitude will be through research efforts to quantify improvements in scientific productivity from improving software quality and sustainability measures. The incentives for which ScSH will advocate will also help alleviate this risk.

How much will it cost?

The pilot project, ScSH, would leverage the experience of current software sustainability activities in ECP (IDEAS, SDKs, E4S, and software ecosystem work such as testing and deployment. Funding for the pilot should also come from the wider SC community, initially at the level of roughly $3--4M/year. This level of funding would enable the pilot project to develop the rubric, evaluate the ECP software portfolio also introduce additional activities such as long-term software planning and to determine whether software sustainability practices from ECP can be extended to 3--4 non-ECP DOE software packages. In the long term, as ECP winds down and ScSH absorbs the sustainability activities under ECP and expands its portfolio to include non-ECP software, the ScSH would require $6--8 M/yr for core operations to provide stewardship for a large, diverse software portfolio, with additional funding for specific collaborative projects.

How long will it take?

By leveraging activities in IDEAS, SDKs, and E4S, which already have formed groups of interested people, ScSH could become operational within 4--6 months. Initial operation would be products being managed, scope and policies defined, roles allocated, and outreach initiated. This schedule assumes that the involved group members are spending a reasonable fraction of their time on this effort and are largely available for meetings and discussion. The pilot would be for two years, and, if successful, the last six months of the pilot would be used to plan the fully operational hub.

What are the mid-term and final "exams" to check for success?

The focus of ScSH is on improving the readiness and distribution of DOE software, conducting research to identify effective techniques for ensuring software quality, and educating the workforce. To this end we have identified the following milestones over the two-year duration of the pilot:

- By the end of the first year:
    - Develop a deeper understanding of the needs in software sustainability within SC through engagement with stakeholders; develop the rubric for evaluation
    - Hold several workshops to expand the community to focus on effective practices for software productivity and sustainability
    - Identify research areas critical to software productivity and sustainability
    - Develop understanding of how best to provide value to the community
    - Start a test engagement with an ECP project to flesh out the details of what it takes to have a software sustainment project
- At the end of the second year:
    - Continue with the workshops for community engagement

- Launch a prototype software sustainment project based on the full process of engaging with stakeholders to define and scope the work (funded by the hub, this effort might extend beyond the pilot period)
- Demonstrate the advantage of stewardship of software projects, via measurable increased productivity
- Identify several target software projects for stewardship
- Prepare proposal for long-term Scientific Software Hub

From inception, ScSH will track key metrics related to community engagement, adoption of effective techniques, software productivity, and research productivity.