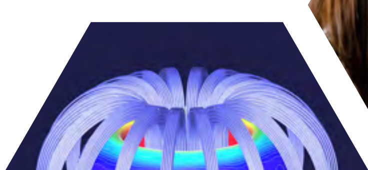
BASIC RESEARCH NEEDS IN

The Science of Scientific Software Development and Use

Investment in Software is Investment in Science



Investment in Software is Investment in Science

Increasingly powerful and affordable computing has revolutionized scientific and scholarly discovery across a broad range of fields. Computing relies on software, which has been rapidly growing in scope, diversity, and complexity. At the same time, the methods, processes, and tools used to produce and utilize this essential software are often ad hoc, and the study and improvement of them is often done without the benefit of direct funding or prioritization. Consequently, concerns are growing about the productivity of the developers and users of scientific software, its sustainability, and the trustworthiness of the results that it produces.

The US Department of Energy Office of Science (DOE-SC) is at the forefront of modern software-enabled scientific discovery across numerous areas of computational, experimental, and observational science, including major investments in national user facilities that support these activities. In December 2021, the DOE-SC Office of Advanced Scientific Computing Research (ASCR) convened a workshop on basic research needs for the Science of Scientific Software Development and Use (SSSDU). Through keynote presentations, lightning talks, and breakout groups, participants discussed the current practice of software development, maintenance, evolution, and use, and considered how the scientific method could be used to examine these practices and develop more evidence-based approaches to enhance the impact of software and computing on all areas of science.

Workshop participants identified three priority research directions (PRDs) and three important crosscutting themes that center on the following overarching insight: software has become an essential part of modern science that impacts new discovery, policy, and technological development. To have full confidence in science delivered via software, we must improve the processes and tools that help us create and use it, and this enhancement requires a deep understanding of the diverse array of teams and individuals doing the work. The full workshop report will be available at https://doi.org/10.2172/1846009.

Priority Research Directions

PRD1: Develop methodologies and tools to comprehensively improve team-based scientific software development and use

Key question: *What practices, processes, and tools can help improve the development, sustainment, evolution, and use of scientific software by teams?*

As the fundamental understanding of scientific software improves, we foresee that the methodologies and tools we need will also change to better match and support how developers, users, and other scientific software stakeholders work toward the goal of accelerating scientific discovery. Although many scientists have extensive intuition about the principles and dynamics of how their community develops, uses, and sustains its software products, research is needed to develop a deeper and broader understanding of software's role in scientific processes. We believe that developing common mental models around fundamentals can help the scientific community as a whole observe, understand, and improve the effectiveness and efficiency of scientific discovery through better methodologies and tools for the development, use, and sustainment of scientific software, particularly in the context of diverse collaborative teams.

PRD2: Develop next-generation tools to enhance developer productivity and software sustainability

Key questions: *How can we create and adapt tools to improve developer effectiveness and efficiency, software sustainability, and support for the continuous evolution of software? How can we support and encourage the adoption of such tools by developers?*

Numerous tools assist developers with their activities. Many tools that are well established in the larger software world might not be widely known or used in scientific software development. We anticipate the rise of future software-generating environments that translate scientific programmer intent into source code fragments that can then be tuned and refined. We also expect advances in tools that assist in generating software tests, documentation, clean and readable source code, and more. The challenges faced by the scientific software community include ensuring that future tools account for the requirements of scientific software that might not be high priorities in other fields and addressing opportunities that may be unique but are still critical to scientific software. Work is also needed to facilitate the adoption of new tools, including effective training, along with help in incorporating these tools into already-complex workflows.

PRD3: Develop methodologies, tools, and infrastructure for trustworthy software-intensive science

Key questions: *How can we facilitate and encourage effective and efficient reuse of data and software from third parties while ensuring the integrity of our software and the resulting science? How can we provide flexible environments that "bake in" the tracking of software, provenance, and experiment management required to support peer review and reproducibility?*

Scientific results are trustworthy only if all aspects of the scientific process can be trusted to produce correct, transparent, reproducible, and replicable results. Although every scientific software developer intends to produce trustworthy results, the current state of the practice varies tremendously from team to team. There are many concerns to consider in ensuring trustworthy computational results, and all of them must be addressed to some degree if we are to make qualitative progress. Our data and software must be managed, archived, and retrievable, and our computational steps must be recorded and available for future use. We must be able to easily detect and correct perturbations in state and execution, and we must be able to preserve provenance down to the finest granularity for assessment and audit.

Crosscutting Themes

Theme 1: We need to consider both human and technical elements to better understand how to improve the development and use of scientific software.

It is common to assume that challenges encountered in software-intensive research can be overcome only by some new technical innovation. In fact, people—individuals and teams—play enormous roles in the development and use of software—often in both the challenges and in the potential solutions. Therefore, we consider this human element essential in our efforts to understand and improve scientific software development and use, including engagement with social, cognitive, and information scientists and others who have not historically been part of the DOE computational research community.

Theme 2: We need to address urgent challenges in workforce recruitment and retention in the computing sciences with growth through expanded diversity, stable career paths, and the creation of a community and culture that attract and retain new generations of scientists.

Developing high-performance scientific software requires the combined contributions of many people with a wide range of skills and backgrounds, most of which are also in high demand in the broader marketplace. Our ability to staff current and future teams requires strong and sustained efforts to educate, recruit, and retain a diverse workforce by cultivating a supportive and inclusive culture within the computing sciences. We urgently need broad and sustained community collaboration to change the culture and demographic profile of scientific computing with multipronged approaches to expand the pipeline and workforce.

Theme 3: Scientific software has become essential to all areas of science and technology, creating opportunities for expanded partnerships, collaboration, and impact.

Not only has powerful and affordable computing revolutionized the scientific enterprise, computing also provides the basis of many technological and engineering products and increasingly informs policy and other consequential decisions. Owing to the broader importance of scientific software, efforts to better understand and improve its development and use should find allies beyond DOE, including practitioners in academia and industry, as well as other government agencies that fund computationally and software-intensive research.

U.S. DEPARTMENT OF ENERGY | Office of Science