Navigating the Future of Scientific Software-Ecosystem Sustainment

Toward a Post-ECP Software Sustainability Organization (PESO)
Draft Version 0.1 August 31, 2023

Report of the PESO Community Workshop (June 8-9, 2023) and Continuing Engagement

Michael Heroux (Sandia National Laboratories; PI)
James Ahrens (Los Alamos National Laboratory)
Todd Gamblin (Lawrence Livermore National Laboratory)
Timothy Germann (Los Alamos National Laboratory)
Xiaoye Sherry Li (Lawrence Berkeley National Laboratory)
Lois Curfman McInnes (Argonne National Laboratory)
Kathryn Mohror (Lawrence Livermore National Laboratory)
Todd Munson (Argonne National Laboratory)
Sameer Shende (University of Oregon)
Rajeev Thakur (Argonne National Laboratory)
Jeffrey Vetter (Oak Ridge National Laboratory)
James Willenbring (Sandia National Laboratories)

Workshop Presenters and Breakout Leads

On behalf of the PESO leadership team, we gratefully acknowledge all community contributions to PESO planning, including people who:

- Provided PESO Planning Input
- Attended online PESO Community Discussions (see https://lssw.io for details)
- Attended the PESO Community Workshop (https://bit.ly/peso-workshop-june2023), see Appendix A for details

We especially thank community members who provided report-back summaries of breakout discussions during the workshop:

- Ann Almgren
- Hartwig Anzt
- Axel Huebl
- Tzanio Kolev
- Mary Ann Leung
- Rob Ross
- Damian Rouson
- Carol Woodward

We also thank community members who spoke during plenary workshop sessions to motivate breakout discussions:

- Richard Gerber
- Berk Gevecki
- Bailint Joo
- Christopher Knight
- Jeff Larkin
- Mary Ann Leung
- Andrew Siegel
- Rick Stevens
- Ulrike Yang

These contributions, along with complementary conversations throughout the past several years, have provided important input about community perspectives and priorities for software sustainment.

Additional Resources

PESO Community Workshop Resource Page:
https://bit.ly/peso-workshop-june2023-sharedcontent

PESO Hub-and-Spoke Resource Page:
https://bit.ly/PESO-Hub-and-Spoke-Kickoff

Table of Contents

1 Executive Summary: Navigating the Future of Scientific Software-Ecosystem Sustainment

Sustainability of the software ecosystem for the US Department of Energy (DOE) leadership scientific computing mission is essential for future scientific progress. In order to fully leverage and extend the advanced software ecosystem developed during the DOE Exascale Computing Project (ECP) to meet the needs of next-generation science, the PESO (Toward a Post-ECP Software-Sustainability Organization) leadership team has engaged the community in determining an effective and flexible community strategy. This report discusses the PESO organizational strategy, emphasizing the distribution of tasks and resources at the complementary levels of individual software product teams, product communities, and a hub that handles overarching coordination.

Under the PESO umbrella, we have adopted a 'Hub and Spoke' approach, fostering self-organizing Software Product Communities (SPCs) and encouraging active engagement in crosscutting efforts. This setup promotes synergy that promises shared design exploration, coordination of complementary capabilities and teams, and more.

Moreover, we focus on the potential for 100X improvements in capabilities enabled by ECP investments over seven years. These improvements have positioned us to exploit advanced computing architectures that feature accelerators from multiple vendors, to prioritize software quality in order to help ensure the integrity of computational results, and to emphasize community engagement as needed to tackle next-generation scientific challenges. ECP's contributions have paved the way for lower costs and higher performance for accelerated platforms.

To further enable a 100X improvement, we discuss opportunities such as migrating from CPUs to GPUs and integrating into larger software communities. We stress the need for engaging with target communities and overcoming impediments to realize this potential. This approach paves the way for our proposed '100X Recipe', which maps out a strategic plan to achieve this improvement.

The report concludes by outlining the multiple paths to leverage the 100X improvement, by achieving high-end science capabilities or by reducing costs, or by a combination of both. The collective goal is to organize the DOE's scientific software community optimally for the highest impact after ECP and beyond.

This report serves as a preliminary guide that underscores the collective commitment toward sustaining and enhancing the scientific software ecosystem. The framework and strategies that it provides help navigate the challenges and opportunities that lie ahead, steering us toward a future where scientific advancements continue to flourish.

2 Introduction: Envisioning a Sustainable Future for Scientific Software Ecosystems

As the Exascale Computing Project (ECP) finishes, leaving as a legacy a large collection of scientific libraries and tools that work portably across three GPU architectures as well as CPUs, the DOE community is poised to enter a new phase in scientific software-ecosystem sustainment, extending the ECP investments beyond merely delivering enhanced capabilities. Our goal is to ensure the sustainability of the software ecosystem that enables leadership scientific computing, leveraging the fruits of the ECP's labors and carrying them forward into a future characterized by continuous evolution and adaptation.

This report describes the initial plans designed to foster a sustainable software ecosystem. The report explores the PESO organizational strategy, hub and spoke approach, and the status of implementation. We also discuss communities of practice and highlight the key takeaways from the PESO input responses.

Furthermore, we examine how ECP investments have spurred a 100-fold improvement in capabilities and explore potential opportunities for leveraging these investments to realize similar improvements in the future. In doing so, we touch upon several ways to attain this 100-fold enhancement. The goal of our efforts is to ensure that the impact of the ECP extends far beyond its lifetime, shaping a future of sustained scientific progress, with flexible coordination across the DOE advanced computing community.

3 The Post-ECP Software-Ecosystem Sustainment (PESO) Project

The Post-ECP Software-Ecosystem Sustainment (PESO) Project is an initiative aimed at supporting and enhancing the sustainability of the Department of Energy's (DOE) open-source libraries and tools used for advanced scientific computing. Acting as a central hub, PESO is committed to facilitating various software-ecosystem sustainment efforts, providing the required infrastructure and support to ensure long-term viability.

A crucial aspect of PESO's approach involves working closely with software project teams to effectively coordinate various development activities. This coordination is designed to promote and ensure long-term sustainability, ultimately delivering substantial benefits to a wide range of stakeholders.

PESO's operations also involve extensive collaboration with software product communities (SPCs, also known as SDKs or “spokes”) and communities of practice (COPs). This collaboration ensures a comprehensive approach to providing crosscutting services and support. The beneficiaries of these services include developers, users, and stakeholders who are actively involved in the scientific software ecosystem.

The project's primary objective is to optimize the effectiveness and efficiency of DOE investments in scientific libraries and tools by creating shared capabilities and establishing synergistic relationships with other entities in the software ecosystem. The PESO Hub-and-Spoke model supports rigorous engagement with commercial software and hardware developers, software product teams outside of DOE, other US agencies, DOE lab management, and DOE sponsors.

In addition to directly facilitating DOE-sponsored software work, PESO takes a broad, strategic view, focusing on key areas such as project growth, software quality, and availability. Through concerted efforts aimed at improving delivery, deployment, and support mechanisms, PESO aims to create a sustainable and dependable software ecosystem.

As a guiding value proposition, PESO is committed to realizing the 100X potential enabled by the Exascale Computing Project (ECP) investments. By optimizing and enhancing the effective use of these investments, PESO aims to significantly improve the software ecosystem, furthering the scientific computing capabilities of the DOE.

3.1 PESO is not ECP

PESO is an initiative designed to sustain and enhance the DOE open-source software ecosystem for advanced scientific computing. While PESO shares a common goal with ECP in improving the scientific computing landscape, the approach and organizational structures of the two projects are significantly different.

The ECP was characterized by a hierarchical structure, centrally controlled financial organization, predetermined set of applications, and defined project scope. ECP was also marked by a 'heavyweight' approach to reporting, a design that was justified given the size and structure of the ECP as a formal federal Project.

In contrast, PESO has been designed with a different approach and structure. A key distinction lies in its focus on peer collaboration, operating as a 'hub' rather than as a hierarchical entity. This approach encourages broader engagement and facilitates more efficient cooperation among various stakeholders in the software ecosystem.

Another significant difference is PESO's financial model. Instead of establishing a centralized financial organization like the ECP, PESO leverages existing institutional financial organizations at DOE labs. This model enables a more versatile and adaptable financial strategy.

PESO also distinguishes itself from the ECP through its dynamic and adaptive scope targets. Unlike the ECP, which had a fixed set of applications and project scope, PESO's scope is adaptable to address emerging needs and challenges in the software ecosystem.

The reporting strategy in PESO is also 'tunable', as opposed to the ECP's 'heavyweight' approach. This flexible strategy allows PESO to adjust its reporting based on requirements, proposing to effectively streamline processes and optimize resources.

Overall, PESO adopts a lighter-weight approach than the ECP. This strategy is designed to ensure that the project can operate more effectively and efficiently for its size, respond more quickly to changes in the software ecosystem, adapt to diverse needs, and maintain a dynamic approach to software ecosystem sustainment.

3.2 PESO is more than Spack and E4S

A potential misconception about PESO is that it is synonymous with two of its major deliverables:

- Spack: a powerful toolset for defining and managing software dependencies, invoking multi-product builds, and assuring robust software installation and testing.
- The Extreme-scale Scientific Software Stack (E4S): A curated stack of approximately 100 libraries and tools, along with all Spack-defined dependencies, to support portable application execution on multiple GPU platforms.

However, the scope and ambition of PESO extends significantly beyond these key outputs.

Spack and E4S indeed form important components of PESO's delivery mechanisms. They serve as significant product delivery conduits and provide platforms for collaboration with various agencies and industry stakeholders. Furthermore, Spack and E4S are instrumental for testing on new and diverse platforms and software environments, contributing significantly to the agility and adaptability of the software ecosystem.

Nevertheless, PESO's reach and impact are intended to extend far beyond these tools. One of PESO's most significant objectives is to maximize the scientific impact through its 100X efforts. This goal signifies an intent to enhance the capabilities and efficiency of the scientific software ecosystem, going far beyond the deliverables of Spack and E4S through advances in individual software projects, software product communities, and communities of practice.

PESO is committed to fostering a culture of collaboration in planning, executing, tracking, and reporting activities. This collaborative approach allows PESO to align work to address the needs of various stakeholders, streamline processes, and optimize resources, contributing to the overall efficiency and effectiveness of the software ecosystem.

Furthermore, PESO prioritizes active engagement with the broader High-Performance Computing (HPC) community, including application developers, computing facilities leadership and staff (including ALCF, NERSC, and OLCF), vendors, and government agencies. By promoting community engagement, PESO ensures the needs and priorities of the broader community are considered and incorporated into its efforts.

PESO also plans to coordinate crosscutting training programs and community engagement efforts, further demonstrating its commitment to sustainability not only of software products but also the community of people who develop and use them, as driven by the needs of next-generation science. These initiatives help build skills, knowledge, and capacity within the software ecosystem and advanced computing community, fostering long-term resilience and adaptability.

3.3 PESO Organizational Strategy

PESO has designed an organizational strategy that recognizes the varying needs and dynamics at different operational levels. This approach ensures effective resource allocation, enhances efficiency, and facilitates coordinated actions to deliver a trustworthy software ecosystem.

Individual Product Team Level
The cornerstone of software development resides within individual product teams, as they handle most of the everyday development work. They are responsible for delivering capabilities that contribute to the whole, alongside testing and product improvement activities. This level of work is essential for the creation of robust, reliable software products that align with stakeholder needs and project goals.

Product Community "Spoke" Level
Some tasks necessitate a broader view, making them more appropriate for execution at the product community level. Such activities include portfolio planning and coordination, holistic tutorial delivery, and design space exploration for next-generation platforms. The collective knowledge and combined expertise within the product community are key to providing these strategic and forward-looking contributions.

Hub Level
Also, a range of tasks are best suited to the hub level, including overarching activities such as software stack management, specialized Continuous Integration (CI) testing, all-team meetings, and coordinated planning across the portfolio. Other community of practice activities, like working with software foundations, improving software skills, and community engagement, also fall under the remit of the hub. These actions ensure alignment and harmony across various software projects, promoting an integrated and coherent software ecosystem and community.

Key Goals:

The PESO organizational strategy is designed with two primary goals in mind:

Goal 1: Allocate budget and work at the level where it can be executed better, faster, and more cost-effectively than elsewhere. This principle supports the efficient use of resources, driving the delivery of high-quality software solutions, while respecting the autonomy of individual software teams in decisions that don't require coordination across products and communities.

Goal 2: Coordinate across levels with an aim to serve product teams, users, and sponsors effectively. This approach enhances stakeholder satisfaction and contributes to the development of a trusted software ecosystem that can meet the challenges of next-generation science.

By adhering to these organizational principles, PESO is positioned to deliver robust, reliable software while fostering a sustainable, inclusive, and efficient scientific software-ecosystem and community.

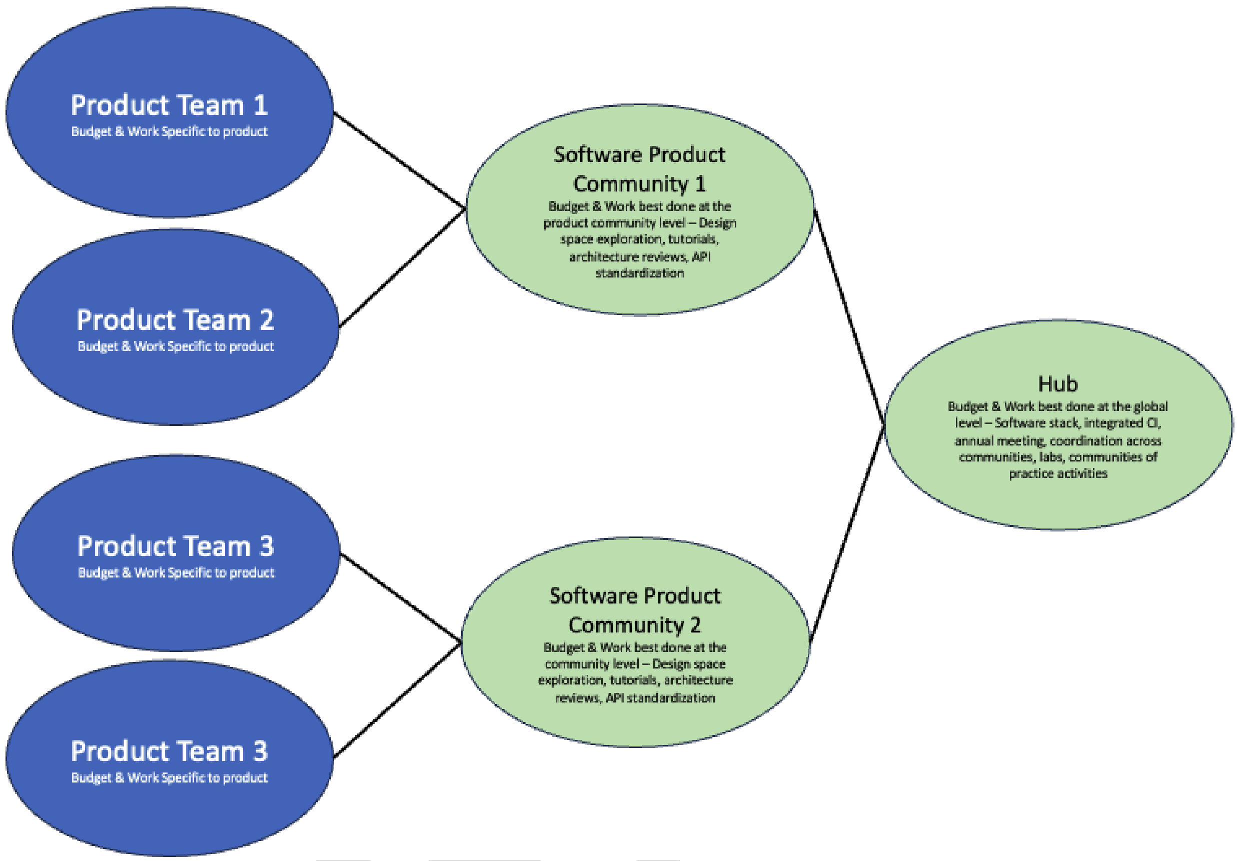

Figure 1: The PESO Hub-and-Spoke model proposes three levels of aggregation. To the left are the individual product teams that contribute to and support the libraries and tools needed by the scientific community. In the middle are software product communities, or "spokes" that bring together compatible and complementary products, including products not sponsored by DOE, for collaborative planning, executing, and outreach to their specific stakeholder communities. To the right is the hub that supports crosscutting activities common to all or most spokes and teams. Hub activities include coordination of funding decisions across spokes, delivery of libraries and tools via E4S, management of the macro software lifecycle, sponsorship of reports and events such as an annual meeting and portfolio reviews, and similar coordinated activities.

3.4 PESO Key Services, Activities, and Value Proposition

PESO will play a crucial role in maintaining and enhancing DOE's open-source software ecosystem for advanced scientific computing. This role is facilitated through a range of key services and activities:

1. **Funding Steering:** PESO is committed to working with spokes and product communities to prioritize and determine funding and work activities. Strategic allocation of resources to address DOE's scientific computing needs is integral to maintaining a robust software ecosystem over the long term.

2. **Delivery and Deployment:** PESO coordinates the delivery and deployment of software through Spack and E4S to DOE Computing Facilities and a diverse range of on-premises and cloud users and developers. This work includes comprehensive services such as Continuous Integration (CI) testing on advanced node types, issue triage, build caches, and quality assurance for software.

3. **Crosscutting Engagement:** PESO actively coordinates crosscutting engagement with DOE Computing Facilities, DOE sponsors, and other stakeholders, including other US agencies, industry partners, and international collaborators. This broad engagement ensures that a variety of perspectives and needs are considered in PESO's initiatives.

4. **Software Lifecycle Support:** PESO provides lightweight processes, models, and tools to support software product communities throughout their software lifecycle. These activities include assistance with annual planning, execution, tracking, and assessment, all underpinned by comprehensive change management strategies.

5. **Community Engagement Infrastructure:** PESO offers basic infrastructure to support community engagement. This work is coordinated with communities of practice for outreach, training, community development, and collaboration with external entities.

6. **Outreach and Workforce Development:** PESO is actively involved in outreach and workforce development initiatives, seeking to incubate new projects, grow the contributor base, and increase external investment in key projects.

The core value proposition of PESO is that by engaging with PESO, each product team and community, as well as sponsors and stakeholders, will be better off than without PESO.

3.5 PESO Financial Model

PESO proposes a financial model aimed at ensuring transparency, low overhead costs, and clear value allocation. This financial model is integral to PESO's sustainability strategy, enabling the project to efficiently and effectively allocate resources to further DOE's open-source software ecosystem for advanced scientific computing.

The primary tenet of the PESO financial model is clear values and transparency. Funding allocations to products and communities are based on a transparent set of criteria applied through an open review and assessment process. This level of transparency ensures fairness in resource allocation and provides all stakeholders with a clear understanding of how and why resources are distributed in specific ways. This understanding not only builds trust within the ecosystem but also promotes a culture of accountability and responsibility.

Furthermore, PESO embraces a low overhead approach to financial management. In practice, this means that funds for all efforts at a particular lab are sent directly from the sponsor as a lump sum to the lab, with itemized amounts earmarked for individual projects at that lab. This direct allocation of funds ensures that a greater portion of resources can be dedicated to productive project activities rather than overhead.

In addition, funds intended for university and industry subcontracts are sent directly from the sponsor to the most appropriate partner lab. This targeted allocation of resources further reduces overhead and ensures the efficient and appropriate use of funds.

Finally, all funds at each lab are managed by the normal lab financial infrastructure, leveraging existing financial management systems and expertise.

3.6 PESO Sustainability Strategy

PESO employs a comprehensive sustainability strategy with an emphasis on the software ecosystem in support of users even as individual products may transition into, out of, and within the ecosystem. Specific strategy elements include:

1. **Sustained Resources:** Central to PESO's sustainability strategy is the sustained provision of resources, encompassing funding, effort, and infrastructure. This consistent resource allocation allows for continuous support, development, and enhancement of the software ecosystem. Sustained resources are essential for building user trust.

2. **Robust User and Developer Base:** PESO will prioritize cultivation of a robust user and developer base, understanding that the human element is crucial to any successful software project. We will invest in the people who are part of our community. We will plan and host regular software webinars for developers and users. We will provide resources for software teams and communities to improve the quality of their software, and we will provide venues for developers and users to gather.

3. **Focused Scope:** PESO will maintain a targeted focus on sustaining the High-Performance Computing (HPC) libraries and tools scientific software ecosystem. This scope includes the expansion and evolution of scientific computing to include machine learning, quantum algorithmic software and other leadership computing requirements as they emerge.

4. **Product Evolution:** PESO is cognizant that while specific products may come and go, critical functionalities must be sustained. Focusing on the ecosystem and planning for product transitions allow flexibility and adaptation, while ensuring the preservation of vital capabilities.

5. **Managed Transitions:** Transitions into, out of, and within the ecosystem are explicitly managed to ensure smooth changes and continuous operation.

6. **Vendor Product Integration:** Vendor products are integral to planning and collaborations. PESO acknowledges the role of industry in driving innovation and providing practical solutions.

7. **Adaptive Evolution:** PESO is committed to evolving to meet community needs, demonstrating responsiveness to changes in the user and developer landscape.

8. **Critical Mass through a Hub and Spoke Approach:** PESO uses a hub and spoke approach to combine efforts and attain a critical mass, thereby allowing the project to leverage its aggregate scale for external influence.

3.7 PESO Project: Promoting Inclusive and Equitable Research (PIER) Plan

PESO is committed to fostering a diverse, equitable, and inclusive environment in scientific software ecosystem sustainment.

The PESO PIER Plan takes a two-pronged, multifaceted approach to advancing diversity, equity, and inclusion:

1. **Internal Activities:** The plan targets activities within the PESO project, seeking to improve inclusion and diversity within our own teams and work.

2. **External Partnerships:** PESO is committed to establishing partnerships with other organizations to drive cultural change within the broader scientific software-ecosystem sustainment community.

Across both dimensions, the PIER Plan addresses the following areas:

1. **Recruitment and Inclusion:** PESO aims to engage diverse individuals from underrepresented groups, making these individuals integral members of our teams and the broader community. The objective is to create a rich, diverse fabric of people who bring different perspectives to the project.

2. **Cultivating Respectful and Professional Work Environments:** PESO recognizes the importance of promoting work environments that foster mutual respect and professionalism. This approach emphasizes sharing best practices and effecting cultural change to create an atmosphere of inclusivity and respect.

3. **Professional and Scholarly Growth:** PESO is dedicated to planning for the scholarly and professional growth of community members. This commitment includes a particular emphasis on supporting Research Software Engineers (RSEs) and early-career staff, providing opportunities for their development and growth within the field.

The PESO Team has a long tradition of engaging the broader community. Members of our team founded the Better Scientific Software (BSSw) Fellowship program and have led the ECP Broadening Participation Initiative and other community initiatives that aim to expand our community by providing opportunities for under-represented groups and creating a culture that welcomes and celebrates diversity as an essential element of a healthy community.

4 PESO Hub and Spoke Approach

The Post-ECP Software-Ecosystem Sustainment (PESO) Project has adopted a "Hub and Spoke" approach to create a robust, coordinated, and sustainable scientific software-ecosystem. This strategy pivots on the interaction between Software Product Communities (SPCs) and Communities of Practice (COPs).

1. Software Product Communities (SPCs)

Known also as SDKs or "Spokes", SPCs form an integral part of the PESO strategy. PESO aims to serve as a hub, forming aggregations with communities composed of the developers of compatible and complementary products. It's anticipated that SPCs will self-organize and establish their own community-specific governance.

The spectrum of SPCs is expected to encompass DOE-sponsored, commercial, and other community software, creating a diverse ecosystem. The value proposition for SPCs includes shared design space exploration and coordination, which facilitates unified progress and fosters collaborative innovation.

2. Communities of Practice (COPs)

PESO will also pursue active engagement with leaders in COPs, a strategy designed to stimulate crosscutting efforts that benefit the broader community. Examples of these communities include:

- **Scientific software developers:** Entities such as the IDEAS project and the HPC Best Practices webinars provide important forums for sharing knowledge and promoting best practices.
- **Community outreach:** Organizations like the Center for Scientific Collaboration and Community Engagement (CSCCE) help to expand the reach of the scientific software community, fostering greater collaboration and engagement.
- **Software foundations:** Groups like NumFOCUS and the Linux Foundation provide vital support for the development and sustainability of open-source software, enhancing the resilience and diversity of the ecosystem.
- **Workforce development:** Initiatives such as the US Research Software Engineering (US-RSE), Better Scientific Software (BSSw) Fellowship Program, and Broadening Participation Initiative (in partnership with Sustainable Horizons Institute) help to build capacity and develop the skills needed for the sustainable management of scientific software.

The PESO Hub and Spoke approach offers a balanced, flexible, and inclusive strategy for software ecosystem sustainment. This approach promotes collaboration, fosters diversity, and ensures sustainable delivery, ultimately bolstering the value and potential of DOE investments in the scientific libraries and tools ecosystem.

4.1 Status of PESO's Hub and Spoke Approach: An Overview

The Post-ECP Software-Ecosystem Sustainment (PESO) Project has made substantial progress in implementing its hub and spoke approach to create a robust scientific software ecosystem. We summarize the status of key components and how they fit into this ecosystem.

1. PESO (Hub)

PESO, acting as the hub, proposes to obtain funding for core software activities that serve the broader community and the establishment and sustenance of cross-community capabilities, including engagements and services that are common to most, if not all, software product communities. By functioning as the nucleus of the system, PESO ensures overall coordination and directs resources where they are most needed.

2. SWAS, STEP, S4PST

SWAS, STEP, and S4PST are funded seed projects focusing on various areas of the software ecosystem. SWAS concentrates on workflows, STEP on tools, and S4PST on programming systems. The PESO team is working with these three seed projects in pursuit of a hub (PESO) and spoke (where SWAS, STEP, and S4PST are each a spoke) model. Of particular interest in our conversations is how to coordinate and collaborate on the selection of work and funding within and across the products, spokes, and the hub.

3. DAV, xSDK

DAV and xSDK are existing ECP Software Development Kits (SDKs), focusing on data/viz and math libraries, respectively, which are well aligned with the PESO approach. Their integration into the hub and spoke model represents the effective utilization of existing resources and underscores the compatibility of PESO with prior ECP endeavors.

4. SciML

SciML represents an unmanaged product community (focusing on machine learning software) that aligns well with the PESO approach. SciML represents potential future integrations of existing communities into the PESO framework, expanding the ecosystem's diversity and capacity.

5. Others

The PESO ecosystem is deliberately designed with room for growth, with some products yet to be identified and incorporated. An ongoing process of discovery and evaluation is in place to ensure that potential contributions to the ecosystem are not overlooked.

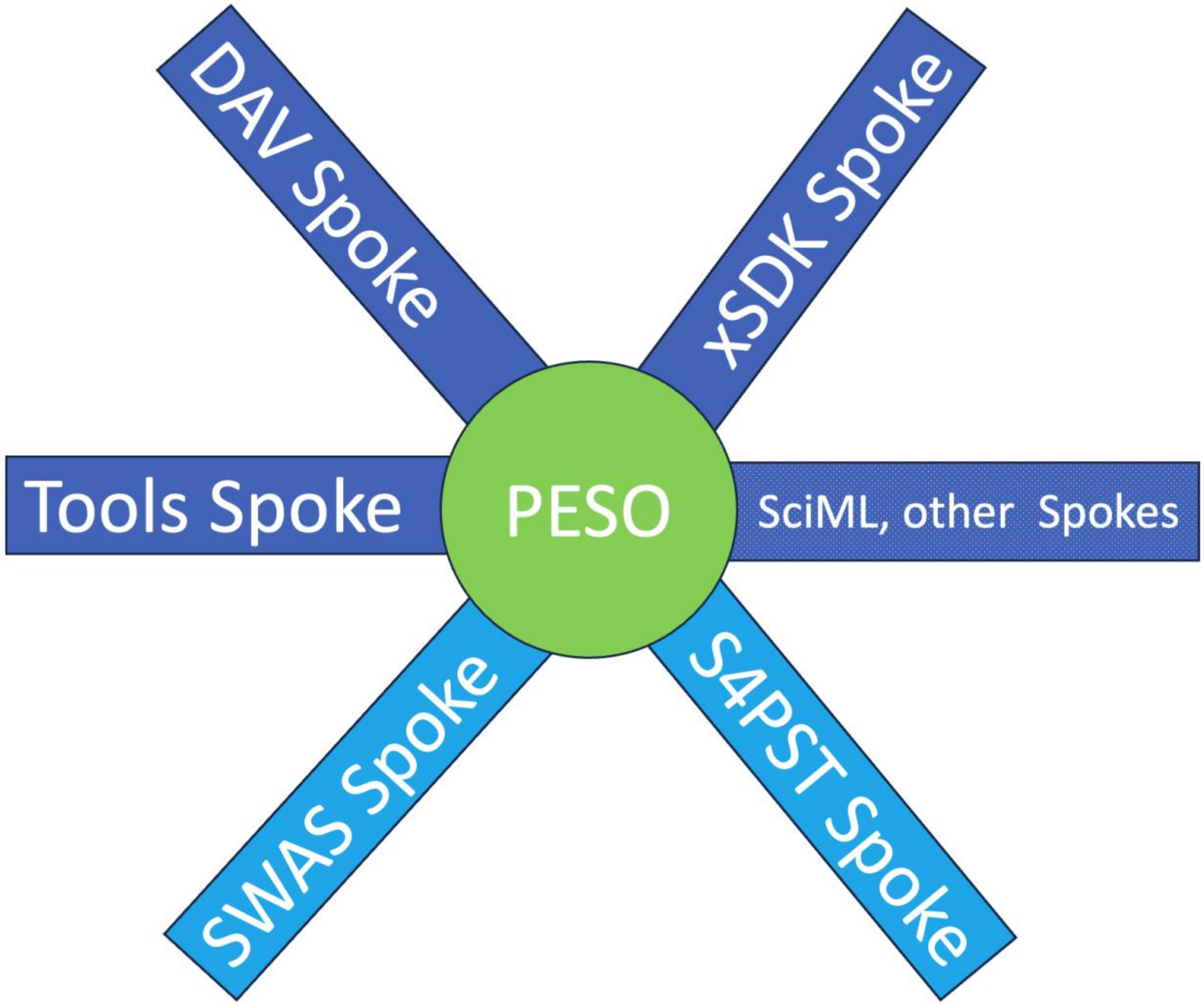

Figure 2: The PESO hub-and-spoke model enables a complementary focus on specific domains and across domains. Within each spoke are teams that represent the collective knowledge, capabilities, challenges, and vision of a particular software product community. Members of a spoke can be funded by DOE or through other means. It is important to have comprehensive representation within a spoke so that DOE-sponsored work is informed, planned, and executed within the broader community. The PESO hub conducts work that supports the spokes by providing the overarching infrastructure that is needed across spokes.

4.2 PESO's Communities of Practice: Enabling Ecosystem-wide Improvement

PESO recognizes that quality improvement in software is an effort that spans across multiple product initiatives. The implementation of best practices in software development often has far-reaching impacts, necessitating a community-based approach to standardization and enhancement.

1. **Cross-Ecosystem Impact:** The improvement in software practices often benefits individual developers and teams across the ecosystem. By addressing common needs and challenges, shared solutions can be derived and applied, fostering an environment of collective growth and development, while retaining team autonomy.

2. **Coordination and Coaching:** Special funding is required primarily for coordination and coaching efforts within these communities of practice. The objective is to streamline collective learning, foster collaboration, and cultivate the sharing of effective strategies and techniques among developers and teams.

Examples of shared initiatives that can be addressed within these communities of practice include, but are not limited to:

- Incorporating generative AI tools and workflows into development processes
- Building an intentional community of engagement
- Considering membership in a software foundation, and understanding the implications thereof

4.3 PESO Governance Model- Draft

Governance for a hub and spoke model requires careful checks and balances that consider many factors, with the goal of leveraging DOE funding to optimize the impact of investments to advance science. The PESO team believes that the overall health of the software ecosystem depends strongly on a well-defined, transparent, and comprehensive governance model that weighs priorities across all DOE-sponsored software efforts.

Each product team, product community, and the hub need to define scope that is consistent and complementary within and across levels and come to agreement about target funding amounts. With multiple and competitive interests, the decision-making process will require clear objectives, transparency, and direct engagement from lab management and DOE sponsors.

External stakeholders are also essential to the process since their informed opinions about the needs and trends in the broader scientific computing community will provide grounding for DOE decisions. These stakeholders can also illuminate the potential for DOE-sponsored efforts to complement, leverage, and expand upon work being done outside of DOE.

4.3.1 Governance Entities

- **Hub and spoke entities:** Figure 1 shows the three levels of aggregation in the hub and spoke model: Product team, Product Community, and Hub. All three entities are essential and distinct entities in governance even though many people will participate in two or three levels and may represent more than one team or spoke.
- **Laboratory operations management:** Each DOE lab will designate a management lead and deputy to participate in all funding decisions and regular discussions about ecosystem activities. The lead and deputy will assure that lab interests, concerns, and plans are informed and consistent with the hub and spoke ecosystem. Spokes may have additional roles for lab management, but the lead and deputy roles described here are intended to represent all hub and spoke efforts at each lab. The leads and deputies will meet with the hub and spoke leaders monthly. This entity will also participate in all funding decisions.
- **Lab leadership board:** Each DOE lab will designate a single senior management representative for quarterly strategic discussion with hub and spoke leaders. The role of this entity is to assure that the long-term direction of efforts is on track. This team will also review funding decisions and assist in major leadership transitions.
- **Industry and agency council:** This entity will be composed of members from major organizations that have an interest in DOE software activities. We expect to recruit members from other US agencies, commercial scientific software providers, hardware vendors, cloud providers, major software foundations, international partners, and other organizations whose interests align with and can inform DOE-sponsored efforts. This council will meet quarterly.

4.3.2 Governance Responsibilities

With these factors in mind, PESO proposes the following governance model to guide the hub-and-spoke ecosystem in conducting its activities. More details will be provided as the PESO team continues its conversations with spoke communities.

- **Spoke governance:** PESO recognizes that each spoke will determine its own detailed governance model. However, we believe that the following should be required for the overall health of the ecosystem:
  - **Spoke leadership:** The spoke will have a leadership team that represents the major institutions expected to receive DOE funding. Typically, the leaders will include a representative from each of the DOE labs that have a major investment in the R&D of the spoke.
  - **Spoke call for work:** The spoke team will work with other spokes and the hub to define a general call for work that is then customized by the spoke to include spoke-specific language and expectations.
  - **Spoke prioritization of proposed work:** Once the teams that are part of a spoke and eligible for DOE-sponsorship will respond to the spoke call for work, the spoke leaders will provide a preliminary prioritization of funding decision for their spoke as their governance model outlines.
  - **Spoke participation in combined funding decision process:** Each spoke leadership team will participate in annual funding decisions and any mid-cycle change management events.
- **Hub governance:** The hub itself will coordinate the combined funding decision process.
  - **Hub leadership:** The hub will have a leadership team that represents all major DOE laboratories.
  - **Hub call for work:** In collaboration with the spokes, DOE sponsors, lab management, and external stakeholders, the hub will compose a call for work by distilling common requirements into a set of expectations common to all spokes and the hub itself.
  - **Hub prioritization of proposed work:** Via real-time, in-person discussions with spoke leaders and laboratory operations management, the hub will coordinate the prioritization and selection of funded projects on an annual basis and coordinate mid-cycle change management events.

4.3.3 Governance events

- **Annual funding decisions:** Each year, hub and spoke leadership teams and laboratory operations management will meet to make funding decisions. DOE sponsors will convey any guidance and requirements prior to the decisions and will be shown the proposed decisions before they are finalized.

- **Annual review:** Hub and spoke teams will organize to conduct an annual independent project review. Hub and spoke leaders will present a summary of work from the past year, present status, and future plans. The review team will be composed of subject matter experts. They will provide findings and recommendations that will be used to steer planning for the next year.
- **Incubation and mid-cycle change management:** New ideas emerge more frequently that once a year. Incubation projects can be considered during the annual cycle and at other times. Other important work may become evident as the year progresses. On a monthly basis, any proposed new work would be reviewed and considered for funding using reserves set aside for this purpose.

Additional governance model details are still emerging.

5 Key Takeaways from the PESO Community Input Responses

The Post-ECP Software-Ecosystem Sustainment (PESO) Project has engaged with its community of stakeholders to understand their perspectives on software-ecosystem sustainment and the associated challenges. Based on their responses, the following key takeaways have emerged.

1. **Awareness of Sustainability Importance**: There is a clear and strong awareness within the community regarding the importance of sustainability in the scientific software ecosystem. Stakeholders recognize that without active and ongoing efforts to maintain and update software tools and libraries, the value provided by these resources could diminish over time. Therefore, the emphasis on sustainability is critical for the continuity of high-quality scientific computing.

2. **Shared Requirements and Strategies:** There is a consensus within the community regarding the shared requirements and strategies needed for software sustainability. These commonalities indicate a unified understanding and approach to achieving long-term success in scientific software development and maintenance.

3. **Potential of PESO-Like Organization:** The collective community responses suggest that the establishment of an organization like PESO could provide significant benefits:

- **Better Quality Output:** The collaborative structure and shared expertise within PESO could improve the quality of the software products generated within the ecosystem. Through joint problem-solving and pooling of resources, the set of end products can be better than they would be in isolation.
- **Faster Processes:** PESO's integrated approach and the efficient division of tasks among its members can expedite software development, testing, and deployment processes, leading to faster delivery of scientific computing solutions.
- **Cost Efficiency:** The consolidation of resources and expertise within PESO can lead to cost savings, as efforts can focus on essential tasks without redundancy. As a result, resources can be directed toward other important aspects of scientific computing and research.

A full summary of the community input responses is available in Appendix B.

6 Impact of ECP Investments: Enabling a 100X Improvement in Capabilities

The Exascale Computing Project (ECP) represents a significant, sustained investment in the development of a robust, accelerated, and cloud-ready software ecosystem. Spanning a course of seven years, this initiative has made notable strides in enhancing scientific computing capabilities.

1. **Multi-Vendor Accelerator Utilization:** ECP positioned the HPC community to utilize accelerators from a range of vendors. This broad compatibility enables us to leverage the best-suited technology for every task, effectively maximizing performance and efficiency.

2. **Software Quality Emphasis:** The ECP has put a strong emphasis on software quality, enhancing testing protocols, improving documentation, and refining design processes. This focus on quality assurance has resulted in software that is more reliable, efficient, and effective.

3. **Community Engagement Prioritization:** ECP prioritized community engagement as a key part of its strategy. Through webinars, Birds of a Feather (BOFs) sessions, tutorials, and more, the project fostered a collaborative, knowledge-sharing environment that has driven innovation and broadened the reach of its efforts.

4. **Credible Portability Layers:** The DOE's portability layers have emerged as the credible solution to building codes sustainable across multiple GPUs. The portability layers help to prevent vendor lock-in, avoid growing divergence, and reduce the need for hand tuning in the code base. These approaches allow for more sustainable, long-term development practices.

5. **Lowered Costs and Increased Performance:** The software products developed through ECP are designed to both lower costs and increase performance for accelerated platforms. This balanced approach ensures the creation of software that is not only powerful, but also efficient and cost-effective.

6**. Broadened Application Class:** The ECP's efforts have allowed the Department of Energy to enable a whole new class of applications and capabilities that utilize accelerated nodes, extending beyond the realm of artificial intelligence. This broadened scope has provided industry and academia with the tools needed to tackle increasingly complex computational challenges.

7. **Legacy of the ECP:** The ECP leaves behind a path and a software ecosystem that others can leverage. This legacy ensures that the benefits of the ECP's efforts will continue to be felt, fueling further advancements in scientific software ecosystem development.

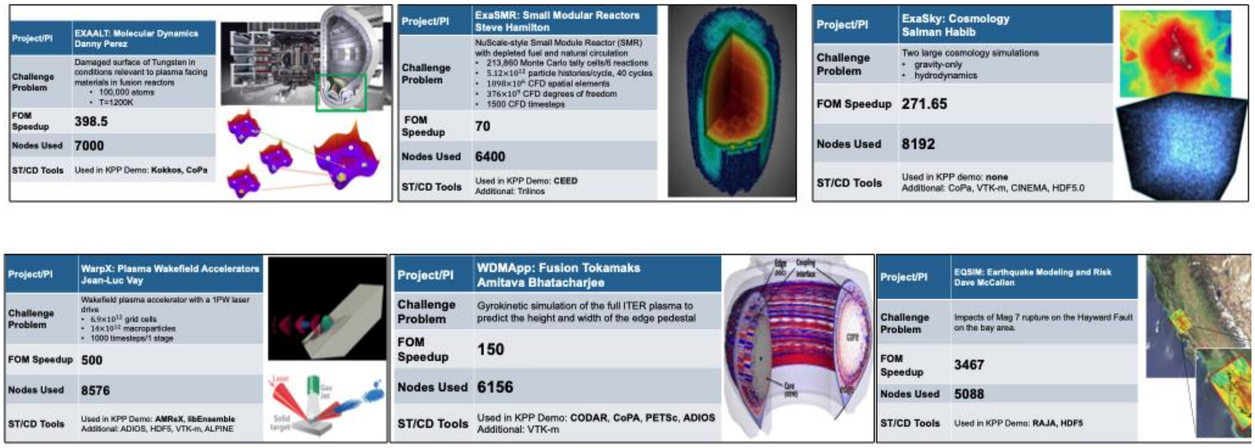

Figure 3: Some ECP application teams were tasked with transforming existing high-performance code bases to run well on the exascale computing systems Frontier and Aurora. These teams established a baseline performance measure at the beginning of ECP. Each team had to achieve a 50-times improvement in their figure of merit to achieve their project goals. Most teams have far exceeded the minimum improvement, as shown in these charts. The approaches used can be leveraged going forward. Part of the success of these teams came from using the libraries and tools that ECP also sponsored. Part of a post-ECP 100X initiative—which would enable many more applications to realize the same performance improvements—is to leverage the libraries, tools, and porting strategies that contributed to ECP application success.

6.1 Leveraging ECP Investments: Enabling a 100X Improvement in Scientific Impact

ECP has generated significant advancements in the field of scientific software-ecosystem sustainment, with potential to further enable a hundredfold improvement in scientific impact.

1. **Room for Improvement:** Despite significant advancements, many scientific software communities continue to rely heavily on home-grown software solutions that lack the sophistication and efficiency of modern software tools and workflows. The following opportunities can be harnessed to improve the situation:
   a. **Migrating from CPU to GPU:** This transition can enable the scaling out to larger problems or scaling in to smaller, GPU-enabled systems, such as laptops. This approach allows for improved computational efficiency and versatility.
   b. **Introducing modern software tools and workflows:** Leveraging the outreach, training, and culture that are focused on improvement can encourage the adoption of modern software tools and workflows, enhancing efficiency and performance.
   c. **Integration into larger software communities:** Incorporating these communities into larger ecosystems, such as the Extreme-scale Scientific Software Stack (E4S) and other software product communities, can lead to wider collaboration, better resource utilization, and improved overall outcomes.

2. **Engagement for 100X Improvement:** The question that arises is how to engage these communities to realize the 100X improvement in science impact. DOE/ECP provides libraries, tools, expertise, and community connections that can be leveraged to achieve this goal. The challenge lies in identifying the best opportunities, overcoming potential impediments, and crafting strategic and tactical plans to bring about this significant improvement.
   a. **Overcoming Impediments:** Promoting the potential 100X impact across the DOE, other agencies, and the industry at large could prove effective. By emphasizing the broader impact and potential advancements these tools and libraries can facilitate, the value proposition becomes more attractive and relatable to a wider audience.
   b. **Leveraging investments and advancements:** Leveraging ECP efforts can bring about a 100X improvement in the scientific impact. With a concerted effort toward modernization, community integration, and an emphasis on the broader impacts, this potential can be realized broadly across the scientific computing community.

6.2 Realizing a 100X Improvement: Opportunities in Scientific Software Ecosystem Sustainment

The Department of Energy's Exascale Computing Project (DOE/ECP) provides a promising pathway to realize a hundredfold improvement in scientific software-ecosystem sustainment. In this context, the following opportunities are potential avenues to achieve this goal:

1. Full Utilization of GPUs: Graphics Processing Units (GPUs) offer superior computational power and are a crucial component in modern high-performance computing. However, current usage often centers on hotspots, which is a start but insufficient for optimal performance. Future GPU devices will have increased scalability, opening up opportunities for superior computational capabilities. Therefore, the porting of scientific software to make full use of GPUs is a significant area to harness.

2. Leveraging the Spack Ecosystem: Spack, a flexible package manager that supports multiple versions, configurations, platforms, and compilers, offers an avenue for major efficiency gains. This ecosystem provides ready access to hundreds of curated libraries and tools, making the integration of sophisticated software more manageable. If Spack recipes are published for specific codes, it eases their consumption, facilitating wider utilization. Moreover, utilizing Spack build caches can result in a tenfold speedup in rebuild times, significantly enhancing productivity.

3. Making Use of the Extreme-Scale Scientific Software Stack (E4S): E4S provides a plethora of resources, including curated libraries, tools, documentation, and build caches. Commercial support is available via ParaTools, allowing access to pre-built containers, binaries, and more. Additionally, the availability of cloud instances for platforms like AWS and Google enables elastic expansion and facilitates neutral collaboration for cross-agency work.

4. Leveraging ECP Team Experience: The ECP team's extensive experience and knowledge in software development for high-performance computing are invaluable resources. By leveraging this expertise, developers can overcome challenges, learn best practices, and ultimately accelerate software improvements.

6.3 Multiple Approaches to Realize 100X Improvement in Scientific Software-Ecosystem Sustainment

The concept of 100X improvement in the realm of scientific software-ecosystem sustainment doesn't pertain solely to the raw enhancement of computing capabilities. Rather, it encompasses a variety of aspects, including exciting new scientific capabilities at the high end, cost reductions, and efficient resource utilization.

High-End Scientific Capabilities

At the upper end of the spectrum, a 100X improvement could be envisaged as groundbreaking new scientific capabilities. These computational advances could be realized through the use of state-of-the-art platforms and software, enabling fundamental breakthroughs in science that were previously unattainable.

Furthermore, leveraging the 100X improvement provides an opportunity for the use of more affordable machines that conveniently fit into current data centers. This approach not only reduces the need for extensive infrastructure changes but also paves the way for enhanced scientific research within existing frameworks.

Cost Reduction

In addition to enhancing capabilities, the concept of 100X also implies achieving similar scientific results at a fraction of the current cost—running the same problems 100X cheaper. This economical approach could lead to significant cost reductions in scientific research, making it more accessible and sustainable.

The migration to accelerated platforms is a key strategy in cost reduction. Such a migration could downsize the requirements from High-Performance Computing (HPC) clusters to more compact systems such as desktop or laptop systems.

Efficient Resource Utilization

An essential aspect of leveraging 100X improvement is efficient resource utilization, especially in the context of energy consumption. As we aim to grow computing capabilities, the aspect of energy efficiency cannot be overlooked.

One way to manage this challenge is by utilizing software ecosystems such as E4S, which are available for container/cloud systems. These systems, while offering high computing capabilities, are more energy-efficient than traditional setups. This approach allows institutions to keep energy costs in check while still growing computing capabilities.

6.4 100X Improvement Recipe: Enhancing Scientific Software Ecosystem Sustainability

The recipe for a hundredfold improvement in scientific software ecosystem sustainability involves an amalgamation of compelling scientific narratives, sufficient funding, dedicated staff, computing resources, training, and the experience and deliverables from ECP, all of which could be delivered via post-ECP organizations such as PESO.

ECP demonstrated that a focused effort on preparing an application code to leverage GPUs can result in two orders of magnitude improvement. Furthermore, ECP efforts show a pathway for other applications to accomplish the same transformation and a collection of libraries and tools that are ready to use on GPU systems. For a given existing scientific application code that runs on CPUs, the following notional "recipe" illustrates the key ingredients and steps to realizing a two order of magnitude improvement.

Ingredients:

1. Compelling science impact stories
2. Funding (ranging from $$ to $$$)
3. Dedicated staff
4. Computing resources
5. Training facilities and materials
6. Deliverables and experiences from DOE/ECP
7. Post-ECP organizations like PESO for the delivery of resources
8. Additional resources as needed

Steps:

1. **Develop Strategy and Plan:** Translate the science story into a clear strategy and execution plan. Leverage the experience from the ECP and other similar projects to ensure effective planning.

2. **Identify Parallelization Strategy:** Determine the node-level parallelization strategy that best suits your needs. Consider options like CUDA, HIP, DPC++, Kokkos, RAJA, OpenMP, and others.

3. **Survey Existing Libraries and Tools:** Familiarize yourself with the existing libraries and tools that can be useful for your project. Look into vendor offerings, resources from the E4S, and others.

4. **Explore Available Platforms:** Consider the range of available platforms that could be used for your project. This might include DOE Facilities, cloud platforms, among others.

5. **Leverage Existing Software Ecosystem:** Use existing software ecosystems to your advantage. This might include using container technologies, Spack, and others to streamline your development and deployment processes.

6. **Engage Software Communities:** Engage with software communities that align with your project's focus. This might involve product communities, communities of practice, among others.

7. **Construct New Codes:** Construct new codes that fit within the broader ecosystem. Ensure they are adaptable, scalable, and compatible with existing systems and libraries.

8. **Produce New Science Results:** Using the developed software, produce new scientific results that enhance our understanding of the world, and demonstrate the effectiveness of your work.

By following this recipe, a project can leverage the power of the scientific software ecosystem to achieve a 100X improvement, enabling more significant scientific breakthroughs.

7 Conclusion: Charting the Course for Scientific Software-Ecosystem Sustainment

The PESO project is dedicated to enabling the overall success of DOE's scientific mission through fostering a robust scientific software ecosystem. Together, the PESO organizational strategy and hub-and-spoke approach foster cooperation, synergy, and efficiency across our software product development teams and communities. The experiences shared by these communities of practice and the insight gained from PESO input responses highlight the collective recognition of sustainability's importance and shared strategic requirements.

This report underscores the transformative impact of ECP's investments. Through diligent development and strategic community engagement, we have accomplished a hundred-fold improvement in software capabilities. The potential for realizing this same magnitude of improvement across various communities is immense, with the utilization of resources such as Spack and E4S offering significant opportunities.

However, realizing this potential requires more than resources; it requires a strategic and coordinated effort across the board. The 100X Recipe presented in this report outlines a framework that leverages the ECP's deliverables and experience, coupled with focused strategies and robust community engagement. Moreover, we acknowledge the various ways that 100X can be leveraged, either by achieving breakthrough science capabilities or by significantly reducing costs.

We are here not just to sustain the scientific software ecosystem but to set it up for success. The strategic and tactical plans, challenges, opportunities, and paths to success outlined herein provide a roadmap for moving forward. By establishing a sustainability effort, we aim to create an ecosystem that can continue to thrive, innovate, and contribute to scientific progress for generations to come. This report serves as a testament to our commitment and a guide for our journey towards a sustainable future for the scientific software ecosystem.

A. June 2023 PESO Community Workshop Details

A.1 Workshop Registered Participants

Ann Almgren, Lawrence Berkeley Lab (LBL)
Hartwig Anzt, University of Tennessee, Knoxville (UTK)
Satish Balay, Argonne National Lab (ANL)
Roscoe Bartlett, Sandia National Labs (SNL)
Pete Beckman, ANL
John Bell, LBL
Daniel Bielich, Ansys
Chris Blanton, NOAA/GFDL
George Bosilca, UTK
Scot Breitenfeld, The HDF Group
Suren Byna, The Ohio State University
Franck Cappello, ANL
Brian Cornille, AMD
Anthony Danalis, UTK
Anshu Dubey, ANL
Alan Edelman, MIT
Jean-Luc Fattebert, Oak Ridge National Lab (ORNL)
Rafael Ferreira da Silva, ORNL
Paul Fischer, ANL
Joerg Gablonsky, The Boeing Company
Ana Gainaru, ORNL
Todd Gamblin, Lawrence Livermore National Lab (LLNL)
Richard Gerber, LBL / NERSC
Berk Geveci, Kitware
Pieter Ghysels, LBL
Roberto Gioiosa, Pacific Northwest National Lab (PNNL)
Yanfei Guo, ANL
Salman Habib, ANL
Paul Hargrove, LBL
Michael Heroux, SNL
Jeffrey Hittinger, LLNL
Bill Hoffman, Kitware
Axel Huebl, LBL
Joseph Insley, ANL
Robert Jacob, ANL
Doug Jacobsen, Google LLC
Balint Joo, ORNL
Christopher Knight, ANL
Tzanio Kolev, LLNL
Jeff Larkin, NVIDIA
Damien Lebrun-Grandie, ORNL
Mary Ann Leung, Sustainable Horizons Institute
Sherry Li, LBL
Meifeng Lin, Brookhaven National Lab (BNL)
Yang Liu, LBL
Li-Ta Lo, Los Alamos National Lab (LANL)
Ray Loy, ANL
Piotr Luszczek, UTK
Victor Mateevitsi, ANL
Lois Curfman McInnes, ANL
Richard Mills, ANL
Misun Min, ANL
Kathryn Mohror, LLNL
David Moulton, LANL
Todd Munson, ANL
CJ Newburn, NVIDIA
DK Panda, The Ohio State University
Michael Papka, ANL
Scott Parker, ANL
Swann Perarnau, ANL
Tom Peterka, ANL
Andrey Prokopenko, ORNL
Kenneth Raffenetti, ANL
Siva Rajamanickam, SNL
Katherine Riley, ANL
Evelyne Ringoot, MIT
Silvio Rizzi, ANL
Jon Rood, NREL
Rob Ross, ANL
Damian Rouson, LBL
Sameer Shende, University of Oregon
Shahzeb Siddiqui, LBL
Andrew Siegel, ANL
Stuart Slattery, ORNL
Rick Stevens, ANL
Miroslav Stoyanov, ORNL
Valerie Taylor, ANL
Keita Teranishi, ORNL
Rajeev Thakur, ANL
Stan Tomov, UTK
Christian Trott, SNL
Jeffrey Vetter, ORNL
James Willenbring, SNL
Michael Wolf, SNL
Carol Woodward, LLNL
John Wu, LBL
Xingfu Wu, ANL
Ulrike Yang LLNL

Workshop Agenda

Thursday, June 8, 2023			
Time (CT)	**Topic**	**Leads**	**Comments**
7:30 am	Registration / Working Breakfast		
8:30 am	Plenary Kickoff	Mike Heroux (PESO PI)	Give overall workshop charge
9:00 am	Applications Perspectives	Andrew Siegel (ECP Applications Development Director)	Challenges and opportunities for increased impact of libraries and tools on application success
9:30 am	Industry Perspectives	Jeff Larkin (NVIDIA), Berk Geveci (Kitware)	Challenges and opportunities for increased impact of libraries and tools in collaboration with industry
10:00 am	Break		
10:30 am	Advanced Computing Facilities Perspectives	Christopher Knight (ALCF), Balint Joo (OLCF), Richard Gerber (NERSC)	Challenges and opportunities for increased impact of libraries and tools in collaboration with computing facilities
11:15 am	Set up for breakouts		Describe charge questions, take Q&A, locate breakout rooms
12:00 pm	Lunch		
12:30 pm	Lunchtime Talk	Ulrike Yang (ECP xSDK Project PI)	How software product communities can enhance the productivity of teams
1:00 pm	Breakout session		
2:45 pm	Break		
3:15 pm	Breakouts resume		
4:00 pm	Report out from breakouts		
5:00 pm	Adjourn		Dinner on your own

Friday, June 9, 2023			
Time (CT)	**Topic**	**Leads**	**Comments**
7:30 am	Working Breakfast		
8:30 am	Challenges and Opportunities for Computing	Rick Stevens (ANL, Assoc Lab Director, Computing, Environment and Life Sciences)	Roles of sustainable software ecosystems in addressing next-generation computing challenges
8:45 am	Hub and Spoke Model, Role of Sustainability Funding, Q&A	Mike Heroux (PESO PI)	
9:30 am	Software Foundations	Todd Gamblin (PESO co-PI)	How we can leverage software foundations for DOE software sustainability
9:45 am	Workforce Development	Lois Curfman McInnes (PESO co-PI), Mary Ann Leung (Sustainable Horizons Institute)	Challenges and opportunities for broadening participation in the HPC workforce
10:00 am	Break		
10:30 am	Breakout		
12:00 pm	Working Lunch		
12:15 pm	Report out from breakouts, discussion		
1:30 pm	Adjourn		

A.2 Breakout Sessions Summary

Background

Much of the time at the June 8 – 9, 2023 PESO Workshop was spent in breakout sessions. We split the group of about 80 participants into four 20-person groups. Each group had the opportunity to address eight questions on Day 1, in round-robin order to assure that each question received sufficient consideration.

On the second day, we asked all teams to address two questions. The first question focused on workforce topics and the second was an open-ended request to discuss topics that had were not specifically stated.

This section of our report lists the question and then provides a summary of all the comments captured during the breakout sessions.

Breakout Session Questions

Group A: Address questions in this order: 1, 2, 3, 4, 5, 6, 7, 8
Group B: Address questions in this order: 3, 4, 5, 6, 7, 8, 1, 2
Group C: Address questions in this order: 5, 6, 7, 8, 1, 2, 3, 4
Group D: Address questions in this order: 7, 8, 1, 2, 3, 4, 5, 6

Day 1:

1. Using the following definitions (Draft PESO Definition of Sustainability) of sustainability as a reference,
    a. **What are we missing in the definition?**
    b. **Can we remove anything?**
2. The success of DOE-sponsored libraries and tools hinges on addressing emerging and anticipated applications, facilities, and other stakeholder requirements in service of DOE's mission.
    a. **What are some ways to assure we are identifying and meeting these needs?**
    b. **How can we optimize our impact in collaboration with stakeholders?**
3. How do we transition activities out of our funding portfolio to create room for new activities?
    a. **How can we transition software products to community ecosystems, vendors, and software foundations?**
    b. **What are other ways to create space for new efforts?**
4. PESO is proposing a decentralized Draft PESO Financial Model.
    a. **Do you see any problems with the model?**

    b. **The model is a high-level sketch. What important details must be considered?**

5. PESO proposes to be a hub for software product communities for these Draft PESO Key Services and Activities that are beneficial across all communities and product teams.
    a. **Do you see any issues with this approach?**
    b. **What are some important details that must be considered?**
    c. **How can we assure the ability to fund new projects even if budgets remain level or grow only modestly?**
6. PESO proposes to support software product communities and teams by fostering crosscutting activities that lead to better practices, processes, tools, and community growth.
    a. **What are some of the most important crosscutting activities PESO should promote and support?**
    b. **How should these activities be organized and provided to the software communities and teams?**
7. PESO proposes to sponsor annual events across the entire community.
    a. **How important is an annual in-person meeting that brings together teams, stakeholders, and key members of the external community?**
    b. **The ECP sponsored a virtual Community BOF Days. Is it useful to continue this event in the future?**
8. Prior to this workshop, the PESO team requested input from the community. The key questions and a summary of the 40 responses are found here: 00-Workshop Input Questions and Responses
    a. **Which topics are most important to consider carefully?**
    b. **From the questions and summaries, what would you change?**

Day 2:

9. PESO is committed to workforce development, especially by reaching out to under-represented groups and creating a culture that is inviting, and by promoting the continued training of workforce members and stability of career paths.
    a. **What are the top three workforce challenges or impediments that you see in your organization(s)?**
    b. **What strategies and activities are currently helping to address workforce challenges? Should these activities be continued?**
    c. **What are the most promising new strategies and activities to address workforce challenges for the future and why?**

A.3 Question 1: Software Ecosystem Sustainment Definitions

What are the key elements in the definition of software sustainability?

Sustainability in the context of the scientific software ecosystem, particularly High-Performance Computing (HPC) libraries and tools, can be defined through the following key elements:

1. Sustained Resources: Availability of funding, human effort, workforce, and infrastructure to support the continued development and maintenance of HPC libraries and tools.
2. Robust User and Developer Base: A healthy and engaged user and developer community, contributing to the software's evolution to meet community needs.
3. Ecosystem Management: Explicit management of transitions into, out of, and within the HPC ecosystem. Vendor products are integral to planning and collaborations. The ecosystem should be designed to sustain critical functionality even as specific products come and go.
4. Critical Mass: Using a hub-and-spoke approach to combine efforts, leveraging the aggregate scale of the community for external influence.
5. Trust: Ensuring the software continues to exist and evolve with user needs. Redundancies in both concepts and concrete implementations are required to maintain trust.
6. Policies & Standards: Establish well-documented API and data interfaces to define/influence standards for compatibility and scalability.
7. User Base Growth and Performance Optimization: Assure that usability concerns do not lead to design choices that inhibit the ability to tune for optimal performance. Ensure that HPC optimization can be performed by a strong core group focused on performance impacting code modules.
8. Relevance: Fit into the emerging workflows of users, adapt to external drivers (new architectures/systems/needs), amplify sustainability through high-level languages, and encourage user extensibility. Balance between generalization and specialization is essential.
9. Sustain Ecosystems: Allow for innovation, transition, and research within the product suite components, rather than focusing on the sustainability of individual products.
10. Healthy Communities and Stable Workforce: Emphasize the creation of a sustainable HPC software ecosystem with a focus on maintaining a stable workforce.
11. Forward-Looking Sustainability: Sustainability should include elements of future planning, not just maintenance or backward-looking aspects.
12. Requirements, Policies, and Processes: These are needed for compatibility and interoperability of updated versions of tools and libraries and for supporting their sustained development.
13. Influence in Broader Community: This includes working with standards bodies and other entities to drive broader implementation of sustainability principles.
14. Engagement with Larger Communities: Engage with larger entities like DOE, ECP, and Advanced Scientific Computing to build a larger contributor base.

15. Leverage Aggregate Scale: Use collective influence for external impact, e.g., influencing vendors.

This definition of sustainability highlights the necessity of long-term planning, collaborative effort, community engagement, and adaptability in maintaining and growing a robust scientific software ecosystem.

A.4 Question 2: Determining Stakeholder Requirements

The success of DOE-sponsored libraries and tools hinges on addressing emerging and anticipated applications, facilities, and other stakeholder requirements in service of DOE's mission.

> ***What are some ways to assure we are identifying and meeting these needs?***
> ***How can we optimize our impact in collaboration with stakeholders?***

Identifying and Meeting Needs:

1. Address the funding challenge by seeking support beyond ECP and SciDAC for the HPC/CS aspect of projects.
2. Implement integrated teams with "multilinguals" who can bridge the gap between applied mathematics, domain science, and CS.
3. Foster industry job creation by creating pipelines from universities and labs into relevant industries.
4. Use real-world examples of AT+ST stacks adopted in startups as proof of meeting needs and driving innovation.
5. Collaborate with industry for mutual benefits, like in-kind contributions and open-source releases.

Optimizing Impact:

1. Communication with Domain Sciences & Applied Science Offices to understand the needs for 100x cases, enabling community knowledge addition, addressing user bottlenecks, and enabling end-to-end autotuning.
2. Consider using foundation concepts from Linux Foundation or NumFOCUS to simplify financing and contributions and establish trust in sustainability.
3. Adopt or build upon compute facility approaches to requirements gathering.
4. Coordinate with DOE program management on upcoming initiatives and maintain close interactions with facilities and vendors.
5. Target broadly used or high-profile software, especially those associated with new instruments.
6. Define metrics and begin data gathering early to aid in evaluation and reprioritization over time.
7. Provide stakeholders and their requirements to project teams to aid in accurate planning.
8. Get early access to testbeds, hardware roadmaps, and procurement plans.
9. Encourage the use of sustainability centers by including language in Funding Opportunity Announcements (FOAs) pointing to these resources.
10. Establish a sustainability center as a liaison between domain scientists and software libraries to understand and communicate their needs.
11. Engage junior members for deeper technical discussions.
12. Maintain a broad focus to prepare software for future DOE needs.

13. Acknowledge that stakeholders are not only users, but also facilities, funding agencies, and developer communities.

Stakeholder-specific Strategies:

1. Funding agencies: Maintain connections to identify and address their challenges. Interact with POCs at labs for updates and to showcase achievements.
2. Facilities: Provide well-tested software on their systems and enable testing at the facilities. Support software packages with knowledgeable developers.
3. Developer communities: Facilitate information sharing and compatibility strategies.
4. Users: Create infrastructure for outreach, learning from the business community on engagement and requirement gathering practices.
5. Broad community: Conduct outreach to a wider audience, including taxpayers.

A.5 Question 3: Creating Room for New Activities

How do we transition activities out of our funding portfolio to create room for new activities?
How can we transition software products to community ecosystems, vendors, and software foundations?
What are other ways to create space for new efforts?

Transitioning software products to community ecosystems, vendors, and software foundations:

1. Define the goals of an ecosystem, rather than focusing on individual products.
2. Encourage innovation, transition, and research within the product suite components.
3. Benchmark and quantify progress and success.
4. Diversify funding sources through superior user support, including potential venues like crowdfunding, GitHub Funding, IssueHunt, NumFOCUS, etc.
5. Consider the potential for lightweight incorporation or formalism.
6. Core-support (ASCR) should continue for research & sustainability.
7. Ascertain that transitioning the product won't pick 'winners' or 'losers' among user communities or other stakeholders.
8. Explore options for transitioning responsibility to the product's user base or an independent software foundation.
9. Gauge user reliance on the software and identify possible alternatives.
10. Evaluate the potential for project growth and stakeholder expansion.

Other ways to create space for new efforts:

1. Encourage incubator projects with concrete timeline goals for adoption and impact.
2. Aim for the cost to maintain a community/ecosystem/product to proportionally decrease over time due to improvements in tools, workflows, automation, better documentation, and integration with other projects or evolving standards.
3. Funding for sustaining a project could decrease annually while funding for new features or disruptive changes could be kept separate.
4. User/R&D steering committees could balance the decisions between sustaining old projects and introducing new ones.
5. End or defund projects that are underperforming or unsuccessful.
6. Standardization and consolidation of some duplicate efforts. Some duplication is good for competition and innovation, but too much can waste resources.

Transitioning out of PESO funding:

1. Decision to ramp down could be made by PESO or by the project itself.

2. Transition "up and out": possibly to a foundation, but still supported by PESO as external parties can engage more actively.
3. PESO could provide seed money to help the project transition to a different funding model.
4. Conduct yearly review of the project to evaluate ramp-up/ramp-down criteria.
5. Develop a comprehensive understanding of the user base. Can the project name its top 10 users?
6. Evaluate projects based on their life-cycle stage.

Maintaining and Transitioning Projects:

1. Communication with stakeholders is essential to understand the impact of ramp-down.
2. The voice of larger users may have more weight.
3. Adoption of software contributes to its sustainment. A representative of user voices needs to be involved in decision making.
4. Funding should be proportional to the size/impact of the project and the need for the project.
5. Projects could be required to run on LC machines and be available through GitLab/GitHub.

Commercialization and Vendors:

1. Never transition a project to a single vendor. It could result in software working only on their hardware, discouraging use by other companies.
2. There can be challenges in getting enough external community buy-in to fully fund projects.
3. Provide training to software teams on commercializing their projects, discussing when it's desirable and how to approach it.

A.6 Question 4: PESO Financial Model

The draft PESO Financial Model is defined below:

Funding is based on a transparent set of criteria applied through an open review and assessment process.

1. *Criteria for product selection will be pre-determined and published openly*
2. *Teams will propose work scope, schedule, and requested budget with milestones*
3. *Budget allocations will be determined annually by a representative team of sponsors, stakeholders, and community experts*
4. *Each team will produce an annual progress report and participate in an annual project review that will feed into budget planning for the following year*
5. *Funds to support product community leadership will be derived from the total funding of a product community*

Criteria for project funding include:

1. *Need for sustaining important products that require DOE-sponsored funds*
2. *Meeting emerging and anticipated needs of application development teams, facilities, and the broader software ecosystem*
3. *Staff and team training and development*

Funds for all efforts at a particular lab are sent directly from the sponsor as a lump sum to the lab with itemized amounts for individual projects at that lab. Funds meant for university and industry subcontracts are sent from the sponsor to the most appropriate partner lab. All funds at each lab are managed by the normal lab funding infrastructure.

1. *PESO will not centralize funds or create a parallel funds management system*
2. *Each lab will have a designated financial point of contact for all sustainability funds*
3. *Reserve funds will be set aside to support in-cycle funding of newly identified gaps and opportunities that might arise within the funding year*

> ***Do you see any problems with the model?***
> ***The model is a high-level sketch. What important details must be considered?***

Funding Process:

1. Uses a transparent set of criteria in an open review and assessment process.
2. The criteria for product selection are pre-determined and openly published.
3. Teams propose work scope, schedule, and budget with milestones.
4. Budget allocations are decided annually by a representative team of sponsors, stakeholders, and community experts.
5. Each team produces an annual progress report and participates in an annual project review feeding into the budget planning for the following year.

6. Funds for product community leadership are derived from the total funding of the product community.

Funding Criteria:

1. The need to sustain important products that require DOE-sponsored funds.
2. Meeting emerging and anticipated needs of application development teams, facilities, and the broader software ecosystem.
3. Staff and team training and development.

Funding Distribution:

1. The funds for all efforts at a particular lab are sent directly from the sponsor to the lab with itemized amounts for individual projects.
2. Funds for university and industry subcontracts are sent from the sponsor to the most appropriate partner lab.
3. All funds at each lab are managed by the normal lab funding infrastructure.
4. No centralized funds or parallel funds management system.
5. Each lab has a designated financial point of contact for sustainability funds.
6. Reserve funds are set aside to support in-cycle funding of newly identified gaps and opportunities within the funding year.

Possible Problems:

1. Lack of clarity in how the model will facilitate long-term funding.
2. The model does not mention how to move money between the labs for single projects that span labs.
3. It's unclear how a company would contribute to support a project, or how the model allows for outside mechanisms to fund projects.
4. Unclear how the model would cover costs for central services like machine for Continuous Integration (CI).

Additional Details Needed:

1. Transparent selection process and yearly milestone setting.
2. Incentives based on product usage across multiple labs and projects.
3. Consideration of moderate sustainability needs ($250-$500K/yr for most).
4. How to handle potential fizzle-out reduction per year.
5. Define relationships between PESO and other bodies like the RSE Institute.
6. Define activities that benefit multiple codes, such as support for CI.
7. The level of sustainability required and how funding will be tied to it.
8. It's crucial to specify that funds sent to each institution will be designated for specific products.

A.7 Question 5: PESO Key Services

PESO proposes to be a hub for software product communities for these [Draft PESO Key Services and Activities] that are beneficial across all communities and product teams.
the draft PESO Key Services and Activities which are:

- *Steer funding to key projects to maintain a robust software ecosystem in the long term.*
- *Delivery and deployment of DOE-sponsored software and dependencies via Spack and E4S to DOE Facilities and other computing centers, via source and containers to cloud users and developers. Services include CI testing on advanced node types, issues triage, build caches, and software quality assurance.*
- *Coordination of cross-cutting engagement with DOE Facilities, DOE sponsors, and other stakeholders such as other US agencies, US industry, and international partners*
- *Lightweight processes, models, and tools to support SPCs with their software lifecycle management activities, including annual planning, execution, tracking, and assessment with change management.*
- *Basic infrastructure for community engagement in coordination with communities of practice for outreach, training, community development, and coordination with external collaborators.*
- *Outreach and workforce development to incubate new projects and grow the contributor base and external investment in key projects.*

Do you see any issues with this approach?
What are some important details that must be considered?
How can we assure the ability to fund new projects even if budgets remain level or grow only modestly?

Addressing PESO Key Services and Activities with DOE-sponsored libraries and tools:

1. Steer funding to key projects for maintaining a robust software ecosystem long term
    - Assessing the role of PESO in choosing software products for support vs. acting as a funding vehicle.
    - Defining and agreeing on responsibilities (facilities, PESO, software teams, etc.) for various aspects (installation, issue tracking, bug fixing, integration testing, etc.).
2. Delivery and deployment via Spack and E4S to DOE Facilities, other computing sites, and cloud users
    - Addition of vulnerability scanning/compliance with automated reporting as part of the release process.
    - Consideration of E4S's relation to ensuring interoperability between tools and applications and exploring other ways to ensure interoperability.
3. Coordination of cross-cutting engagement with stakeholders
    - Ensuring transparent information flow about stakeholder and sponsor needs to product teams.

- Expanding engagement coordination to include interactions with developer teams.

4. Supporting SPCs with software lifecycle management activities
    - Avoiding one-size-fits-all statements, establishing minimum requirements, and facilitating additional aspects as much as possible.
    - Evaluating different standards for different types of software, ensuring these standards are meaningful.
5. Basic infrastructure for community engagement
    - Recognizing the need for a common services project(s) benefiting multiple code projects.
    - Acknowledging the difference in standards for different tiers within PESO.
6. Outreach and workforce development
    - Nurturing new projects and growing the contributor base and external investment in key projects.
    - Recognizing the need for experts to connect users with developers.

Issues with this approach:

1. Lack of developer involvement in the engagement with DOE facilities, sponsors, and other stakeholders.
2. Lack of clarity around PESO's role in 'steering funding' and in maintaining a robust software ecosystem long term.

Important details to consider:

1. Dynamic potential for new members to join communities or projects.
2. Clarity around who would serve as the hub, e.g., higher-level functionality provider & vertical integration.

Assuring the ability to fund new projects:

1. Reducing funding for sustainment over time.
2. Earmarking resources for new projects.
3. Assisting projects in finding external funding.
4. Approaching with a task-centric model for finer grain allocations.
5. Developing criteria for sunsetting projects.
6. Providing short-term "feasibility funding" or incubator funding for projects to demonstrate their worth/feasibility for inclusion in the portfolio.

A.8 Question 6: Crosscutting Activities

PESO proposes to support software product communities and teams by fostering crosscutting activities that lead to better practices, processes, tools, and community growth, considering the following questions:

> ***What are some of the most important crosscutting activities PESO should promote and support?***
> ***How should these activities be organized and provided to the software communities and teams?***

PESO's proposed support for software product communities and teams via fostering crosscutting activities for better practices, processes, tools, and community growth:

Cross-Cutting Activities PESO should promote and support:

1. Regular meetings to bring users/apps and ST developers/tools together, with potential for overarching events such as the "DOE Scientific Software Conference".
2. Birds of a Feather (BoFs) & Tutorials for each Ecosystem, focused on user-friendliness, feature overviews, and synergy exploration.
3. Encourage workforce development and diversity within the software community.

Organization and provision of these activities to the software communities and teams:

1. Adopt the Salishan Conference Model, with one plenary session per day, encouraging much dialogue and cross-industry sponsorships. Co-organization by PESO, STEP, and other seedlings.
2. Continuous Integration services, both centralized and focused, for projects.
3. Enabling product communities/teams to scope out cross-cutting feature implementation, including allocating developers to multi-project integrative tasks and cross-project integration CI.
4. Comprehensive training programs, covering a broad range of topics like general HPC background, best practices, software management, lifecycle training, and workflow reproducibility. This can be achieved through hackathons, collaboration with facilities, and cloud platforms.
5. Assistance in developing tutorials and course materials, as well as encouraging application to key conferences (e.g., SC, ISC).
6. Community engagement through Slack, Discord, and other similar channels, fostering a space for open question-and-answer exchanges.
7. Encouragement of standardization within product communities, like defining common formats for data structures (e.g., sparse matrices) and interfaces that let different products interact.
8. Promotion of Developer Relations (DevRel) activities, including inviting non-DOE individuals to give training on non-DOE products.

9. Early access to hardware for community members, achieved through collaboration with vendors.
10. Digital marketing efforts such as website building, web services, podcasts, PR spotlights, etc.
11. Monitoring of project metrics for community engagement (e.g., GitHub issues and Slack questions responded to).
12. Sponsorship of awards, fellowships, special issues, and other activities that recognize "best software product".
13. Broader efforts to promote scientific software as a legitimate, recognizable academic activity.
14. Provision of opportunities for projects to gain new contributors/users.

Important crosscutting activities:

1. Training in essential tools (CI, documentation, packaging, workflows, repositories, etc.), programming models, math libraries, and best practices.
2. Developing infrastructure to support community development.
3. Planning and executing user engagement activities.
4. Refining community policies for extensibility, interoperability, and sustainability.

How these activities should be organized and provided:

1. Organizing Hackathons and targeted workshops within and beyond DOE entities.
2. Conducting webinars for community-wide learning and interaction.

A.9 Question 7: Annual Events

PESO proposes to sponsor annual events across the entire community, considering the following questions:

How important is an annual in-person meeting that brings together teams, stakeholders, and key members of the external community?

The ECP sponsored a virtual Community BOF Days. Is it useful to continue this event in the future?

Annual In-Person Meeting Importance:

1. Essential for team, stakeholder, and community engagement.
2. Should be hosted at an easily accessible location near an airport hub.
3. Provides a platform for project governance discussion and determining future direction.
4. In-person stakeholder meetings highly beneficial for community interaction, project collaboration, and training.
5. Consideration for a broader DOE HPC annual meeting involving different domains and facets of scientific discovery through advanced computing.
6. Need for pre-planning with dates known at least 8+ months in advance.
7. Possibility of every other year being fully remote to accommodate participants who cannot travel.
8. Should be scheduled after funding is received for all project members.

Continuation of Virtual Community BOF Days:

1. Generally viewed as useful with 3/17 people affirming its usefulness and none opposing.
2. Provides an accessible platform for updates and communication.
3. Opportunity for showcasing results, successes, and generating interest through highlight events, press releases, and newsletters.
4. Platform name should be more overarching and representative, like LSSW.io.
5. In-person interactions perceived as more valuable than virtual presentations.
6. Possibility to time this event 6-months from SC for keeping teams connected with users.
7. Should not be tied to an in-person event to ensure online participation feels equally important.

Other Suggestions:

1. Organize meetings to spread best practices, trainings, and foster connections between projects.
2. Need for substantial interaction time, avoiding too many PowerPoint presentations.
3. Consider co-locating with other meetings, especially a sustainability-focused meeting.
4. Recognize the value of user-facing meetings, tutorials, BOFs, and activities that help build community around products.
5. Value seen in pooling resources for event logistics and promotion.

A.10 Question 8: Key Topics from PESO Input Request Responses

Prior to this workshop, the PESO team requested input from the community. The key questions and a summary of the 40 responses are found here: 00-Workshop Input Questions and Responses

> ***Which topics are most important to consider carefully?***
> ***From the questions and summaries, what would you change?***

Important Topics:

Encourage user transition into developers:

1. Cater to different stages of user growth: new user, power user, developer, and core contributor.
2. Expand Research Software Engineer (RSE) career opportunities.
3. Increase faculty entry positions.

Improving software sustainability:

1. Identify top actions for better software sustainability.
2. Define key factors for software product selection and assessment.
3. Determine needed annual funding for software development, delivery, and user support.
4. Define successful stewardship for Department of Energy (DOE) software.

Criteria for project funding:

1. Include not only established projects but also those with emerging potential.
2. Fund competing projects with similar functionality.
3. Maintain balance between mature and emerging projects.
4. Define criteria for "emerging projects" worthy of sustainment.

Successful stewardship of DOE software:

1. Provide structure, success metrics, advertising, and cross-training.
2. Host and archive virtual training.
3. Guarantee long-term, reliable funding while keeping project teams productive.
4. Report to a Subject Matter Expert (SME) who understands project contributions.
5. Allow project adaptability according to changes in priorities or environment.
6. Train for sourcing alternate funding.
7. Avoid micromanagement.
8. Assure users of product longevity.
9. Facilitate innovation.

Suggested Changes/Additions:

1. Open the DOE-sanctioned RSE career path for Early Career Research Program (ECRP).
2. Facilitate a clear transition path from users to core contributors.
3. Prioritize and define criteria for emerging projects.
4. Establish a sustainable and reliable funding model.
5. Refine selection criteria for balanced representation between mature and emerging projects.
6. Implement a solid stewardship structure, with emphasis on training, success metrics, productivity, and effectiveness.
7. Encourage an environment that supports innovation and assures product longevity.

A.11 Day 2 Question 1: Workforce Development

PESO is committed to workforce development, especially by reaching out to under-represented groups and creating a culture that is inviting, and by promoting the continued training of workforce members and stability of career paths.

> ***What are the top three workforce challenges or impediments that you see in your organization(s)?***
> ***What strategies and activities are currently helping to address workforce challenges? Should these activities be continued?***
> ***What are the most promising new strategies and activities to address workforce challenges for the future and why?***

Workforce Challenges:

1. Need for exceptional breadth of knowledge amongst staff.
2. Uncertainty after the end of ECP; VISA issues, hiring stops, and potential employee transition to other divisions or industry.
3. Need for the growth of workforce both inside DOE and in the broader ecosystem.
4. Perceptions and potential negative impacts of staff transitioning to industry.
5. Difficulty maintaining diversity, attracting mid-career professionals, and hiring foreign nationals.
6. The risk of losing highly specialized staff, such as GPU experts, to competition from industry and universities post-ECP.
7. Lack of coherent strategy for career progression/development.
8. Challenge of outreach and HPC on-ramps for students from underrepresented groups.

Current Workforce Development Strategies:

1. Deep technical training / onboarding in HPC (ATPESC), including internships and fellowships that attract future hires, particularly underrepresented groups.
2. Strong, long-term mission at DOE labs (e.g., ECP, SciDAC program).
3. Emphasis on open-source development and contributions, which has led to external hires.
4. Creation of pipelines via student and postdoc positions, particularly through established connections.
5. Using mission and impact to attract individuals, despite potentially lower salaries compared to industry.
6. Efforts to improve diversity through programs like FAIR and requirements for PIER plans in proposals.
7. Activities initiated by the ECP Broadening Participation Initiative to expand the pipeline and workforce for DOE computing sciences: a multi-lab approach to the Sustainable Research Pathways Program, an Intro to HPC Bootcamp, and the HPC Workforce Development and Retention Action Group

Future Workforce Development Strategies:

1. Establishing attractive training opportunities for Deans and Senior Faculty.
2. Adopting new development/productivity tools/workflows (e.g., GitHub, AI/ML assistants).
3. Marketing mission, projects, tools, and career paths.
4. Expanding remote work options to broaden the hiring pool.
5. Leveraging Google Summer of Code and creating immersive experiences for undergraduates.
6. Developing a curriculum around HPC based in the cloud to lower barriers to entry.
7. Advocating for Research Software Engineers (RSEs).
8. Providing more funding for education and workforce development.
9. Leveraging platforms like YouTube for content dissemination and self-learning.
10. Collaborations with non-traditional institutions and underrepresented groups.
11. Encouraging staff teaching appointments at diverse institutions.
12. Developing materials for educators focusing on software sustainability and engineering for science.
13. Seeking more stability/funding predictability from ASCR, considering longer project durations and staggering project end dates to avoid simultaneous workforce reductions.
14. Expanding and refining work initiated by the ECP Broadening Participation Initiative.

A.12 Day 2 Question 2: Important Software-ecosystem Sustainment Questions Not Addressed

1. Recognizing that existing legislation in some states can complicate Diversity, Equity, and Inclusion (DEI) efforts.
2. Viewing software as an infrastructure that needs long-term support and adoption/extension.
3. The rising importance of software citation and reproducibility, especially given the increased need for transparency as large data sets often cannot be shared.
4. Leveraging industry partnerships for adoption, growth, and community services. This includes SBIR/STTR for product creation and sharing ownership through foundations for international collaborations and funding.
5. Prioritizing risk mitigation and guidance for teams. This entails having redundancies/alternatives in the ecosystem such as multiple C++ compilers and standardized APIs.
6. Exploring potential partnerships with OSTI for secure hosting services.
7. Acknowledging the crucial role of Continuous Integration (CI) as a service that PESO can provide, and the need for broader socialization of CI issues.
8. Recognizing the importance of supply chain security, which good CI can help ensure.
9. Determining how PESO can effectively reach out to applications in other offices and coordinating with other Department of Energy (DOE) funding sources on how the software stack enables their work.
10. Discussing the parts of the ecosystem not directly tied to specific sustainability calls, such as applications.
11. Understanding the necessity for not just good infrastructure, but also training for teams to adopt new technologies.
12. Recognizing the need for an AI component within PESO and exploring how AI can be leveraged as a tool for sustainment.
13. Encouraging increased interaction with the academic world via communication, publications, conferences, workshops, invited talks, etc.
14. Addressing the uncertainties about the plan for the "seedlings", including timelines for projects and expectations regarding new funding.
15. Discussing collaboration with the industry to alleviate funding challenges, with potential industry contributions to open-source tools and integration of scientific software with the broader software community.
16. Questioning whether PESO will have an advisory board and who should be included in it.
17. Considering PESO as a "sales" organization for the community, facilitating connections with industry and other stakeholders, and providing approved resources like slide decks for use.
18. Defining how PESO differs from a SciDAC Institute.
19. Identifying a mission goal for PESO to strive for and developing a motto or tagline to represent it.

B. Summary of Responses to PESO Input Requests

In April 2023, the PESO Project requested input from the scientific software community on key topics related to software-ecosystem sustainment. This document provides a summary of those responses.

B.1 What factors do you think should be considered when selecting and assessing a software product as part of a software sustainability organization portfolio?

When selecting and assessing a software product as part of a software sustainability organization portfolio, a multitude of factors should be considered. These can be broadly categorized into:

1. User Base & Community: Assess the current and potential user base, both within specific domains (like the Department of Energy - DOE) and across wider scientific and industrial applications. In addition, consider the size of the developer community, the participation of the product developers in the software community, and the vibrancy of the community that maintains and develops the product.

2. Relevance & Importance: Evaluate the product's relevance and importance to its users and to the organization's mission, such as its criticality to technology, level of usage in high-performance computing (HPC) communities, and its inclusion in significant initiatives like large DOE hardware procurement or Software Development Kits (SDK).

3. Software Quality & Maintainability: Look at the quality of the product, its maturity, and best software development practices. This includes unit testing, continuous integration/continuous delivery (CI/CD) practices with high test coverage, code quality, clear documentation, a good build system, and responsive, active development. The design of the product should encourage sustainable development.

4. Usability & Scalability: Ensure the product is easy to use and can scale well to large problems and machines. This includes considering installation procedures, user support mechanisms, and deployability.

5. Performance & Portability: The product should exhibit superior performance and be portable on future systems, considering the overall performance, scalability, and programmability of the software.

6. Support & Sustainability Plan: Consider the level of support for the product, such as superior customer support, a supporting ecosystem around the product, and a dependable developer community. Furthermore, there should be a sustainable plan in place for long-term sustainment.

7. Innovation & Potential for Growth: Evaluate the potential for the product to fill an unmet need in the community and its potential for growth, especially in emerging areas. A quick adoption rate could indicate high future impact.

8. Alignment with Existing Portfolio: The product should be assessed in context with the existing portfolio and its alignment with the mission of the organization's portfolio. It's crucial to avoid unnecessary overlap with other products.

9. Openness & Extendibility: Open-source products with numerous contributors, adherence to standards, and a well-maintained development community are highly preferred. Moreover, the software should be extensible and maintainable.

10. Uniqueness & Added Value: The product's unique capabilities, either in performance or functionality, should be assessed, as well as the added value it brings to the community.

Lastly, while assessing the above factors, it's crucial to ensure a balance between supporting successful, mature products and nurturing innovative, emerging ones.

B.2 What factors do you think should be considered when selecting and assessing a software product as part of a software sustainability organization portfolio? Facilities' perspective

The factors to consider when selecting and assessing a software product for a software sustainability organization portfolio, as identified by various facility staff, include:

1. Interaction with Vendors: The relationship and communication channels with the software's creators or suppliers.
2. Performance Portability: The software should have the ability to efficiently operate across different hardware configurations or operating environments.
3. User and Project Demand: The software should have a significant user demand or meet the requirements of major projects.
4. Broad and Strategic Impact: The software should have a widespread influence, particularly on strategic objectives such as enabling portability or enhancing key investments.
5. Support and Sustainability Models: The mechanisms for maintaining and updating the software, and its long-term viability.
6. Utilization by the HPC Applications Community: The software's usage rate and acceptance within the high-performance computing (HPC) community.
7. Community Interest and Participation: The software should generate interest from the user community, who should be actively involved in its maintenance and enhancement.
8. User Base and Potential for Growth: The software should have a large user base or the potential for one, and it should be vital enough to generate organic growth.
9. Impact and Value Added: The software's influence on its user community and the added value it brings.
10. Infrastructure: The software should provide valuable infrastructure to support its user base and future growth.
11. Vision for the Future: The software should be forward-looking in all aspects, anticipating and accommodating future needs and trends.

B.3 What are the critical factors that determine whether you will use an open-source library or tool?

The critical factors that determine the use of an open-source library or tool are:

1. Licensing: The library or tool must have acceptable license constraints and restrictions. It should be compatible with the project's or organization's licensing terms.

2. Functionality: The tool or library should be able to fulfill specific needs and improve productivity.

3. Developer and Community Support: The tool or library should have ongoing support from its developers, including regular updates and bug fixes. A robust user and developer community is also crucial for problem-solving and help.

4. Maturity and Documentation: The tool or library should have reached a level of maturity, proven stability and should have comprehensive documentation to assist its users.

5. Compatibility and Portability: The library or tool should support the programming languages used in the project and be portable across different platforms.

6. Quality: The library or tool should have a high standard of development quality, including extensive testing and active maintenance on bug reports and pull requests.

7. Ease of Use and Learning Curve: The tool or library should be easy to use, have a short learning curve, and be capable of integration with other tools and libraries.

8. Active and Welcoming Development Team: The team behind the library or tool should value user feedback and provide a supportive environment for users.

9. Performance: The library or tool should deliver high performance and low latency.

10. Sustainability: There should be recent development activity and a long-term support model, indicating a high likelihood that development will continue.

11. Vendor Support: For some users, particularly in High Performance Computing (HPC) environments, vendor support may be a significant factor.

12. Part of a Larger Ecosystem: The tool or library should preferably be part of a larger, friendly user base, like Stack Overflow or other forums, which can offer additional support and resources.

Overall, the decision to use a particular open-source library or tool is a balance of these factors, weighed according to the specific needs of the project or organization.

B.4 What are the top 3 things that your software project should do to improve its sustainability that a software sustainability organization could provide support for in some way?

The common themes in improving software project sustainability based on the responses are:

1. Testing and Quality Assurance: There's a strong emphasis on enhancing the testing infrastructure, including unit testing, continuous integration (CI), and deployment testing. This also involves establishing testing environments similar to user environments and using diverse application sets for testing. Also mentioned is the creation of a comprehensive test suite and regular code refactoring. Formalization and automation of development practices and procedures are seen as crucial.

2. Portability and Maintenance: Several responses highlighted the importance of making the software compatible with newer languages and architectures, as well as improving portability on future systems. This includes maintenance activities for tools such as repositories, issue tracking, and CI, as well as humans for code reviews, bug fixing, training, and documentation. Maintenance of the project on major systems for testing and support was also mentioned.

3. Documentation and Training: This includes improving and developing user and developer documentation, online short courses for users, tutorials, and other training materials. Some suggested that a software sustainability organization could provide a venue for sharing and highlighting documentation and tutorial development.

4. Community Engagement and User Support: Responses suggested expanding the developer community, using modern community communication tools, and organizing training and outreach events. Efforts should also be made to increase the software's visibility, encourage more registrations, and identify users' potential needs. A suggestion was made to foster collaborations through a software sustainability organization to smoothen the user experience.

5. Integration and Interoperability: This involves improving modularity of various features, integrating the software into larger communities, and providing interfaces to other packages. Better architectural documentation and support for application developers were also considered important.

6. Funding and Support: Dedicated funding for software maintenance, quality assurance, documentation development, porting, and testing was recommended. If sustainability organizations provide personnel, the preference is for the personnel to work with the project team.

7. Forward Planning and Innovation: Respondents mentioned feasibility studies for emerging technology and software engineering practices, as well as the need to innovate to address potential roadblocks such as language sustainability. Early access to upcoming technologies was also viewed as important.

8. Standards and Automation: Some respondents advocated for driving standards and automating more aspects of the developer workflow, like code formatting and updates.

9. Development of Advanced Features: The need for more efficient implementations for advanced hardware support, more fine-grained power monitoring, and support for modern workflows consisting of AI, Big Data, and Data Science workloads were also pointed out

B.5 Which kinds of libraries and tools in the DOE software stack have the greatest potential for broader adoption and use by your customers?

The responses from facility staff suggest several libraries and tools in the DOE software stack that could be beneficial for broader adoption and use by their customers:

1. Spack - a package management system for supercomputers, Linux, and macOS.

2. Workload managers, specifically Flux - a new generation resource management framework.

3. Portability frameworks like Kokkos and Raja - these aim to provide performance portable parallel programming models for manycore and multicore architectures.

4. Performance focused libraries, with Trilinos being mentioned - it is a collection of algorithms and data structures for large-scale scientific computing.

5. Basic Linear Algebra Subprograms (BLAS) - a software library for numerical linear algebra.

6. Solvers - these could refer to mathematical or computational solvers used for solving equations or optimization problems.

7. Scalable math libraries - these help in performing complex mathematical computations across a range of computing architectures.

8. Various tools for visualization, performance tuning, and mesh generation - these are essential components of many scientific computing workflows.

B.6 How does the user base for your package grow over time, and how could a sustainability organization help?

1. The growth of the user base for different packages seems to occur through various methods including collaborations, leveraging existing communities, and word of mouth. Growth is also supported by the needs of different industries and the availability and quality of performance and functionalities of the software.

2. Performance, robustness, usability/portability, and documentation & user support are seen as the four pillars of successful projects, and sustainability organizations could provide guidance and resources in these areas.

3. Trust plays a significant role in the adoption of these software packages. If a sustainability organization supports a product, it could provide implied trust, suggesting that the product is reliable and will be supported in the future.

4. Several respondents highlight the potential of sustainability organizations to promote and raise awareness of the software packages. This could include hosting training sessions, workshops, community engagement activities, or promoting the software in newsletters or at conferences.

5. For some, growth is difficult to quantify as downloads or usage isn't tracked. However, user feedback and surveys can be valuable tools in identifying challenges and potential areas for improvement.

6. A common concern is the need for sustained support and funding. The sustainability of the software is linked to its user base and functionality, and support from sustainability organizations can help ensure this.

7. New feature development and expanding to different fields or industries are mentioned as other methods of increasing the user base. For example, expanding the use of HDF5 for high energy physics (HEP) use cases could significantly increase its user base.

8. Lastly, some mention the concept of organic growth - the incremental expansion of use cases, users, and developers creating a sustainable ecosystem. Support in continued education and expansion, such as through marketing and workshops, is seen as a method that sustainability organizations could use to promote this kind of growth.

B.7 What kind of outreach activities are you doing to grow your user base, and how could a sustainability organization help?

The responses describe a variety of outreach activities being conducted to grow the user base, such as:

1. Tutorials and Workshops: Many respondents mentioned conducting tutorials at workshops and conferences, including specialized events like the Snowbird meetings. Some have suggested the possibility of small workshop-like meetings to bring together Application Development and Software personnel.

2. Annual User Meetings: These are also a common way of engaging with the user base, often coupled with graded quality assurance and engineering service for testing.

3. Community Engagement and Support: Programs like "Call the Doctor", Birds of a Feather (BoF) sessions, and User Group Meetings have been mentioned. In addition, organizations like the ExaHDF5 team organize specific community events such as the HDF5 User Group meetings. They have also released resources such as the h5bench for users to test performance and reuse code.

4. Online Resources: Online tutorials related to new features are provided by some organizations. Other online activities include running a YouTube channel, and some organizations have expressed the need for more regular online training events.

5. Conference Activities: Participation in conferences via presentations, talks, posters, live demos, and hosting tutorials are frequently mentioned outreach methods.

6. Virtual Training and Advocacy: Some organizations provide virtual training and promote their tools within specific networks, such as the Department of Defense (DoD) programs.

7. Developer Collaboration: Some respondents mentioned the importance of working directly with developers, possibly through hackathons and workshops.

In terms of how a sustainability organization could help:

1. Organization and Support: Assistance in organizing and supporting these activities is frequently requested. This includes support for staff and infrastructure, coordinating courses and workshops, and helping with technical aspects like containerization, integration through Jupyter notebooks, advertising, etc.

2. Endorsement and Advocacy: Some organizations could benefit from official endorsements, or advocacy within specific networks and programs.

3. Expanding User Base: A sustainability organization could help identify new groups that could benefit from these tools or assist in diversifying the user base by reaching out to communities like AI, Big Data, and Data Science.

4. Increasing Visibility: There's a desire for sustainability organizations to help increase the visibility of these tools beyond current circles.

5. Development Support: This could involve helping with the addition of functionalities, documentation, and improvement of build systems.

B.8 What level of annual funding is needed for incremental development, delivery, and user support of your product?

The responses indicate a wide range of necessary annual funding for the incremental development, delivery, and user support of various products. Here's a summary:

1. FTE Requirements: Several respondents suggest a specific number of Full-Time Equivalents (FTEs) required for the maintenance and development of their products, ranging from 0.2 to 6 FTEs. This includes roles such as developers, architects, research scientists, and postdocs. The specific division of labor (maintenance, new development, user support) varies.

2. Monetary Requirements: The annual funding needed varies widely, from $300k to $2M per year. Most respondents indicate a need for funding between $500k to $1M per year.

3. Size and Complexity: Several responses suggest that the funding needs depend on the size and complexity of the product. More complex or larger projects, or those that need to keep pace with emerging technologies (e.g., MPI standard evolving, new interconnect technologies), tend to require higher funding.

4. Continuation vs Expansion: Some responses differentiate between maintaining current functionality and expanding or enhancing the product. Continued work in the absence of new hardware or software generally requires less funding, whereas funding for expansion or to avoid a single point of failure needs to be higher.

5. Specific Projects: Certain project names are mentioned, including Kokkos, OpenACC for Clang/Flang, and OpenMP V&V, along with their specific funding needs.

6. Community Support: A few responses highlight that different levels of community support can influence development and user base growth, suggesting that funding may be used to foster this community interaction.

Overall, the necessary funding depends on a range of factors, including the size and complexity of the product, the scope of maintenance and development tasks, and the desired level of user support.

B.9 What barriers to adoption might be addressed, at least in part, by a software sustainability effort?

Staff from various facilities have identified a range of barriers to adoption that might be addressed by a software sustainability effort.

1. Long-Term Viability Concerns: There are reservations about the durability of open-source solutions for industry issues. Open-source projects often depend on community support and development, and their sustainability can be uncertain. A dedicated software sustainability effort can alleviate these concerns by ensuring ongoing support, development, and maintenance of these projects.

2. Adapting to Evolving Hardware: With rapidly changing hardware platforms, software obsolescence is a significant concern. Sustainable software initiatives can help address this issue by ensuring that software remains compatible and efficient with the latest hardware advancements.

3. Software Licensing Issues: Some pointed out the need for "Apache or better" software licenses that permit free commercial use. This barrier suggests a need for clear and open-source friendly licensing, allowing companies to adopt software without concerns over legal ramifications or costs. Software sustainability efforts can help ensure that the licensing of software promotes its widespread use and longevity.

In summary, a software sustainability effort could address concerns about the longevity of open-source solutions, adaptability to evolving hardware, and issues related to software licensing.

B.10 What are the most common impediments to you using more open-source libraries and tools?

The common impediments to using more open-source libraries and tools based on the responses include:

1. Lack of Maintenance and Support: Users are uncertain about the longevity and reliability of these tools. They are concerned about receiving timely bug fixes, support for migrations to new systems, and general assistance.

2. Startup Difficulty: Discovering what tools are available, as well as installing and evaluating them in restricted environments, is challenging.

3. Dependency Management: Managing dependencies and creating portable builds is a significant concern. While tools like Spack might help, they are considered somewhat fragile.

4. Poor Documentation and Testing: Users often find that open-source libraries and tools lack sufficient documentation and testing. Bugs may not be addressed in a timely fashion.

5. Compatibility Issues: There can be lack of support for specific platforms, languages, or other tools, making the library or tool difficult to integrate into existing workflows.

6. Licensing Concerns: Some users have issues with restrictive or nonstandard licenses, preferring libraries that use common open-source licenses like variants of BSD or the LGPL.

7. Refactoring Existing Codebase: Refactoring an existing application codebase to use a library instead of hard-coded solutions is a common issue.

8. Uncertainty About Fit: Users are often uncertain about whether a library's solution will be suitable for their particular problem and how long it will take to determine its suitability.

9. Lack of Stability: There is a concern about the stability of open-source libraries and tools, including the ability to get modifications, improvements, and bug fixes.

10. Lack of Sustaining Community: Users are worried about whether the open-source community behind a library or tool is strong and mature enough to sustain its development.

11. Installation Difficulties: Some libraries and tools can be hard to install, and they may lack an intuitive interface.

12. Unfamiliarity and Fear: Some users are unfamiliar with certain tools and libraries, and there can be apprehension about the learning curve involved in adopting new tools.

13. Awareness: There's difficulty in finding out about the existence of certain libraries or tools, suggesting the need for a central database.

14. Inter-tool Compatibility: Users also mention problems in ensuring that different open-source tools work together seamlessly.

15. Inadequate Test Coverage: Some open-source libraries and tools lack comprehensive test coverage, leading to frequent breakage.

B.11 What activities are you undertaking to grow the contributor base for your tool, and how could a sustainability organization help?

The responses indicate a wide array of strategies being employed to grow contributor bases for various tools:

1. Staff Engagement and Recruitment: Some are engaging internal junior staff, postdocs, or external parties for contributions, and hope that a sustainability organization can help attract more contributors.

2. Community Events: Annual events like "Birds of a Feather" (BoF) type events, hackathons, training events, and weekly team meetings are being utilized to grow the community.

3. Open-source Development: Many respondents have adopted fully open-source development processes and are inviting institutions and users to participate in the core development. They hope a sustainability organization can help establish standards and practices, maintain community contributions, and provide infrastructure for engaging the community.

4. Documentation: Contributors are working on architectural documentation, developer guides, and tutorials, and wish that a sustainability organization could provide support and examples for these documentation efforts.

5. Integration: Some are focusing on integrating their tools with existing production tools, which they find to be a slow process due to resolving corner cases. They hope for funding to expedite integration and development of tools that ease this process.

6. Engineering Support & Quality Assurance: Some respondents express a need for engineering support to increase successful use-cases and quality assurance, hoping for help from sustainability organizations in these areas.

7. Modular Design & Standardization: A few responses mention improving plug-in capability, building SDKs for external contributors, and refactoring frameworks to make the addition of new components less complex. They hope a sustainability organization could help with defining clear interfaces and APIs.

8. Code Organization & Continuous Integration: Some contributors are focused on improving code organization, adding internal documentation, and setting up CI/CD processes.

9. Collaboration with Vendors and User Application Teams: Direct collaboration with vendors, user application teams, and external researchers is seen as a crucial strategy to grow the contributor base.

10. Visibility: Contributors are moving their projects to platforms like GitHub to increase visibility and make it easier for new contributors to participate.

11. Student Involvement: Some are striving to increase collaborative projects with summer students and postdocs, looking to establish pipelines from university programs and offer career paths for developers.

12. Outreach Activities: There's a focus on outreach activities and building infrastructure tools to perform outreach.

Most respondents see the role of a sustainability organization in helping organize and fund these activities, establishing standards and practices, maintaining community contributions, providing infrastructure for community engagement, helping with documentation efforts, defining clear interfaces and APIs, and helping align project needs with student abilities for internships.

B.12 What are the top things a software sustainability organization should (or should not) do to be a successful steward of DOE software?

The responses from facility staff on what a software sustainability organization should do to be a successful steward of DOE software can be summarized into key themes:

1. Funding and Stability: Ensure a steady level of funding to support and sustain software products over the long term. It will help in retaining the talented team members and attracting new ones to participate and contribute.

2. Visibility and Engagement: Increase visibility of software products and carefully choose engagement levels to avoid overstretching resources. Facilitate interactions between different groups developing software and help developers connect with users to understand their needs. Reach out to application developers and users actively.

3. Sustainability and Long-term Planning: Acknowledge the fear of products not being sustained in the HPC community. Address this by explicitly sustaining products and assuring users that products will be maintained in the long run.

4. User Support and Usability: Strive for a balance between user support and technological advancement. Ensure software is robust, has repeatable build and install instructions, and provides a clear description of usage expectations. Prioritize excellent user support in developing research libraries or tools.

5. Documentation and Best Practices: Enforce good documentation practices, CI/CD practices, and circulate best practices across software packages. Advocate for best practices and provide conventions or standards for a uniform experience for application developers.

6. Innovation and Avoiding Monoculture: Encourage innovation and avoid a situation where only a single dominant package/product is supported in any given category. Promote open-source software and ensure flexibility.

7. Adaptation to Technological Changes: Adapt the software to changes in operating systems and GPU vendor software stacks. Improve software reliability, scalability, and performance based on user needs.

8. Outreach and Collaboration: Foster community development, facilitate outreach to grow user base beyond traditional DOE HPC users, and establish mechanisms for meaningful collaborations.

9. Portability and Composability: Ensure portability on future systems (including emerging architectures and programming models/languages), ensure composability with third-party tools.

10. Integrated Approach: The organization should encompass various libraries and tools in an integrated manner so that end users and system administrators can deploy the stack easier.

11. Future Orientation: Have a clear vision of the scientific communities' needs in the next decade and invest in highly performant sustainable software ecosystems.

12. Education and Awareness: Raise awareness of the need for dedicated support and guidelines for best practices. The organization could also drive sustainability and adoption at the university education level.

13. Quality Assurance and Simplified Installation: Provide as much quality assurance as possible and maintain simplified installation methods.

B.13 How can you (facilities) help to broaden the contributor base for DOE software packages?

Facilities staff proposed the following strategies to broaden the contributor base for DOE (Department of Energy) software packages:

1. Partnership with Independent Software Vendors (ISVs) and Industrial Customers: Engaging with these parties, some of whom may contribute to DOE software packages to enhance the quality of their own software.

2. Co-design with the DOE: This involves taking into account the needs of software applications and how they might interact with forthcoming hardware features.

3. Promotion and Confidence Building: Enhancing awareness about the DOE's software packages and building confidence in the artifacts (the produced software and related files).

B.14 How can your facility staff contribute to a software sustainment effort?

The facility staff can contribute to a software sustainment effort in various ways:

1. They act as intermediaries between software vendors and developers, responsible for reporting bugs and following up on their resolution. Their proximity to performance engineering teams allows them to contribute to benchmarking and optimization of performance. They also utilize early access to hardware samples, compiler prototypes, and drivers to conduct initial testing.

2. They provide frontline support and problem management, install software packages, assist with Continuous Integration (CI), create site-specific documentation, and contribute to training.

3. The staff can give feedback on the functionality and performance of the software, reporting any issues that they encounter during their use of the software.

4. Even though direct funding for their participation might be lacking, they can make targeted contributions to the components that they use and that require maintenance or enhancement.

5. The facility staff can also help in facilitating the adoption of new software technologies, given their crucial role in deciding what software can be used on High-Performance Computing (HPC) infrastructure. They can provide best practices for specific programming languages (like Julia), engage with the broader programming language communities to address HPC specific concerns, and collaborate with academic institutions like JuliaLab at MIT to develop and improve the language for specific compute architectures.

B.15 What are ways that you (facilities) can contribute to the ongoing sustainment of these DOE software packages?

Staff from various facilities suggest several methods for contributing to the ongoing sustainment of Department of Energy (DOE) software packages:

1. The creation of a highly configurable and scalable test and validation platform. This will ensure that the software packages are thoroughly tested and validated for optimum performance.

2. Collaboration with code teams to work on porting the software. The focus would be on making the software portable and sustainable to support new hardware features and technologies. This would enhance the software's adaptability and longevity.

Offering in-kind support for troubleshooting and improving the software packages. There's also a potential for paid support. This provides an avenue for technical assistance and continuous improvement of the software packages.