Open Scientific Software Foundation

Gregory R. Watson, Oak Ridge National Laboratory (Principal Investigator)
Addi Malviya-Thakur, Oak Ridge National Laboratory (Co-Investigator)
Elaine M. Raybourn, Sandia National Laboratories (Co-Investigator)
Bill Hoffman, Kitware, Inc. (Co-Investigator)
John Kellerman, Eclipse.org Foundation, Inc. (Co-Investigator)
Clark Roundy, Eclipse.org Foundation, Inc. (Co-Investigator)
Dana Robinson, The HDF Group (Co-Investigator)
Daniel S. Katz, University of Illinois Urbana Champaign (Co-Investigator)

The long-term viability of scientific software has never been more critical than today, especially in light of the huge investment made under the auspices of the Exascale Computing Project (ECP), and the impending conclusion of this project. In addition, there is also a broad community across government, nonprofits, industry, and academia, involved in the development of scientific software who are ready and willing to leverage a software sustainability strategy. A successful governance model will therefore be necessary to secure the Department of Energy's Office of Science (DOE SC) long-term vision of software sustainability.

We believe that it will be imperative for any successful strategy for scientific software sustainability to encompass this whole community. Our proposal centers around the creation of a unique partnership between the stakeholders in the form of a self-sustaining not-for-profit open-source organization that will facilitate the long-term stewardship of scientific software. Key to this approach is the fundamental importance of ensuring that the environment that supports the software is sustainable. This means creating a community that is invested, works toward common goals, shares important ideas, and provides common services and infrastructure that benefit the whole community. A community working this way can result in something that is greater than the sum of the individual parts.

Our assertion is that the best way to achieve this is a foundation model, where a not-for-profit organization acts as an open and transparent facilitator allowing a collection of teams, groups, and organizations to advance scientific software through the shared ecosystem. This type of model has proved highly successful for open-source software, where foundations espouse neutrality, open participation, meritocracy, collaboration, and are self-maintaining. They have clearly demonstrated that sustainable funding models are not only achievable, but are fundamental to ensuring that the open source software ecosystem continues to proliferate.

Contents

Background/Introduction

The long-term viability of scientific software, in particular software that is used for computational science and engineering (CSE) and to support experimental and observational science (EOS), has never been more critical than today, especially in light of the huge investment made under the auspices of the Exascale Computing Project (ECP), and the imminent need to ensure that this software continues to be maintained and improved for the foreseeable future. In addition, there is also a broad community across government, nonprofits, industry, and academia, involved in the development of scientific software who are ready and willing to leverage a software sustainability strategy – a successful governance model is necessary to secure the Department of Energy's Office of Science (DOE SC) long-term vision of software sustainability.

We believe that it will be imperative for any successful strategy for scientific software sustainability to encompass this whole community. Our proposal centers around the creation of a unique partnership between the stakeholders in the form of a self-sustaining not-for-profit open-source organization that will facilitate the long-term stewardship of scientific software. Key to this approach is the fundamental importance of ensuring that the environment that supports the software is sustainable. If viewed as a diverse ecosystem, the objective is not to ensure that any specific piece of software or software project survives, but that the ecosystem as a whole thrives, provides value to its users, innovates, and inspires confidence in stakeholders about its stability and sustainability over time. This means creating a community that is invested, works toward common goals, shares important ideas, and provides common services and infrastructure that benefit the whole community. A community working this way can result in something that is greater than the sum of the individual parts.

Our assertion is that the best way to achieve this is a *foundation model*, where a not-for-profit organization acts as an open and transparent facilitator allowing a collection of teams, groups, and organizations to advance scientific software through the shared ecosystem. This type of model has proved highly successful. For example, the two-decade-old Linux Foundation supports over 200 communities and 3,400 project source code repositories. The Apache Software Foundation, established in 1999, has grown into an ecosystem with over 300 projects. The Eclipse Foundation, which was established in 2004, now supports over 400 projects and 20 working groups and hosts around 300M lines of code. The more recent NumFOCUS, Inc. established in 2012, has 45 sponsored and 51 affiliated projects. These foundations and the projects they include espouse neutrality, open participation, meritocracy, collaboration, and are self-maintaining. They have clearly demonstrated that sustainable funding models are not only achievable, but are fundamental to ensuring that the open source software ecosystem continues to proliferate.

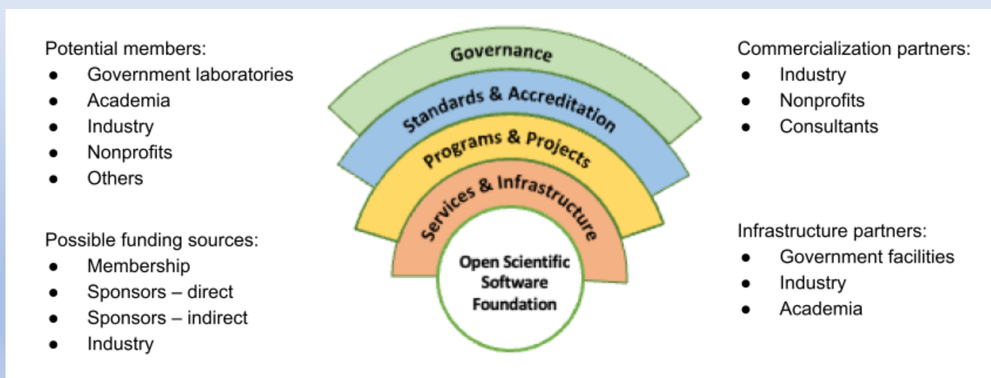
We have assembled a highly skilled team to provide a steering committee that will be responsible for all aspects of establishing an open scientific software foundation based on such a model. Our team brings together a unique blend of experience from national laboratories, nonprofits, academia, and industry that will be necessary to achieve such an outcome. This team represents a broad segment of the software community and brings significant experience in the development of highly successful partnerships of this nature. The following table summarizes the skills each team member brings to the project.

Gregory Watson	Corporate governance, public/private partnerships, foundation establishment, and operation as past Chair of .au Domain Administration and past President of the Internet Society of Australia; 25 years computer science research & development; teaching & education; management
Addi Malviya-Thakur	Software engineering research & development; software ecosystem research; technical leadership
Daniel S. Katz	Software sustainability; community leader; sustainability research
Elaine Raybourn	Social scientist applying diffusion of innovations to incentivizing software productivity and

	software quality. Training & learning, team science, teams of teams, virtual communities, new media, communication science, early career mentorship, diversity, equity, and inclusion, inter-agency collaboration
Dana Robinson	Software engineering sustainability & development, physical and biological sciences research and community engagement, open-source software development, leadership, and community management, non-profit management and operations
Bill Hoffman	25 years of experience supporting, developing, community building, and promoting Open Source scientific software at Kitware. Founder of the defacto standard C++ build tool CMake. Experience with creating sustainable software processes that have survived the test of time with projects like CMake, VTK, ITK, Slicer, ParaView and others that are thriving after 20+ years of development.
John Kellerman	Open source development and community building including industry collaboration, foundation governance, intellectual property management, standards compliance.
Clark Roundy	Open source community building; software product/project/program management; brand building and product marketing; HPC/scientific computing industry background

Project Objectives

Our vision is to create an open-source scientific software foundation that embraces the principles of openness through governance, standards, affiliated programs and projects, and supporting services and infrastructure.



Upon award, our plan is to hit the ground running with a gap analysis which explores a mix of various funding, organizational and governance models that accelerate scalability, work towards the ongoing development and support of software projects, as well as support the growth and sustainability of a community ecosystem around the foundation activities. We envision creating a charter for this foundation that will encourage membership from government laboratories, academia, industry, other not-for-profits, as well as similar foundations. We will explore multifaceted collaboration, and commercial and non-commercial partnership opportunities by engaging with other not-for-profits, foundations, industry, and commercial consultants. By leveraging our close partnerships with government laboratories, industry, practitioners, and academia, we will accelerate the formation of new alliances with organizations, centers, and institutes involved in similar activities to provide expertise, support for and adoption of open-source scientific software development, and leadership-class facilities for infrastructure services. We will then create a business and operational plan that will provide key details on the parameters necessary for bootstrapping the organizations and how the business operations will begin as soon as these conditions have been met.

In the scientific software ecosystem, there is a complex interplay between the variety and nature of the software, the roles of software developers and domain scientists, the availability of funding sources, publications and other sources of recognition, and the quality and efficacy of the software itself. Scientific software may be developed indirectly as a result of research, directly to fulfill a particular scientific purpose, or for commercial or other reasons [1]. This complexity results in some distinctive characteristics of the scientific software ecosystem that are not present in other open-source ecosystems. These include the close connection with government and government sponsored laboratories, software that are domain specific and specialized to cater to high impact science, the huge variation in scaling needs, and the ability to run on custom hardware such as supercomputers. These unique aspects must be considered when creating a governance structure and business plan that will ultimately lead to a self-funding, and hence self-sustaining organization. There is also an urgent need to address the sustainability needs of the large software base that is currently being developed by the ECP. Our approach will be to leverage the ECP software as a core component of the foundation portfolio by providing a transitional path for funding and infrastructure from the current ECP structure into the new organization.

A foundation model provides a wide array of benefits that can address the many of the sustainability needs of the scientific software community. In addition to securing stable funding through partnerships, grants, sponsorship, and other sources, there are opportunities for the creation of policies relating to coding standards, quality, security, and reproducibility that are required to ensure software is of high quality and maintainable in the long term. These policies could build on existing approaches used by the ECP Extreme-scale Scientific Software Stack (E4S) Community Policies, the Linux Foundation, the development process at Apache and Eclipse, the Association for Computing Machines (ACM) Artifact Review and Badging, the Software Sustainability Institute (SSI), Better Scientific Software (BSSw), and other organizations to provide incentives and quality metrics through tiered structures and badging. Other benefits could also extend to the accreditation of training and consulting services that could be offered either by the foundation, by partnering with organizations such as The Carpentries, CodeRefinery, Universe-HPC, INTERSECT, or via commercial offerings. Services, such as legal and intellectual property, Digital Object Identifier (DOI) minting, archival storage, and FAIR data could also be explored. The foundation will be expected to maintain essential infrastructure to support software development, continuous integration, deployment, and other operations. We also see it responsible for ensuring the integrity of software supply-chain security – an area that has not been a focus for scientific software to date but has critical implications for the security of national infrastructure. Ultimately, we see the *Open Scientific Software Foundation* as a singular solution of robust partnerships forming a coalition of the willing, that can ensure a diverse, inclusive, and equitable community of stakeholders that will result in a thriving and self-sustaining ecosystem for all participants.

Proposed Approach

The primary goal of this effort is to create an organizational structure with strong stakeholder and community support, along with a detailed funding strategy and implementation plan, so that it is possible to transition to an operational mode in as short a space of time as possible when the appropriate conditions are met. To achieve this, we plan to undertake six core activities: realize a foundation structure; determine the sustainability objectives and create a strategy for meeting them; engage with stakeholders; engage with sponsors; create a transition plan for key projects; and establish a plan for standing up the organization. In the following sections, we discuss each of these aspects in more detail.

Realizing the foundation structure

There are many potential organizational models that could be adopted for this effort, so the purpose of this activity will be to determine which of these models will bring the most benefits to this community and be most likely to accelerate a successful outcome. However, since we have the opportunity to build a model from scratch, we plan to utilize our experience in this area to ensure that the best aspects of existing models are adopted where appropriate.

Before undertaking any evaluation, our plan is to address a number of fundamental questions about the proposed organization. We intend to deliver a report that answers the following questions:

1. What are the guiding principles for the foundation?
2. What is the vision and mission of the organization?
3. What are the key objectives the foundation must accomplish, and how will it accomplish them?
4. Who are the stakeholders and what are their needs?
5. How will stakeholders be involved in the organization?
6. What are the known and potential funding sources?
7. Are there threats to this approach?
8. What could existing foundations have done differently to scale faster?
9. What practices, technologies, or policies are in place in existing foundations that we might leverage, as well as those they would change if they were to start their foundations in 2023?
10. What are the risks to success?

Once these questions have been answered, we will undertake an analysis to determine what mix of various funding, organizational and governance models will be most effective in accelerating scalability, working towards the ongoing development and support of projects, as well as supporting the growth and sustainability of a community ecosystem around the foundation activities. The outcome of this analysis will be a proposed foundation structure that will be used to underpin subsequent planning activities.

Figure 2 illustrates one such possible organizational structure for the foundation. In this example, the organization comprises two main aspects: management and meritocracy. The management aspect of the organization is responsible for the provision of services, day-to-day operation, financial and corporate governance, implementation of policies including diversity, equity, and inclusion, provision and operation of infrastructure, interaction with external partners and commercial services, public relations, and

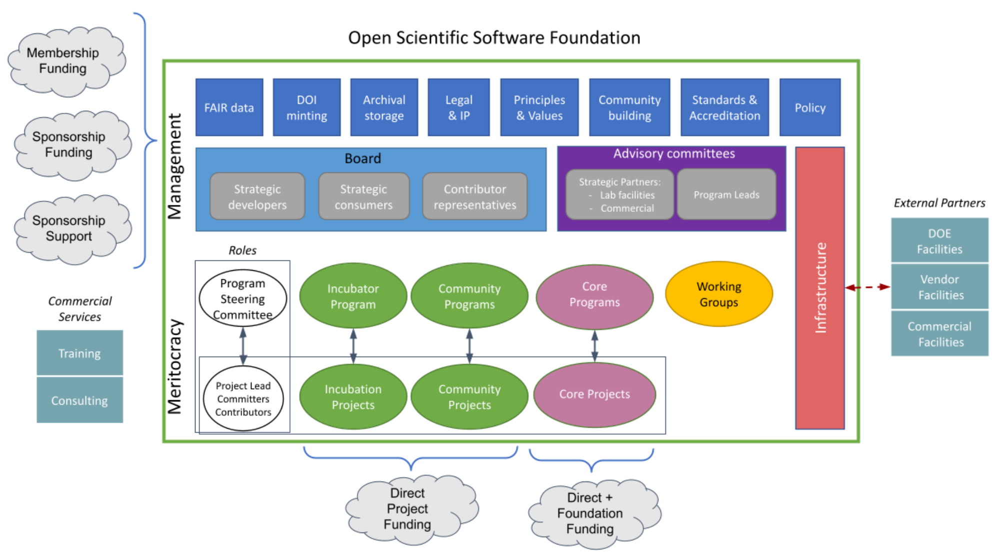


Figure 2: Potential Foundation Structure

developing funding opportunities. The meritocratic aspect of the organization is responsible for developing policies, creating and overseeing programs, creating and overseeing projects, liaison with working groups, and ensuring projects meet required policies. We envision the meritocratic model fostering "teams of teams" among members and contributors through the alignment of goals, use of liaisons, and transparent communication. Structuring the organization in this way ensures a clean separation between funding and operation of the organization, and leadership and operation of working groups, program development, and technical development of the projects themselves.

A key decision will be whether to create a new organization from scratch, or leverage an existing foundation that matches the characteristics we have identified. As broad community support and enthusiastic sponsor engagement is critical to sustaining an ecosystem, any proposed entity (new or existing) will need to be perceived as being neutral and unbiased toward all stakeholders. Our experience with other foundations has shown that this can be a significant factor in excluding sections of the community and impacting on the success of such an endeavor. Part of our analysis will therefore include market research to establish stakeholder impressions of proposed organizational models.

Determine the sustainability objectives

For software in use today to be sustainable, there must be a reasonable expectation that it will continue to be supported and improved, and be available in the future. Research has shown that open-source projects typically transition through four distinct stages: introduction, growth, maturity, and decline/revive [2]. This model is shown in Figure 3. The introduction stage is typically when one or more developers encounter a gap or need, and then self-organize to produce an initial version. The growth stage occurs as more users become aware of the project and start providing feedback in the form of feature requests, bug reports, etc. Many of these users are often also developers, and will contribute code to resolve issues, add features, etc. Often the increased size of the project requires a more formalized structure in order to coordinate the efforts of the community better. Once the project starts to reach a critical size, it enters the maturity stage, where the central focus is on maintenance and sustaining the project. As users lose interest or competitive projects provide better solutions, the project will enter the declining stage where only a small group of developers remain, and the primary focus is on support of existing functionality. Sometimes the project community will revive in response to an external trigger, in which case the project may enter a new growth stage.

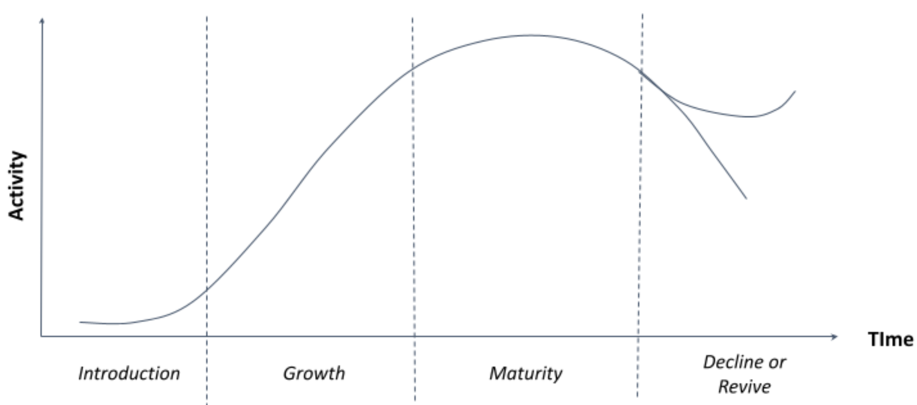


Figure 3: Open-Source Software Lifecycle

Knowing this is the case, it is essential to understand what the objectives of a sustainability organization are. For example, is it to provide common infrastructure and support services to projects in order to prolong their growth/maturity stages? Is it to encourage projects to enter the introduction stage to encourage a thriving ecosystem of mutually supportive projects? Is it to provide additional resources or other triggers to kick declining projects back into a new growth stage? For scientific software, other factors are at play here too. Reproducibility in the scientific method relies on the ability to recreate the results of previous experiments. Computational scientists and engineers rely on the software being available and viable for this to happen. Some funding organizations also have an expectation that the software created as a result of their investment will be around for a considerable amount of time. These are additional factors that will drive the sustainability objectives of the organization [3].

To better understand the motivations of software projects to achieve sustainability and how they could leverage an organization such as this, it is informative to look at some case studies. In the first case study, a grant funded project is in the process of transitioning to a sustainable model via an existing foundation with similar principles to the one being proposed here. In the second, a commercial organization has been maintaining a widely used tool through a variety of commercialization and government funding sources and would derive a number of benefits from the foundation. In the third, a project has been intently focused on a particular community (the primary funding source) but must now expand its reach in order to diversify this funding base.

Case Study 1: Parsl

Parsl is a system co-led by co-PI Katz that allows Python functions (including wrappers for external applications) to be run asynchronously, remotely (e.g., on HPC, cloud), and potentially in parallel, depending on data dependencies determined automatically at runtime [4]. Parsl's initial development was funded over 5 years by the National Science Foundation (NSF). At the end of the initial NSF funding, Parsl was being used by a diverse community, including individual researchers, large research consortia, and in industry, spanning domains such as astrophysics, biology, materials science, and many others.

The challenge for Parsl is that, in spite of its growth and impact, it is not currently sustainable. As usage grows, so too does the work of the core Parsl team. While Parsl is supported by a combination of external funding from projects and collaborations, volunteer efforts, and commercial organizations, these do not provide sufficient resources for the core team. Provided and volunteer resources are useful for adding features to Parsl, and support a limited number of use cases, but are not sufficiently coordinated or aligned to fully support Parsl's core needs over multiple years. The funded Parsl team is responsible for managing the community, reviewing code contributions, fixing bugs, supporting users, developing new features, releasing new versions of the software, among other activities.

To address these issues, the Parsl team has recently won two new awards to transform Parsl into a sustainable project over two years. This includes a community-centered approach to quantify and reduce the costs of core project activities (reduce technical debt), and to identify resources to support these activities into the future. This work will follow the successful sustainability model of Astropy [5] and yt. It will work on community engagement, governance (as a NumFOCUS fiscally-sponsored project with an Astropy-like governance charter and roles), funding streams (balancing in-kind contributions, collaborative partnerships, and financial support, with funding going to NumFOCUS, not the current team's institutions, so that these funds can be directed to the places where work can best be performed without being biased by institutional conflicts), innovation (ensuring that new CS research is supported by separate research awards to the Parsl team or to others), training (including via leveraging computing centers to publish center-specific resources describing use of Parsl), and outreach/engagement.

Case Study 2: CMake

From the inception of compiled programming languages, building software has been recognized as an important and challenging process. The makefile (and associated Unix make utility) was invented to control the construction of program executables. Subsequently other make-related tools such as autoconf have been developed to manage the build process. The open-source, cross-platform CMake family of tools designed to build, test, and package software is emerging as a dominant tool with 55% of C++ projects now using CMake, and large vendors such as Microsoft embracing it within their development environment [6]. CMake has become popular for a multitude of reasons including cross-platform support, speed, and integration with multiple IDEs/compiler environments.

The CMake project was initially funded by National Institute of Health (NIH) as part of the creation of the Insight Toolkit [7] in the early 2000s. It has since grown to be the de facto standard for building C++ projects and is the preferred build tool for ECP projects. Kitware has kept the CMake project alive by finding different commercial and government funding sources over the past 22 years, based on philanthropically wanting to ensure the continuity of this tool. This has required considerable ingenuity and effort on the part of Kitware. The scientific computing community's needs for CMake differ from much of CMake's user base, however. Although the standard supported platforms serve the needs of much of the community, HPC requires specialized compilers and environments that require additional resources to develop and maintain. A foundation like the one envisioned in this proposal could distribute the necessary work across more organizations to provide a much more stable and sustainable approach to keeping critical software infrastructure like CMake moving forward and addressing the specific needs of the scientific computing community. This foundation could raise and then provide funding that would allow for CMake to continue to work on the latest HPC compilers and tools, as well as act as a central resource for CMake training and support that could be used by projects to best leverage this build technology. The foundation could provide recommended testing and standardization for projects using CMake which would in turn improve the usability and reproducibility of other software that is part of the foundation.

Case Study 3: VTK-m

VTK-m is a toolkit of scientific visualization algorithms for emerging processor architectures. VTK-m supports the fine-grained concurrency for data analysis and visualization algorithms required to drive extreme scale computing by providing abstract models for data and execution that can be applied to a variety of algorithms across many different processor architectures. The current visualization workhorses for the scientific community such as ParaView and VisIt have traditionally targeted CPU architectures and support coarse grained parallelism through shared memory and distributed memory programming models. As such, this approach cannot fully leverage the performance of new generation architectures that heavily rely on fine grained parallelism of GPUs. VTK-m aims to close this gap.

The VTK-m project started in 2014 when the three DOE-funded efforts to develop fine grained parallel visualization toolkits, Dax [8], Piston [9] and EAVL [6], joined efforts. The project continued under DOE research funding until being replaced by funding through ECP. VTK-m adopted many of the software processes necessary for a sustainable community and ecosystem, including a cross-platform build system based on CMake, issue tracking using GitLab, support for continuous integration testing through Kitware's and ECP Gitlab-CI, and extensive documentation. Since 2017, the VTK-m team has focused solely on the needs of ECP as its main funding source, however this limited funding base is now becoming the major sustainability issue. This is particularly problematic given ECP's extensive needs, and the extreme challenge of supporting multiple GPU architectures. When the ECP project ramps down, it is unlikely that the VTK-m team will be unable to sustain continued maintenance and development.

VTK-m's short-term needs for sustainability will be resources (mainly funding) for maintenance of the code repository and testing infrastructure, core development to continue to address newly discovered issues and to support changes in architecture, and management of community contributions through review of merge requests etc. In the longer term, VTK-m needs to diversify its funding sources. An open scientific software foundation such as the model proposed here, would help VTK-m to enhance the ease of use and accessibility of VTK-m in order to create a broader user and developer community. This would increase the value of VTK-m to a larger community and extend access to a larger group of stakeholders. A foundation could then provide the resources and impetus to engage with a larger group of stakeholders (i.e. funding sources) including the industry and other government agencies both within and without the US.

Engage with stakeholders

Engagement with stakeholders is a crucial part of establishing a software foundation, not only for understanding stakeholder needs and requirements, but also for sharing information about the foundation with others on a regular basis to garner support, buy-in, and adoption. We will collect stakeholder input and requirements, such as current pain points, funding shortfalls, infrastructure and service requirements, training and educational needs, etc. using appropriate forums. These forums will also provide valuable opportunities for open discussion on the sustainability objectives and strategies for meeting them. In addition, we plan to develop a report as part of this process that would delineate the strategies for stakeholder engagement.

We also intend to begin the process of convening working groups of stakeholders from common disciplines in order to begin the process of building communities of interest. These will be communities that are building likeminded software or have common interests in working together towards a common goal. Our plan is that successfully established working groups will eventually transition into operational programs within the foundation. In the establishment phase of the working groups, there will invariably be a number of existing software projects that would form the initial set of projects once the transition to the foundation model has been effected. Foundation programs are one place that lifecycle management for projects could take place. In this scenario, programs would own the assessment and monitoring of project health and vitality, and could utilize a range of metrics to determine project lifecycle stages and corresponding funding levels. Examples of the types of metrics that could be used for this purpose include the software sustainability matrix [3], criticality score, OpenSSF Scorecard, and others.

Engage with sponsors

The sustainability of the foundation will require finding and executing new funding opportunities, in addition to maintaining existing funding sources. This can be done by understanding the sponsors' priorities and future growth opportunities. It is important to find a balance where the sponsor is actively involved in the success of the foundation, promotes the funded project within the foundation, helps remove any roadblocks, and gives the foundation more visibility as a whole. To make sure they are involved in the important decisions and to show the impact it can have in the context of a larger foundation portfolio, it is important to define a shared understanding and the purpose of the engagement. This also includes every funded project having its own structure, and roles and responsibilities being clearly defined and agreed upon and within the sponsors' expectations. This engagement will require meetings, workshops, and visits at a regular cadence to keep sponsors informed and involved throughout the course of the funded projects. Sponsors will be proactively informed about impending activities and deadlines, and thereafter progress will be monitored and shared widely with the sponsors for transparency purposes. A tiered model for engagement where project managers will be established, with the expectation that foundation personnel will engage with sponsors at various levels of decision-making and information exchange and use different communication methods to offer flexibility and reinforce focus on the funded efforts.

Create transition plans

Our expectation is that, at a minimum, there will be a number of ECP projects that will need to start transitioning into the new foundation structure on day one. There may also be other non-ECP projects that wish to do the same. We intend to create a series of transition plans that deal with the individual requirements for each of these. In the case of the ECP projects, this transition plan will need to take into account the resourcing needs of the projects, both in terms of personnel and infrastructure, as well as the current development process, management structure, and funding sources. The plan will also need to determine the development requirements for each project (e.g. maintenance-only, new development, etc.) along with existing infrastructure and any specialized infrastructure needs. Transition plans will detail how these requirements will be supported and the timeframe for the transition. Each project is anticipated to have a sustenance profile that would include the required funding and essential resources needed to carry out the work and innovation.

Create a business plan

The content of the foundation business plan will the necessary detail for starting and operating the organization, both from a management perspective, and to begin transitioning projects into their new home.

It will provide a strategy and implementation plan along with a financial plan containing revenue and expense projections, both for bootstrapping and steady state. The plan will also provide three key policy documents: a Development Process, an Intellectual Property Policy, and a Committers Agreement.

Financial & Implementation Plans

The financial plan will outline an investment budget, a financial budget, an operating budget, and a cash flow budget. This will include preliminary sponsorship deals backed by a memorandum of understanding for each of the main sponsors.

The implementation plan will provide a blueprint and run sheet that describes commencement of business operations, infrastructure deployment, and community engagement and outreach for fast startup and scaling. This plan will be executed once subsequent funds become available for bootstrapping and startup costs. It is expected that this plan will include such topics as company formation, domain name registration, establishment of web and development infrastructure, and initial outreach activities in the form of advertising/marketing to stakeholder communities.

Key Policy Documents

Development Process. The development process establishes rules and procedures that ensure openness, transparency and meritocracy. Generally, it meets the needs of three communities: developers, adopters, and users. The process covers at a minimum, rules for committers, code and releases, intellectual property (IP) records, and community awareness. The development process also establishes a technical hierarchy for who can make what kind of decisions. It outlines the roles and responsibilities of contributors (those making suggested improvements to a project) and the committers (those with the earned authority to actually make changes to project source code). The development process may also establish a lifecycle for projects. The foundation may define a preferred development process, but will accept others as well in recognition of the diversity of existing practices in today's scientific open-source software.

IP Policy. The IP policy generally does two things. It establishes processes and procedures, including reviews and record keeping requirements, designed to assure the integrity of the source code in the repositories under the control of the foundation. It also establishes the licensing constraints under which the open-source projects are offered and the acceptable licenses of their dependencies. This policy is a byproduct of the foundation vision and mission and can be influenced by the business plan.

Committers Agreement. The committers agreement must be signed by anyone in order to get write access to a project's source code repositories. It is an assertion that all changes the signee makes they have the right to commit to the repository. An example of a committers agreement that is being actively used is the Eclipse Foundation Committer's Agreement (ECA). This agreement was derived from an agreement that originated with the Linux Foundation.

Milestones and Deliverables

Milestone 1: Realize the foundation structure

By month 2, the team will complete the following:

- Identify and describe the key organizational objectives that are relevant to the foundation structure. Develop a set of guidelines, scope, and objectives for the foundation
- Develop a report describing the foundation key structure and its core principles

Milestone 2: Determine the sustainability objectives for the foundation

By month 4, the team will complete the following:

- Identify and describe the key sustainability objectives that are relevant to the foundation. Organize lesioning sessions, workshops, and seminars to identify broader working knowledge of the foundation and develop its key milestones and objectives.
- Develop a sustainability report and prepared to act on information learned from the reporting

Milestone 3: Stakeholders engagement

By month 6, the team will complete the following:

- Identify key stakeholders based on the foundation immediate to long-term strategies
- Develop an engagement strategy and act on it

Milestone 4: Sponsor engagement

By month 8, the team will complete the following:

- Identify key sponsors based on the foundation immediate to long-term strategies
- Develop an engagement strategy and act on it

Milestone 5: Transition plans

By month 10, the team will complete the following:

- Identify key aspects of the foundation that are relevant for its operation
- Prioritize projects to transition based on stakeholder input
- Develop step by step plans to perform the transitioning, identify key patterns

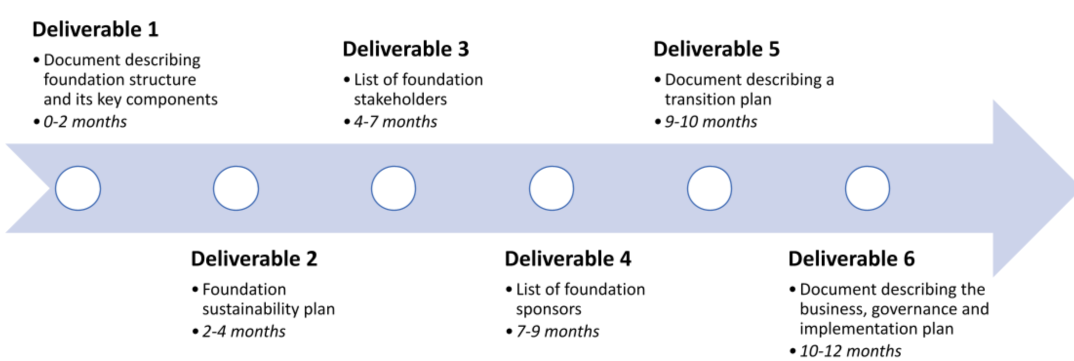
Milestone 6: Plans for standing up the foundation

By month 12, the team will complete the following:

- Define and codify the standards and governing rule for the foundation
- Develop a business plan
- Develop a plan for the practice of diversity, equity, and inclusion
- Develop a plan for establishing the foundation

Deliverables

The tangible output of the proposal correlates with the milestone described earlier. The main goals of this proposal are a step toward creating a successful Open Scientific Software Foundation. The foundation's governance and self-sustenance will be inspired by existing foundations such as Apache and the Eclipse Foundation. Furthermore, a set of deliverables that includes a documented list of stakeholders and sponsors of this foundation, as well as key aspects of funding, will allow the foundation to sustain itself in the foreseeable future. In the figure below, we have depicted the key deliverables that correspond to the milestones, as well as the estimated time for delivery in relation to the milestones.



APPENDIX 1: BIBLIOGRAPHY & REFERENCES CITED

[1] J. Cohen, D. S. Katz, M. Barker, N. Chue Hong, R. Haines and C. Jay, "The Four Pillars of Research Software Engineering," *IEEE Software,* vol. 38, no. 1, pp. 97-105, 2021.

[2] D. E. Wynn, "Organizational structure of open source projects: a life cycle approach," in *Proceedings of 7th Annual Conference of the Southern Association for Information Systems*, 2004.

[3] W. Schroeder, M. Grauer, M. McCormick, M. D. Hanwell and J.-C. Fillion-Robi, "How Sustainable is Your Software?," Jan 2020. [Online]. Available: https://www.kitware.com/how-sustainable-is-your-software/.

[4] Y. Babuji, A. Woodard, Z. Li, D. S. Katz, B. Clifford, R. Kumar, L. Lacinski, R. Chard, J. M. Wozniak, I. Foster, R. Wilde and K. Chard, "Parsl: Pervasive Parallel Programming in Python," in *Proceedings of the 28th International Symposium on High-Performance Parallel and Distributed Computing (HPDC '19)*, New York, NY, USA, 2019.

[5] The Astropy Collaboration et. al., "The Astropy Project: Sustaining and Growing a Community-oriented Open-source Project and the Latest Major Release (v5.0) of the Core Package," *ApJ,* vol. 935, no. 167, 2022.

[6] J. S. Meredith, J. Ahern, S. Ahern and R. Sisneros, "EAVL: The extreme-scale analysis and visualization library," in *Eurographics Symposium on Parallel Graphics and Visualization (EGPGV)*, 2012.

[7] T. S. Yoo, "The Insight Toolkit: An Open-Source Initiative in Data Segmentation and Registration," in *The Visualization Handbook*, Amsterdam, Elsevier, 2005, pp. 733-48.

[8] K. Moreland, U. Ayachit, K. Geveci and K. Ma, "Dax Toolkit: A proposed framework for data analysis and visualization at Extreme Scale," in *1st IEEE Symposium on Large-Scale Data Analysis and Visualization 2011*, 2011.

[9] L. Lo, C. Sewell and J. Ahrens, "PISTON: A portable cross-platform framework for data-parallel visualization operators," in *Eurographics Symposium on Parallel Graphics and Visualization*, 2012.