

## FOURTH WORKSHOP ON SUSTAINABLE SOFTWARE FOR SCIENCE: PRACTICE AND EXPERIENCES (WSSSPE4)

DANIEL S. KATZ<sup>(1)</sup>, KYLE E. NIEMEYER<sup>(2)</sup>, SANDRA GESING<sup>(3)</sup>, LORRAINE HWANG<sup>(4)</sup>,  
WOLFGANG BANGERTH<sup>(5)</sup>, SIMON HETTRICK<sup>(6)</sup>, RAY IDASZAK<sup>(7)</sup>, JEAN SALAC<sup>(8)</sup>,  
NEIL CHUE HONG<sup>(9)</sup>, SANTIAGO NÚÑEZ-CORRALES<sup>(10)</sup>, ALICE ALLEN<sup>(11)</sup>, R. STUART GEIGER<sup>(12)</sup>,  
JONAH MILLER<sup>(13)</sup>, EMILY CHEN<sup>(14)</sup>, ANSHU DUBEY<sup>(15)</sup>, PATRICIA LAGO<sup>(16)</sup>

**ABSTRACT.** This article summarizes motivations, organization, and activities of the Fourth Workshop on Sustainable Software for Science: Practice and Experiences (WSSSPE4). The WSSSPE series promotes sustainable research software by positively impacting principles and best practices, careers, learning, and credit. This article includes the mission and vision statements that were drafted at the workshop and finalized shortly after it; the keynote and idea papers, position papers, experience papers, demos, and lightning talks presented during the workshop; and a panel discussion on best practices. The main part of the article discusses the set of working groups that formed during the meeting, along with contact information for readers who may want to join a group. Finally, it discusses a survey of the workshop attendees.

---

<sup>(1)</sup> National Center for Supercomputing Applications (NCSA) & Department of Computer Science & Department of Electrical and Computer Engineering & School of Information Sciences (iSchool), University of Illinois, Urbana-Champaign, IL, USA; d.katz@ieee.org; ORCID: 0000-0001-5934-7525.

<sup>(2)</sup> School of Mechanical, Industrial, and Manufacturing Engineering, Oregon State University, Corvallis, OR, USA; kyle.niemeyer@oregonstate.edu; ORCID: 0000-0003-4425-7097.

<sup>(3)</sup> Center for Research Computing & Department of Computer Science and Engineering, University of Notre Dame, IN, USA; sandra.gesing@nd.edu; ORCID: 0000-0002-6051-0673.

<sup>(4)</sup> University of California, Davis, CA, USA; ljhwan@ucdavis.edu; ORCID: 0000-0002-1021-3101.

<sup>(5)</sup> Department of Mathematics, Colorado State University, Fort Collins, CO, USA; bangerth@colostate.edu; ORCID: 0000-0003-2311-9402.

<sup>(6)</sup> The Software Sustainability Institute, Southampton, UK; s.hettrick@software.ac.uk; ORCID: 0000-0002-6809-5195.

<sup>(7)</sup> RENCI, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA; rayi@renci.org; ORCID: 0000-0002-3444-7615.

<sup>(8)</sup> University of Virginia, Charlottesville, VA, USA; jeansalac@virginia.edu; ORCID: 0000-0001-5971-9333.

<sup>(9)</sup> University of Edinburgh, UK; N.ChueHong@epcc.ed.ac.uk; ORCID: 0000-0002-8876-7606.

<sup>(10)</sup> Illinois Informatics Institute (I3) and National Center for Supercomputing Applications (NCSA), University of Illinois at Urbana-Champaign, IL, USA; nunezco2@illinois.edu; ORCID: 0000-0003-4342-6223.

<sup>(11)</sup> Astrophysics Source Code Library, Astronomy Department, University of Maryland, College Park, MD, USA; aallen@ascl.net; ORCID: 0000-0003-3477-2845.

<sup>(12)</sup> University of California, Berkeley, Berkeley Institute for Data Science (BIDS), Berkeley, CA, USA; stuart@stuartgeiger.com; ORCID: 0000-0001-7215-0532.

<sup>(13)</sup> Perimeter Institute for Theoretical Physics & Department of Physics, University of Guelph, Guelph, ON, Canada; jmiller@perimeterinstitute.ca; ORCID: 0000-0001-6432-7860.

<sup>(14)</sup> University of Illinois at Urbana-Champaign, USA; echen35@illinois.edu; ORCID: 0000-0003-2578-0296.

<sup>(15)</sup> Mathematics and Computer Science Division, Argonne National Laboratory, Lemont, IL, USA & University of Chicago, Chicago, IL, USA; Adubey@anl.gov; ORCID: 0000-0003-3299-7426.

<sup>(16)</sup> Vrije Universiteit Amsterdam, The Netherlands; p.lago@vu.nl; ORCID: 0000-0002-2234-0845.

## 1. INTRODUCTION

The Fourth Workshop on Sustainable Software for Science: Practice and Experiences (WSSSPE4)<sup>1</sup> was held over 2 1/2 days on 12–14 September 2016 in Manchester, England. This location and date was selected so that WSSSPE4 immediately preceded the First Research Software Engineers (RSE) Conference. Previous events in the WSSSPE series include WSSSPE1<sup>2</sup> [1, 2], held in conjunction with the SC13 conference; WSSSPE1.1<sup>3</sup>, a focused workshop organized jointly with the SciPy conference<sup>4</sup>; WSSSPE2<sup>5</sup> [3, 4], held in conjunction with SC14; WSSSPE2.1<sup>6</sup>, a focused workshop organized again jointly with SciPy<sup>7</sup>; and WSSSPE3<sup>8</sup> [5], held in Boulder, Colorado, USA. The first WSSSPE workshop was named “Working towards Sustainable Software for Science: Practice and Experiences,” which remains the meaning of the WSSSPE group, but the workshops after that were named “Workshop on Sustainable Software for Science: Practice and Experiences.” Together these reflect that WSSSPE is both a community and a set of workshops.

The WSSSPE4 workshop included multiple mechanisms for participation and encouraged team building around solutions. WSSSPE4 strongly encouraged participation of early-career scientists, postdoctoral researchers, graduate students, early-stage researchers, and those from underrepresented groups, with funds provided to the conference organizers by the National Science Foundation (NSF), the Gordon and Betty Moore Foundation, the Alfred P. Sloan Foundation, and the Software Sustainability Institute (SSI) to support the travel of potential participants who would not otherwise be able to attend the workshop. These funds allowed 29 additional people to attend and participate. A subset of the organizing committee reviewed 44 applications for travel support and competitively selected the awardees, including 11 students and five early-career researchers. In addition, the travel award subcommittee tried to increase the diversity of applicants by forwarding the notice of travel support to organizations including PyLadies<sup>9</sup>, Django Girls<sup>10</sup>, Women Who Code<sup>11</sup>, and Women in HPC<sup>12</sup>. WSSSPE4 also included two professional event organizers/facilitators who helped plan the workshop agenda, and who engaged with participants during the workshop.

At the workshop, we introduced a Code of Conduct (CoC).<sup>13</sup> The CoC was conceived for the workshop and a guideline for the community of scientists that WSSSPE supports, including their personal and online interactions (e.g., on Twitter, in email lists, in the Slack team). The WSSSPE4 CoC is based on the FORCE11 conference CoC [6], which is in turn based on the Code4Lib CoC [7]. The main guidelines of the CoC are:

WSSSPE events are community events intended for networking and collaboration as well as learning. We value the participation of every member of the community and want all attendees to have an enjoyable and fulfilling experience. Accordingly, all attendees are expected to show respect and courtesy to other attendees throughout the event and in interactions online associated with the event.

The WSSSPE event organizers are dedicated to providing a harassment-free experience for everyone, regardless of gender, gender identity and expression, age, sexual orientation, disability, physical appearance, body size, race, ethnicity, religion (or lack thereof), technology choices, or other group status.

---

<sup>1</sup><http://wssspe.researchcomputing.org.uk/wssspe4/>

<sup>2</sup><http://wssspe.researchcomputing.org.uk/wssspe1/>

<sup>3</sup><http://wssspe.researchcomputing.org.uk/wssspe1-1/>

<sup>4</sup><https://conference.scipy.org/scipy2014/participate/wssspe/>

<sup>5</sup><http://wssspe.researchcomputing.org.uk/wssspe2/>

<sup>6</sup><http://wssspe.researchcomputing.org.uk/wssspe2-1/>

<sup>7</sup><http://scipy2015.scipy.org/ehome/115969/286469/>

<sup>8</sup><http://wssspe.researchcomputing.org.uk/wssspe3/>

<sup>9</sup><http://www.pyladies.com>

<sup>10</sup><https://djangogirls.org>

<sup>11</sup><https://www.womenwhocode.com>

<sup>12</sup><http://www.womeninhpc.org>

<sup>13</sup><http://wssspe.researchcomputing.org.uk/wssspe4/code-of-conduct/>

To make clear what is expected, everyone taking part in WSSSPE events and discussions—speakers, helpers, organizers, and participants—is required to conform to the following Code of Conduct.

- Communicate appropriately for a professional audience including people of many different backgrounds. Sexual language and imagery are not appropriate for any event.
- Be kind to others. Do not insult or put down other attendees. Be mindful of jargon, which can sometimes exclude others from engaging in the discussion.
- Behave professionally. Remember that harassment and sexist, racist, ageist, or exclusionary behavior are not appropriate.

The CoC was introduced at the beginning of WSSSPE4, with the CoC subcommittee and a general email address introduced for reporting concerns or incidents, or asking questions. There was one concern mentioned to the CoC subcommittee after the first half day, regarding how part of the workshop was being run, and we changed the workshop to address this.

This paper is a summary of a longer report [8] on the events at the workshop and the submitted materials. The remainder of the paper includes the call for papers (§2); the WSSSPE mission and vision (§3); a set of presentations that included a keynote, papers, lighting talks, and a panel (§4); and the activities of a set of working groups (§5). One full day of the workshop was spent with participants in the working groups, which occurred in parallel with each other. And each of the working groups left with a plan for how they could move forward. This report also mentions the Slack channel created for further discussions (§6), and it highlights an attendee survey (§7) before concluding (§8). An appendix lists the organizers and program committee members (§A).

## 2. CALL FOR PARTICIPATION

WSSSPE4 was based on the work done at previous WSSSPE meetings, but aimed at producing working groups that better continued working after the workshop ended. In addition, based on feedback after WSSSPE3, it became clear that attendees had two distinct motivations: to make a better future for research software, and to immediately do better research software development. This led to the idea of WSSSPE4 being partially divided into two tracks:

**Track 1 – Building a sustainable future for open-use research software:** defining a vision of the future of open-use research software, and in the workshop, initiating the activities that are needed to get there.

**Track 2 – Practices & experiences in sustainable scientific software:** improving the quality of today’s research software and the experiences of its developers by sharing best practices and experiences.

The call for participation requested: idea papers, implementable proposals related to Track 1; position papers, longer, not previously published papers discussing what we can do to improve sustainable scientific software in the short term; experience papers, longer papers that discuss current experiences and how they have been used to improve the quality of today’s research software and/or the experiences of its developers; demos, brief papers describing a tool or process that would be demonstrated that improves the quality of today’s research software and/or the experiences of its developers; and extended abstracts describing lightning talks, where each author could make a brief statement about work that either had been done or was needed.

Suggested contribution topics included:

- Development and Community: best practices for developing sustainable software; models for funding specialist expertise in software collaborations; software tools that aid sustainability; academia/industry interaction; refactoring/improving legacy scientific software; engineering design for sustainable software; metrics for the success of scientific software; and adaptation of mainstream software practices for scientific software
- Professionalization: career paths; RSE as a brand; RSE outside of the UK or Europe; and Increase incentives in publishing, funding and promotion for better software

- Training: training for developing sustainable software; and curriculum for software sustainability
- Credit: making the existing credit and citation ecosystem work better for software; future credit and citation ecosystem; software contributions as a part of tenure review; case studies of receiving credit for software contributions; and awards and recognition that encourage sustainable software
- Software publishing: journals and alternative venues for publishing software; and review processes for published software
- Software discoverability/reuse: proposals and case studies
- Reproducibility and testing: reproducibility in conferences and journals; and best practices for code testing and code review

Submissions to WSSSPE4 comprised 19 lightning talks, 4 idea papers, 3 position paper, 5 experience papers, and 3 demos. Most of the submissions were accepted, since the goal of WSSSPE is always to take in as many relevant inputs as possible, and to encourage their authors to participate in sharing and implementing their ideas. Specifically, 19 lightning talks, 4 idea papers, 3 position papers, 2 experience papers, and 3 demos were accepted, as mentioned in Section 4. (Two of the submitted lightning talks and one submitted experience paper were rejected, while two more submitted experience papers were accepted as lightning talks.) The papers and lightning talks have been published as a volume in the CEUR Workshop Proceedings [9].

### 3. MISSION AND VISION

Going into WSSSPE4, WSSSPE had no formal mission or vision statement. The organizers developed a strawhorse, which was presented to the participants early in the meeting. The presentation included guidelines as well as examples from other similar communities. The guidelines were that in general, an organization’s mission should stand the test of time and state what the organization does; its vision should imagine what the world would look like if the organization is successful; and from which, focus areas could be used to establish the scope of the organization along with its goals.

The participants were given time after the presentation to write down their comments. Seven people volunteered to work on revising the mission and vision statements, incorporating these comments and presenting back to the community. Based on this feedback, the committee redrafted the mission and vision. The committee added focus areas to clarify the organization’s breadth and to keep the mission and vision simple and long-lasting. Comments after the close of the meeting were incorporated into the statements.

A final draft was put on GitHub and advertised via the WSSSPE mailing list, Facebook group, and Twitter. After two weeks without suggested changes, the final statements, as shown below, were added to the WSSSPE web page (<http://wssspe.researchcomputing.org.uk/about-wssspe/>).

**Mission.** WSSSPE is an international community-driven organization that promotes sustainable research software by addressing challenges related to the full lifecycle of research software through shared learning and community action.

**Vision.** We envision a world where research software is accessible, robust, sustained, and recognized as a scholarly research product critical to the advancement of knowledge, learning, and discovery.

**Focus areas.** WSSSPE promotes sustainable research software by positively impacting:

- **Principles and Best Practices.** Promoting best practices in sustainable software
- **Careers.** Developing and supporting career paths in research software development and engineering
- **Learning.** Engaging in activities to promote peer learning and interaction
- **Credit.** Ensuring recognition of research software as an intellectual contribution equal to other research products

**Definitions:** *Sustainable software* has the capacity to endure such that it will continue to be available in the future, on new platforms, meeting new needs. The *research software lifecycle* includes: acquiring and assembling resources (including funding and people) into teams and communities,

developing software, using software, recognizing contributions to and of software, and maintaining software.

#### 4. PRESENTATIONS

The keynote was given by Patricia Lago and entitled “The legacy of unsustainable software”. Sustainability is broadly associated with natural ecologic systems. When we translate the notion of sustainability to software solutions, however, we often confuse *impact in a certain point in time* with *balanced and durable effects*. In addition, software sustainability adds a fourth dimension to environmental, social and economic aspects: technical sustainability, and hence higher complexity [10]. Lago’s keynote discussed the challenges (and some related ongoing research) of combining technical and environmental sustainability, providing a complementary angle to the workshop discussions.

WSSSPE4 included the presentation of 12 10-minute talks (based on four idea papers, three position papers, two experience papers, and three demos) that addressed a wide range of topics around sustainability for software in science. These talks covered three main areas.

The following four papers belong to the first area: the academic environment. They address diverse aspects ranging from incentives for quality software to advocating a professional society for research software to roles and degrees for research software engineers. For talks with multiple authors, an \* indicates the author who presented the talk.

- Michael Heroux: *Idea paper: Sustainable & Productive: Improving Incentives for Quality Software* [11].
- Gabrielle Allen: *Idea Paper: Establishing a Professional Society for Research Software* [12].
- Olivier Philippe\*, Simon Hettrick, and Neil Chue Hong: *Experience Paper: Preliminary analysis of a survey of UK Research Software Engineers* [13].
- Christopher Gwilliams: *Demo: Using industrial engagement to create and develop research ties within academia* [14].

The second main area, concerned with characteristics and needs of research software, includes the following five papers:

- Anshu Dubey\* and Katherine Riley: *Experience Paper: Software Engineering and Community Codes Track in ATPESC* [15].
- Neil Chue Hong: *Position Paper: Why do we need to compare research software, and how should we do it?* [16].
- Anshu Dubey\* and Lois Curfman McInnes: *Idea Paper: The Lifecycle of Software for Scientific Simulation* [17].
- Francisco Queiroz\* and Rejane Spitz: *Position Paper: Collaborative Gamification Design for Scientific Software* [18].
- Bruce Childers\*, Jack Davidson, Wayne Graves, Bernard Rous, and David Wilkinson: *Position Paper: Active Curation of Artifacts is Changing the Way Digital Libraries will Operate* [19].

Three contributions focused on the third main area: elucidating lessons learned from or visions for use cases of scientific software and communities working with and/or on a software package.

- Debashis Ganguly, William C. Garrison III, David Wilkinson, Bruce R. Childers, Adam J. Lee, and Daniel Mosse\*: *Demo: Composing, Reproducing, and Sharing Simulations* [20].
- Sameer Shende\* and Allen Malony: *Demo: Using TAU for Performance Evaluation of Scientific Software* [21].
- Janos Sallai\*, Christopher Iacovella, Christoph Klein, and Tengyu Ma: *Idea Paper: Development of a Software Framework for Formalizing Forcefield Atom-Typing for Molecular Simulation* [22].

There were 19 lightning talks presented at WSSSPE4, as follows (in order of presentation).

- Shoaib Sufi: *The Software Sustainability Institute Fellowship programme; supporting the social side of research software* [23].

- Sandra Gesing\*, Maytal Dahan, Linda B. Hayden, Katherine Lawrence, Suresh Marru, Marlon Pierce, Nancy Wilkins-Diehr, and Michael Zentner: *The Science Gateways Community Institute - Supporting Communities to Achieve Sustainability for Their Science Gateways* [24].
- Karthik Ram\*, Noam Ross, and Scott Chamberlain: *A model for peer review and onboarding research software* [25].
- Frank Löffler and Steven Brandt\*: *A vision of computing in 10+ years* [26].
- Daniel S. Katz, Kyle E. Niemeyer\*, Arfon M. Smith, and the FORCE11 Software Citation Working Group: *Software Citation: Process, Principles, and Implementation* [27].
- Ray Idaszak\*, David G. Tarboton, Hong Yi, Michael Stealey, Pabitra Dash, Alva Couch, Daniel P. Ames, Jeffery S. Horsburgh, Tony Castronova, Jon Goodall, Mohamed Morsy, Venkatesh Merwade, Mauriel Ramirez, Tian Gan, Drew (Zhiyu) Li, Jeff Sadler, Shawn Crawley, Zhaokun Xue, Lan Zhao, Carol Song, Christina Bandaragoda: *HydroShare - A Case Study in Software Engineering Best Practices and Culture Change for Developing Sustainable Community Software* [28].
- Lorraine Hwang\*, Wolfgang Bangerth, Timo Heister, and Louise Kellogg: *ASPECT: Hackathons as an Example of Sustaining an Open Source Community* [29].
- Carole Goble: *A Simple Profiling Framework for Software User-Producer Reciprocity Review* [30].
- Stephan Druskat: *A proposal for the measurement and documentation of research software sustainability in interactive metadata repositories* [31].
- Mistral Contrastin, Matthew Danish, Dominic Orchard\*, and Andrew Rice: *Supporting software sustainability with lightweight specifications* [32].
- Alice Allen, Cecilia Aragon, Christoph Becker, Jeffrey Carver, Andrei Chiş, Benoit Combemale, Mike Croucher, Kevin Crowston, Daniel Garijo, Ashish Gehani, Carole Goble\*, Robert Haines, Robert Hirschfeld, James Howison, Kathryn Huff, Caroline Jay, Daniel S. Katz, Claude Kirchner, Kateryna Kuksenok, Ralf Lämmel, Oscar Nierstrasz, Matt Turk, Rob van Nieuwpoort, Matthew Vaughn, and Jurgen Vinju: *"I solemnly pledge": A Manifesto for Personal Responsibility in the Engineering of Academic Software* [33].
- Neil Chue Hong\*: *Making it easier to understand research software impact* [34].
- Willem Robert van Hage\*, Jason Maassen and Rob van Nieuwpoort: *Software Impact Measurement at the Netherlands eScience Center* [35].
- Gabrielle Allen, Emily Chen\*, Ray Idaszak, and Daniel S. Katz: *Report on Software Metrics for Research Software* [36].
- Eric L. Seidel\* and Gabrielle Allen: *Bringing Techniques from Software Engineering into Scientific Software* [37].
- Iain Emsley\* and David De Roure: *Sustaining the social: Connecting the lives of Drupal community* [38].
- Jack Dongarra, Sven Hammarling, Nicholas J. Higham, Samuel D. Relton\*, Pedro Valero-Lara, and Mawussi Zounon: *Creating a Standardised Set of Batched BLAS Routines* [39].
- Lukas Breitwieser\*, Roman Bauer, Alberto Di Meglio, Leonard Johard, Marcus Kaiser, Marco Manca, Manuel Mazzara, Fons Rademakers, Max Talanov: *The BioDynaMo Project* [40].
- Aseel Aldabjan, Robert Haines, and Caroline Jay\*: *How should we measure the relationship between code quality and software sustainability?* [41].

The final WSSSPE4 presentation was a panel of five experts with different perspectives on "Best Practices for Scientific Software." These were: Alice Allen, Editor, Astrophysics Source Code Library, who brought an understanding of the difficulties of organizing a community and curating their software; Mike Croucher from the University of Sheffield and Rob Haines from the University of Manchester who are both Research Software Engineers with decades of experience of writing code for researchers; Patricia Lago from Vrije Universiteit Amsterdam, who presented a keynote at the workshop and brought a fresh perspective on software sustainability from the point of view of its impact on society and business; and Karthik Ram from the Berkeley Institute for Data Science,



who brought his perspective on the practicalities of using scientific software to conduct his research as a data scientist.

## 5. WORKING GROUPS

After most of the lightning talks and other presentations, WSSSPE used three areas of a large room to let attendees use sticky flip charts and sticky notes on the walls to suggest a vision on any aspect of the work, a gap or challenge, or a project idea. Next, attendees organically formed groups around the flip charts, and 12 groups emerged. Summaries of each group’s activities follow; for more details about the group discussions see [8].

**5.1. Verifying best practices & metrics for sustainable research software.** Many open source projects for research software document their best practices that contributed to the sustainability of the software. Many of these projects also document software metrics they use to define their research software project’s success. This group will take the outputs of the WSSSPE efforts to identify best practices for creation of sustainable software for science/academia, and also the outputs of WSSSPE efforts to identify metrics for sustainable software for science/academia and cross-reference these with current open source research software that successfully uses modern software engineering. This will allow the group to identify gaps on both sides. This approach will also allow the group to hypothesize how successful open source projects can be further improved, verify that recommended approaches for software engineering for science/academia are sufficient and valid, and that metrics for software engineering for science/academia are relevant and useful. The group has a specific objective of getting a good cross-sampling of disparate software including community model codes, community cyberinfrastructure, community analytic tools, etc.

Each group member will volunteer to take on one to two projects, ideally ones in which they are already interested. This effort will evaluate an initial 5–10 projects. Once the group documents the workflow by which these evaluations are performed, other people can choose to follow this workflow and contribute their own evaluations of additional software.

To join this group or obtain more information about it, please send an email to the verifying best practices & metrics for sustainable research software group: <wssspe4-verify-best-practices@googlegroups.com>.

**5.2. Software Sustainability Alliance.** The Software Sustainability Alliance working group aims to establish an alliance between organizations interested in improving the sustainability of research software. Such an alliance would ease the collaboration between member organizations to improve the sharing of expertise, resources and best practices. These organizations are funded groups or teams that aim to advance research software sustainability beyond their local university or community. More specifically, this working group aims to define the scope of this alliance and provide clear distinction with WSSSPE. The group also seeks to understand the incentives for members of this alliance and identify its key activities. It is currently seeking feedback on potential member organizations, as well as the aims and scope of this alliance.

Currently, point-to-point collaboration exists between organizations, but this inadvertently results in competition or redundancy within the sustainable software community. An alliance of software sustainability organizations would ease inter-organization collaboration and the promotion of software sustainability. This alliance would also improve the pooling of competencies and the sharing of expertise. Furthermore, with the international scope of this alliance, it could support an organization who wants to hold an event in a country where another organization exists.

If interested, visit <http://softwaresustainability.org/> or email Neil Chue Hong (<N.ChueHong@software.ac.uk>) and Jean Salac (<jeansalac@virginia.edu>)

**5.3. Scientific Software Prototyping Infrastructure (S2PI).** There is a productivity bottleneck—yet to be solved—in HPC from the human perspective, first identified in the 1980s [42]. Of all domain-specific scientists, only a percentage of those use simulations due to perceptions mostly related to the difficulty of using and developing those tools. XSEDE and other resources

have become a gateway for successfully increasing the basis of HPC scientific users by facilitating their access to infrastructure and tools [43], but the development challenge remains. Of all scientists who use simulations, only a small percentage know how to develop prototypes or full applications that can be later scaled by computer scientists and engineers. Writing new parallel applications is unusual for domain scientists and thus has created a complex dependency on specialized scientific programmers, who are scarce [44]. However, the capacity to quickly envision and prototype applications by domain experts—later to be fully implemented into by scientific programmers—is critical for acquiring maturity and proficiency in the grand challenges that Exascale attempts to solve, and moreover for allowing scientists to develop their own ideas quickly and productively [45]. The latter, in addition, would increase service utilization time in HPC systems, a critical measure of efficiency. Considering the existing code base in scientific computing, software infrastructures that allow a clean transition from legacy systems to new Exascale platforms while preserving flexible prototyping towards the future are central [46].

This group deems the ability to rapidly construct software artifacts that can be trusted in terms of the methods from their design up, easily discarded when wrong and extended when right at low human and computational cost to be one aspect of scientific software sustainability. In essence, a motto for scientific software prototyping is *fail hard, fail fast*. The group’s objective is development of a prototype tool that allows domain scientists to generate simulation prototypes through a simple, yet expressive declarative strongly-typed programming language close to equational expressions. The result of that specification is both an executable artifact as well as a specification for scientific programmers to later flesh out completely and adapt to particular infrastructures. This tool will start in the domain of computational physics, and (probably) extend to other domains, and the code will be open source. The underlying hypothesis of this project is that a strongly typed, declarative problem-specification language with adequate constructs, to be implemented through stencils and algorithmic skeletons tied to numerical methods, captures a subset of useful and interesting problems in physics and other domains. The group plans to develop a functional (meta-)prototype before WSSSPE5, and if useful, then develop a larger software infrastructure tied to particular cyberinfrastructure resources.

To join this project, please visit the GitHub repository at <https://github.com/nunezco2/S2PI> or join the WSSSPE Slack channel `#wg-sci-soft-proto`.

**5.4. CodeMeta.** Research software is often not shared; that which is shared may not have much metadata associated with it, and that which does exist often does not travel further than the website on which the software resides. The CodeMeta project<sup>14</sup> wants to incentivize software developers to release their software, encourage the development of metadata for it, enable credit assignment and citation of research software, increase its discoverability, more easily track dependencies, and enable reuse of software metadata, all goals that WSSSPE attendees have great interest in supporting. CodeMeta seeks in part to create a “Rosetta Stone” for software metadata to facilitate retaining such metadata between repositories, services, registries, indexers, publishers, citation managers, and other entities that create, ingest, use, and/or store metadata about software. The project also wants to establish a JSON-LD schema as a tool for making metadata machine-readable [47].

The primary objective of this working group, which included people who had previously been working on the CodeMeta project, is to find ways to help the project come to fruition. The working group plans to engage with those already working on CodeMeta, to examine the existing CodeMeta crosswalk table to see what improvements and additions might be made, and to determine how to engage the community and provide ongoing social engagement and structure. Further, the group wants to assist in the implementation of the specifications and, by providing an outsider’s view, contribute suggestions for more understandable project documentation. Finally, the group seeks a better way or ways to present the crosswalk table to make it more easily understood and consumable by research software communities. While at WSSSPE4, the group greatly expanded the CodeMeta project README file to include a description of the project geared to those with little or no prior

<sup>14</sup><http://codemeta.github.io/>



knowledge of the project, a list of contributors, information on how one can get involved, a brief project history and who is managing the project, and links to additional information.

Though a Google group mailing list has been established for the working group, the easiest way to engage with the CodeMeta project is through its Github repository<sup>15</sup>.

**5.5. White paper on developing sustainable software.** Many diverse aspects and dimensions (such as economic, technical, environmental, and social) of developing sustainable software can be investigated. This group aims to write white papers that will focus on scientific environments and their implications, targeted at developers and project managers of scientific software. Given the complexity of this field, it is important to select a subset of sustainability aspects for the white papers. The idea is to create a series of papers instead of trying to tackle all important topics in one paper. While there are already a few papers available on best practices and sustainability of scientific software in general, the group’s goal is to create a series of papers that lead to consensus in the community, tackle many diverse aspects, and stay up-to-date with new trends. For the first white paper, the group aims to set the stage with successful use cases and an analysis of why they have been successful. Further topics will include community-related practices, government and management, funding, metrics, tools, and usability. The resulting draft white paper will be distributed via the WSSSPE email list. After collecting feedback on it, the group’s plans are to attract a wider community to contribute to an extended journal paper version and to investigate whether more people would like to be involved in the white paper series.

The GitHub repository for the white paper can be found at <https://github.com/WSSSPE/WG-Best-Practices>. For more information and requests, join the `#wg-best-practices` channel at WSSSPE’s Slack team.

**5.6. Social science for scientific software.** There is more and more academic research being done on topics related to software sustainability, including work on software engineering practices and management of open source projects. However, academic research in general is often siloed for many reasons, and work on topics relevant to software sustainability is no exception. There are many academic studies and projects which may be relevant for practitioners working in this area, and many projects and initiatives by practitioners that may be relevant for social scientists. However, there is a gap between these two in research and practice. This working group is motivated by the goal of building better connections between academic researchers who are studying topics in or relating to software sustainability with practitioners, managers, and administrators who are working in the area of software sustainability. In any domain, bringing research and practice closer together a mutually beneficial goal, but also has its challenges. The group aims to give social scientists and practitioners working on scientific software, software sustainability, and open source and a better understanding of each other’s work, as well as help them connect and coordinate on specific projects of mutual interest. Furthermore, the group also recognize that academic social science and research software engineering are not monolithic, and there is a need to connect people who care about research-driven best practices inside of these two domains with each other as well.

The working group plans to briefly survey existing literature to get a better sense of the academic research landscape, facilitate some initial dialog between academic researchers and practitioners at WSSSPE4, then identify needed actions that would be mutually beneficial to researchers and practitioners. This plan resulted in the creation of the following SMART steps listed in the next section.

To join the group, join its mailing list at <https://groups.google.com/forum/#!forum/researchsoftwarestudies>

**5.7. Software best practices for undergraduates.** This working group was motivated by the perceived prevalence of so-called “hidden code” in scientific communities: code written by individual researchers in an unsustainable way that is never shared with the larger community. Participants

---

<sup>15</sup><https://github.com/codemeta/codemeta>

recalled their own experience working with colleagues who write hidden code or inheriting unsustainable hidden code from collaborators. The question is, then, how to prevent researchers from writing hidden code? The participants hypothesized that the best strategy is to catch researchers while they are still in training and teach them software best-practices. Therefore, this working group’s goal is to develop courses on software best practices aimed at undergraduate students studying domain science. The program might be similar to a Software Carpentry or Data Carpentry workshop but focused for domain scientists.

While implementation of a course aimed at each domain science is a long term goal, a short-term, achievable goal, is to develop a curriculum for a course aimed at a single domain science. Since the participants in the working group have expertise in physics, this is a natural target. The development of a successful curriculum relies on the expertise of software engineers to describe the best practices, domain experts to describe model problems and work-flow, and instructors to formulate the pedagogy. Ideally these people will be brought together for a short workshop or hackathon with the goal of drafting the curriculum. The hackathon will be implemented and a draft of the curriculum will be written within the next year. Later, organizational partners will be required to actually implement the program.

Anyone interested in contributing should get in touch via one of the following methods:

- Join the Google Group/ mailing list [48].
- Join the WSSSPE Slack (§6) and the `#wg-undergraduatecourse` channel.
- Ask to join the WSSSPE4-undergraduate-course organization [49].

**5.8. Meaningful metrics for sustainable software.** This group aims to increase the visibility of the quality of scientific software, facilitate the reusability of scientific software, and promote the best software practices by standardizing metrics via interviews with scientific software developers. This working group believes improving the current software metrics system will increase software sustainability. Currently, there are inefficiencies regarding software duplication, sustainability, and selection, as well as others, within the scientific software community. In order to address these inefficiencies, the group aims to create a goal-oriented method to collecting productive metrics by focusing on the developer side of software.

In the context of this aim, the group hopes to find an efficient solution for streamlining the process of collecting and utilizing metrics to benefit software sustainability, as well as minimize the current inefficiencies within the scientific software community. This will improve software evaluation and comparison, thus reducing the effort spent on seeking scientific software. The group plans on surveying and interviewing scientific software developers to form metrics from the goals they have for their software, then to publicize the results.

For more information, please contact Emily Chen at <echen35@illinois.edu>.

**5.9. Coordinating access to continuous integration (CI) for research software.** Each developer of software with uncommon needs (hardware, software, libraries, data sets), non-public code, or tests that exceed time limits for free plans must acquire, setup, and maintain their own continuous integration systems because their needs make them ineligible for popular free services such as Travis CI. For example, software groups that develop BIOUNO, CI4SI, or GROMACS have done this. (Note that Debian provides a testing infrastructure<sup>16</sup> that is mature and supports hardware and other requirement specifications.)

This group is interested in reducing the burden of different projects having to build and maintain their own continuous integration systems (when publicly available CI are not a fit), by coordinating and sharing this burden across multiple projects. The scope of the group’s interest is any type of testing, though interactive access for troubleshooting would be of particular interest beyond just automated testing.

However, there are a lot of open issues:

---

<sup>16</sup><https://ci.debian.net/doc/>

- Some similar work has already been done. How can this group apply and/or learn from that work, rather than reinventing the wheel?
- How to ensure this will work across disciplines?
- Meaningful CI for large projects may need hundreds of CPU hours per day

Some possible goals include:

- Acquire additional hardware such as GPUs, Xeon PHI, FPGAs and add/share them to e.g. Debian's testing infrastructure or to a shared Jenkins-based infrastructure
- Extending Debian's scope to include published but not mature software
- Implementing the same interface but with specialized hardware or available to non-public codes.
- <https://reproducible-builds.org> but for CI

For more information, see <https://groups.google.com/forum/#!forum/continuous-integration-for-research-software>

**5.10. Software engineering processes tailored for research software.** This working group is concerned with identifying processes that are not adequately covered by general software engineering, starting with verification and testing. Computational science and engineering applications have many moving parts that need to interoperate with one another. The accuracy and reliability of results produced by scientific software depends not only on the individual components behaving correctly, but also on the validity of their interactions. As scientific understanding grows, the corresponding computational software models are refined, leading to more complex codes. Increasing complexity makes them more prone to defects, not only in individual code units, but also in interaction among units. Therefore, a strong verification process combined with a rigorous testing regime plays a critical role in the prevention of generating incorrect scientific results. However, most science teams struggle to find a good solution for themselves.

One cause is lack of exposure to the practices. While good developers will test their code to verify that it operates as expected, they may not appreciate that without regular testing, defects can be introduced inadvertently. An even bigger challenge is that those who understand the importance of regular testing do not often find much help from software engineering literature. There is a significant gap between the testing gospel and its applicability to computational science. This gap leads to frustration and abandonment of the good with the bad. Some relevant literature exists, in particular experiences from practitioners in computational science who developed their own solutions. However, this literature is scattered among many different forums, and can be challenging to find. This working group aims to address this gap by curating the existing content and contributing content where none exists.

This working group will (1) conduct a literature survey to gauge the extent of awareness of the issue in general, (2) generate content useful for the community where needed, and (3) curate the collected and added content for the use of the community.

Readers interested in getting more information should use the working group's channel in the WSSSPE Slack (§6), called `#wg-testing-in-science`. Additionally, a git repository (<https://github.com/WSSSPE/WG-Best-Practices.git>) exists for contributing content and reference to, and curation of the existing literature on this topic.

**5.11. Open research index.** The aim of this group is to investigate the building of an index of research products in an open sustainable manner. Its goal is not to eliminate commercial products, but to build on what is there and provide data and services that are missing. The Open Research Index should take in all research products (papers, software, datasets, workflows, etc.) from their publishers and recorders (journals, societies, domain repositories, government [open access] repositories, preprint servers, general repositories [e.g., figshare, zenodo]) and other services (CrossRef, ORCID). Each product should list authors and citations and allow people to search the resulting network. Users should also be able to interact with their own record and edit it, like Google Scholar allows.

The working group’s plans are relatively simple to express, though quite complex to undertake. They are to first determine a plan to build an open research index that allows various stakeholders to satisfy their needs, then to determine if the plan is feasible. At the time of the meeting, and today as well, it is unclear who has the time and energy to pursue this idea. Thus, a leader needs to be identified. With a leader and feasible plan, the group would then obtain resources, and then build the index. Because Google and others provide some similar services today, though these services (and the underlying data) could be removed at any time and the community cannot build new services, these companies would ideally be involved in the activities of the group.

Alternatively or additionally, the group could build a mailing list for us and discuss further, depending on how receptive others are to this idea. At WSSSPE4, there seemed to be enough interest to do this, so the group set up a Slack channel within the WSSSPE team (§6), called `#wg-open-research-idx` for its members and any future members.

**5.12. Letters of evaluation for computational scientists.** Scientists working on scientific software are often located in disciplinary departments, depending on whether their software originates from the mathematical, physical, chemical, or other disciplines. As a consequence, they are frequently outside the core areas of their science, and their contributions are typically to both the research activity their software enables, as well as on algorithm and implementation aspects. This presents issues when letters of evaluation for hiring, tenure, and promotion do not specifically cover how this is relevant to the discipline.<sup>17</sup>

This group believes that the authors of scientific software provide important services to departments that are no less than strictly disciplinary research. Consequently, leveling the playing field with more disciplinary candidates for hires, tenure, and promotion requires that letter writers be aware of how their letters will be read by committees. It also requires that committees be aware that such letters often look different and may provide a different perspective of how a candidate’s achievements should be assessed. For example, in mathematics a typical candidate would be evaluated on the difficulty and depth of the statements she may have proven in their papers. On the other hand, an author of mathematical software would likely be evaluated based on the impact of her software, or the number of citations of the publications that describe it. She may also be evaluated by how widely the software is used *outside* mathematics, a criterion that is rarely used for more disciplinary mathematicians.

Addressing this problem likely requires that letter writers pay particular attention to who exactly the audience of such letters is, and tailor the message by specifically highlighting how the work of a candidate benefits the department and discipline that is considering a candidate. On the other hand, committee members also need to pay attention to the fact that there are areas that are important to the mission of their department and professional community in which different standards for evaluation hold. There are essentially two important strategies that build on each other: (i) raising awareness of the problem beyond just those who are affected by it (namely, computational scientists working on scientific software), and (ii) providing letter writers, letter readers, and evaluating committees with guidance on what criteria are relevant in assessing scientific software authors.

Concrete guidance is most valuable if it comes from respected bodies such as professional societies or established and respected organizations. Getting these to act will only be possible if the problem is widely acknowledged, and so the first of the goals above should be the current focus. The group will attempt to address it by organizing sessions at conferences that address the career problem, as well as writing editorials that can be published in the magazines of professional societies. These editorials ought to outline best practices for letter writers that specifically (i) make clear the contribution of a candidate to their interdisciplinary area, (ii) the relevance to their home discipline, and (iii) why writing software is good for the discipline itself.

Ultimately, building a community large enough to affect change is important; contact Wolfgang Bangerth at <bangerth@colostate.edu> if you are interested in helping.

---

<sup>17</sup>The extensive use of such letters, and the problems that are associated with it, may be an issue specific to the United States.

## 6. SLACK TEAM AND CHANNELS

As part of the discussion at WSSSPE4, we created a Slack team for WSSSPE [50]. To join it, go to <https://wssspe.signup.team/>. A number of the working groups also created channels within the team; see the subsections in §5 for the names of these channels.

## 7. ATTENDEE SURVEY

All participants present on the last day of the meeting, September 14, 2016, were asked to complete the online survey administered through Qualtrics<sup>18</sup>. 44 responses were recorded, with one respondent indicating that they took the survey twice. The survey URL was also later distributed to attendees by email and advertised on twitter for attendees who were not there on the last day to have a chance to respond, and one additional response was recorded after the meeting. Hence, the total number of unique individuals responding to the survey is likely 44. The survey contained nine multiple choice and open-text answer questions. Text responses have been alphabetize to preserve anonymity. The survey had a 100% completion rate.

In general, the respondents were highly satisfied with the meeting and interested in continuing and participating in WSSSPE activities. The mix of topics and types of interactions—talks, panels and discussion—was well balanced with several indicating a desire for time for questions and answers following talks as well for discussions in general. Many respondents indicated that they will remain engaged in WSSSPE working group initiatives. However, demand for a professional organization encompassing WSSSPE interests was weaker with over half interested in joining and a large percentage willing to consider the idea. Respondents were grateful for the opportunity to network, explore new collaborations and for travel support to the conference.

The survey results indicate that future WSSSPE conferences should consider the balance of the attendees. This year’s meeting had many first time and early career participants who would have benefited from a review of basic concepts and terminology including what it means for software to be sustainable and the roles of research software engineers. Additional emphasis and topics to explore would be the inclusion of more case studies, focus on the decision making process in developing and using software, deeper dives into selected topics, tutorials, software training within and outside of STEM, lowering the barriers to implementing best practices in software development, and progress towards executing WSSSPE’s vision.

## 8. CONCLUSIONS

In WSSSPE4, we heard about a number of interesting projects and ideas, and used those ideas to create and form working groups, intended to start at the workshop and then continue afterwards, to address challenges that arose from the workshop ideas. This workshop has reinforced the lesson from WSSSPE3 that it is relatively easy to get motivated people to attend a meeting and productively spend their time there both doing work and planning more work, but it is very hard to get that additional work after the meeting to take place. The main problem seems to be one of time. Once the attendees have agree to spend their time at the workshop, they put their energy into doing so productively, but they have not really committed themselves to anything more than this, so their energy and effort trails off, as all of their other commitments (particularly, those they are funded to do as part of their jobs) come back to the fore. Without a process to resolve this concern, the utility of further multi-day WSSSPE workshops is unclear.

## ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation (ACI-1648293, ACI-1547611), by the Gordon and Betty Moore Foundation (GBMF#5620), and by the Alfred P. Sloan Foundation (G-2016-7214).

---

<sup>18</sup><https://www.qualtrics.com>.



## APPENDIX A. ORGANIZATION

WSSSPE4 organizers were:

Gabrielle Allen	University of Illinois Urbana–Champaign, USA
Jeffrey Carver	University of Alabama, USA
Sou-Cheng (Terrya) Choi	Illinois Institute of Technology, USA
Tom Crick	Cardiff Metropolitan University, UK
Michael R. Crusoe	Common Workflow Language Project
Sandra Gesing	University of Notre Dame, USA
Robert Haines	University of Manchester, UK
Michael Heroux	Sandia National Laboratories, USA
Lorraine J. Hwang	University of California, Davis, USA
Daniel S. Katz	University of Illinois Urbana–Champaign, USA
Kyle E. Niemeyer	Oregon State University, USA
Manish Parashar	Rutgers University, USA
Colin C. Venters	University of Huddersfield, UK

The WSSSPE4 program committee comprised:

David Abramson	University of Queensland, Australia
Lorena A. Barba	George Washington University, USA
Ross Bartlett	Sandia National Laboratories, USA
Christoph Becker	University of Toronto, Canada
Ewout van den Berg	IBM Watson, USA
David Bernholdt	Oak Ridge National Laboratory, USA
Stefanie Betz	Karlsruhe Institute of Technology, Germany
Coral Calero	Universidad Castilla La Mancha, Spain
Ishwar Chandramouli	National Cancer Institute, National Institutes of Health, USA
Ruzanna Chitchyan	University of Leicester, UK
Karen Cranston	Duke University, USA
Ewa Deelman	Information Sciences Institute, University of Southern California, USA
Charlie E. Dibsedale	O-Sys, Rolls Royce PLC, UK
Anshu Dubey	Argonne National Laboratory, USA
Nadia Eghbal	Independent Researcher (via Ford Foundation), USA
Peter Elmer	CERN, Switzerland
Martin Fenner	DataCite, Germany
David Gavaghan	University of Oxford, UK
Mike Glass	Sandia National Laboratories, USA
Carole Goble	University of Manchester, UK
Joshua Greenberg	Alfred P. Sloan Foundation, USA
Michael K Griffiths	University of Sheffield, UK
Sarah Harris	University of Leeds, UK
James Hetherington	University College London, UK
Fred J. Hickernell	Illinois Institute of Technology, USA
Neil Chue Hong	Software Sustainability Institute, University of Edinburgh, UK
Caroline Jay	University of Manchester, UK
Rafael C. Jimenez	ELIXER, Cambridge, UK
Matthew B. Jones	University of California Santa Barbara, USA
Nick Jones	New Zealand eScience Infrastructure (NeSI), NZ
Sedef Akinli Kocak	Ryerson University, Canada
Jong-Suk Ruth Lee	National Institute of Supercomputing and Networking, KISTI, Korea
James Lin	Shanghai Jiao Tong University, China
Frank Löffler	Louisiana State University, USA
Gregory Madey	University of Notre Dame, USA
Ketan Maheshwari	University of Pittsburgh, USA
Steven Manos	University of Melbourne, Australia
Chris A. Mattmann	NASA JPL; University of Southern California, USA
Abigail Cabunoc Mayes	Mozilla Science Lab, USA
Robert H. McDonald	Indiana University, USA
Lois Curfman McInnes	Argonne National Laboratory, USA
Alberto Di Meglio	CERN, Switzerland
Chris Mentzel	Gordon and Betty Moore Foundation, USA
Peter Murray-Rust	University of Cambridge, UK
Christopher R. Myers	Cornell University, USA
Jarek Nabrzyski	University of Notre Dame, USA
Cameron Neylon	Curtin University, Australia

Aleksandra Pawlik	New Zealand eScience Infrastructure (NeSI), NZ
Fernando Perez	Lawrence Berkeley National Laboratory; University of California, Berkeley, USA
Marian Petre	The Open University, UK
Marlon Pierce	Indiana University, USA
Andreas Prlic	University of California, San Diego, USA
Karthik Ram	University of California, Berkeley, USA
Morris Riedel	Juelich Supercomputing Centre, Germany
Dave De Roure	Oxford e-Research Centre, University of Oxford, UK
Norbert Seyff	University of Zurich, Switzerland
Arfon Smith	GitHub Inc, USA
Borja Sotomayor	University of Chicago, USA
Edgar Spalding	University of Wisconsin, USA
Maria Spichkova	RMIT University, Australia
Victoria Stodden	University of Illinois Urbana-Champaign, USA
Matthew Turk	University of Illinois Urbana-Champaign, USA
Nancy Wilkins-Diehr	San Diego Supercomputer Center, University of California, San Diego, USA
James Willenbring	Sandia National Laboratories, USA
Scott Wilson	Cetis LLP, UK
Theresa Windus	Iowa State University and Ames Laboratory, USA

## REFERENCES

- [1] Katz DS, Allen G, Chue Hong N, Parashar M, Proctor D. First Workshop on Sustainable Software for Science: Practice and Experiences (WSSSPE): Submission and Peer-Review Process, and Results. arXiv; 2013. 1311.3523. Available from: <http://arxiv.org/abs/1311.3523>.
- [2] Katz DS, Choi SCT, Lapp H, Maheshwari K, Löffler F, Turk M, et al. Summary of the First Workshop on Sustainable Software for Science: Practice and Experiences (WSSSPE1). Journal of Open Research Software. 2014;2(1). Available from: <https://doi.org/10.5334/jors.an>.
- [3] Katz DS, Allen G, Chue Hong N, Cranston K, Parashar M, Proctor D, et al. Second Workshop on Sustainable Software for Science: Practice and Experiences (WSSSPE2): Submission, Peer-Review and Sorting Process, and Results. arXiv; 2014. 1411.3464. Available from: <http://arxiv.org/abs/1411.3464>.
- [4] Katz DS, Choi ST, Wilkins-Diehr N, Chue Hong N, Venters CC, Howison J, et al. Report on the Second Workshop on Sustainable Software for Science: Practice and Experiences (WSSSPE2). Journal of Open Research Software. 2016;4(1):e7. Available from: <https://doi.org/10.5334/jors.85>.
- [5] Katz DS, Choi ST, Niemeyer KE, Hetherington J, Löffler F, Gunter D, et al. Report on the Third Workshop on Sustainable Software for Science: Practice and Experiences (WSSSPE3). Journal of Open Research Software. 2016;4(1):e37. Available from: <http://doi.org/10.5334/jors.118>.
- [6] Ilik V, Thaney K, Katz DS, others. FORCE11 Conference Code of Conduct; 2016. Accessed: 2016-09-15. <https://www.force11.org/meetings/force2016/code-conduct>.
- [7] Code4Lib. CodeOfConduct4Lib; 2015. Accessed: 2016-09-15. Git hash 4270594fdb2e8eeb45256ea6304c8bd800e6a86. [https://github.com/code4lib/antiharassment-policy/blob/master/code\\_of\\_conduct.md](https://github.com/code4lib/antiharassment-policy/blob/master/code_of_conduct.md).
- [8] Katz DS, Niemeyer KE, Gesing S, Hwang L, Bangerth W, Hettrick S, et al. Report on the Fourth Workshop on Sustainable Software for Science: Practice and Experiences (WSSSPE4). arXiv; 2017. 1705.02607. Available from: <http://arxiv.org/abs/1705.02607>.
- [9] Allen G, Carver J, Choi SCT, Crick T, Crusoe MR, Gesing S, et al., editors. Proceedings of the Fourth Workshop on Sustainable Software for Science: Practice and Experiences (WSSSPE4). Aachen; 2016. Urn:nbn:de:0074-1686-8. Available from: <http://ceur-ws.org/Vol-1686/>.
- [10] Lago P, Koçak SA, Crnkovic I, Penzenstadler B. Framing sustainability as a property of software quality. Communications of the ACM. 2015;58(10):70–78.
- [11] Heroux M. Sustainable & Productive: Improving incentives for quality software. In: Allen G, Carver J, Choi SCT, Crick T, Crusoe MR, Gesing S, et al., editors. Workshop on Sustainable Software for Science: Practice and Experiences. No. 1686 in CEUR Workshop Proceedings. Aachen; 2016. Urn:nbn:de:0074-1686-8. Available from: [http://ceur-ws.org/Vol-1686/WSSSPE4\\_paper\\_10.pdf](http://ceur-ws.org/Vol-1686/WSSSPE4_paper_10.pdf).
- [12] Allen G. Establishing a professional society for research software. In: Allen G, Carver J, Choi SCT, Crick T, Crusoe MR, Gesing S, et al., editors. Workshop on Sustainable Software for Science: Practice and Experiences. No. 1686 in CEUR Workshop Proceedings. Aachen; 2016. Urn:nbn:de:0074-1686-8. Available from: [http://ceur-ws.org/Vol-1686/WSSSPE4\\_paper\\_24.pdf](http://ceur-ws.org/Vol-1686/WSSSPE4_paper_24.pdf).
- [13] Philippe O, Hettrick S, Chue Hong N. A survey of Research Software Engineers in the UK. In: Allen G, Carver J, Choi SCT, Crick T, Crusoe MR, Gesing S, et al., editors. Workshop on Sustainable Software for Science: Practice and Experiences. No. 1686 in CEUR Workshop Proceedings. Aachen; 2016. Urn:nbn:de:0074-1686-8. Available from: [http://ceur-ws.org/Vol-1686/WSSSPE4\\_paper\\_19.pdf](http://ceur-ws.org/Vol-1686/WSSSPE4_paper_19.pdf).
- [14] Gwilliams C. Using industrial engagement to create and develop research ties within academia. In: Allen G, Carver J, Choi SCT, Crick T, Crusoe MR, Gesing S, et al., editors. Workshop on Sustainable Software for Science: Practice and Experiences. No. 1686 in CEUR Workshop Proceedings. Aachen; 2016. Urn:nbn:de:0074-1686-8. Available from: [http://ceur-ws.org/Vol-1686/WSSSPE4\\_paper\\_1.pdf](http://ceur-ws.org/Vol-1686/WSSSPE4_paper_1.pdf).
- [15] Dubey A, Riley K. Software engineering and community codes track in ATPESC. In: Allen G, Carver J, Choi SCT, Crick T, Crusoe MR, Gesing S, et al., editors. Workshop on Sustainable Software for Science: Practice and Experiences. No. 1686 in CEUR Workshop Proceedings. Aachen; 2016. Urn:nbn:de:0074-1686-8. Available from: [http://ceur-ws.org/Vol-1686/WSSSPE4\\_paper\\_17.pdf](http://ceur-ws.org/Vol-1686/WSSSPE4_paper_17.pdf).
- [16] Chue Hong N. Why do we need to compare research software, and how should we do it? In: Allen G, Carver J, Choi SCT, Crick T, Crusoe MR, Gesing S, et al., editors. Workshop on Sustainable Software for Science: Practice and Experiences. No. 1686 in CEUR Workshop Proceedings. Aachen; 2016. Urn:nbn:de:0074-1686-8. Available from: [http://ceur-ws.org/Vol-1686/WSSSPE4\\_paper\\_29.pdf](http://ceur-ws.org/Vol-1686/WSSSPE4_paper_29.pdf).
- [17] Dubey A, McInnes LC. The lifecycle of software for scientific simulation. In: Allen G, Carver J, Choi SCT, Crick T, Crusoe MR, Gesing S, et al., editors. Workshop on Sustainable Software for Science: Practice and Experiences. No. 1686 in CEUR Workshop Proceedings. Aachen; 2016. Urn:nbn:de:0074-1686-8. Available from: [http://ceur-ws.org/Vol-1686/WSSSPE4\\_paper\\_16.pdf](http://ceur-ws.org/Vol-1686/WSSSPE4_paper_16.pdf).
- [18] Queiroz F, Spitz R. Collaborative gamification design for scientific software. In: Allen G, Carver J, Choi SCT, Crick T, Crusoe MR, Gesing S, et al., editors. Workshop on Sustainable Software for Science: Practice and Experiences. No. 1686 in CEUR Workshop Proceedings. Aachen; 2016. Urn:nbn:de:0074-1686-8. Available from: [http://ceur-ws.org/Vol-1686/WSSSPE4\\_paper\\_6.pdf](http://ceur-ws.org/Vol-1686/WSSSPE4_paper_6.pdf).

- [19] Childers B, Davidson J, Graves W, Rous B, Wilkinson D. Active curation of artifacts is changing the way digital libraries will operate. In: Allen G, Carver J, Choi SCT, Crick T, Crusoe MR, Gesing S, et al., editors. Workshop on Sustainable Software for Science: Practice and Experiences. No. 1686 in CEUR Workshop Proceedings. Aachen; 2016. Urn:nbn:de:0074-1686-8. Available from: [http://ceur-ws.org/Vol-1686/WSSSPE4\\_paper\\_8.pdf](http://ceur-ws.org/Vol-1686/WSSSPE4_paper_8.pdf).
- [20] Ganguly D, Garrison W, Wilkinson D, Childers B, Lee A, Mosse D. Composing, reproducing, and sharing simulations. In: Allen G, Carver J, Choi SCT, Crick T, Crusoe MR, Gesing S, et al., editors. Workshop on Sustainable Software for Science: Practice and Experiences. No. 1686 in CEUR Workshop Proceedings. Aachen; 2016. Urn:nbn:de:0074-1686-8. Available from: [http://ceur-ws.org/Vol-1686/WSSSPE4\\_paper\\_33.pdf](http://ceur-ws.org/Vol-1686/WSSSPE4_paper_33.pdf).
- [21] Shende S, Malony A. Using TAU for performance evaluation of scientific software. In: Allen G, Carver J, Choi SCT, Crick T, Crusoe MR, Gesing S, et al., editors. Workshop on Sustainable Software for Science: Practice and Experiences. No. 1686 in CEUR Workshop Proceedings. Aachen; 2016. Urn:nbn:de:0074-1686-8. Available from: [http://ceur-ws.org/Vol-1686/WSSSPE4\\_paper\\_12.pdf](http://ceur-ws.org/Vol-1686/WSSSPE4_paper_12.pdf).
- [22] Sallai J, Iacovella C, Klein C, Ma T. Development of a software framework for formalizing forcefield atom-typing for molecular simulation. In: Allen G, Carver J, Choi SCT, Crick T, Crusoe MR, Gesing S, et al., editors. Workshop on Sustainable Software for Science: Practice and Experiences. No. 1686 in CEUR Workshop Proceedings. Aachen; 2016. Urn:nbn:de:0074-1686-8. Available from: [http://ceur-ws.org/Vol-1686/WSSSPE4\\_paper\\_35.pdf](http://ceur-ws.org/Vol-1686/WSSSPE4_paper_35.pdf).
- [23] Sufi S. The Software Sustainability Institute Fellowship programme; supporting the social side of research software. In: Allen G, Carver J, Choi SCT, Crick T, Crusoe MR, Gesing S, et al., editors. Workshop on Sustainable Software for Science: Practice and Experiences. No. 1686 in CEUR Workshop Proceedings. Aachen; 2016. Urn:nbn:de:0074-1686-8. Available from: [http://ceur-ws.org/Vol-1686/WSSSPE4\\_paper\\_27.pdf](http://ceur-ws.org/Vol-1686/WSSSPE4_paper_27.pdf).
- [24] Gesing S, Dahan M, Hayden LB, Lawrence K, Marru S, Pierce M, et al. The Science Gateways Community Institute - Supporting communities to achieve sustainability for their science gateways. In: Allen G, Carver J, Choi SCT, Crick T, Crusoe MR, Gesing S, et al., editors. Workshop on Sustainable Software for Science: Practice and Experiences. No. 1686 in CEUR Workshop Proceedings. Aachen; 2016. Urn:nbn:de:0074-1686-8. Available from: [http://ceur-ws.org/Vol-1686/WSSSPE4\\_paper\\_32.pdf](http://ceur-ws.org/Vol-1686/WSSSPE4_paper_32.pdf).
- [25] Ram K, Ross N, Chamberlain S. A model for peer review and onboarding research software. In: Allen G, Carver J, Choi SCT, Crick T, Crusoe MR, Gesing S, et al., editors. Workshop on Sustainable Software for Science: Practice and Experiences. No. 1686 in CEUR Workshop Proceedings. Aachen; 2016. Urn:nbn:de:0074-1686-8. Available from: [http://ceur-ws.org/Vol-1686/WSSSPE4\\_paper\\_13.pdf](http://ceur-ws.org/Vol-1686/WSSSPE4_paper_13.pdf).
- [26] Löffler F, Brandt S. A vision of computing in 10+ years. In: Allen G, Carver J, Choi SCT, Crick T, Crusoe MR, Gesing S, et al., editors. Workshop on Sustainable Software for Science: Practice and Experiences. No. 1686 in CEUR Workshop Proceedings. Aachen; 2016. Urn:nbn:de:0074-1686-8. Available from: [http://ceur-ws.org/Vol-1686/WSSSPE4\\_paper\\_9.pdf](http://ceur-ws.org/Vol-1686/WSSSPE4_paper_9.pdf).
- [27] Katz DS, Niemeyer KE, Smith AM, FORCE11 Software Citation Working Group. Software citation: Process, principles, and implementation. In: Allen G, Carver J, Choi SCT, Crick T, Crusoe MR, Gesing S, et al., editors. Workshop on Sustainable Software for Science: Practice and Experiences. No. 1686 in CEUR Workshop Proceedings. Aachen; 2016. Urn:nbn:de:0074-1686-8. Available from: [http://ceur-ws.org/Vol-1686/WSSSPE4\\_paper\\_31.pdf](http://ceur-ws.org/Vol-1686/WSSSPE4_paper_31.pdf).
- [28] Idaszak R, Tarboton DG, Yi H, Stealey M, Dash P, Couch A, et al. HydroShare: A case study in software engineering best practices and culture change for developing sustainable community software. In: Allen G, Carver J, Choi SCT, Crick T, Crusoe MR, Gesing S, et al., editors. Workshop on Sustainable Software for Science: Practice and Experiences. No. 1686 in CEUR Workshop Proceedings. Aachen; 2016. Urn:nbn:de:0074-1686-8. Available from: [http://ceur-ws.org/Vol-1686/WSSSPE4\\_paper\\_5.pdf](http://ceur-ws.org/Vol-1686/WSSSPE4_paper_5.pdf).
- [29] Hwang L, Bangerth W, Heister T, Kellogg L. ASPECT: Hackathons as an example of sustaining an open source community. In: Allen G, Carver J, Choi SCT, Crick T, Crusoe MR, Gesing S, et al., editors. Workshop on Sustainable Software for Science: Practice and Experiences. No. 1686 in CEUR Workshop Proceedings. Aachen; 2016. Urn:nbn:de:0074-1686-8. Available from: [http://ceur-ws.org/Vol-1686/WSSSPE4\\_paper\\_7.pdf](http://ceur-ws.org/Vol-1686/WSSSPE4_paper_7.pdf).
- [30] Goble C. A simple profiling framework for software user-producer maturity review. In: Allen G, Carver J, Choi SCT, Crick T, Crusoe MR, Gesing S, et al., editors. Workshop on Sustainable Software for Science: Practice and Experiences. No. 1686 in CEUR Workshop Proceedings. Aachen; 2016. Urn:nbn:de:0074-1686-8. Available from: [http://ceur-ws.org/Vol-1686/WSSSPE4\\_paper\\_2.pdf](http://ceur-ws.org/Vol-1686/WSSSPE4_paper_2.pdf).
- [31] Druskat S. A proposal for the measurement and documentation of research software sustainability in interactive metadata repositories. In: Allen G, Carver J, Choi SCT, Crick T, Crusoe MR, Gesing S, et al., editors. Workshop on Sustainable Software for Science: Practice and Experiences. No. 1686 in CEUR Workshop Proceedings. Aachen; 2016. Urn:nbn:de:0074-1686-8. Available from: [http://ceur-ws.org/Vol-1686/WSSSPE4\\_paper\\_20.pdf](http://ceur-ws.org/Vol-1686/WSSSPE4_paper_20.pdf).
- [32] Contrastin M, Danish M, Orchard D, Rice A. Supporting software sustainability with lightweight specifications. In: Allen G, Carver J, Choi SCT, Crick T, Crusoe MR, Gesing S, et al., editors. Workshop on Sustainable Software for Science: Practice and Experiences. No. 1686 in CEUR Workshop Proceedings. Aachen; 2016. Urn:nbn:de:0074-1686-8. Available from: [http://ceur-ws.org/Vol-1686/WSSSPE4\\_paper\\_23.pdf](http://ceur-ws.org/Vol-1686/WSSSPE4_paper_23.pdf).

- [33] Allen A, Aragon C, Becker C, Carver J, Chiş A, Combemale B, et al. I solemnly pledge: A manifesto for personal responsibility in the engineering of academic software. In: Allen G, Carver J, Choi SCT, Crick T, Crusoe MR, Gesing S, et al., editors. Workshop on Sustainable Software for Science: Practice and Experiences. No. 1686 in CEUR Workshop Proceedings. Aachen; 2016. Urn:nbn:de:0074-1686-8. Available from: [http://ceur-ws.org/Vol-1686/WSSSPE4\\_paper\\_15.pdf](http://ceur-ws.org/Vol-1686/WSSSPE4_paper_15.pdf).
- [34] Chue Hong N. Making it easier to understand research software impact. In: Allen G, Carver J, Choi SCT, Crick T, Crusoe MR, Gesing S, et al., editors. Workshop on Sustainable Software for Science: Practice and Experiences. No. 1686 in CEUR Workshop Proceedings. Aachen; 2016. Urn:nbn:de:0074-1686-8. Available from: [http://ceur-ws.org/Vol-1686/WSSSPE4\\_paper\\_30.pdf](http://ceur-ws.org/Vol-1686/WSSSPE4_paper_30.pdf).
- [35] van Hage WR, Maassen J, van Nieuwpoort R. Software impact measurement at the Netherlands eScience Center. In: Allen G, Carver J, Choi SCT, Crick T, Crusoe MR, Gesing S, et al., editors. Workshop on Sustainable Software for Science: Practice and Experiences. No. 1686 in CEUR Workshop Proceedings. Aachen; 2016. Urn:nbn:de:0074-1686-8. Available from: [http://ceur-ws.org/Vol-1686/WSSSPE4\\_paper\\_4.pdf](http://ceur-ws.org/Vol-1686/WSSSPE4_paper_4.pdf).
- [36] Allen G, Chen E, Idaszak R, Katz DS. Report on software metrics for research software. In: Allen G, Carver J, Choi SCT, Crick T, Crusoe MR, Gesing S, et al., editors. Workshop on Sustainable Software for Science: Practice and Experiences. No. 1686 in CEUR Workshop Proceedings. Aachen; 2016. Urn:nbn:de:0074-1686-8. Available from: [http://ceur-ws.org/Vol-1686/WSSSPE4\\_paper\\_26.pdf](http://ceur-ws.org/Vol-1686/WSSSPE4_paper_26.pdf).
- [37] Seidel EL, Allen G. Bringing techniques from software engineering into scientific software. In: Allen G, Carver J, Choi SCT, Crick T, Crusoe MR, Gesing S, et al., editors. Workshop on Sustainable Software for Science: Practice and Experiences. No. 1686 in CEUR Workshop Proceedings. Aachen; 2016. Urn:nbn:de:0074-1686-8. Available from: [http://ceur-ws.org/Vol-1686/WSSSPE4\\_paper\\_28.pdf](http://ceur-ws.org/Vol-1686/WSSSPE4_paper_28.pdf).
- [38] Emsley I, De Roure D. Sustaining the social: Connecting the lives of Drupal community. In: Allen G, Carver J, Choi SCT, Crick T, Crusoe MR, Gesing S, et al., editors. Workshop on Sustainable Software for Science: Practice and Experiences. No. 1686 in CEUR Workshop Proceedings. Aachen; 2016. Urn:nbn:de:0074-1686-8. Available from: [http://ceur-ws.org/Vol-1686/WSSSPE4\\_paper\\_34.pdf](http://ceur-ws.org/Vol-1686/WSSSPE4_paper_34.pdf).
- [39] Dongarra J, Hammarling S, Higham NJ, Relton SD, Valero-Lara P, Zounon M. Creating a standardised set of batched BLAS routines. In: Allen G, Carver J, Choi SCT, Crick T, Crusoe MR, Gesing S, et al., editors. Workshop on Sustainable Software for Science: Practice and Experiences. No. 1686 in CEUR Workshop Proceedings. Aachen; 2016. Urn:nbn:de:0074-1686-8. Available from: [http://ceur-ws.org/Vol-1686/WSSSPE4\\_paper\\_3.pdf](http://ceur-ws.org/Vol-1686/WSSSPE4_paper_3.pdf).
- [40] Breitwieser L, Bauer R, Meglio AD, Johard L, Kaiser M, Manca M, et al. The BioDynaMo Project. In: Allen G, Carver J, Choi SCT, Crick T, Crusoe MR, Gesing S, et al., editors. Workshop on Sustainable Software for Science: Practice and Experiences. No. 1686 in CEUR Workshop Proceedings. Aachen; 2016. Urn:nbn:de:0074-1686-8. Available from: [http://ceur-ws.org/Vol-1686/WSSSPE4\\_paper\\_14.pdf](http://ceur-ws.org/Vol-1686/WSSSPE4_paper_14.pdf).
- [41] Aldabjan A, Haines R, Jay C. How should we measure the relationship between code quality and software sustainability? In: Allen G, Carver J, Choi SCT, Crick T, Crusoe MR, Gesing S, et al., editors. Workshop on Sustainable Software for Science: Practice and Experiences. No. 1686 in CEUR Workshop Proceedings. Aachen; 2016. Urn:nbn:de:0074-1686-8. Available from: [http://ceur-ws.org/Vol-1686/WSSSPE4\\_paper\\_22.pdf](http://ceur-ws.org/Vol-1686/WSSSPE4_paper_22.pdf).
- [42] Barstow D, Duffey R, Smoliar S, Vestal S. An automatic programming system to support an experimental science. In: Proceedings of the 6th international conference on Software engineering. IEEE Computer Society Press; 1982. p. 360–366.
- [43] Towns J, Cockerill T, Dahan M, Foster I, Gaither K, Grimshaw A, et al. XSEDE: accelerating scientific discovery. Computing in Science & Engineering. 2014;16(5):62–74.
- [44] Post DE, Votta LG. Computational science demands a new paradigm. Physics today. 2005;58(1):35–41.
- [45] Vinter B, Kristensen MR, Lund SA, Blum T, Skovhede K. Prototyping for Exascale. In: Proceedings of the 3rd International Conference on Exascale Applications and Software. University of Edinburgh; 2015. p. 77–81.
- [46] Hwu WM, Chang LW, Kim HS, Dakkak A, El Hajj I. Transitioning HPC software to exascale heterogeneous computing. In: Computational Electromagnetics International Workshop (CEM), 2015. IEEE; 2015. p. 1–2.
- [47] Jones MB, Boettiger C, Mayes AC, Smith A, Slaughter P, Niemeyer K, et al.. CodeMeta: an exchange schema for software metadata. KNB Data Repository; 2016. Available from: <https://doi.org/10.5063/SCHEMA/CODEMETA-1.0>.
- [48] WSSSPE4 Software Best Practices for Undergraduates Working Group. WSSSPE4 Software Best Practices for Undergraduates Google Group; 2015. Accessed: 2015-12-11. <https://groups.google.com/forum/#!forum/wssspe4-software-best-practices-for-undergraduates>.
- [49] WSSSPE4 Software Best Practices for Undergraduates Working Group. WSSSPE4 Software Best Practices for Undergraduates GitHub Organization; 2015. Accessed: 2015-12-11. <https://github.com/orgs/WSSSPE4-undergraduate-course>.
- [50] WSSSPE Organization. WSSSPE Slack; 2016. Accessed: 2016-12-11. <https://wssspe.signup.team/>.