

## # Software Team Experiences and Challenges: A Report from Day 3 of the 2021 Collegeville Workshop on Scientific Software

### ### Contributors

- Anshu Dubey, Argonne National Laboratory, GitHub: adubey64
- Nur Fadel, Swiss National Supercomputing Centre, GitHub: nuraiman
- Elaine Raybourn, Sandia National Laboratories, GitHub: elaineraybourn
- Ben Cowan, Pilot AI, GitHub: benc303
- Sarah Osborn, Lawrence Livermore National Laboratory, GitHub: osborn9
- Stan Tomov, University of Tennessee, GitHub: stomov
- Sarah Knepper, Intel Corporation, GitHub: sknepper
- Charles Ferenbaugh, Los Alamos National Laboratory, GitHub: cferenba
- Vadim Dyadechko, ExxonMobil, GitHub: **tbd**
- Ulrike Meier Yang, Lawrence Livermore National Laboratory, GitHub: ulrikeyang
- David Moulton, Los Alamos National Laboratory, GitHub: jd-moulton
- Han Yong Wunrow, Sandia National Laboratory, GitHub: hwunrow
- Benjamin Sims, Los Alamos National Laboratory, GitHub: bhsims
- Jay Lofstead, Sandia National Labs, GitHub: gflofst
- Johanna Cohoon, University of Texas Austin, GitHub: jlcohoon
- Reed Milewicz, Sandia National Laboratories, GitHub: rmmilewi
- Todd Munson, Argonne National Laboratory, GitHub: tmunson
- Sam Yates, Swiss National Supercomputing Centre; GitHub: halfflat

### ### Editors:

- Daniel S. Katz, University of Illinois at Urbana-Champaign, GitHub: danielskatz
- Pat Quillen, Mathworks, GitHub: **tbd**
- Michael A. Heroux, Sandia National Laboratories, GitHub: maherou

### ## Background: The Collegeville Workshop Series

The Collegeville Workshop Series on Scientific Software is intended to bring together three communities of scientific software contributors: academia, industry and laboratories. While there are existing exchanges between these communities, we are dedicated to improving awareness of common needs, unique contributions and career paths that span these communities. Workshop contributions include short white papers, video interviews, and a three-day live event with panels, small-group discussions, and teatime sessions for themed conversations.

### ## Collegeville 2021 Theme: Software Teams

The Collegeville 2021 theme was scientific software teams. The first day of live discussion focused on software team definitions and challenges; the second day on technical strategies for improvement; and the third on cultural approaches for improvement.

Little scientific software is developed by individual scientists. Instead, teams with diverse skills collaborate on producing and using software to advance scientific discovery and understanding. Understanding how teams function and how teamwork can be improved represent two of the

frontiers in improving the impact of software on science. Software team skills and cultures can vary. A scientific software team will have science domain experts but increasingly has expertise in computer science, mathematics and software engineering. As we increase our focus on understanding and improving software teams, we see growing value in including expertise in social and cognitive sciences.

### ## Workshop Small Group Discussions

During each of the live discussions, small groups gathered to discuss the topic of the day, creating a shared notes file. This blog is the third in a series of three that summarize the output from these discussions

### ## Day 3: Cultural Approaches to Improved Software Teams

After talking about teams and technical approaches to improving them in the first two days, we spent Day 3 talking about cultural approaches to improving teams. Two-thirds of the discussion participants were from research labs, with the others split between universities and industry. 18 participants chose to receive attribution for their contributions.

### ### Summary of participant software team experiences

Discussion participants from labs and universities represented a number of well-known open-source software projects. Others came from industry, representing the oil & gas sector, and technical computing software providers. Finally, a number of participants were from the social and cognitive science communities, where their domain of study includes scientific software teams and developers.

In aggregate, the discussion participants have hundreds of years of collective software development and software project leadership experience, ranging from individual contributors to leaders of large, multi-team efforts. Together, these participants provide software to thousands of users throughout the world. Participant experiences further represent approximately four decades of focused study of scientific software teams via methodologies from the social and cognitive sciences.

### ### Key Challenges

In the remainder of this report, we summarize the key challenges identified during the small group discussions. The detailed notes from these discussions are available on the [Collegeville 2021 Workshop website](<https://collegeville.github.io/CW21>).

**\*\*Motivating team members:\*\*** We need to use a combination of multiple approaches to motivate team members, recognizing that each person may have a different set of both intrinsic and extrinsic motivations for their actions and activities. Carrots (e.g., peer recognition, better research results) and sticks (e.g., journal requirements, funding requirements) absolutely influence culture and individual behavior, but a carrot to one person can be a stick to another. Better understanding researchers' motivations can help effect culture change.

**\*\*Motivating the team:\*\*** It's important to make sure that the team's work is recognized. We need to develop good ways of judging the overall success of a software project. Citations can be part of the answer, but this is not a complete answer since some users don't write public papers about their

work and some lower-level software is typically not cited in papers. Similarly, asking the users is not a complete answer, since the team typically doesn't know all of the users. Collecting information from a variety of sources is useful, like emails/private communications or user group meetings. Once these metrics are gathered, it's important to ensure that people higher up in institutions as well as funding agencies use them.

**\*\*Affecting cultural change:\*\*** Organization culture as set from the top (the lab director, provost, etc.) plays a strong role in the evolution of organizational culture over time, but this alone is not sufficient; affecting change also takes bottom-up effort and a willingness to change in the middle. We need to have buy-in at both ends to hope to achieve lasting cultural change. New ways of thinking and doing have to overcome the inertia of the status quo, but when certain teams demonstrate visible successes, others are likely to follow (e.g., version control becoming the norm among scientific software developers). Change can also come from within a team: new members can introduce new and good culture to a group.

**\*\*Building a culture of diversity and respect:\*\*** Different views and backgrounds can provide a benefit, but for this to happen, we need to listen to and respect each other. There are known methods to help us do this, such as team building activities that involve informal discussions, active listening, and empathy development (e.g., small-team design challenges and games that have nothing to do with one's regular job). These activities really help by adjusting people's perspectives and open their mind to relate to other people's needs and perspectives.

**\*\*Recognizing that all jobs are important:\*\*** This includes team members such as communicators, technical writers, sysadmins, software engineers, and scientists, and who are both full- and part-time. While some work may be seen by some as "boring" (documentation), and other work may be seen as "exciting" (speeding up a function), all of the team's work is generally needed for a project to be successful. In particular, work that is seen as less exciting is less likely to be done or done well. For example, documentation is still a weak link as it's often not directly funded, although it is becoming a recommended policy in more research software teams. In order to increase usage of documentation, teams need to first be provided the resources (e.g., weekly protected time for documentation) so that they feel supported in incorporating documentation practices. A general approach that is applicable to other cultural approaches is to lead by example. If the team lead displays the benefits and quick wins that documentation can lead to, others may be more open to incorporate this practice into their daily schedule. Balancing publication and milestones versus documentation continues to be a challenge. Public recognition (e.g., awards, badges) of good documentation could be a potential solution.

**\*\*Value the team's time:\*\*** Make sure that communications happens in multiple ways that best meet the needs and style of the team members. In particular, limit meetings to those that are necessary, and make sure the time is used effectively through meeting best practices, e.g., give sufficient notice before calling or cancelling a meeting, having an agenda in advance that allows participants to add to it, agree on meeting rules, etc.

**\*\*Recognizing the value of RSEs and their work:\*\*** This can be done through a combination of top-down and bottom-up activities. Top-down examples include creation of departments and positions (with appropriate titles). Bottom-up examples include acceptance/roles on teams, teams

value of reproducibility and sustainability in their code base, advancing processes for recognition (DOIs, awards), etc.

**\*\*Valuing and enabling learning:\*\*** Researchers are continually reinforcing and learning the skills and values of their profession. Likewise, improving team performance is a long-term project that builds on current practices. When extrinsic motivations (e.g., employee evaluations) are not aligned with intrinsic motivations (e.g., a desire for better team communication), teams are unlikely to make changes. Despite such obstacles, there are also times when learning would be best facilitated, such as when a new member is joining the team. Applying insights from psychology might help us identify those opportunities where introducing new practices can yield the greatest return. Taking such an approach would mean not only teaching best practices, but doing so under the most favorable circumstances.

### **## Final Remarks from Day 3 Discussions**

Day 3 discussions at the Colledgeville 2021 Workshop represent the input of a diverse and experienced group of scientific software developers and leaders, and their colleagues from the social and cognitive sciences. We hope that the challenges summarized in this blog resonate with the reader and help the scientific software community when prioritizing efforts to improve the quality and impact of software in the pursuit of scientific discovery.