

Lighting Talk - ASPECT: Hackathons as an Example of Sustaining an Open Source Community

Lorraine J. Hwang
Computational Infrastructure for Geodynamics
University of California, Davis
Davis, California USA
ljhwang@ucdavis.edu

Timo Heister
Clemson University
Clemson, South Carolina USA
heister@clemson.edu

Wolfgang Bangerth
Colorado State University
Fort Collins, Colorado USA
bangerth@colostate.edu

Louise H. Kellogg
Computational Infrastructure for Geodynamics
University of California, Davis
Davis, California USA
lhkellogg@ucdavis.edu

Abstract—Successful scientific open source projects must not only have good technical but also good social structure. Hackathons can be used to engage and grow a software community by providing geographically distributed communities as well as novices and experts an opportunity to work side-by-side. The community built around the mantle convection code ASPECT (Advanced Solver for Problems in Earth ConvecTion) has held multi-day hackathons for the past 3 years. Held over approximately 10 days, these hackathons build capacity by fostering new users, providing access to domain experts, and promoting new collaborations.

Index Terms—Hackathon, open source software, software sustainability, ASPECT.

I. INTRODUCTION

The ecosystem surrounding a successful scientific open source software package combines both social and technical aspects. Much thought has been given to the technology side of writing sustainable software for large infrastructure projects and software libraries, but less about building the human capacity to perpetuate scientific software used in computational modeling. Scientific software is challenging as it requires more advanced and specialized skills on top of an interest in software development. This often requires the community to cope with a diverse set of skills. It also differs from commercial software in the way its continued development is funded, as well as a frequent lack of formal supervisor/supervisee relationship among many participants.

II. FOSTERING SOFTWARE COMMUNITIES

A software package does not thrive without a community that engages in roles as maintainers, contributors, and users. A person may engage in one or multiple roles. The community is dynamic, as roles evolve over time and as members arrive and depart from the community. Users are attracted to a

software package when they perceive it as usable and applicable. They become contributors, reporting bugs, submitting patches, and implementing new features as the nature of their problem outpaces the existing development efforts. Becoming a maintainer requires different technical and social skills as they support the infrastructure. This includes not only reviewing code but also growing and nurturing the user base, eventually attracting new maintainers.

From our experience running multiple open source projects, one of the most effective forms for fostering community is regular multi-day hackathons. Hackathons are events, typically lasting several days, in which a large number of people meet to engage in collaborative software development. Scientific hackathons are distinctly different from commercial hackathons. They bring together a group of science domain users and scientific software contributors to make progress on a specific software package. Innovation comes through the chance to work within established and newly formed collaborations. Especially in the domain sciences with small, international communities, hackathons give geographically distributed scientists an opportunity to connect face-to-face with people who previously were a faceless “@”. They foster lively discussions amongst scientists with different expertise, promote new collaborations, and increase transparency to common problems in the both the technical and scientific aspects of code development.

In practice, hackathon participants range from novice to expert users, predominantly from the academic sector. At the beginning of the hackathon, leaders establish project specific conventions for development, demonstrate the workflow for code contributions, and review relevant technical skills (version control, etc.). They lower the bar to entry by suggesting easy initial tasks accessible to novice users. These starter projects can be something as simple as reformatting something, easy bug fixes, adding to cookbooks, improving

documentation, or writing up improved installation instructions for a particular platform. Sometimes this takes novice users working side-by-side with advanced users or a simple shout across the room. Maintainers set aside ego and encourage users by providing supportive comments, accepting “less than perfect” initial commits, and in some cases may rewrite contributions as one method of mentoring. The goal of a hackathon over the course of several days is that all participants become comfortable contributing to the repository, such that when the hackathon ends, they continue to contribute.

III. EXAMPLE FROM THE ASPECT COMMUNITY

Using this model, we specifically comment on our experience with ASPECT (Advance Solver for Problems in Earth ConvecTion), an open source, parallel, extensible finite element code to simulate thermal convection, that began development in 2011 as a community driven open source project under the Computational Infrastructure for Geodynamics (CIG). It is based on the deal.II finite element software library. Initial development of ASPECT was undertaken by a faculty and two postdocs and today has grown to 33 contributors. The bulk are early career, graduate students or postdocs spread over three continents who are domain experts as well as computational scientists. ASPECT’s first hackathon in 2014 hosted 14 participants. From 2014 to 2015, the project removed one and added two maintainers, allowing the number of hackathon participants to increase to 22 in 2016. Over the 3 years hackathons have been offered, 58% of the participants have attended more than one hackathon, providing stability and continuity. Of these, 21% have participated all 3 years, representing 58% of the inaugural participants. All participants make contributions during the event, strongly encouraged by the organizers; 52% of participants continue to make code contributions after the event.

IV. GOING FORWARD

In running hackathons, we have learned that they should:

- be multi-day events allowing participants to make significant progress on their research problem (~10-days) and to get to know each other well;
- have participants with multidisciplinary expertise;
- include a mix of expert and novice users to provide mentoring;

- have dedicated and experienced code reviewers provide immediate feedback and keep pace with workflow, even if that means that they will not contribute much own work during the hackathon;
- have daily group updates communicating work plans and allowing collaborations to form freely;
- be organized to promote group interactions; and
- enforced free time to prevent burn out.

The participants in our hackathons are still young (median age 29) and the project is rapidly growing. Leaders currently include early career faculty, tenured faculty, and postdoctoral fellows. They are motivated by believing science and society are better served by open source software. As there are no established paths for those vested heavily in developing scientific software, we anxiously watched as this cohort’s careers advance. Interests lie in both the traditional faculty path and paths focused on software development, for both of which working on academic software may not be the best path toward long-term and stable employment. The need to create career paths and recognition for contributions to software is widely recognized. We have given careful consideration in this community towards providing citable metrics for all participants, and for fostering environments in which younger scientists can show their skills and make connections that will be useful for the next stage of their careers.

CIG hopes to build other software communities on this model, but anticipates each to bring their own unique challenges. The origins of projects, the leadership, size of community and software type varies widely within the group of projects CIG encompasses. Codes interested in holding hack events range from software developed by a single early career scientist to ones that were initially developed by a group of graduate students and/or postdocs. The potential user and contributors base for one code is small, the other large; both groups are internationally distributed. In common, we foresee that retention and depth of talent is an issue that will require different tweaks to the above hackathon model.

V. ACKNOWLEDGMENT

We thank the ASPECT hackathon participants and leaders for their contributions to the community. We thank National Science Foundation award NSF-0949446 for their support and the support of the code ASPECT which is hosted by CIG (www.geodynamics.org).