Software Training in HEP

Sudhir Malik[1,*], *Samuel* Meehan[2], *Kilian* Lieret[3], *Meirin Oan* Evans[4], *Michel H.* Villanueva[5], *Daniel S.* Katz[6], *Graeme A.* Stewart[2], and *Peter* Elmer[7]

[1]University of Puerto Rico Mayaguez, USA
[2]CERN, Switzerland
[3]Ludwig Maximilian University of Munich, Germany
[4]University of Sussex, Brighton, UK
[5]University of Mississippi, Oxford, MS, USA
[6]University of Illinois, Urbana, IL, USA
[7]Princeton University, Princeton, NJ, USA

Abstract. Long term sustainability of the high energy physics (HEP) research software ecosystem is essential for the field. With upgrades and new facilities coming online throughout the 2020s this will only become increasingly relevant throughout this decade. Meeting this sustainability challenge requires a workforce with a combination of HEP domain knowledge and advanced software skills. The required software skills fall into three broad groups. The first is fundamental and generic software engineering (e.g. Unix, version control, C++, continuous integration). The second is knowledge of domain-specific HEP packages and practices (e.g., the ROOT data format and analysis framework). The third is more advanced knowledge involving more specialized techniques. These include parallel programming, machine learning and data science tools, and techniques to preserve software projects at all scales. This paper discusses the collective software training program in HEP and its activities led by the HEP Software Foundation (HSF) and the Institute for Research and Innovation in Software in HEP (IRIS-HEP). The program equips participants with an array of software skills that serve as ingredients from which solutions to the computing challenges of HEP can be formed. Beyond serving the community by ensuring that members are able to pursue research goals, this program serves individuals by providing intellectual capital and transferable skills that are becoming increasingly important to careers in the realm of software and computing, whether inside or outside HEP.

1 Introduction

Particle physics in the coming decades will continue to explore the fundamental workings of the universe. This requires upgrading existing major facilities like the Large Hadron Collider (LHC) to the High Luminosity LHC (HL-LHC, https://home.cern/science/accelerators/high-luminosity-lhc) and building new facilities like the Long-Baseline Neutrino Facility (LBNF) [1] and Deep Underground Neutrino Experiment (DUNE, https://lbnf-dune.fnal.gov/), among many others. To realise the full physics potential of this work, an equivalent investment must be made into the software required to collect, process, and analyse the

*e-mail: sudhir.malik@upr.edu

deluge of the data recorded. Recent efforts like the HSF (https://hepsoftwarefoundation.org/) and IRIS-HEP (https://iris-hep.org/) are facilitating cooperation and common efforts in HEP software and computing worldwide to develop state-of-the-art software cyberinfrastructure required to meet the challenges of the upcoming HEP experiments' data intensive scientific research. The rapid evolution of computing technology with a concomitant increase in the complexity of software algorithms for analysis requires developers to acquire a broad portfolio of programming skills in order to enable future discoveries. It is critical that all stakeholders across HEP make a major effort to provide a strong foundation for new researchers entering the field. The researchers must be brought up to date with new software technologies, concurrent programming and artificial intelligence, as well as maintaining, improving, and sustaining the existing HEP software. However, young researchers graduating from universities worldwide currently do not receive adequate preparation in the modern computing practices to respond to growing needs of the above experimental challenges. A community white paper [2] outlined the initiatives to address training needs and issues that need to be taken into account for these to be successful. In the last two years, a software training program has been developed under the umbrella of HSF collectively with IRIS-HEP and FIRST-HEP (https://first-hep.org/) and partnering with the Carpentries (https://carpentries.org/). It provides a training path from a researcher's first steps through more regular and active contribution. The specific goals of this group have been focused on two specific efforts: (1) developing and introductory HEP software curriculum and (2) teaching this curriculum to HEP scientists. Thus far, over 1000 people in HEP and related computing areas have been trained. This paper describes the activities, the curriculum and future directions of HEP software training.

2 Organisation

The organisation of training is an objective of the HSF Training Working Group. The group, led by three co-conveners, engages with educators and facilitators from different experimental collaborations and initiatives such as IRIS-HEP and FIRST-HEP and the Software Carpentries. It prepares training material and coordinates activities for the common good with a strong community of both instructors and participants, and with a feeling of community ownership. This vision is centered around the philosophy that people are the key to successful software. The training focuses on common software material across HEP and ranges from basic core software skills needed by everyone to advanced training required by specialists in software and computing. The training style is student-centric and experiment agnostic, with reusable material that is openly accessible. The style and pedagogy of the training material is inspired by the the Carpentries. The training group has weekly public meetings [3] to plan and assess progress. This is where ideas and proposals are discussed and events planned. Live notes are maintained and the meeting is done remotely using Zoom with ease for everyone to contribute. Training events are announced via several email lists and events can be accessed via Indico [4]. Training material is hosted in GitHub [5] and can be accessed via lesson websites [6], of which there is one per individual training topic. The procedure to request and organise a training is documented [7].

As creating training material and teaching requires a lot of commitment and time, it is of great importance to acknowledge the efforts of everyone involved. Currently this is is mostly achieved by listing helping community members on the pages of the relevant training and on a central community page. Providing financial incentives and compensations could further increase both quantity and quality of the training activities, but currently no direct funds are available to compensate for time invested in preparation of material. Finally, there are Blueprint workshops [8] and hackathons [9] organised to brainstorm the curriculum, content

development, and new topics for training. The travel cost for educators and video captioning of training material have been supported by the IRIS-HEP and FIRST-HEP.

3 Curriculum

An initial survey of software and training needs of the HEP community was conducted in February of 2019 [10]. This was followed by development of "prototype" course modules and pilot training events from which feedback from participants was solicited concerning the course content. Based on the surveys and the experiences gathered at the events, the course structure was extended into a full curriculum and the guidelines for the development of the modules and the procedure for training events was formalized. Each training module is independent from the others (but for some clearly marked requirements), such that students can prioritize certain skills before others. This is especially important in academia because students are often expected to directly work towards scientific results with minimal time given for acquiring software knowledge or best practices, oftentimes leading to poor quality software.

The most basic skill set (Unix shell, python, git) is covered by modules directly developed by Software Carpentry. A large module that covers the basics of modern (!) C++ is currently in development and other modules focusing on development in C++ such as CMake have already been taught with great success.

This is complimented by a series of broader software engineering topics, such as continuous integration on the example of both GitHub Actions and GitLab CI. These modules are also part of a broader category of skills particularly relevant for analysis preservation, for which modules covering domain specific software such as reana are in development.

A lesson on machine learning and a lesson specifically targeting machine learning with GPUs started a section on data analysis techniques. Similarly important are HEP specific tools, especially ROOT and integrations such as uproot.

Finally development is ongoing for modules that cover advanced topics that are important for students striving to become core developers such as code documentation, performance optimization and parallel programming.

The module list and the material evolves continuously depending on the input from participants and person-power available; it is open source and welcomes merge requests from any interested stakeholders. The entry barrier required to contribute to the material is fairly low, as all lessons are written in the easy-to-learn Markdown format, so only knowledge of git is required to contribute.

4 Training

During the initial period of training, 150 people received "introductory" software skills training at Fermilab (FNAL), Argonne National Lab (ANL), Lawrence Berkeley Lab (LBNL), and CERN [11–14]. Over 50 people received more advanced "computing bootcamp" training at the CoDaS-HEP school (http://codas-hep.org/). National labs are the hub of the HEP community and provide an environment where it is easier to reach a diverse population of participants with good infrastructure for in-person training. However, the COVID-19 pandemic necessitated rapid adjustment to virtual platforms that evolved throughout the course of 2020 as we gained experience. To date, nearly 100 educators have taught over 1000 participants in about a dozen training events. The educators are typically volunteers from various HEP experiments and in many cases were participants in previous training events. By actively engaging participants in this way and including them throughout the training community, we

aim to sustainably nurture a culture of intentional learning that is not yet wholeheartedly embraced in HEP. Valuable lessons have been learnt regarding in-person and virtual training. While in-person training limits participants to a few dozen due to costs, and a long lead time for planning logistics of travel, room booking, and space constraints, it also offers opportunities for active and efficient engagement of participants and community building. However, this can also make in-person training a bit exclusive. Participants need extra preparation time to travel to the venue. Hosts have to book specially arranged/equipped rooms with multiple projectors and screens to simultaneously project teaching materials and slides. Our in-person events have been managed by around five educators to allow for the "hands-on" aspect to be successful, and need a large commitment of their time as well; they cannot just present their material and leave.

On the other hand, virtual training events allow for a broader reach of participant attendance with registrations upwards of 300 for single events. These events are a considerably more equitable service to the community by allowing individuals to participate fully in the event regardless of funding, given that there are no travel costs. Furthermore, because the teaching materials are fully preserved via lesson creation and YouTube videos beforehand, an inability to attend during the scheduled time does not considerably degrade learning. Finally, these video materials are captioned to ensure inclusivity to those with hearing impairments, which itself is considerably more economical when comparing the cost of a hired sign language interpreter (~$1000/day) to that of captioning videos for a week-long event (~$50/day). Organizationally, the lead time to plan a training is reduced considerably due to reduced logistics (i.e., room booking is not needed). The disadvantage, however, is that it is difficult for educators and participants to interact closely - you just can't recreate the in-person environment on Zoom. Educators/mentors/participants have to plan and resolve in the best possible way their spread across time zones. It is also challenging to keep everyone engaged and on the same page due to the pervasive culture of "multi-tasking" within HEP. Due to this issue, although initial registration for these events are very high, the actual attendance at these events is typically only 50% of those who have registered. This is attributed to the combination of reduced barriers and the widespread desire to engage in this training, but an institutional culture that dedicated professional development is not something that should detract from research time and so standard activities are typically prioritized instead. However, it should be noted that this does not mean that there is a lesser degree of learning occurring at the training event. Tools like Mattermost, discord, and Slack have been effectively deployed for asynchronous communication, both during and after the event.

There is very clear and detailed guidance for anyone willing to host, request or organize a training while staying aligned with the approach, philosophy and code of conduct of the HSF-Training group so as to make the tools and techniques that are developed persistent, reuseable, and broadly accessible. If the community has new ideas on training modules, suggestions and improvements, there are opportunities to discuss them at the weekly training meetings and brainstorming sessions.

5 Feedback

Feedback is required for us to evaluate if we are effectively facilitating learning and to ensure the success of future training. Every training has a pre- and post- survey to collect self-reported feedback from the participants. They include a set of baseline questions pertaining to demographics and questions to assess the quality and method of training. These questions can be adapted to the nature and topic of each training. Additionally, it has become the status-quo to organize a "post-mortem planning" session among the educators to discuss the successes and failures of the training activity. This typically occurs after completion of the results of the

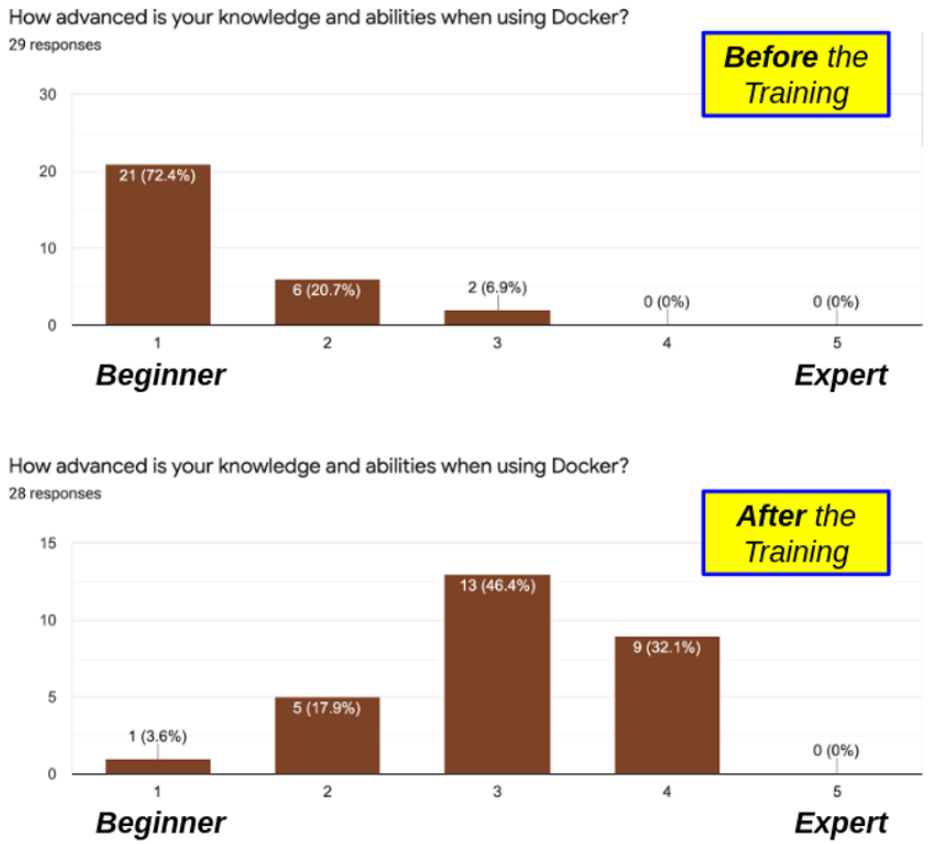

Figure 1. The self-reported pre- and post- training level of knowledge on the topic of Docker (a software container technology)

post- (and pre-) workshop surveys, which guide the discussion. Finally, a short presentation about this training experience is presented at the HSF-Training weekly meeting and/or at the HSF all-working-groups planning meeting. Figure 1 shows feedback on a training event involving containerization with Docker [15]. It is clear that training made a difference to the skills researchers had. However, we have recognized that this type of a "learning evaluation" does not fully encompass the impact that our training has on the research community, as it only probes the perceived and self-reported learning of a skill. Instead, what is needed is a survey that is conducted at a date sufficiently later than the training to understand whether the learned skill is being applied in the research context.

6 Community

A workforce trained in a range of software skills with a combination of HEP domain knowledge and advanced software skills is the critical ingredient from which solutions to computing challenges can grow. To fulfill our mission, we rely on active community members [16] to support us in various ways, all of which, at this time are carried out on a voluntary basis but typically with dedication and enthusiasm; our community is intrinsically motivated. The

members come from different HEP collaborations. This diversity adds great value to the training, as each brings their own flavor of experience from a different computing environment with a common goal to create, teach, and sustain a common set of skills across HEP. Recently, the Nuclear Physics community also became interested in our work and we hope that they will benefit from it given their similar software needs. While training can impart and sharpen participants' software skills, it also gives tutors an opportunity to develop their pedagogical skills and professional profile. Given that the two-thirds of the HEP workforce eventually works outside of HEP, such as in the software industry and in data science, the HSF training will make a difference in robust preparation for such careers in terms of software knowledge and experience, and will enhance their chances of employability. The skills learnt, like Python, machine learning and data analysis, align with the needs of industry and strengthen the job profile of a physicist to work in industry. The HSF community page carries links to the profile of each tutor that contributes to the training and serves as public proof of their capability and skills.

7 Sustainability

Sustainable software [17] gives HEP software developers an important skill that is essential to careers in the realm of software, inside or outside HEP. A sustainable training program is key to pursuing this goal. While continuing the existing work, it will be essential to spread the training events and training expertise geographically to keep the costs low and move to an online training model to reduce financial burdens that accompany in-person training. In parallel, it is important that as the curriculum grows it begins to include material specifically aimed at making software sustainable. Training should be structured so that a minimal set of people are needed to keep the training infrastructure running and identify additional costs for additional events. We need a long term funding model and one that migrates training from voluntary work rewarded by a picture on a webpage to tangible incentives, financial or otherwise. Mentoring the trainers/mentors to grow the community is an important aspect of sustaining the workforce. In addition, giving them recognition can keep the community vibrant, motivated and help in careers. In this way, the community should recognize the broader value in our software training which prepares a workforce to solve computing challenges that are essential to advance our field and society at large.

To be able to lead software training across HEP and related communities over the long run, we must focus on building a community of individuals who are educators, mentors, facilitators, participants (learners), instructors, experts and hosts. At the center must be a core team (supported directly by the HEP community for a long term) whose main focus is to support the overall mission of HEP software training. To achieve this, we need to build regional and local mentorship and leadership to train HEP communities that are guided and supported by the core team. Specifically, while we have started the following set of activities, we need to scale up by:

- Engaging more HEP labs, institutes, and universities in this endeavour.
- Promote equity, diversity, inclusion and accessibility in participation across HEP communities and be mindful of under-resourced institutions in different geographical regions.
- Establish a mechanism to get feedback from our communities and improve the training.
- Ensure that our core team and volunteers are afforded opportunities to grow professionally and have career paths.
- Explore ways to manage a financial support model to share costs in the long term.

8 Broader Impacts

HSF-led training is multilayered with a basic HEP software curriculum progressing to HEP specific physics tools. Integrated with this is a growing outreach program that is essential to building an influx of software workforce and training young minds, catching them early in their educational development. In addition, pursuing training in computing skills needed for researchers, several outreach events are organised on introducing python programming to K-12 teachers [18] under IRIS-HEP and FIRST-HEP. The teachers can turn this into a classroom experience for their students where physics, astronomy and math courses can have problem solving components which integrate programming with python. In outreach events the teachers analyze physics data with Python programming using Google Colab and interpret their findings by making plots. Workshops teaching the basics of Machine Learning to school teachers are also organised [19]. We plan to scale this experience by partnering with other stakeholders in HEP outreach, for example, Quarknet (https://quarknet.org/), which already has a well developed network of teachers and schools taking part in HEP outreach programs.

9 Outlook

HSF and IRIS-HEP are creating software training and ensuring sustainability of software in HEP for years to come. The training material is open source and shared publicly via GitHub. This allows anyone to join the discussion and make contributions by proposing changes, thereby continuously improving the available material guided by continual feedback solicited from those engaged with the material and their implementation in training events. Finally, we have established a growing community of educators to broadly promote a culture within HEP that goes beyond valuing software skills, but also values the teaching of those skills to others. In doing so, we aim to foster a more active, inclusive, and diverse scientific community. By leading software training across HEP and related communities, we will be able to meet the challenges in the field and beyond.

Acknowledgments

This work is supported in part by National Science Foundation Cooperative Agreement OAC-1836650 and grants OAC-1829707 and OAC-1829729.

References

[1] V. Papadimitriou, K. Ammigan, J.A.J. au2, K.E. Anderson, R. Andrews, V. Bocean, C.F. Crowley, N. Eddy, B.D. Hartsell, S. Hays et al., *Design of the lbnf beamline*, https://arxiv.org/abs/1704.04471 (2017), 1704.04471

[2] HEP Software Foundation, D. Berzano, R.M. Bianchi, P. Elmer, S.V. Gleyzer, J. Harvey, R. Jones, M. Jouvin, D.S. Katz, S. Malik et al., *HEP software foundation community white paper working group - training, staffing and careers*, https://arxiv.org/abs/1807.02875 (2019), 1807.02875

[3] *HSF training and careers working group meetings*, https://indico.cern.ch/category/10294/

[4] *HSF training events*, https://indico.cern.ch/category/11386/

[5] *HSF training and educational material repository*, https://github.com/hsf-training

[6] *Towards a HEP software training curriculum*, https://hepsoftwarefoundation.org/training/curriculum.html

[7] *How to host an HSF training event*, https://hepsoftwarefoundation.org/training/howto-event.html

[8] *IRIS-HEP and HSF training blueprint meeting*, https://indico.cern.ch/event/889665/ (2020)

[9] *The HSF training hackathon*, https://indico.cern.ch/event/997485/ (2021)

[10] D. Lange, *Selected results from HSF training survey*, https://indico.cern.ch/event/759388/contributions/3315848/attachments/1816082/2968198/training_how2019.pdf (2019)

[11] *Software carpentry workshop (Fermilab)*, https://indico.fnal.gov/event/20233/ (2019)

[12] *FIRST-HEP/ATLAS training (Argonne)*, https://indico.cern.ch/event/827231/ (2019)

[13] *FIRST-HEP/ATLAS training (LBNL)*, https://indico.cern.ch/event/827232/ (2019)

[14] *Software carpentry workshop (CERN)*, https://indico.cern.ch/event/834411/ (2019)

[15] *HSF virtual Docker training*, https://indico.cern.ch/event/934651/ (2020)

[16] *The HSF training community*, https://hepsoftwarefoundation.org/training/community.html

[17] D.S. Katz, S. Malik, M.S. Neubauer, G.A. Stewart, K.A. Assamagan, E.A. Becker, N.P. Chue Hong, I.A. Cosden, S. Meehan, E.J.W. Moyse et al., *Software sustainability & high energy physics*, https://doi.org/10.5281/zenodo.4095837 (2020)

[18] *Data analysis for STEM teachers*, https://indico.cern.ch/event/927162/ (2020)

[19] *Machine learning basics for STEM teachers*, https://indico.cern.ch/event/998732/ (2021)