Snowmass2021 Letter of Interest : Coherent Vision for Enabling Software Training in HEP

Daniel S. Katz[8], Clemens Lange[1], Kilian Lieret[7], Sudhir Malik[2], Samuel Meehan[1], Kevin Nelson[6], Robin Newhouse[5], Meirin Oan Evans[10], Adam Parker[11], Mason Proffitt[9], Eduardo Rodrigues[14], Amber Roepe[3], Giordon Stark[4], Graeme Stewart[1], Sadhana Verma[12], Leonora Vesterbacka[13], and Claire David[15]

[1]CERN
[2]University of Puerto Rico
[3]The University of Oklahoma
[4]University of California Santa Cruz
[5]University of British Columbia
[6]University of Michigan
[7]LMU Munich
[8]University of Illinois Urbana-Champaign
[9]University of Washington
[10]University of Sussex
[11]California State University East Bay
[12]Indian Institute of Technology
[13]New York University
[14]University of Liverpool
[15]York University

Thematic Areas:

☐ CommF1: Applications & Industry
☐ CommF2: Career Pipeline & Development
☐ CommF3: Diversity & Inclusion
■ CommF4: Physics Education
☐ CommF5: Public Education & Outreach
☐ CommF6: Public Policy & Government Engagement
■ Computing Frontier

Contact Information:
Samuel Meehan: *samuel.meehan@cern.ch*, Sudhir Malik: *Sudhir.Malik@cern.ch*

Abstract: Computing and software development skills are continually more critical for success in the field of HEP. However, our approach to cultivating these skills within our community is piece-meal and lacks a uniform pedagogy. We propose here to create a more robust mechanism/program by which individuals entering HEP can learn the essential skills they need to succeed, independent of their formal education. A central aspect of this is creating a ”Core Computing Curriculum”.

Software skills are an integral part of the toolkit of any successful HEP physicist. These skills are at the center of every experiment from fully leveraging the capabilities of the detectors and hardware to transforming the dataset collected with these detectors into meaningful interpretations and, eventually, physics publications. Furthermore, they are playing increasingly important roles in making robust theoretical predictions. Finally, the knowledge-base acquired while practicing physics in this computer-centric environment serves as a highly-transferable skill set for individuals who pursue a career outside of HEP and/or academia.

Unfortunately, there is a critical deficiency in the manner by which users learn and practice these skills. Individual universities vary widely in the emphasis that they place on providing students with computing skills both formally and informally prior to their research. Then, once a student begins their research program, there is rarely a formal continuation of technical training to augment their diverse skill set. This can affect marginalized groups in particular and amplify the challenges they are facing due to other aspects of their career progression. These challenges in performing technical research are compounded in many cases due to domain-specific aspects of applying software skills that are convoluted with the process of learning the generic aspects of computing. Thus far, our community has largely embraced the approach of "no one size fits all" and allowed training to occur either through self-guided learning by each individual or through ad-hoc and unstructured guidance by a research mentor. Though successful in some cases, this approach suffers from being inequitable and favoring those who are already successful and have the opportunity to access resources that can help them augment their own knowledge base. Moreover, with the increasing complexity of the computing environment required to be successful in HEP but the inability to increase the duration of a PhD proportionally, improving the efficiency in which expert-level proficiency is obtained will be crucial to the future success of our field. And what about senior researchers who, due to the changing software landscape have to pick up a new "core" skill later in their career. For example, someone who suddenly has to work with Python but was only educated in C++, or someone who is now required by their collaboration to implement unit tests and continuous integration. Education is a continuous process and simply because someone is more senior does not automatically mean they are more knowledgeable or better at learning. It is our responsibility as a community to facilitate and include all who want to learn in this process, thereby increasing the efficacy of the entire field and not just "the next generation".

One solution in HEP is to exploit our diverse community and organize training within and between our research domains. Such efforts have recently been pioneered by IRIS-HEP, FIRST-HEP and the HEP Software Foundation (HSF). Most recently, the HSF has formalized this pedagogical approach by creating a group dedicated to training[1] with the mission *to help the research community provide training in the computing skills needed for researchers to produce high quality and sustainable software*. This group develops training materials and hosts workshops with a pedagogical approach similar to that of the Software Carpentries[2], rooted in five principles : [1] hands-on, [2] student-centric, [3] experiment agnostic, [4] re-useable, and [5] open and accessible. These lessons developed in an open-source hsf-training github space[3] such that any individual can contribute. Mature lessons are complemented by training videos housed on the HSF YouTube channel[4]. These materials are used as the focus for self-study material as well as both in-person (Nov 2019, Feb 2020) and virtual (June 2020, July 2020) training events that aim to be hands-on and expose participants to the generic and core tools and concepts while also helping them apply these techniques to an application of their choice (e.g. "their analysis").

The proposal of this LoI is to reflect on this approach to developing software and computing expertise in our field and how this expertise can be imparted to relevant fields in the context of programs such as the

[1] https://hepsoftwarefoundation.org/workinggroups/training.html
[2] ://software-carpentry.org/
[3] https://github.com/hsf-training
[4] https://www.youtube.com/channel/UCv4hukXGkCYvBClQMKzypnQ/playlists

Carpentries Incubator[5]. This will involve understanding what aspects of training in computing "factorizes" and can be implemented through augmentation of university curricula. It will also involve developing a "HEP Core Computing Curriculum", taking advantage of the existing work from the HSF Curriculum[6] and organizing a series of "Develo-Hackathons" throughout Snowmass during which a broader community can profit from and contribute to this curriculum. Finally, we propose to explore how a concrete commitment, through the creation of career paths and other incentives with an enhanced focus on training, will facilitate more effective progress in the future. Snowmass can play a role in bringing together a focus on these efforts to foray into sustainability and scalability and initiate learning early on in the process and which continues throughout one's career, to more fully prepare a software-equipped particle physicist community.

Facets of Proposed Study

- Survey and understand how the formal curricula of a physics degree (bachelors and PhD) emphasize and approach computing education
- Execute multiple "develo-hackathons" to develop a "HEP Core Computing Curriculum" to more efficiently train PhD students and researchers in essential skills
- Explore ways in which the field can augment the approach to career progress to better support individuals who wish to devote a more significant portion of their career to educating the community
- Explore ways in which in which the field can support researchers transitioning to industry

[5] https://carpentries.org/involved-lessons/
[6] https://hepsoftwarefoundation.org/training/curriculum.html