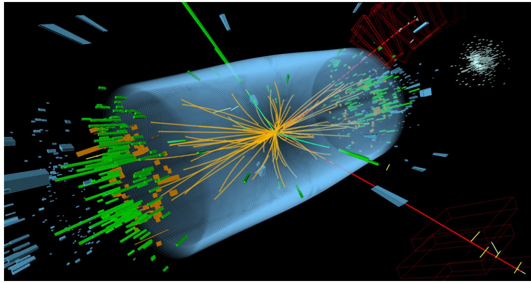
Software Training in HEP

Sudhir Malik
University of Puerto Rico Mayaguez
(on behalf of HSF/IRIS-HEP training group and all contributors to the training)

Co-authors:
Samuel Ross Meehan (CERN)
Kilian Lieret (Ludwig Maximilian University Munich)
Meirin Oan Evans (University of Sussex (GB))
Michel Hernandez Villanueva (University of Mississippi)
Daniel S. Katz (University of Illinois)
Graeme A Stewart (CERN)
Peter Elmer (Princeton University (US))
And many more at: https://hepsoftwarefoundation.org/training/community.html

Software a key to HEP success



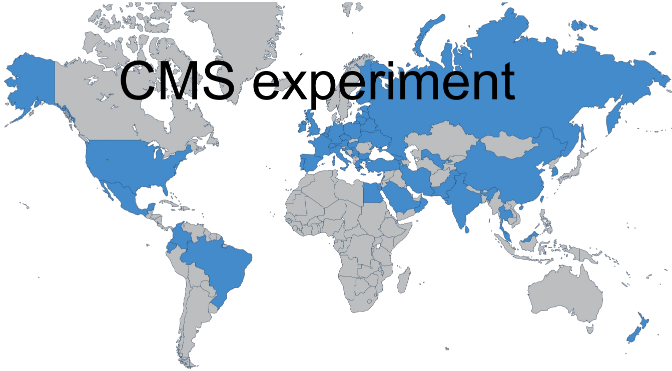
- Solving software challenges integral to the success of current and future HEP experiments (HL-LHC, DUNE, etc.)

- Software and computing systems are key subsystems of our experiments, involve significant budget

- Maximizing science from the hardware investments increasingly relies critically on software

- Software skills essential for a successful HEP physicist, and for career evolution for people trained inside HEP, seeking career in industry

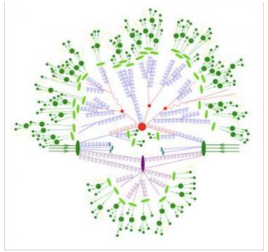
Scientific Collaborations are big and growing



- Current examples (estimate stats)
  - **BELLE II** - 1200 collaborators/121 institutes/26 countries
  - **CMS** - 4000 collaborators/200 institutes/50 countries
  - **ATLAS** - 3000 collaborators/174 institutes/38 countries
  - **LHCb** - 1200 collaborators/76 institutes/16 countries
  - **ALICE** - 1000 collaborators/100 institutes/30 countries
  - **DUNE** - 1000 collaborators/180 institutions/30 countries
  - **LIGO** - 1200 collaborators/100 institutions/18 countries
- Past
  - **DZero** - 540 collaborators/90 institutions /18 countries
  - **CDF** - 600 collaborators, 30 institutions/12 countries

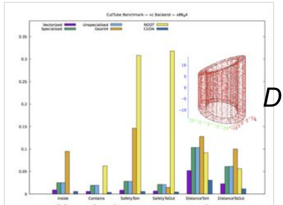
HEP software ecosystem

Software and Physics analysis are intertwined



Physics Event Generators

Data, Software,
Analysis Preservation

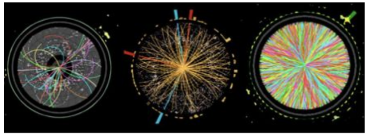


Vectorised geometry speed-ups,
Mihaela Gheata

Detector Simulation

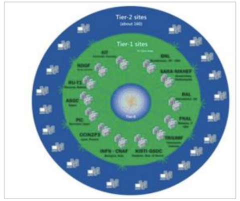
Security

Software Development



Trigger,Event Reconstruction

Lots of challenges

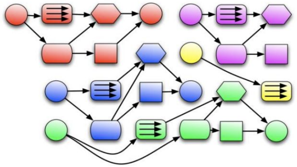


Facilities, Distributed
Computing

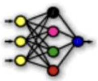
Data Analysis, Interpretation, Simulation

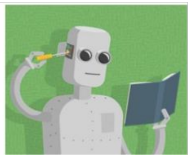
Visualization

Data Processing
Frameworks



Machine Learning

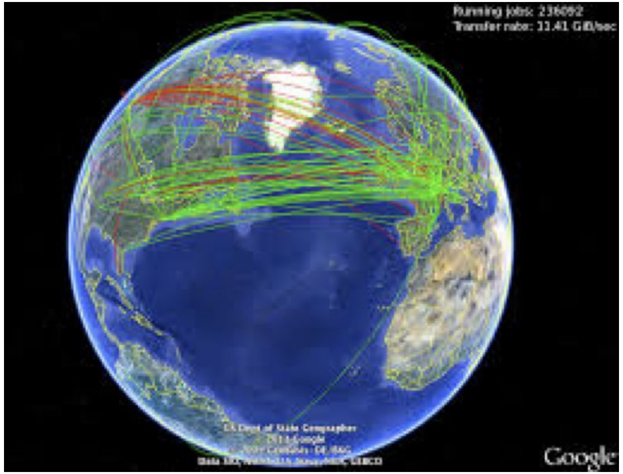




Data Management
Organisation, Access

HEP Paradigm

- Paradigm for HEP users
  - Knowledge of complex computing and physics analysis tools intertwined
  - Software challenges wrt to data rates, processing and analysis
- Long life span of the experiment ~ 30 years
- Enormous data rate
- Most users not resident at host laboratory
  - Financial and logistic constraints to be at host lab (e.g. CERN)
- Highly distributed environment for
  - Computing (Grid)
  - Physics analysis
- Physics/Computing Support
  - Should reach every user wherever they may be
  - Should be taken up in organized and central way



Training Challenge

- Training is a prerequisite to meet data and software challenges
- Funding agencies and institutions may not have the
  same priority for software training and education
  as for building/operating detectors, physics analysis, etc.
- Training activities are severely undervalued in making career steps
- Individual universities do not uniformly provide training today prior to a Ph.D. student beginning their research career
- Volunteers can usually only dedicate their time in specific career phases as a side “hobby” project
- Training materials are a moving target as technology evolves
- Separating “Experiment” specifics (e.g. computing environments or dedicated software) from HEP wide “common” usable material is important, but doesn’t always happen
- Are training materials a common good or an individual product? Even if individuals do want to contribute to a common good, how do they do so?

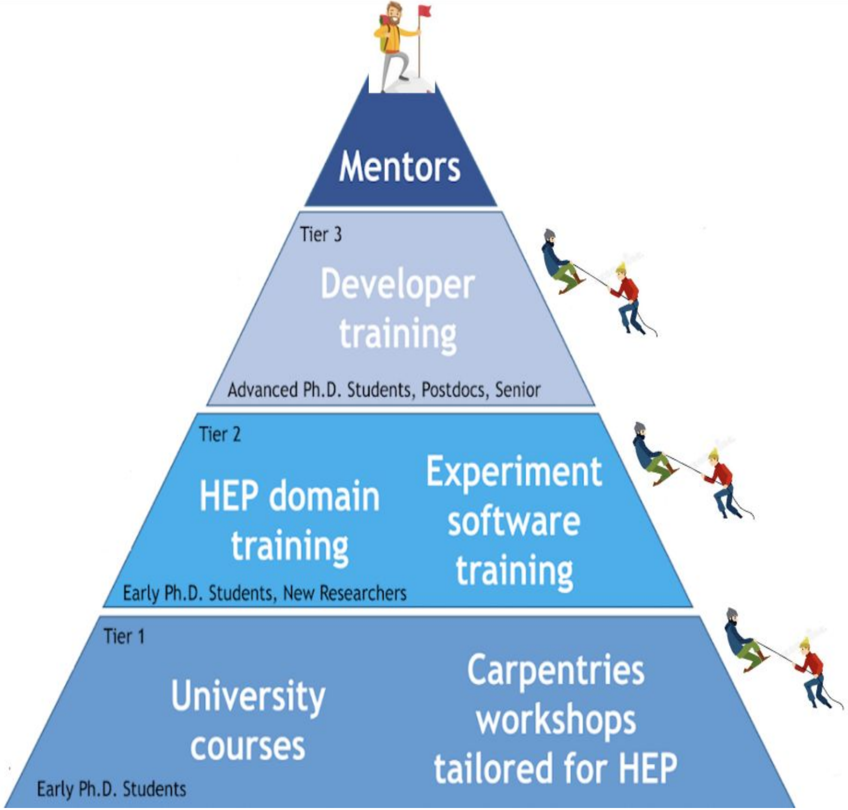
Training Vision

- Provide training in the computing skills to produce high quality sustainable software, solve HEP challenges, software-trained workforce

- Train every new HEP entrant in the related Software and tools

- Build community for scalability and Sustainability

- Training style - Hands-on, Student-centric, Experiment Agnostic, Reuseable, Open and Accessible


Mentors
Tier 3
Developer training
Advanced Ph.D. Students, Postdocs, Senior
Tier 2
HEP domain training
Experiment software training
Early Ph.D. Students, New Researchers
Tier 1
University courses
Carpentries workshops tailored for HEP
Early Ph.D. Students

- Broader impact - reach out to STEM aspirants in high schools via their teachers and involve students early on

Training Organisation

- Led by HSF training group, established ~3 years ago

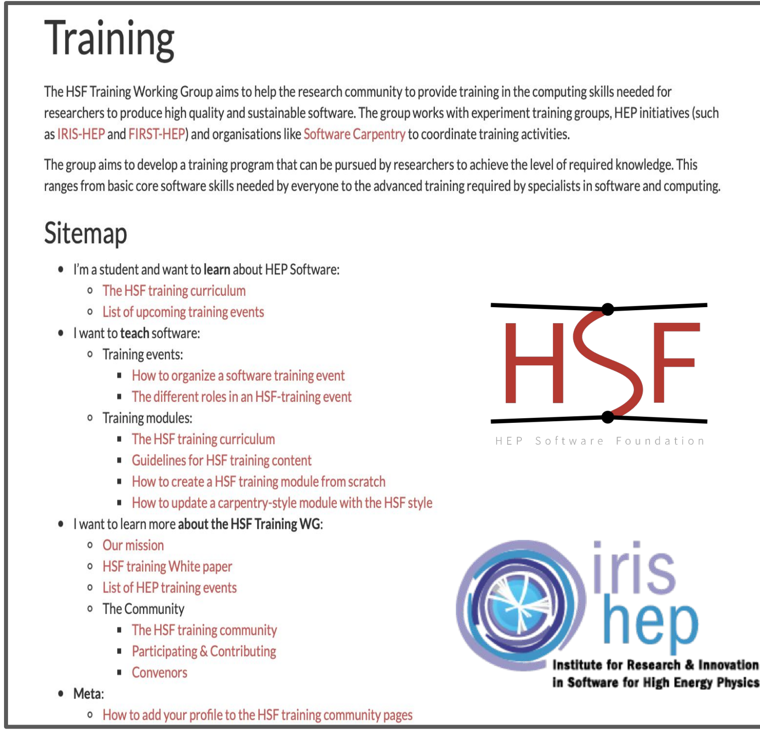
- 3 co-convenors, engaging with educators from different collaborations

- Strong partnership with IRIS-HEP, FIRST-HEP and the Carpentries

- Prepares training material and coordinates activities for the common good

- Strong community of instructors and participants, feeling of community ownership

- Focuses on common software material across HEP, ranges from basic core software skills needed by everyone to advanced training required by specialists in software and computing



Training

The HSF Training Working Group aims to help the research community to provide training in the computing skills needed for researchers to produce high quality and sustainable software. The group works with experiment training groups, HEP initiatives (such as IRIS-HEP and FIRST-HEP) and organisations like Software Carpentry to coordinate training activities.

The group aims to develop a training program that can be pursued by researchers to achieve the level of required knowledge. This ranges from basic core software skills needed by everyone to the advanced training required by specialists in software and computing.

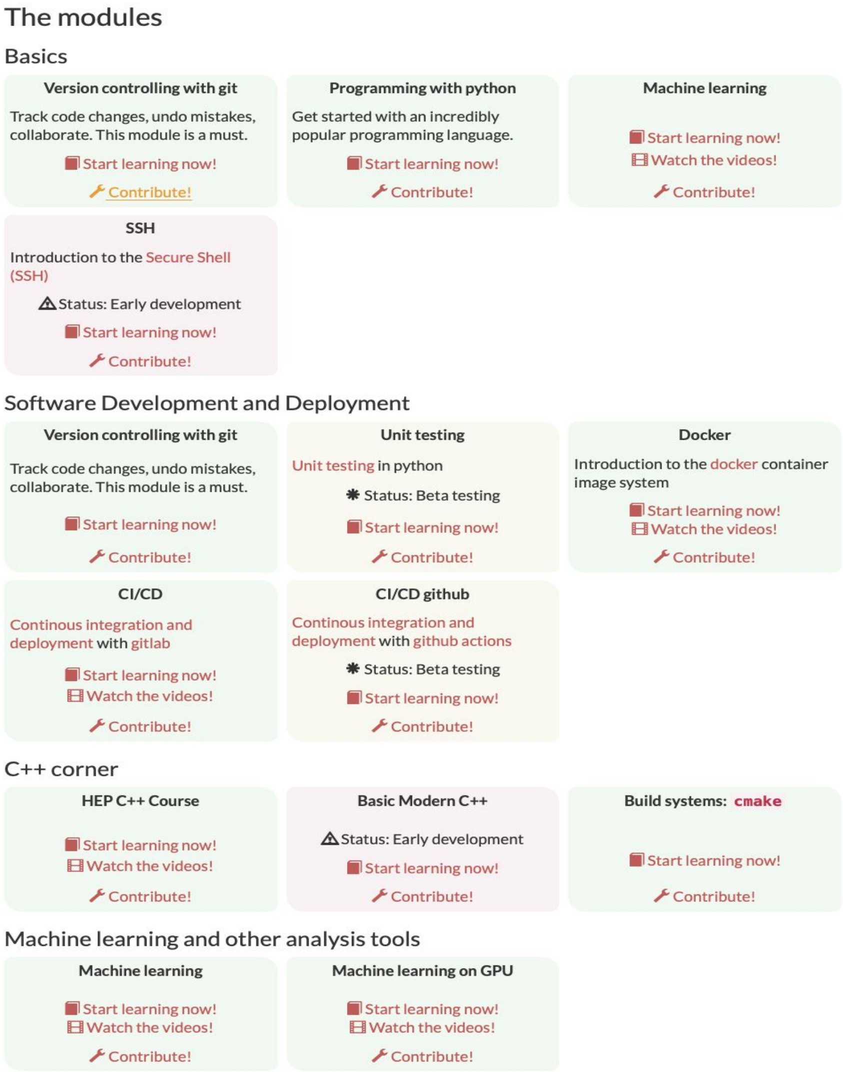
Sitemap

- I'm a student and want to **learn** about HEP Software:
  - The HSF training curriculum
  - List of upcoming training events
- I want to **teach** software:
  - Training events:
    - How to organize a software training event
    - The different roles in an HSF-training event
  - Training modules:
    - The HSF training curriculum
    - Guidelines for HSF training content
    - How to create a HSF training module from scratch
    - How to update a carpentry-style module with the HSF style
- I want to learn more **about the HSF Training WG**:
  - Our mission
  - HSF training White paper
  - List of HEP training events
  - The Community
    - The HSF training community
    - Participating & Contributing
    - Convenors
- Meta:
  - How to add your profile to the HSF training community pages

HSF
HEP Software Foundation
iris hep
Institute for Research & Innovation in Software for High Energy Physics

Curriculum

- The pilot phase of training events was based on an initial survey across HEP community in 2019
- This survey and experience and feedback gathered at the events lead course structure into a full curriculum
- Guidelines for the development of the modules and the procedure for training events are formalized
- Each training module is independent from the others, students can prioritize certain skills before others
- All software material is open source



The modules

Basics

Version controlling with git

Track code changes, undo mistakes, collaborate. This module is a must.

Start learning now!

Contribute!

Programming with python

Get started with an incredibly popular programming language.

Start learning now!

Contribute!

Machine learning

Start learning now!

Watch the videos!

Contribute!

SSH

Introduction to the Secure Shell (SSH)

⚠ Status: Early development

Start learning now!

Contribute!

Software Development and Deployment

Version controlling with git

Track code changes, undo mistakes, collaborate. This module is a must.

Start learning now!

Contribute!

Unit testing

Unit testing in python

✱ Status: Beta testing

Start learning now!

Contribute!

Docker

Introduction to the docker container image system

Start learning now!

Watch the videos!

Contribute!

CI/CD

Continous integration and deployment with gitlab

Start learning now!

Watch the videos!

Contribute!

CI/CD github

Continous integration and deployment with github actions

✱ Status: Beta testing

Start learning now!

Contribute!

C++ corner

HEP C++ Course

Start learning now!

Watch the videos!

Contribute!

Basic Modern C++

⚠ Status: Early development

Start learning now!

Contribute!

Build systems: `cmake`

Start learning now!

Contribute!

Machine learning and other analysis tools

Machine learning

Start learning now!

Watch the videos!

Contribute!

Machine learning on GPU

Start learning now!

Watch the videos!

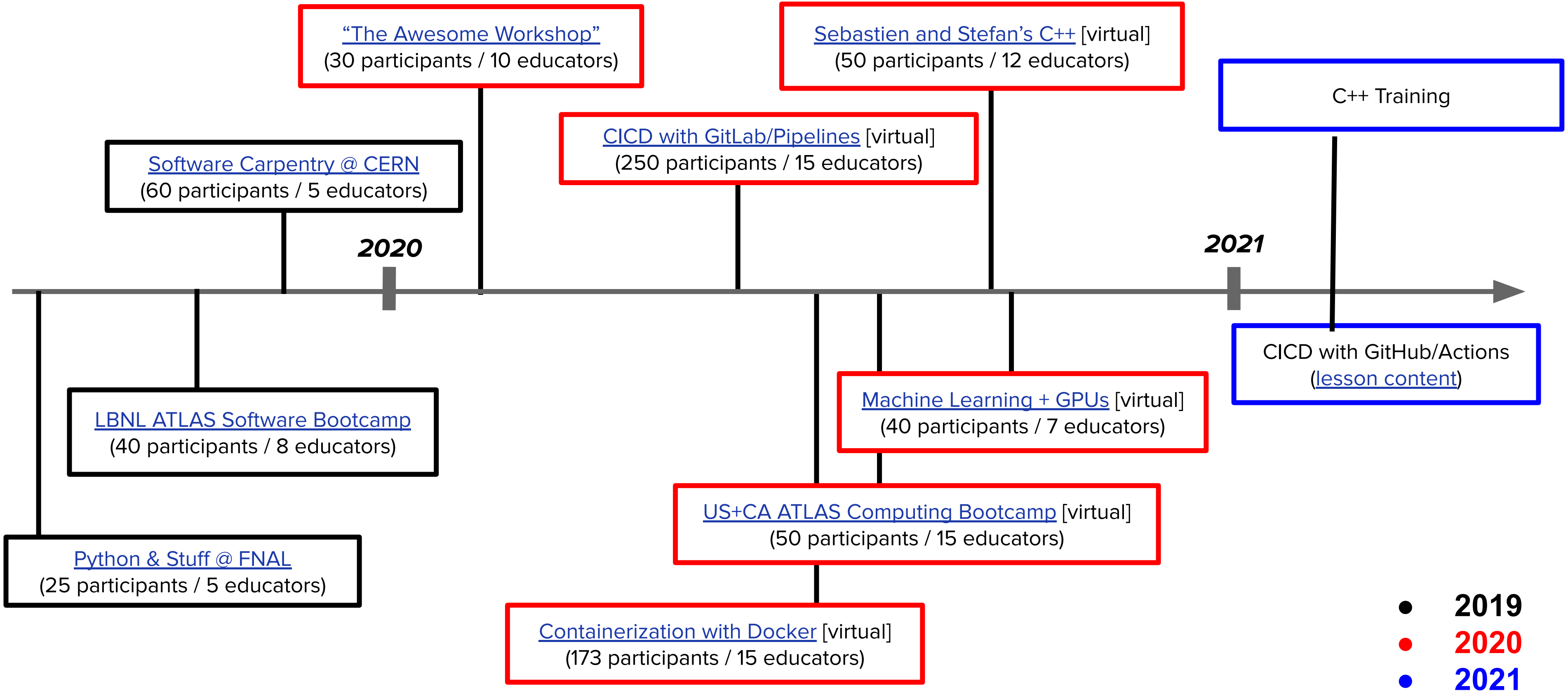
Contribute!

Accomplishments

- **Software modules**
  - Basic software curriculum
    - Introductory software training curriculum serving all HEP entrants
  - Intermediate modules, some specific to HEP
  - All modules are open source
- **Training events** (last 2 years)
  - 13 events, 1500 participants
  - In-person, Online (Covid Impact)
  - Out of these, 5 are Basic Curriculum
    - 400 attendees

- **120 instructors involved**

Trainings to Date



Python & Stuff @ FNAL
(25 participants / 5 educators)

LBNL ATLAS Software Bootcamp
(40 participants / 8 educators)

Software Carpentry @ CERN
(60 participants / 5 educators)

2020

“The Awesome Workshop”
(30 participants / 10 educators)

CICD with GitLab/Pipelines [virtual]
(250 participants / 15 educators)

Containerization with Docker [virtual]
(173 participants / 15 educators)

US+CA ATLAS Computing Bootcamp [virtual]
(50 participants / 15 educators)

Machine Learning + GPUs [virtual]
(40 participants / 7 educators)

Sebastien and Stefan’s C++ [virtual]
(50 participants / 12 educators)

2021

CICD with GitHub/Actions
(lesson content)

C++ Training

- ● 2019
- ● 2020
- ● 2021

You want to have a high impact and advance HEP? - Training might be your most effective choice!

In person training

- **Attendance** : few ***dozen***
- **Advantage**
  - Active/efficient engagement of participants
  - Professional networking and additional “events”
- **Limitations**
  - Travel costs (education should not be exclusive)
  - Long lead time for planning logistics
    - Related to travel/room booking
  - Requires participant “sacrifice”
- **Important things**
  - Room setup is crucial
    - Two projects/screens
    - Not an auditorium
    - Ample power
- **Suggested Ratio of Participant** : Educator <= 5
  - This is *essential* to allow for the “hands on” aspect of the workshop to be successful

- **Large time commitment** on behalf of the educators
  - Can’t just “do your talk” and then leave

Virtual training

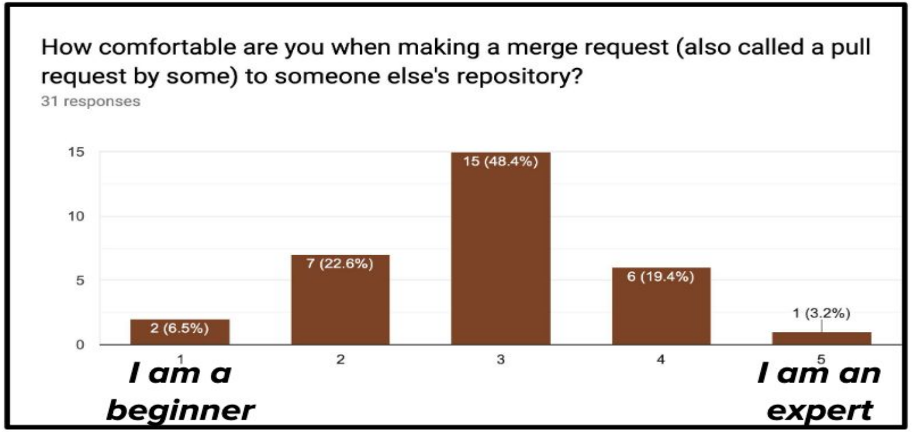
- Pivot to remote training due to COVID
  - Adapted quickly
  - 7 online training workshops (last 12 months)
- Attendance : few ***hundred***
- Positives
  - Broader reach, more participants: >100 registrants
  - No travel costs → critical for some supervisors
  - Easier logistics, easy to reach all timezones
  - Materials are recorded and archived (videos)

- **Limitations**
  - Active/meaningful interactions
  - Mentors in different time zones
  - Keep everyone engaged
- **Important things**
  - Clearly-defined roles for instructors
  - Effective chat application is essential
    - mattermost/discord/slack

Training Works !!

- We do our best to diligently collect before/after data via surveys
  - Pre-survey
    - Demographics
    - How much do you know?
  - Post-survey
    - How much do you **now** know?
    - What can we do better next time?
  - Would like to have further out “follow up” surveys (takes more work …)
- Self-reported learning *does* happen!

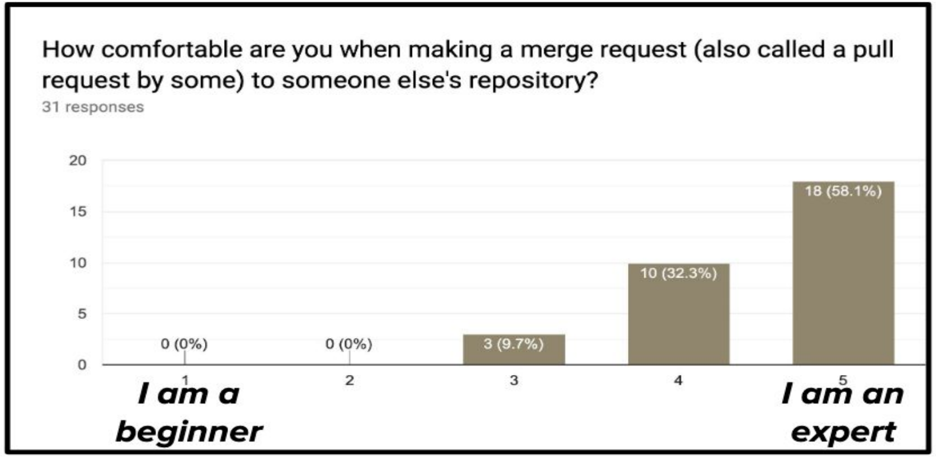
Before



How comfortable are you when making a merge request (also called a pull request by some) to someone else's repository?

31 responses

After



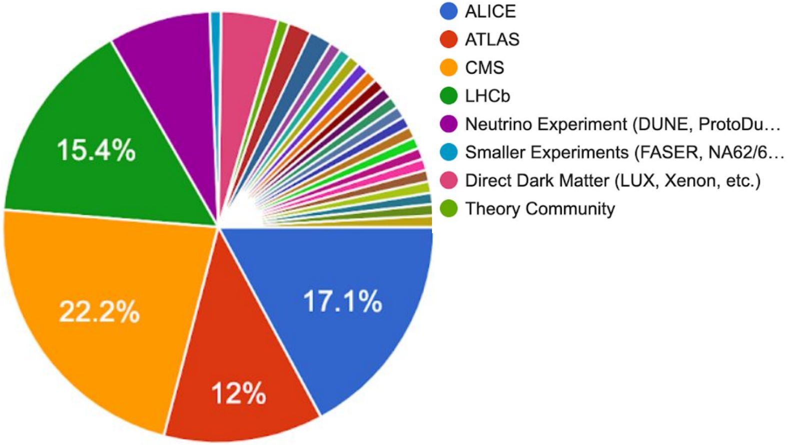
How comfortable are you when making a merge request (also called a pull request by some) to someone else's repository?

31 responses

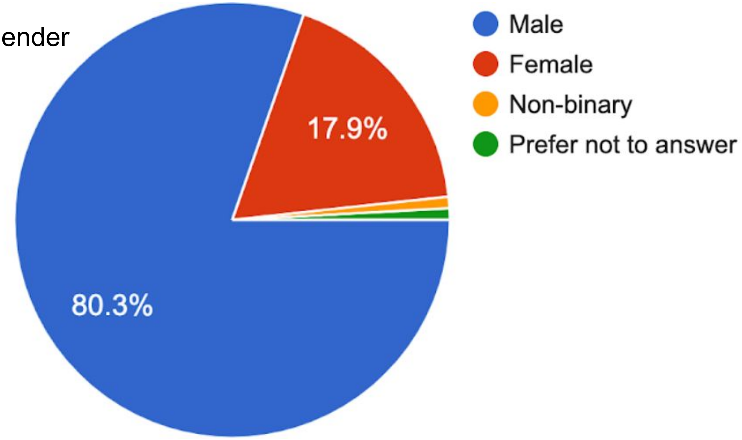
Impact and diversity

GitHub CI/CD Training Example (Feb 2021)

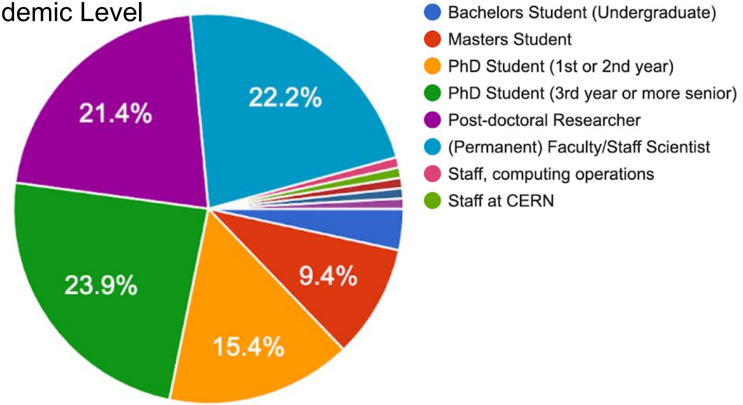
Experimental Collaboration



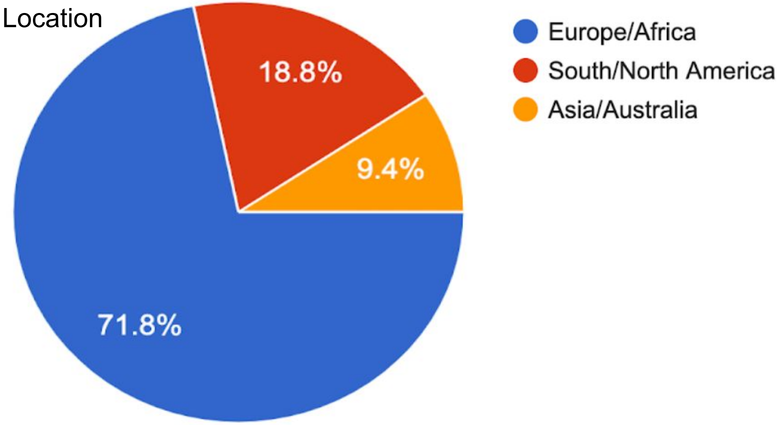
Gender



Academic Level

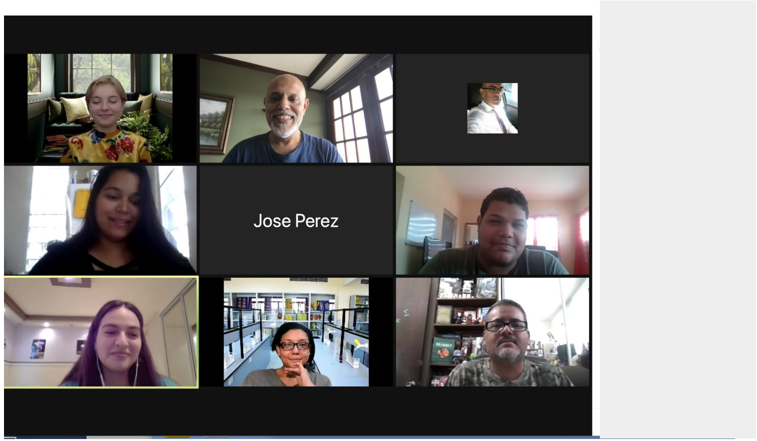


Location



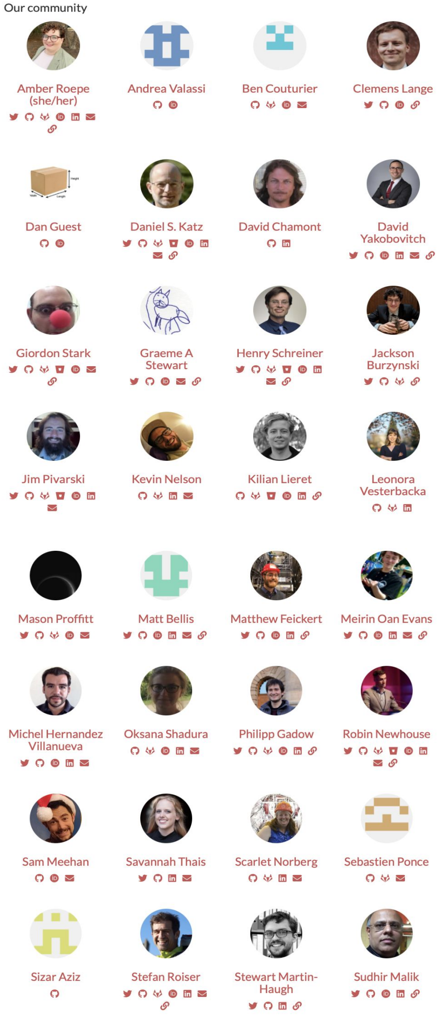
Lessons Learned

- Advantages and limitations of in-person and virtual trainings
- Build a community around training
  - Incentivize and compensate instructors
  - Core team to support the training mission
- Scale up training (new formats)
  - **Core format: we organize and teach** ✓
    - In-person, Online
    - Fund instructors to travel and teach
    - To scale up, need to expand to other formats
  - **DIY (Do-it-yourself)** 🚧
    - Minimal help from us, basically organise yourself, using training material (no expense involved)
    - In-person, Online
  - **Asynchronous (Anytime/Anywhere)** 🚧
    - Coursera-type, small professional videos (~10 mins.), Q&A assessment
    - Use current material to extend training to this format



Community

- Active community members to support training
- Time dedicated on voluntary basis, great dedication and enthusiasm
- Tutors come from different HEP collaborations
  - This diversity adds great value to the training
  - Brings flavor of experience from a different computing environment
  - Common goal to create, teach, and sustain a common set of skills across
- Prepare for careers in software, strengthen job profile and enhance chances of employability in industry
- Profile of each tutor that contributes to the training on HSF page
  - public proof of their capability, skills and contribution



Our community

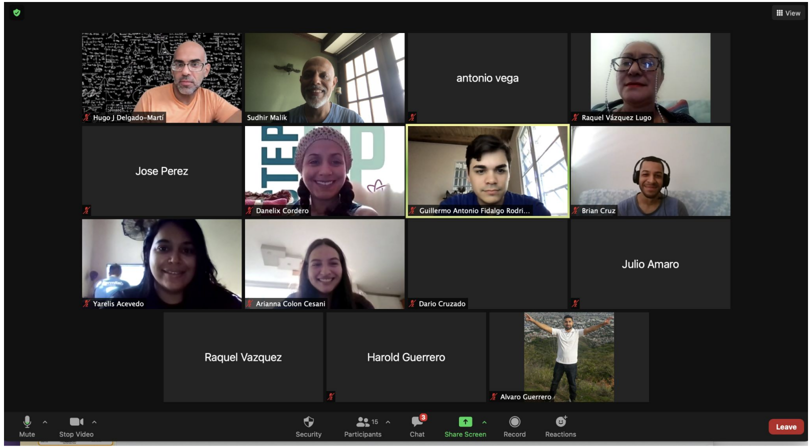
Amber Roepe (she/her)
Andrea Valassi
Ben Couturier
Clemens Lange
Dan Guest
Daniel S. Katz
David Chamont
David Yakobovitch
Giordon Stark
Graeme A Stewart
Henry Schreiner
Jackson Burzynski
Jim Pivarski
Kevin Nelson
Kilian Lieret
Leonora Vesterbacka
Mason Proffitt
Matt Bellis
Matthew Feickert
Meirin Oan Evans
Michel Hernandez Villanueva
Oksana Shadura
Philipp Gadow
Robin Newhouse
Sam Meehan
Savannah Thais
Scarlet Norberg
Sebastien Ponce
Sizar Aziz
Stefan Roiser
Stewart Martin-Haugh
Sudhir Malik

Sustainability

- **Training model**
  - Training without borders
  - Strong community would lead to minimal human resources needed to keep training infrastructure running
  - Long-term financial support
- **Build regional and local capacity**
  - Empower HEP communities
    - Local mentorship and leadership
    - Engage more HEP labs and universities
- **Opportunities to grow professionally**
  - Career paths, strengthen job profile
- **Equity, diversity, inclusion and accessibility**
  - Participation across HEP communities, under-resourced, underrepresented institutions, communities in different geographical regions
    - Serve as a role model
    - Open source is a step in this direction

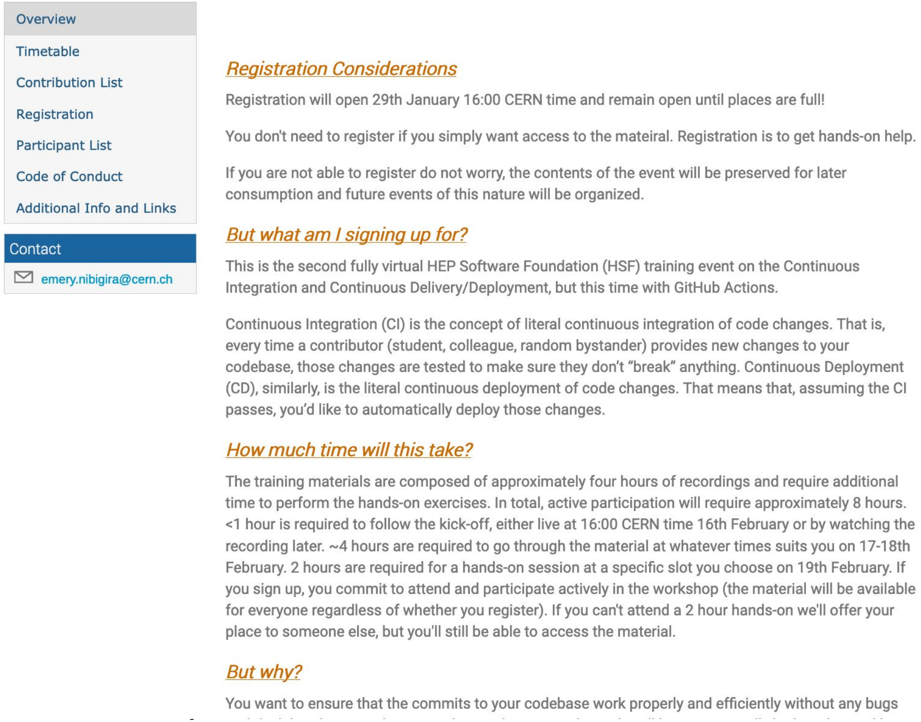
Broader Impacts

- **Organized 6 outreach events**
  - Programming
  - HEP data preview
  - CMS Open Data
  - Machine Learning basics
  - Machine Hackathon
  - Events in-person and online
- **Future Outreach**
  - More events per year in HEP-related communities
    - Keep in mind: Teachers available only at end of semesters
  - Develop short video modules for teachers and students to learn software anytime
  - Supplement workshops (in-person or online)
  - More engagement with Quarknet



Upcoming Events

- Basic curriculum training
  - July, September, December 2021
- Brainstorming sessions
  - June and November 2021
  - April 2021
- Matplotlib for HEP (October 2021)
- Modern C++ (September 2021)
- Introduction to Singularity (Novem
- HEP data analysis
  - Advanced analysis tools in HEP
  - February 2022



Registration Considerations

Registration will open 29th January 16:00 CERN time and remain open until places are full!

You don't need to register if you simply want access to the mateiral. Registration is to get hands-on help.

If you are not able to register do not worry, the contents of the event will be preserved for later consumption and future events of this nature will be organized.

But what am I signing up for?

This is the second fully virtual HEP Software Foundation (HSF) training event on the Continuous Integration and Continuous Delivery/Deployment, but this time with GitHub Actions.

Continuous Integration (CI) is the concept of literal continuous integration of code changes. That is, every time a contributor (student, colleague, random bystander) provides new changes to your codebase, those changes are tested to make sure they don't "break" anything. Continuous Deployment (CD), similarly, is the literal continuous deployment of code changes. That means that, assuming the CI passes, you'd like to automatically deploy those changes.

How much time will this take?

The training materials are composed of approximately four hours of recordings and require additional time to perform the hands-on exercises. In total, active participation will require approximately 8 hours. <1 hour is required to follow the kick-off, either live at 16:00 CERN time 16th February or by watching the recording later. ~4 hours are required to go through the material at whatever times suits you on 17-18th February. 2 hours are required for a hands-on session at a specific slot you choose on 19th February. If you sign up, you commit to attend and participate actively in the workshop (the material will be available for everyone regardless of whether you register). If you can't attend a 2 hour hands-on we'll offer your place to someone else, but you'll still be able to access the material.

But why?

You want to ensure that the commits to your codebase work properly and efficiently without any bugs

Summary

- Software training is making a difference
- Organized several training events in-person and virtual
- Learnt valuable lessons
- Virtual training has increased impact
- Developed basic software curriculum modules
- Intermediate and advanced level modules are populated
- Next step is to scale up training activities
- Synergy among HEP experiments (including neutrino) and Nuclear Physics community exists
- Preparing trained workforce pipeline
- Broader Impacts, inclusiveness and diversity are integral part

Training Information

- **Training events**: https://indico.cern.ch/category/11386/
- **Material**: All the training modules developed so far resides: https://hepsoftwarefoundation.org/training/curriculum.html
- **Community**: Our training community is listed here: https://hepsoftwarefoundation.org/training/community.html
- **Procedure**: how to request and organize a training: https://hepsoftwarefoundation.org/training/howto-event.html
- **Funding**: Funding for training events is provided by the IRIS-HEP/FIRST-HEP
- **Blueprint**: First blueprint on training https://indico.cern.ch/event/889665/
- **Videos**: https://www.youtube.com/c/HEPSoftwareFoundation/videos
- **Training, Education, Outreach** - https://iris-hep.org/ssc.html

Acknowledgments

Our community is growing and credit goes to many individuals, especial thanks to:

- The US **National Science Foundation** through grants OAC-1829707 and OAC-1829729 (FIRST-HEP), Cooperative Agreement OAC-1836650 (IRIS-HEP)

- Hosts and partners of training events mentioned in the talk:

    - CERN
    - Fermi National Accelerator Laboratory
    - Argonne National Laboratory
    - Lawrence Berkeley National Laboratory
    - The Carpentries
    - The US-ATLAS
    - Princeton University
    - University of Manchester
    - University of Puerto Rico at Mayaguez