EPJ Web of Conferences **214**, 05044 (2019) https://doi.org/10.1051/epjconf/201921405044
CHEP 2018

Software training for the next generation of physicists: joint experience of LHCb and ALICE

Dario Berzano[1,2,*], *Chris* Burr[3,**], *Hans* Beck[4], *Violaine* Bellée[5], *Redmer Alexander* Bertens[6], and *Albert* Puig Navarro[7]
on behalf of the ALICE and LHCb Collaborations

[1]European Organization for Nuclear Research (CERN)—Genève, Switzerland
[2]Istituto Nazionale di Fisica Nucleare (INFN)—Italy
[3]The University of Manchester—United Kingdom
[4]DXC Technology—Frankfurt, Germany
[5]École polytechnique fédérale de Lausanne—Switzerland
[6]University of Tennessee—USA
[7]Universität Zürich—Switzerland

Abstract. The need for good software training is essential in the HEP community. Unfortunately, current training is non-homogeneous and the definition of a common baseline is unclear, making it difficult for newcomers to proficiently join large collaborations such as ALICE or LHCb. In the last years, both collaborations have started separate efforts to tackle this issue through training workshops, via Analysis Tutorials (organized by the ALICE Juniors since 2014) and the Starterkit (organized by LHCb students since 2015). In 2017, ALICE and LHCb have for the first time joined efforts to provide combined training by identifying common topics, such as version control systems (Git) and programming languages (e.g. Python). Given the positive experience and feedback, this collaboration will be repeated in the future. We will illustrate the teaching methods, experience and feedback from our first common training workshop. We will also discuss our efforts to extend our format to other HEP experiments for future iterations.

1 Introduction

LHCb [1] and ALICE [2] are two major experiments at the Large Hadron Collider (LHC), CERN. They count around 800 and 1800 members, respectively. Every year, dozens of new students from all over the world at different points in their career (bachelor, master, Ph.D.) join the two collaborations.

Basic computing training from their home universities is diverse: some of them have a solid computing training tradition and even specific classes with examples from Particle Physics experiments, but there exist universities where programming languages are not part of the training.

When students join the collaboration, supervisors cannot rely on their computing skills: quite frequently, students are given coding examples to copy from, not always coming from

*e-mail: dario.berzano@cern.ch
**e-mail: christopher.burr@cern.ch

authoritative sources, and they are asked to "learn by example", with mixed results—which ends up negatively impacting on the work they are meant to do, and their supervisor's as well.

Against this backdrop, the need for some form of introductory training in computing for High Energy Particle Physicists is clear. The LHCb and ALICE collaborations have started addressing this problem separately, as we will see in Section 2, before finally joining forces in 2017. In October 2017, the first edition of the joint LHCb–ALICE training event called "Starterkit" was organised at CERN: it is a yearly 5 day long introductory crash course with various computing classes, both generic (such as programming languages) and experiment-specific. From the success of the first edition, a second one is imminent at the time of writing (November 2018), with the involvement of a third experiment, SHiP[3]. An overview on the training topics is given in Section 3. The way the training material is written, published and managed is discussed in Section 4. How the organisation of the first joint Starterkit took place is covered in Section 5, and a discussion on the students' feedback and their demographics is presented in Section 6. Finally, we conclude with some lessons learned from our previous experiences (Section 7), and a discussion on the sustainability of our approach (Section 8).

2 Training initiatives in LHCb and ALICE

2.1 Training in LHCb

In 2015, some young LHCb members felt the need for computing training and started organising a 5 day workshop[4]. The name "Starterkit" (see Figure 1) was attributed to the event.

Figure 1. The LHCb Starterkit logo

The Starterkit [5] is traditionally organised once per year in Autumn, and it covers basic topics. In springtime a more advanced three days workshop takes place: it is called the "Impactkit" and includes a *hackaton* as well. Usually, some students become teachers the year after they are taught, and they become event organisers two and a half years later.

The way the Starterkit is organised, and the training topics, are the same as today, where the ALICE collaboration also takes part, and are therefore discussed in the rest of the article.

2.2 Training in ALICE

The ALICE training efforts were started in 2014 by the ALICE Juniors community [6]. A special group called the "Analysis Tutorial Committee" was thus created to this respect, with the mandate of organising tutorials during the collaboration meetings (the "ALICE Weeks"), in order to mitigate the travel expenses: there are three ALICE Weeks a year, typically at CERN, and a large fraction of the collaboration travels there for the full week in any case.

As the name "Analysis Tutorial" suggests, those training events revolve around how to perform physics analyses in ALICE, from the grounds up. Since the ALICE software is built on the users' laptops, there is a specific software installation class (using aliBuild [7], the ALICE build tool). Analysis on the Grid in ALICE is generally organised in so-called "trains" [8]: a specific tutorial on how to write an analysis task and how to launch it on the Grid is thus the most important part. Organised analysis implies having a single Git software repository for all users [9] with daily releases: for this reason, there are classes on how to contribute through Git and GitHub Pull Requests. Regularly, advanced topics are also covered, such as memory optimisation, debugging and profiling in C++. Thematic classes are also available for users doing specific analyses. A special session of the Analysis Tutorial is also organised for ALICE CERN Summer Students at the beginning of their stay.

Since ALICE teachers are in most cases the creators of the tools or procedures they are teaching, Analysis Tutorials have been used in multiple occasions to prepare the whole collaboration, and not just the youngest members, to important changes in their workflow. This was the case, for instance, when ALICE software was migrated to GitHub, and Pull Requests with automatic tests were introduced. Tutorials are also useful for collecting feedback on the usability on the aforementioned tools and procedures.

The ALICE collaboration has a considerable number of members outside Europe who cannot easily travel to CERN. For this reason, Analysis Tutorials can always be attended remotely using Vidyo [10]. Remote participation is always higher than the local one.

3 Training topics

The first problem when organising an event together was to agree on the set of skills to teach: LHCb and ALICE have different analysis workflows and experiment-specific tools, but there are also many similarities. In both experiments, for instance, Git is used for version control. Both experiments assume a Unix-like environment for working (Linux, macOS) therefore the Unix shell is used. Python is the main programming language for LHCb, and there is a growing interest in it in ALICE (even if it currently uses C++), mostly due to the expectations from the LHC Run 3 framework (currently in preparation [11]) and an increasing interest in machine learning, where Python is the most popular language.

Based on the above arguments, LHCb and ALICE have decided to split the first joint Starterkit into two parts. The first two days are dedicated to what are considered topics of general interest: Git, the Bash shell and Python. The remaining three days are organised independently by the two experiments, and experiment-specific topics and procedures are illustrated (Figure 2).

Starterkit students are expected to bring their laptops. Given that the time is short, students receive instructions on how to prepare their work environment in advance. A small amount of time during the first days is in any case dedicated to help the students who had troubles doing it at home.

4 Training material

LHCb and ALICE share the same policy for publishing their training material, making it easy to reach an agreement on this point. Training material is available online, free for everyone to read (no access credentials), and it is presented in form of web pages. Web pages are favoured over PDF documents or slides because they allow for an easier consultation (including copying commands and examples) and because any update to them is immediately visible: with PDF copies there is a higher chance that users may refer to an outdated, offline copy.

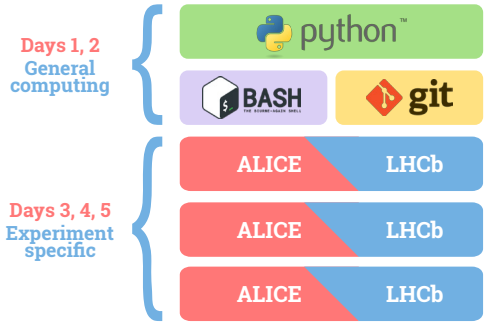

Figure 2. The 5 day Starterkit program

The training material serves as a base for the Starterkit lessons, where web pages are projected on the big screen. It is also supposed to be a reference for later, especially for advanced topics that could not be covered during the Starterkit. In order to allow to easily contribute, the documentation is written in Markdown [12] and hosted on GitHub, where users are encouraged to contribute through Pull Requests. The final web pages are generated through GitBook [13]. Instructions on how to contribute are available on the same documentation pages. The documentation is fully reviewed by the teachers at least once per year, prior to the Starterkit. Moreover, during the Starterkit, students are asked to take note of any problem they have found in the documentation, and part of the last day is dedicated to fixing it. All those aspects contribute to make sure the documentation is always fresh and up to date.

Along with LHCb-[14] and ALICE-specific [15] documentation pages, experiment-independent material has been extracted [16] and it is now being maintained as part of the HEP Software Foundation (HSF) [17] working group on Training, Staffing and Careers [18]. This material represents the main source of documentation for both the LHCb and the ALICE collaborations.

Sharing common documentation through experiments is beneficial as it puts it under the scrutiny of communities with different needs. For example, the original Git documentation from the LHCb Starterkit did not cover a collaborative workflow based on Pull or Merge Requests (depending on whether GitHub or GitLab is used), which is however very important for ALICE. The missing part has been added to the *common* documentation, with particular care of making it not experiment-oriented, and it is now available to everybody.

4.1 Licensing

The original LHCb Starterkit material is a derivative copy from Software Carpentry [19]. Software Carpentry is a project managed by volunteers that aims to teach basic computing skills to researchers. The original documentation is available through the open *Creative Commons Attribution license (CC-BY)* [20]: it can be copied and modified with the only fair obligation of crediting the original source. All the LHCb, ALICE and joint Starterkit material is shared under the same license, even if Software Carpentry does not require it (i.e. it does not have the *Share Alike* clause). Code snippets (such as code examples and commands) embedded in the documentation are licensed under the MIT license [21].

4.2 Automatic checks of the documentation

As said, the way to contribute to the documentation is through Pull Requests on GitHub. When a Pull Request is opened, it triggers some automatic tests that check if the page can be built with GitBook (i.e. there is no syntax error), and whether all hyperlinks and cross-references are valid and reachable. Tests are automatically run using the free tier of Travis CI [22]: if tests show errors, the Pull Request cannot be merged.

An extra level of tests has been added to the ALICE-specific documentation: shell commands are extracted from selected pages, and they are executed in order to check whether they work. ALICE has a specific documentation explaining what to install on the user's laptop in order to use the experiment software [23]: those commands are executed in a Docker container of the reference platform (for instance, CentOS 7) by automatically generating a Dockerfile [24]. As a byproduct, a reference, ready to use Docker container is generated and published on Docker Hub [25].

This test is run at every pull request, and every day. This is useful, for instance, in case a suggested package name to install is misspelled, or in case a certain package combination cannot be installed any longer: maintainers are immediately notified and they can take proper actions. Evaluations on how to extend this approach to more use cases, *e.g* testing the Python documentation, is underway.

5 Organisation and lessons

As we have seen, the Starterkit is five days long and it is organised at CERN once per year. There are two main organisers per experiment: LHCb has two new organisers every year, whereas ALICE has an established organising committee. Usually, a week close to other relevant events is chosen in order to contain the travel costs. Students are asked to pay a small fee (25 CHF) that covers the costs of coffee breaks and the social dinner on ThurWednesday. It is worth noting that both the coffee breaks and the social event are organised by volunteers.

Registration occurs online. Unfortunately, only a limited number of persons can be accommodated: the first joint Starterkit had a 45 participants limit for LHCb, and 25 for ALICE as a trial. Given the positive outcome of the first edition, ALICE has raised the limit to 40 for the 2018 edition. A waiting list is put in place in case someone cancels their registration. Participants are split over three rooms, each one with one teacher and at least two assistants, called "helpers" in the Software Carpentry context. During the first two days, where common topics are covered (see Section 3), LHCb and ALICE students and teachers are mixed.

Lessons follow the training material (Section 4) at a slow pace: students are encouraged to type every single command or code snippet and check the results before continuing, as opposed of copying-pasting them. There are also exercises to complete at certain points. For these, students are provided with two sticky notes of different colours and they are asked to put the green one on display when the exercise is complete, or the red one in case there is a problem, as we can see in Figure 3. Sticky notes, in combination with the helpers, are very useful for the engagement, and ensure everybody is catching up and little time is wasted. The sticky notes approach was adopted from Software Carpentry [19] as well.

Remote participation is discouraged and was never an option for LHCb-only Starterkits (Section 2.1), since it makes it difficult to interact with the remote students and assess their level of knowledge. Moreover, the importance of building a community by allowing for students and teachers to network is one of the key points of the Starterkit, and it is hard to do without meeting in person. The option to remotely participate has been however requested by ALICE with the motivations discussed in Section 2.2. As a compromise, the first joint Starterkit had one out of three classrooms broadcast using Vidyo, but the remote participants

Figure 3. Students are provided with coloured sticky notes to be put on display: they are used to request assistance *(red)* or to communicate that they are ready for the next topic *(green)*

had no possibility to intervene using their microphones. A separate chatroom was created with Mattermost [26], allowing remote users to type questions, and a dedicated ALICE helper collected them and reported them to the classroom when relevant. This approach effectively allowed remote participants to follow without disrupting the live lesson, and will be repeated in future Starterkits. Remote participants are not required to pay a fee.

6 Demographics and feedback

The Starterkit format is consolidated in the LHCb community: newcomers know about its existence and they are encouraged to subscribe by their supervisors and by former Starterkit students, who may also take part in teaching and helping. Every year about 50% of newcomers to LHCb participate in the Starterkit, all of them being in the early stages of their academic career. The first joint event showed a more diverse demographics for ALICE: a small fraction of the students were not young, but they participated in order to consolidate their field experience.

A strong sense of community emerges from the feedback from both experiments. ALICE students used a dedicated students mailing list and the Mattermost channel originally meant for remote participation (as seen in Section 5) to report problems and exchange information, before gradually migrating to the main mailing lists of the experiment. Starterkit helps students to become part of the experiment and feel at ease when asking for support.

At the end of every Starterkit, students are provided with an anonymous survey to complete. From the survey, the most appreciated aspect of the Starterkit is the one-to-one help:

students use it successfully even when they have questions that go beyond the scope of the lesson. The survey also shows how the lessons are generally considered well paced: there are corner cases of students already knowledgeable in some topics that find the lessons to be too slow, but in most cases even such category manages to learn new tricks. Most of the students feel like the Python classes are too short: it is, indeed, very difficult to teach a new programming language in less than two days. We took the feedback to split the Python documentation into a basic and an advanced part, in order to allow the students to better pursue their learning in autonomy after the Starterkit. Such a feedback stresses the importance of having a good documentation that can be followed offline too.

We had a very positive feedback concerning the networking between LHCb and ALICE: the very fact that students and teachers are mixed the first two days, so that ALICE teachers can teach LHCb students and vice versa, contributed in creating a broader sense of community that goes beyond the scope of one's experiment.

We are evaluating the possibility to run a second survey one year after the Starterkit to better understand, in hindsight, how much the Starterkit helped and where it did not.

7 Major lessons learned

From the organisational perspective, the training efforts were started by volunteers both in LHCb (Section 2.1) and in ALICE (Section 2.2), after coming to the conclusion that untrained students negatively impact the productivity of the collaborations. Volunteering calls for a flat organisational structure, as opposed to the hierarchical working group organisation typical of the LHC experiments. A relaxed approach makes the experience of taking part in the Starterkit pleasant and rewarding, both from the professional and the human perspective.

Maintaining the documentation requires a continuous effort. Once the documentation has been written, it needs to be constantly vetted and updated to new needs and technologies. There is no way around the time that needs to be dedicated to the process, therefore our efforts are towards eliminating the frustrations and splitting the workload evenly. TWiki pages and presentations in PDF were historically used by both LHCb and ALICE, and they held people back from contributing: the Markdown language, GitHub and Pull Requests have helped increasing the number of contributors—by also providing them a public space to discuss the modifications iteratively beforehand. Finally, asking teachers to review their part of the material prior to the lesson ensures that there is a regular checkpoint every year at the latest.

Teaching is not an easy task. Ideally, a mixture of experience levels is required: young teachers know what are the steepest parts of the learning curve, whereas experienced teachers master the topic and efficiently deliver the message. Having both teachers and helpers allows the Starterkit to have a good assortment of experience levels in every teaching room. Picking good teachers is not easy either, and we have found ourselves already in situations where some teachers were not completely suited for the task. This is one of the few downsides of an organisation made of volunteers: given the flat structure, it is difficult for the organisers to make sure teachers are prepared and exclude them in case they are not. We need to remind all the potential teachers that we count on their ability of self-assessing their knowledge, and invite them to rehearse the lessons beforehand: the ones with the highest communication skills must help the others into building confidence by motivating them.

The techniques adopted to ensure students' engagement have proven to be very effective. Interactivity is promoted over lecturing, and this is why remote participation is allowed only in special cases and with some limitations (Section 5). We have mentioned the use of sticky notes for requesting assistance: even though they represent a discreet form of communication, making students more comfortable into asking questions, there is still a fraction of students

that do not dare to ask. For this reason, helpers need to take an active role, that implies walking around the classroom and asking one by one if everything is fine.

8 Sustainability and recognition

High Energy Physics is an environment with a high turnover: many people leave the field every year, therefore it is extremely important to ensure continuity by building a community of teachers as large as possible. Everybody is invited to give even a small contribution, such as proofreading parts of the documentation: the organisation of next year's Starterkit is frequently mentioned in order to involve students right away.

We have already discussed all the aspects that make writing and maintaining documentation hard. We believe that the most sustainable approach consists in treating the documentation as a common and public resource: documentation does not belong to its authors. We make sure documentation is hosted in public spaces, in a way that everybody can contribute, and that new pairs of eyes check the documentation every year. By taking inspiration from Software Carpentry, we have merged common LHCb and ALICE documentation [16] in order to reduce the maintenance costs even further, and we have adopted the same GitBook structure in all our documentation websites.

There currently are limitations in organising a single yearly event at CERN: LHCb and ALICE have a relatively small size allowing them to work together and schedule the event in a way that works for both experiments. When more experiments join us, we will incur into organisational issues. In order to make the Starterkit scale and to overcome geographical limitations we need to reach out and foster decentralised events organised by communities away from CERN, similarly to what TEDx [27] events are in relation to TED [28]. Incidentally, this would also allow for more frequent Starterkits.

How much training activities are recognised as an actual work by the HEP community is, at the moment, a serious issue that holds back potential teachers from taking part. As teaching is by no means a side task, it is inevitable that the required time is taken away from other work activities. This means that teaching might not only not be rewarding for your career, but also impact it negatively in some cases, unless it is properly recognised. The Starterkit experience has taught us that teaching needs to be officially recognised by HEP experiments and the participating institutes, maybe as a "service task": we believe that everybody in HEP should try the teaching experience at least once as a service to the others, and as a way to consolidate one's knowledge as well.

Starterkit organisers are actively participating to the HEP Software Foundation efforts on training. The HSF is providing us with practical support by hosting our common documentation [16] on an experiment-independent website. More importantly, it is giving us the chance to advertise our positive joint experience, which represents one of the core points of the Community White Paper on training activities [18], and to get advice from other experiments.

References

[1] *LHCb Collaboration*, https://lhcb-public.web.cern.ch/lhcb-public/en/Collaboration/Collaboration-en.html
[2] *ALICE Collaboration*, http://alice.web.cern.ch/
[3] *SHiP experiment*, https://ship.web.cern.ch/
[4] A. Puig Navarro (LHCb), PoS **EPS-HEP2017**, 565 (2017)
[5] *The LHCb Starterkit*, https://lhcb.github.io/starterkit/

[6] H. Beck, *The Junior Community in ALICE*, in *EPS Conference on High Energy Pyhsics 2017* (2017), https://indico.cern.ch/event/466934/contributions/2589553/
[7] *The ALICE build tool: aliBuild*, https://alisw.github.io/alibuild
[8] M. Zimmermann, A. collaboration et al., *The ALICE analysis train system*, in *Journal of Physics: Conference Series* (IOP Publishing, 2015), Vol. 608(1), p. 012019
[9] *The ALICE analysis repository: AliPhysics*, https://github.com/alisw/AliPhysics
[10] *The Vidyo web conferencing tool*, https://www.vidyo.com/
[11] D. Berzano, R. Deckers, C. Grigoras, M. Floris, P. Hristov, M. Krzewicki, M. Zimmermann, *The ALICE Analysis Framework for LHC Run 3*, in *Computing in High-Energy and Nuclear Physics 2018* (2018), https://indico.cern.ch/event/587955/contributions/2938126/
[12] *Markdown*, https://daringfireball.net/projects/markdown/
[13] *GitBook*, https://www.gitbook.com/
[14] *LHCb Starterkit lessons*, https://lhcb.github.io/analysis-essentials
[15] *ALICE Starterkit and Analysis Tutorial lessons*, https://alice-doc.github.io/alice-analysis-tutorial
[16] *Common HEP lessons hosted by the HEP Software Foundation*, https://hsf-training.github.io/analysis-essentials
[17] *The HEP Software Foundation*, https://hepsoftwarefoundation.org/
[18] D. Berzano, R.M. Bianchi, P. Elmer, S.V. Gleyzer, J. Harvey, R. Jones, M. Jouvin, D.S. Katz, S. Malik, D. Menasce et al., *HEP Software Foundation Community White Paper Working Group - Training, Staffing and Careers*, arXiv:1807.02875 [physics.ed-ph] (2018)
[19] *Software Carpentry*, https://software-carpentry.org/
[20] *Creative Commons Licenses*, https://creativecommons.org/licenses/
[21] *MIT license*, https://opensource.org/licenses/MIT
[22] *Travis CI*, https://travis-ci.org/
[23] *ALICE software build instructions*, https://alice-doc.github.io/alice-analysis-tutorial/building/
[24] *Dockerfiles*, https://docs.docker.com/engine/reference/builder/
[25] *DockerHub*, https://hub.docker.com/
[26] *Mattermost*, https://mattermost.com/
[27] *TEDx*, https://www.ted.com/about/programs-initiatives/tedx-program
[28] *TED*, https://www.ted.com/