PROCEEDINGS OF SCIENCE

The LHCb Starterkit initiative

Albert Puig Navarro[*][†]
Universität Zürich (Switzerland)
E-mail: albert.puig@cern.ch

The vast majority of high-energy physicists use and produce software every day. Software skills are usually acquired on the go and dedicated training courses are rare. The LHCb Starterkit is a new training format for getting LHCb collaborators started in effectively using software to perform their research. The initiative, combining courses and online tutorials, focuses on teaching basic skills for research computing, as well as LHCb software specifics. Unlike traditional tutorials we focus on starting with basics, performing all the material live, with a high degree of interactivity, giving priority to understanding the tools as opposed to handing out recipes that work "as if by magic". The LHCb Starterkit was started by young members of the collaboration inspired by the principles of Software Carpentry, and the material is created in a collaborative fashion using the tools we teach. Three successful entry-level workshops, as well as two advanced ones, have taken place since the start of the initiative in 2015, and were taught largely by PhD students to other PhD students.

EPS-HEP 2017, European Physical Society conference on High Energy Physics
5-12 July 2017
Venice, Italy

[*] Speaker.
[†] On behalf of the LHCb Collaboration.

1. Introduction

The LHC [1] is one of the most complex machines ever build; data collected by its experiments is therefore of an extreme complexity. However, in the beginnings of the field of high-energy physics (HEP), the process of analyzing data looked very different than it does today: events were counted by hand, and discoveries were made using oscilloscopes and photographs; physical tools, such as rulers, were used to determine properties of photographed tracks. As the knowledge of particles and their interactions deepened, data taking and analysis techniques evolved and manual methods were replaced, on a large scale, by digital readout and analysis using computers. As a consequence, also the size of the collected datasets has increased over time—in parallel with the affordable computing capacity—and with it the complexity of the software stacks needed to guarantee their efficient processing.

Currently, high-energy physicists have to deal with several programming languages, such as *Python* [2] or *C++* [3], scientific software packages, such as *ROOT* [4], *Mathematica* [5], *numpy* [6] or *scikit-learn* [7], and even version control, *e.g.*, *Git*^{TM} [8], on top of each of the experiment's own software stack. They have thus become computer experts, and for people to take that step confidently, training becomes indispensable.

It is much too common, however, that computing training in high-energy physics experiments is lacking. Often, newcomers—already overwhelmed not only by new software to be learned, but also by the conventions used in each experiment—have to go through broken or incomplete tutorials, trying to access outdated, or even completely missing, documentation. Hours are wasted by many people running into the same types of problems, and experts answering the same questions, over and over again, and more often than not the final solution is to modify "inherited" scripts that are known to work, without completely understanding why.

This can be mitigated by adopting a centralized setup to provide working, up-to-date tutorials and answers to frequently asked questions. The LHCb Starterkit [9] aims to accommodate such a setup by replacing the traditional static, often-unmaintained web sites[1] by a modern, collaborative approach.

2. The LHCb Starterkit

The LHCb Starterkit [9] is an initiative started in 2015 with the aim to provide software training, by and for the LHCb collaboration members, with the following goals:

- Give a *solid starting point* to new members of the collaboration—usually young PhD students or even MSc students—in the most used software in LHCb, both general and specific to the collaboration.

- *Improve software literacy* in the experiment. Since new collaboration members come from many different backgrounds and have varying levels of experience, it is important to get everyone on the same level to make communication and problem-solving easier for everyone.

[1] TWiki pages [10], editable by all members of the collaboration, are the most used way in the LHC experiments to collect documentation and tutorials.

- *Teach good practices*, putting emphasis on the importance of documentation, in order to improve collaboration between scientists. This contributes, in the long run, to the quality and maintainability of the LHCb software stack.
- Help newcomers *socialize and integrate* in the collaboration through the organization of workshops. This allows young collaboration members to get to know each other, as well as some of the experts in key areas of the software; this is further effectuated through social events during those workshops. Achieving a good socialization fosters a climate of communication and makes the young members feel welcome to ask questions and share their ideas.

These goals are achieved through the creation and maintenance of up-to-date tutorials and the organization of workshops where participants obtain hands-on experience with these tutorials under the supervision of more experienced collaborators.

The project was started (and is still run) by early career scientists, mostly at PhD and MSc level, and all work is done on a voluntary basis; this implies a heavy rotation of the people involved. In this situation, knowledge transfer is ensured by using industry-standard tools—*Slack* [11], *Mattermost* [12], *Google Docs* [13], *Github* [14], etc—to publish the tutorials and keep track of organizational details. Additionally, to ensure the survival of the setup, former attendees of the workshops are encouraged to take responsibilities within the initiative in subsequent editions. Knowledge and experience is continuously transferred, and the project endures.

While one of the main outcomes of the Starterkit initiative is a central tutorial repository, the organization itself is entirely decentralized. There is no strict hierarchy within the group of people working on the Starterkit—decisions are made democratically. Because there is no single person in charge, work can always continue with people present at any particular time.

2.1 The tutorials

The LHCb software, while based on relatively simple building blocks, is quite intimidating to newcomers and has a steep learning curve. A very common practice for new students is to receive scripts from their supervisors or colleagues and hack them to try to adapt them to their needs; this approach is very error-prone and results in many questions on mailing lists that could have been easily avoided with a good basic training on the principles of the software stack. To solve this problem, the Starterkit software tutorials focus on teaching the building blocks of the software, giving the attendees/readers the tools necessary to build on and perform complex tasks later on. To do so, they start from the basics and move on, step by step, to more complicated material, always providing full examples of working code.

The tutorials, inspired—both in structure and content—by the well-established Software Carpentry tutorials [15], are freely accessible, both within and outside of the LHCb collaboration [16]. They are hosted as *Github* webpages [17], available under a Creative Commons Attribution License [18], and anyone can (and is encouraged to) submit issues and pull requests to update them. This open and collaborative approach offers two advantages: on one side, its openness makes it very easy to collaborate, and as a consequence many LHCb experts, as well as users, have contributed to its improvement; on the other, the public availability of the tutorials and the use of a tool like *Github* to maintain them has ensured that they have stayed up-to-date and in an optimal shape, even when significant changes have occurred in the LHCb software stack. In total, more than 250 issues and more than 100 pull requests have been received from around 50 contributors.

2.2 The workshops

Once per year, the *Starterkit workshop* is held at CERN. It has two main organizers, which change every time and are usually drafted from previous Starterkit workshop participants, with the support from 10–15 volunteer instructors and assistants. The target audience of the workshop, limited to 40 students, are PhD and MSc students who are new to the collaboration; as the LHCb collaboration has about 80 new students each year, the workshop accepts enough students to make a noticeable impact on the collaboration as a whole.

The workshop, which takes four days, is divided in two sections: the first section covers general computing tools, while the second one focuses on LHCb-specific software. A hands-on approach, inspired by the Software Carpentry Workshop Operations guidelines [19], is followed: the participants follow a main instructor in completing each of the tutorials, always prioritizing the interactivity, and can ask for help at any time from the 3–4 assistants[2] in the room. The high ratio of experts to students, one per 4–5 students, ensures this interactivity and level of help can be maintained even in the most complex lessons, at the cost of limiting the attendance to the workshop. A social event is also held during the workshop, with the aim to help establish crucial contacts among peers and to ease their way into a new collaboration.

Since 2016, a follow-up workshop, called the *Impactkit*, is held once a year roughly six months after the Starterkit, to cover advanced LHCb-specific software tutorials. Twenty students, mostly participants of previous workshops, attend advanced lectures on LHCb computing topics, taught in a similar way as in the Starterkit workshop, and participate in a *hackathon*. In it, they are tasked, guided by experts, with solving real problems involving the LHCb software over the course of one day, with the final aim to reach solutions to be used in production. These challenges represent the crystallization of the teachings and goals of the Starterkit initiative: having started six months prior in the Starterkit, students are able to make meaningful, well-documented contributions to the very complex LHCb software stack in an autonomous way.

3. Conclusion

The Starterkit initiative, now a consolidated project within the LHCb collaboration, has been a huge success and has received overwhelmingly positive feedback, proving the need for a centrally organized, functioning educational system within HEP collaborations.

It has eased new collaboration members into their career within HEP, and has prevented them from feeling lost in the myriad of software that is used in the everyday life of an experimentalist. The tutorials are being actively followed and updated, and the workshops, which have already received over a hundred participants, are always oversubscribed.

As the complexity of software continues to increase and the requirements on experimental physicists continue to become to become more demanding, we hope similar initiatives arise in other experiments.

[2] *Helpers* in Software Carpentry language.

4. Aknowledgements

The Starterkit Team wants to thank all the people that have contributed to making it a reality, both through tutorial writing, checking and updating, and by helping or teaching in the workshops. Special thanks also go to the LHCb Secretariat, which have helped in many organizational aspects. A.P. was supported by the SNSF Ambizione Grant PZ00P2_168169.

References

[1] L. Evans and P. Bryant, *LHC Machine*, *JINST* **3** (2008) S08001.

[2] "Python Software Foundation. Python Language Reference." Available at `http://www.python.org`.

[3] International Organization for Standardization (ISO), *International Standard ISO/IEC 14882:2014(E) – Programming Language C++*, 2014.

[4] R. Brun and F. Rademakers, *ROOT: An object oriented data analysis framework*, *Nucl. Instrum. Meth.* **A389** (1997) 81–86.

[5] Wolfram Research, Inc., "Mathematica, Version 11.2."

[6] S. van der Walt, S. C. Colbert and G. Varoquaux, *The NumPy Array: A Structure for Efficient Numerical Computation*, *Comput Sci Eng* **13** (2011) 20–30.

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel et al., *Scikit-learn: Machine Learning in Python*, *Journal of Machine Learning Research* **12** (2011) 2825–2830.

[8] "Git distributed version control system." Available at `https://git-scm.com/`.

[9] "LHCb Starterkit." `http://lhcb.github.io/starterkit`.

[10] "TWiki – the Open Source Enterprise Wiki and Web Application Platform." `http://twiki.org/`.

[11] "Slack." `https://slack.com`.

[12] "Mattermost private cloud messaging." `https://about.mattermost.com/`.

[13] "Google Docs." `https://www.google.com/docs/about/`.

[14] "GitHub." `http://www.github.com`.

[15] G. Wilson, "Software carpentry: Lessons learned." `http://f1000research.com/articles/3-62/v2`. 10.12688/f1000research.3-62.v2.

[16] "The LHCb Starterkit lessons." `https://lhcb.github.io/starterkit-lessons/`.

[17] "The LHCb Starterkit lessons Project." `https://github.com/lhcb/starterkit-lessons`.

[18] "Creative Commons Attribution 4.0 International License." `https://creativecommons.org/licenses/by/4.0/legalcode`.

[19] "Software Carpentry Workshop Operations." `https://software-carpentry.org/workshops/operations/`.