

Publication of the Belle II Software

Thomas Kuhr^{1,*}, *Fabrizio Bianchi*^{2,**}, *Giacomo De Pietro*^{3,***}, *David Jaffe*^{4,****}, *Frank Meier*^{5,†}, *Martin Seviour*^{6,‡}, and *Philipp Urquijo*^{6,§}

¹Ludwig-Maximilians-Universität München, Germany

²Universita di Torino and INFN, Italy

³Karlsruhe Institute of Technology (KIT), Germany

⁴Brookhaven National Laboratory, USA

⁵Duke University, USA

⁶The University of Melbourne, Australia

Abstract. The Belle II software was developed by a few hundred individual contributors over several years. Following the rising desire of making it publicly available, the collaboration established open source software policies and procedures. The political and technical challenges and their solutions at Belle II are discussed in this article. With the publication of the Belle II software, basf2, on GitHub and Zenodo in 2021 an important milestone towards open science was reached.

1 Introduction

Belle II [1] is a next-generation B factory experiment at the e^+e^- collider SuperKEKB [2] in Tsukuba, Japan. The collaboration was officially founded in 2008 with the goal to study B mesons, charm hadrons, and tau leptons [3] with a dataset about two orders of magnitude larger than the one collected by its predecessor. At about that time, also a coordinated development of software for the experiment started. Initially, only a few people contributed to the software that was maintained in a subversion repository.

In the first years, while the topic gained attention in the particle physics community, there were some discussions about making the software available outside the collaboration under an appropriate open source license. While the general idea was received well by many contributors to the software, there were also concerns about the required effort and the implications, in particular regarding the use of external (proprietary) software. To not be limited or impeded by any external constraints it was decided at that time to not publish the Belle II software.

*e-mail: Thomas.Kuhr@lmu.de

**e-mail: fabrizio.bianchi@to.infn.it

***e-mail: giacomo.pietro@kit.edu

****e-mail: djaffe@bnl.gov

†e-mail: frank.meier@duke.edu

‡e-mail: martines@unimelb.edu.au

§e-mail: purquijo@unimelb.edu.au

2 Task Forces

Seeing more and more experiments following the trend of making their software open source, the topic was rediscussed at Belle II. In 2017 a Licensing Task Force was created with members Carl Rosenfeld (chair), Martin Ritter (expert for external software), Martin Seviar, Michel Jouvin, Rok Pestotnik, Thomas Kuhr, Torben Ferber (expert for generators software), Thomas Hauth, Frank Meier, and Marko Bracko. One of the main results was an overview of the licenses of the packages on which the Belle II software relies. The current list can be found in the *externals* git repository [4]. The list is long and contains many different licenses, including permissive ones like MIT and Apache, copyleft licenses like GPL, and custom licenses like Geant4 and EPICS. This makes it difficult to achieve compatibility with all of them.

The Licensing Task Force also established contact with colleagues at CERN. One of the concepts that had proven to work well for the LHC experiments was to transfer the ownership of the software to CERN. This makes it easier to change the license in the future, if needed. At Belle II a transfer of ownership to KEK, the host laboratory of the experiment, or DESY, that hosts the collaborative services including the software repositories, or even CERN, was considered. But none of these routes turned out to be viable. One concern was liability for changes over which the laboratory has no direct control.

Nevertheless, the topic was pushed further by an Open Source Task Force founded in May 2019 with members Fabrizio Bianchi, David Jaffe, Thomas Kuhr (chair), Martin Seviar, and Phillip Urquijo. It was mandated by the Belle II Executive Board to draft a policy that defines software categories, documents advantages and disadvantages of making Belle II software public, and describes procedures for making Belle II software partially or fully public.

While there are of course different categories of software it turned out that an explicit definition is not really needed for the policy. Advantages of making Belle II software public that are mentioned in the task force report are:

- it gives the deserved credit to (in particular young) people,
- it allows proper reuse of code,
- it allows collaboration with people outside Belle II,
- it follows good scientific practices of publishing the used methods,
- it may improve software quality,
- it gives access to open source services,
- it provides access without authentication and searchability with search engines,
- it is a prerequisite for open data,
- it makes the software available to those who funded it.

The discussed disadvantages are:

- scientific intellectual property may be exposed in a harmful way for Belle II,
- it may pose security risks,
- there may be licensing or copyright issues,
- inappropriate content may be published,
- it may increase the reluctance to commit code.

The task force also performed a survey to get an impression of what the collaboration thought about open source. As can be seen in Fig. 1, there was a strong support for open source software.

What is your opinion on open source software?

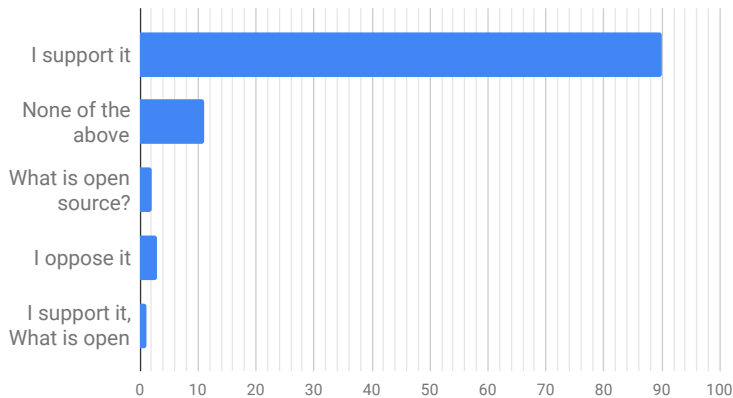


Figure 1. Result of a survey on open source software among Belle II members.

3 Belle II Policy

The policy drafted by the Open Source Task Force aims on the one hand at making sure that the benefits for the collaboration and its members are preserved. On the other hand the procedure for making Belle II software public should be feasible with reasonable effort. The policy that was finally approved by the Belle II Institutional Board in June 2020 and can be found on the public Belle II pages [5] reads as follows:

Belle II strongly profits from, recognizes the value of, and supports open source software. By committing code to a Belle II repository the right to be recognized as original author is granted and the author allows the members of the collaboration to use and modify it.

A Belle II repository can be declared public if all contributors agree that it can be distributed under an open source license. By committing code to a public Belle II repository contributors agree to the publication of their code and affirm compliance with the code of conduct. The copyright remains with the contributors.

The solution of leaving the copyright with the contributors is not only imposed by the lack of an institute willing to take the ownership, but also seems more likely to be compatible with the different rules of the various institutes, employers, funding agencies, and countries.

4 Contributors Agreement

After the policy was approved the next step was to follow the procedure defined by it for making basf2 [6] — the Belle II software for simulation, reconstruction, and analysis, used offline and on the high-level online triggering system — public. The scripts to install and set up basf2, to build its external dependencies, and to manage its versions were developed by only a few people. So it was easy to get their approval for a publication under the MIT license. However, the basf2 software was developed by a few hundred individual contributors over many years. To get their approval a merge request was set up that explained the reasons

and the implications of a publication (or non-publication). The contributors agreement they could decide on reads

By contributing to this repository you

- *affirm that your contribution to the Belle II software complies with the Belle II code of conduct [7],*
- *grant the members of the Belle II collaboration the right to use and modify your contribution,*
- *grant the Belle II collaboration the right to distribute your contribution under an open source license of its choice,*
- *provide the contribution "as is" without warranty.*

As many contributors already left the collaboration and did not have access to the merge request any more it was tried to reach them by email. In total there were 258 approvals and no disapprovals. No reply was received from 94 persons who contributed together less than 10% of the code. To proceed with the publication of basf2 it was brought up for vote in the Belle II Institutional Board; 72 voted in favor, 3 opposed, and 9 abstained.

5 Technical Aspects

Having passed the political hurdles, the repository must be prepared for the publication. A license file was added and a license statement added or updated in the source files:

```
/*
 * basf2 (Belle II Analysis Software Framework)
 * Author: The Belle II Collaboration
 *
 * See git log for contributors and copyright holders.
 * This file is licensed under LGPL-3.0, see LICENSE.md.
 */
```

To make sure there are no issues with generator code that was included in basf2 the authors of the code were contacted. Markdown files giving credit to the original authors were added where appropriate. A potential license conflict was identified by Xilinx code in the repository. Fortunately, it was not used any more in the current code, but it was still in the history.

To avoid that unwanted code got published in the history the following solution was implemented, which is also illustrated in Fig. 2. A new repository is started from the clean code at the head. This repository is then to be published. To connect it with the history in the original repository the `git replace` command is used to replace the first commit in the new repository with the last one in the original one. A `git log` command then shows the full history. A copy of the original repository is cleaned with the `git-filter-repo` tool [8] to remove the unwanted files. The cleaned history can then be published and connected to the new one in the same way with the `git replace` command. The tool `b2code-history` is provided for users to easily get and connect the history. One disadvantage is that the history replacement only works for locally checked out versions and not for web-based tools such as GitHub or GitLab.

The worry that inappropriate content in the code or commit messages may be made public was paid particular attention to. The tools `profanity_check` [9] and `better_profanity` [10] were used to search for such content. As they produce many false positives and do not take into account the context (e.g. for the word "strip" in the context of

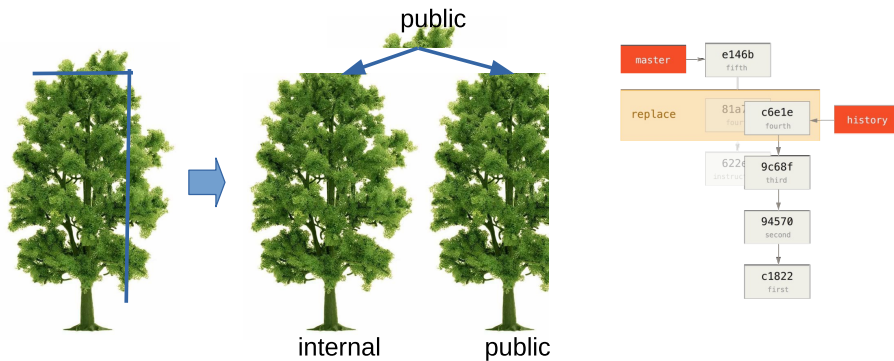


Figure 2. Illustration of history cleaning solution for the publication of the Belle II software.

a silicon detector) a manual review of the reported issues is mandatory. In addition, names of collaborators were searched for in the code and it was checked that they did not get mentioned in ways they would feel uncomfortable with. Because what is considered inappropriate depends on the cultural background the Belle II Diversity Officers helped to assess the found cases. No very serious cases were found, but a few improvements were implemented. It seems that in general the required code reviews for merging changes is effective in avoiding inappropriate content.

6 Release

The Belle II software was finally released on GitHub and Zenodo in July 2021. Together with the tools, the externals, and the versioning, it can be found under <https://github.com/belle2>.

The cleaned history of the main branch can be found in the `history` branch. On GitHub only the main branch and releases are published. This is done manually from time to time and it is planned to automate the procedure in the future. The development is still done in a repository only accessible to Belle II members. While this may lower the barrier for committing code and discussing issues in merge requests, it makes it harder for non-members to contribute. A further adjustment of the software development organization may be discussed in future to address this shortcoming.

As the Sphinx and Doxygen documentation of the Belle II software can be generated from the source code, it was decided to also make the centrally produced documentation publicly available. It can be found at <https://software.belle2.org/>. Not all of it has been adjusted for using the public instead of the internal software repository. Some work remains to be done in this instance.

7 Impact

To our knowledge there were no negative consequences of publicly releasing the Belle II software. The risks discussed in Sec. 2 have not hit us in any substantial way so far.

On the positive side we see citations of the software itself [11]. It gives more accurate and up-to-date credits to contributors and better shows what was actually used, compared to the citation of the paper describing the core components of basf2 [6]. Because the software citations are not tracked in tools such as InspireHEP it is, however, harder to have an overview of, and count, the citations.

Another benefit of having the software stack open source is to be able to participate in the Google Summer of Code [12] where students are paid by Google to work on open source software projects. Belle II participated under the HEP Software Foundation [13] as Organization in 2022 and 2023. In 2022 the student implemented significant improvements in the validation framework [14] and afterwards started a PhD in Belle II. In the following year the student implemented the first web-based Belle II event display [15, 16] using the Phoenix framework [17].

There were also requests to update the repository on GitHub to support job applications. Whether this had a positive effect we don't know.

8 Conclusions

It is generally known that publishing software gets harder the longer one waits. What can be learned from the Belle II experience is that it can still be done even for software that was developed by hundreds of people over several years. It is a lot of effort, but we think it was worth it. It is a step towards open science and the basis for going further steps in that direction.

References

- [1] Abe, T. *et al.* (Belle II Collaboration), *Belle II technical design report*, KEK-REPORT-2010-1, arXiv:1011.0352 (2010).
- [2] Akai, K., Furukawa, K., Koiso, H. (SuperKEKB Accelerator Team), *SuperKEKB collider*, Nucl. Instrum. Methods Phys. Res., Sect. A **907**, 188 (2018).
- [3] Kou, E. *et al.*, *The Belle II physics book*, Prog. Theor. Exp. Phys. **2019**, 123C01 (2019).
- [4] *basf2 Externals / Included Software Packages*, <https://github.com/belle2/externals/blob/main/README.md#included-software-packages> (visited on 2023/11/10)
- [5] *Belle II Open Source Software Policy*, <https://confluence.desy.de/display/BI/Belle+II+Open+Source+Software+Policy> (visited on 2023/11/10).
- [6] Kuhr, T., Pulvermacher, C., Ritter, M., Hauth, T. Braun, N., *The Belle II Core Software*, Comput. Softw. Big Sci. **3**, 1, 1 (2019).
- [7] *Belle II Code of Conduct*, <https://confluence.desy.de/download/attachments/230767468/Belle%20II%20bylaws.pdf> (visited on 2023/11/10).
- [8] *git filter-repo*, <https://github.com/newren/git-filter-repo> (visited on 2023/11/10).
- [9] *profanity-check*, <https://github.com/vzhou842/profanity-check> (visited on 2023/11/10).
- [10] *better-profanity*, https://github.com/snguyenthanh/better_profanity (visited on 2023/11/10).
- [11] Belle II Collaboration, *Belle II Analysis Software Framework (basf2)*, <https://doi.org/10.5281/zenodo.5574115>.
- [12] *Google Summer of Code*, <https://summerofcode.withgoogle.com/> (visited on 2023/11/10).
- [13] *HEP Software Foundation*, <https://hepsoftwarefoundation.org/> (visited on 2023/11/10).
- [14] Arul Prakash, *GSoC blog: Advanced Belle II Software Validation*, https://hepsoftwarefoundation.org/gsoc/blogs/2022/blog_Belle2_ArulPrakash.html (visited on 2023/11/10).
- [15] Le Cong Minh Hieu, *GSoC blog: Belle II Event Display with Phoenix*, https://github.com/HieuLCM/hsf.github.io/blob/main/_gsocblogs/2023/blog_Belle2_HieuLeCongMinh.md (visited on 2023/11/10).

- [16] Le Cong Minh Hieu, <https://github.com/belle2/display> (visited on 2023/11/10).
- [17] Moyse, E., Ali, F., Cortina, E., Bianchi, R. M., Couturier, B., *The Phoenix event display framework*, EPJ Web Conf. 251 (2021) 01007; Ali, F. *et al.*, *HSF/phoenix*, <https://doi.org/10.5281/zenodo.2586998>.