Funded by
the European Union

Training and Documentation: LHCb’s new approaches

Andy Morris

Aix Marseille Univ, CNRS/IN2P3, CPPM, Marseille, France

Andy.M@cern.ch – [23/Jun/25]

LHCb produces a lot of software!

However this is common to any large physics collaboration

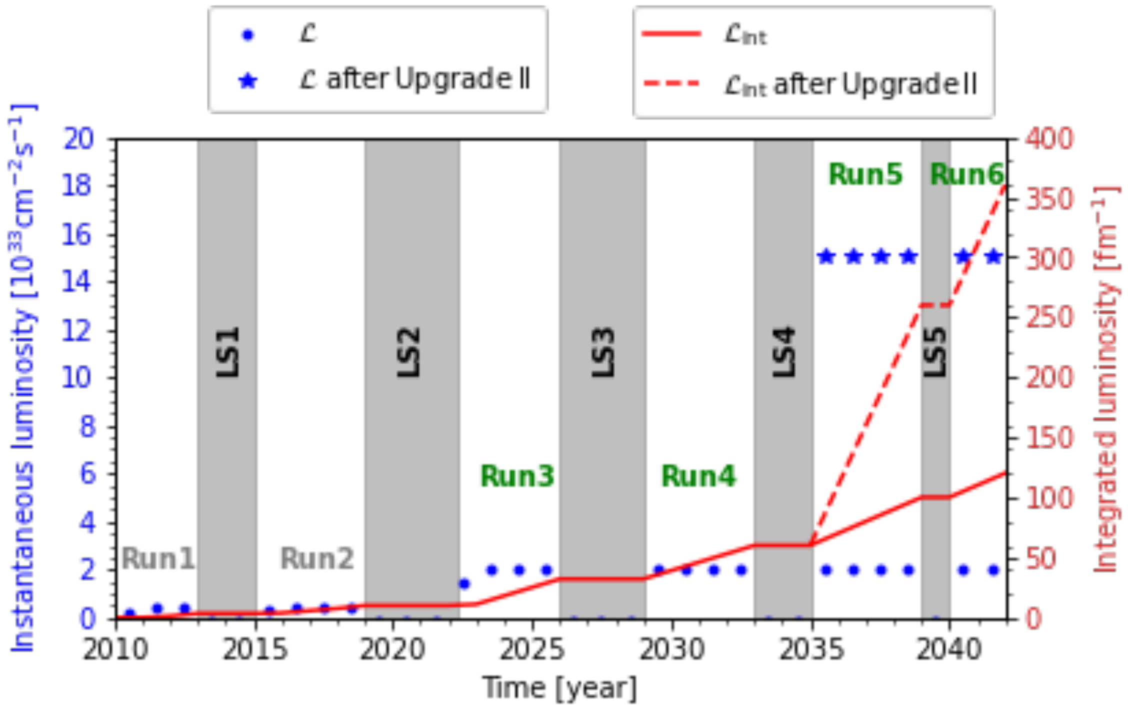
- Software to process the data
- Software to monitor data taking
- Software to analyse the data
- Software to produce simulation
- Software to reweight simulated samples

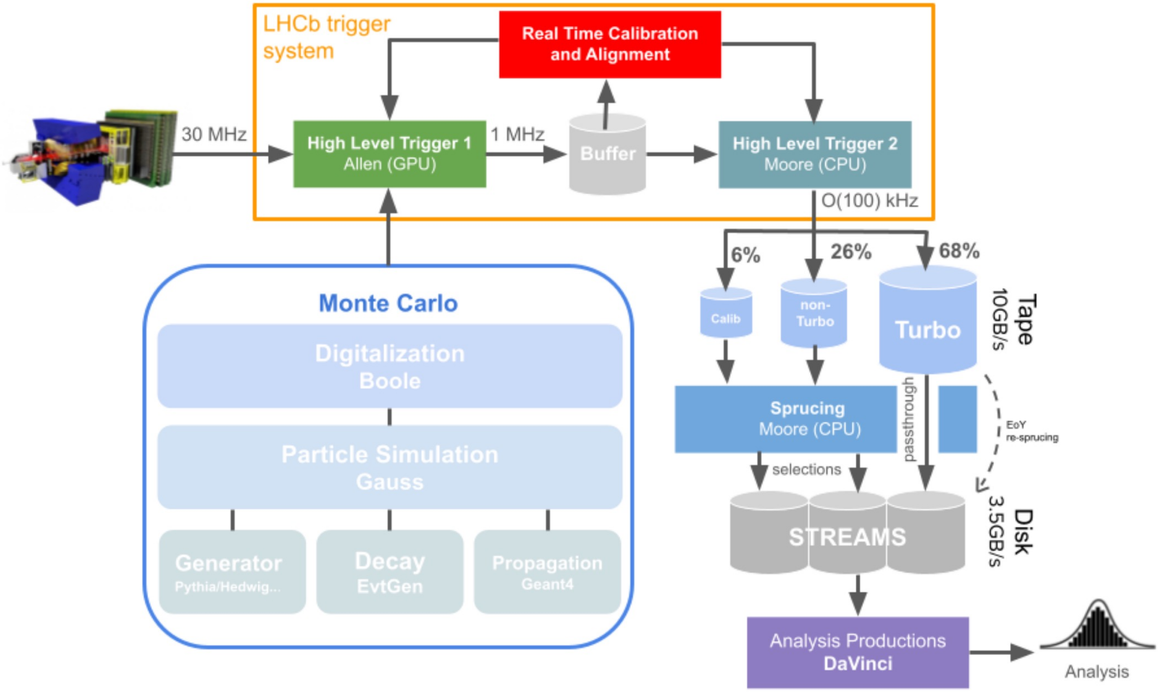
⋮

All of this needs documentation!

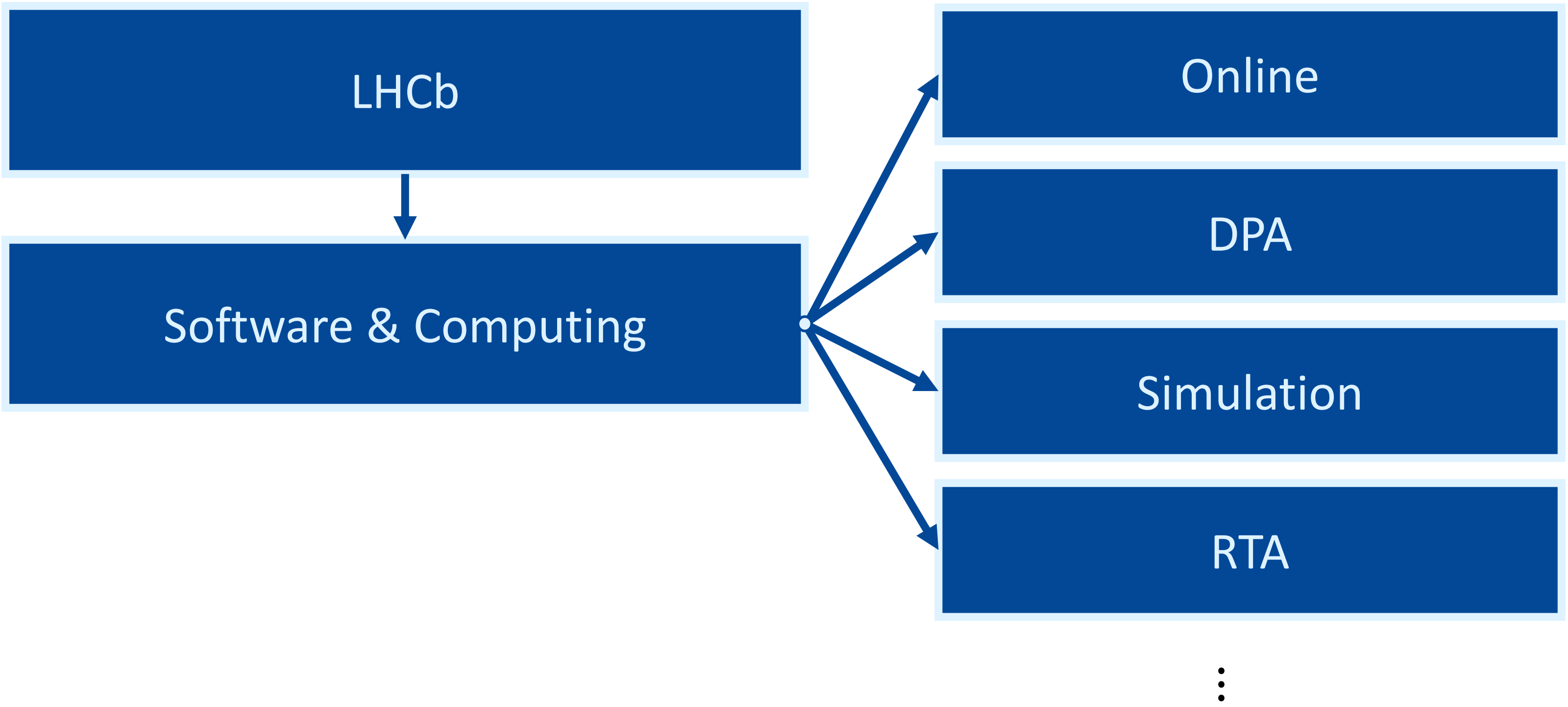
LHCb had an upgrade!

- For Run 3 of the LHC (2022-2026) LHCb was upgraded
  - This allowed for higher luminosity to be measured
  - However this also necessitated mostly new software



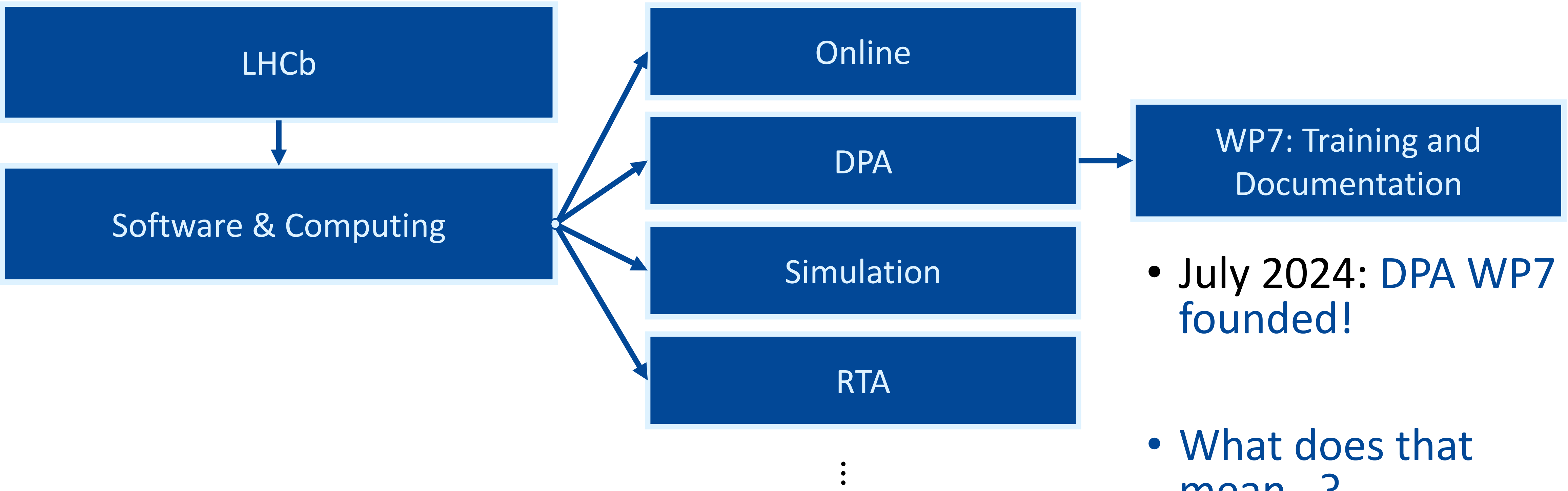


The structure of software projects in LHCb



- LHCb produces a lot of software through its S&C groups
  - These split into 6 work packages (WPs)
- Software needs documentation or end users will struggle to use it!
- For a long time training and documentation was done fairly ad hoc…

DPA WP7?



- July 2024: DPA WP7 is founded!

- What does that mean…?

DPA WP7!

- The remit was given:

Responsibilities:

- *Manage the StarterKit(s) material and oversee events ensuring their suitability/quality as teaching resources*
    - *Maintenance of the Run 1/2 StarterKit*
        - *Prompting experts to update outdated material*
        - *Reviewing and merging pull requests*
    - *Coordinate the creation of Run 3 StarterKit*
        - *Organising the effort, structure and direction (not writing the material itself)*
    - *Ensuring the delivery of quality StarterKit events*
        - *Run by younger collaboration members*
        - *Aligned with software pedagogical best practices*
- *Advocate for and facilitate software training within LHCb (events, materials, help channels…)*

Work with:

- *HSF training group and other LHC experiments for resources and events*
- *RTA, DPA, Simulation, Computing for creation and upkeep of material*

DPA WP7!

- The remit was given:

Responsibilities:

- *Manage the StarterKit(s) material and oversee events ensuring their suitability/quality as teaching resources*
    - *Maintenance of the Run 1/2 StarterKit*
        - *Prompting experts to update outdated material*
        - *Reviewing and merging pull requests*
    - *Coordinate the creation of Run 3 StarterKit*
        - *Organising the effort, structure and direction (not writing the material itself)*
    - *Ensuring the delivery of quality StarterKit events*
        - *Run by younger collaboration members*
        - *Aligned with software pedagogical best practices*
- *Advocate for and facilitate software training within LHCb (events, materials, help channels…)*

Work with:

- *HSF training group and other LHC experiments for resources and events*
- *RTA, DPA, Simulation, Computing for creation and upkeep of material*

STARTERKIT

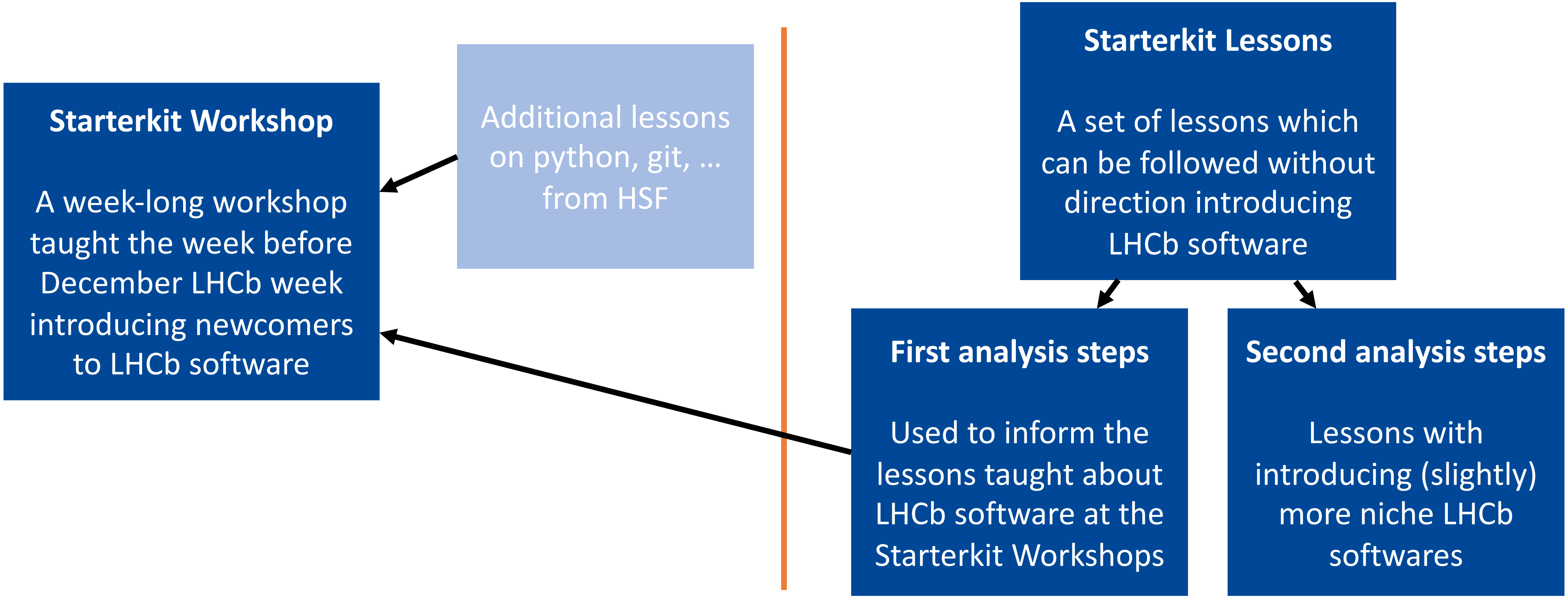
LHCb

Estd. 2015

- Make sure the Starterkit stays up-to-date
    - The primary method of onboarding in LHCb
- Produce the Run 3 Starterkit and make sure *that* stays up-to-date too
- Make sure the Starterkit events are high quality

… What’s the Starterkit?

- Two things! Both aimed at onboarding



Starterkit Workshop

A week-long workshop taught the week before December LHCb week introducing newcomers to LHCb software

Additional lessons on python, git, … from HSF

Starterkit Lessons

A set of lessons which can be followed without direction introducing LHCb software

First analysis steps

Used to inform the lessons taught about LHCb software at the Starterkit Workshops

Second analysis steps

Lessons with introducing (slightly) more niche LHCb softwares

… What's the Starterkit?

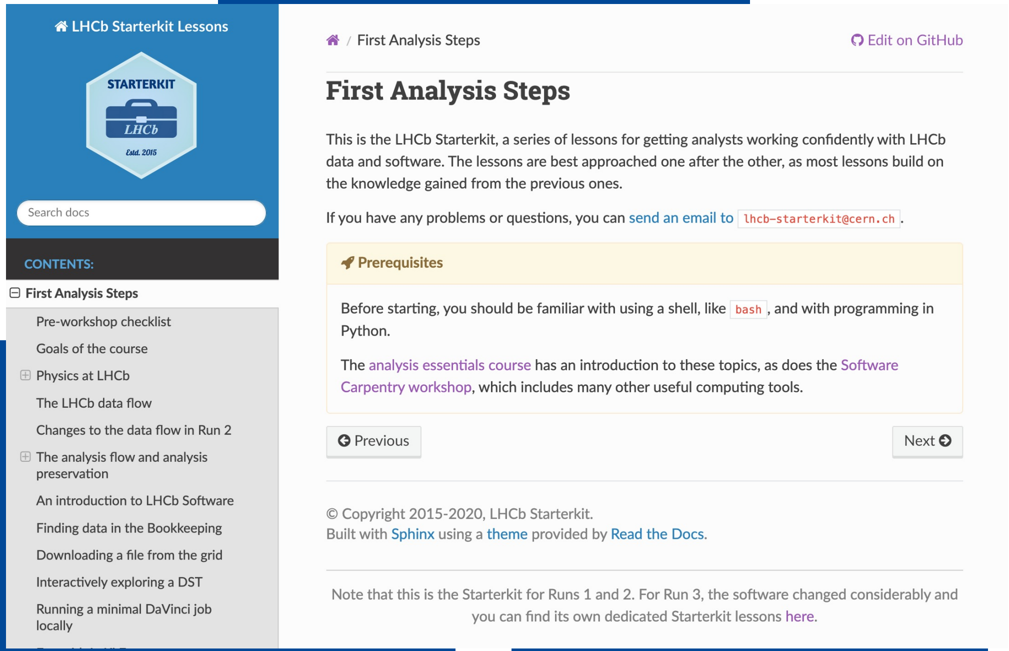
- Two things! Both aimed at onboarding

S

A
ta
D
in

ons
, …

Starterkit Lessons



LHCb Starterkit Lessons

Search docs

CONTENTS:

- First Analysis Steps
  - Pre-workshop checklist
  - Goals of the course
  - Physics at LHCb
  - The LHCb data flow
  - Changes to the data flow in Run 2
  - The analysis flow and analysis preservation
  - An introduction to LHCb Software
  - Finding data in the Bookkeeping
  - Downloading a file from the grid
  - Interactively exploring a DST
  - Running a minimal DaVinci job locally

First Analysis Steps

Edit on GitHub

First Analysis Steps

This is the LHCb Starterkit, a series of lessons for getting analysts working confidently with LHCb data and software. The lessons are best approached one after the other, as most lessons build on the knowledge gained from the previous ones.

If you have any problems or questions, you can send an email to `lhcb-starterkit@cern.ch`.

Prerequisites

Before starting, you should be familiar with using a shell, like `bash`, and with programming in Python.

The analysis essentials course has an introduction to these topics, as does the Software Carpentry workshop, which includes many other useful computing tools.

Previous | Next

© Copyright 2015-2020, LHCb Starterkit.
Built with Sphinx using a theme provided by Read the Docs.

Note that this is the Starterkit for Runs 1 and 2. For Run 3, the software changed considerably and you can find its own dedicated Starterkit lessons here.

Starterkit Workshops

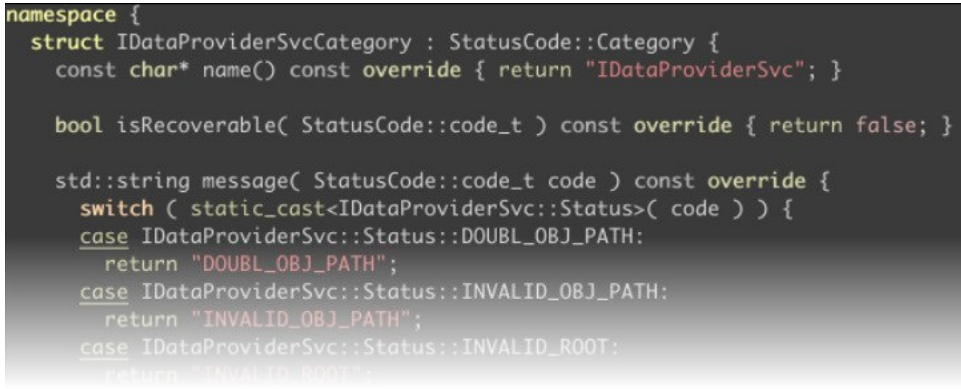
softwares

Today:

- Let’s discuss training and documentation in LHCb
  - Where it’s good, where it’s less good, and how it’s improved

- Let’s discuss training and documentation in other places
  - How and why it might be different to our approach at LHCb – what general steps can apply to everyone

What people think is hard



```
namespace {
  struct IDataProviderSvcCategory : StatusCode::Category {
    const char* name() const override { return "IDataProviderSvc"; }

    bool isRecoverable( StatusCode::code_t ) const override { return false; }

    std::string message( StatusCode::code_t code ) const override {
      switch ( static_cast<IDataProviderSvc::Status>( code ) ) {
      case IDataProviderSvc::Status::DOUBL_OBJ_PATH:
        return "DOUBL_OBJ_PATH";
      case IDataProviderSvc::Status::INVALID_OBJ_PATH:
        return "INVALID_OBJ_PATH";
      case IDataProviderSvc::Status::INVALID_ROOT:
        return "INVALID_ROOT";
```

What’s actually hard

```
// Writing a single comment
```

Disclaimer

- Today’s talk will be mostly opinion-based

- The opinion is informed from having worked with LHCb’s training and documentation for a long time but it’s not objective fact

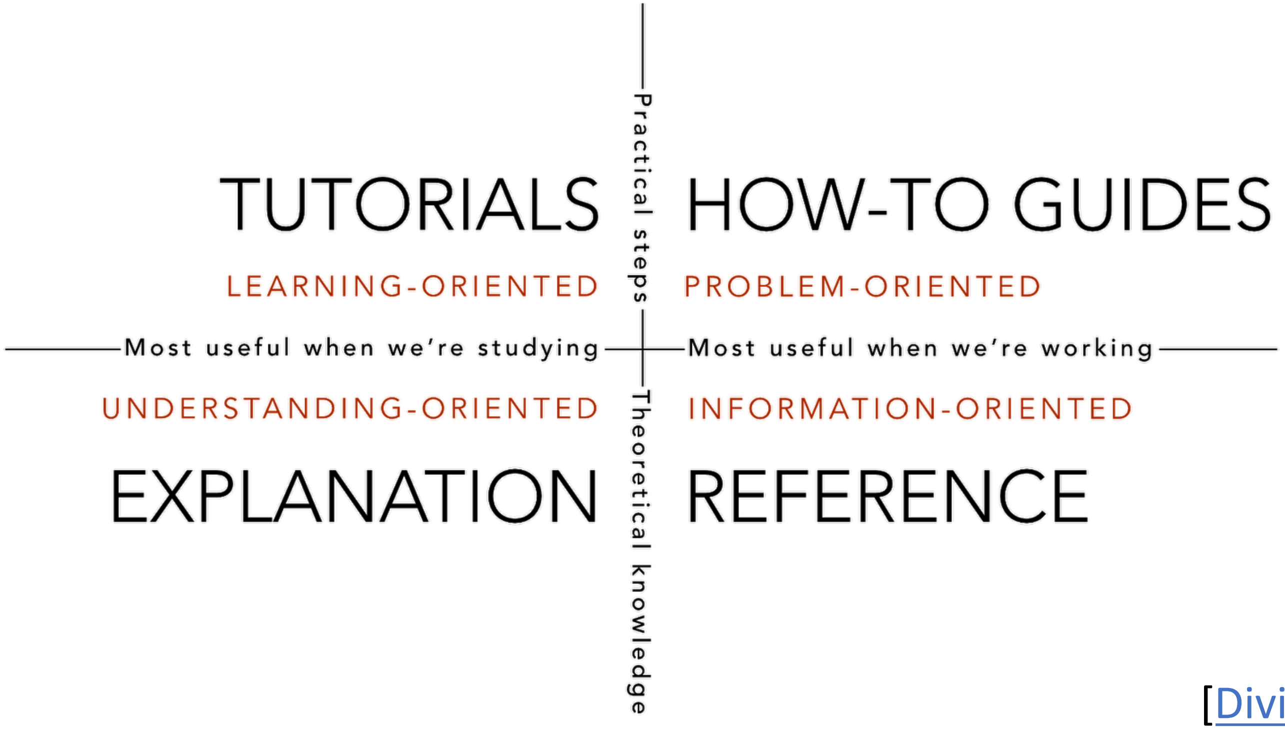
- The views and opinions in this talk do not necessarily reflect those of other members of LHCb-DPA or LHCb generally – they are purely my own. This talk is for educational purposes only with the intention that you may reflect on how your collaboration handles its training and documentation

Documentation

We’ll go over training in a bit

It’s a broad topic!

- … Even if we limit it purely to software



TUTORIALS

LEARNING-ORIENTED

HOW-TO GUIDES

PROBLEM-ORIENTED

Practical steps

Most useful when we’re studying

Most useful when we’re working

UNDERSTANDING-ORIENTED

EXPLANATION

INFORMATION-ORIENTED

REFERENCE

Theoretical knowledge

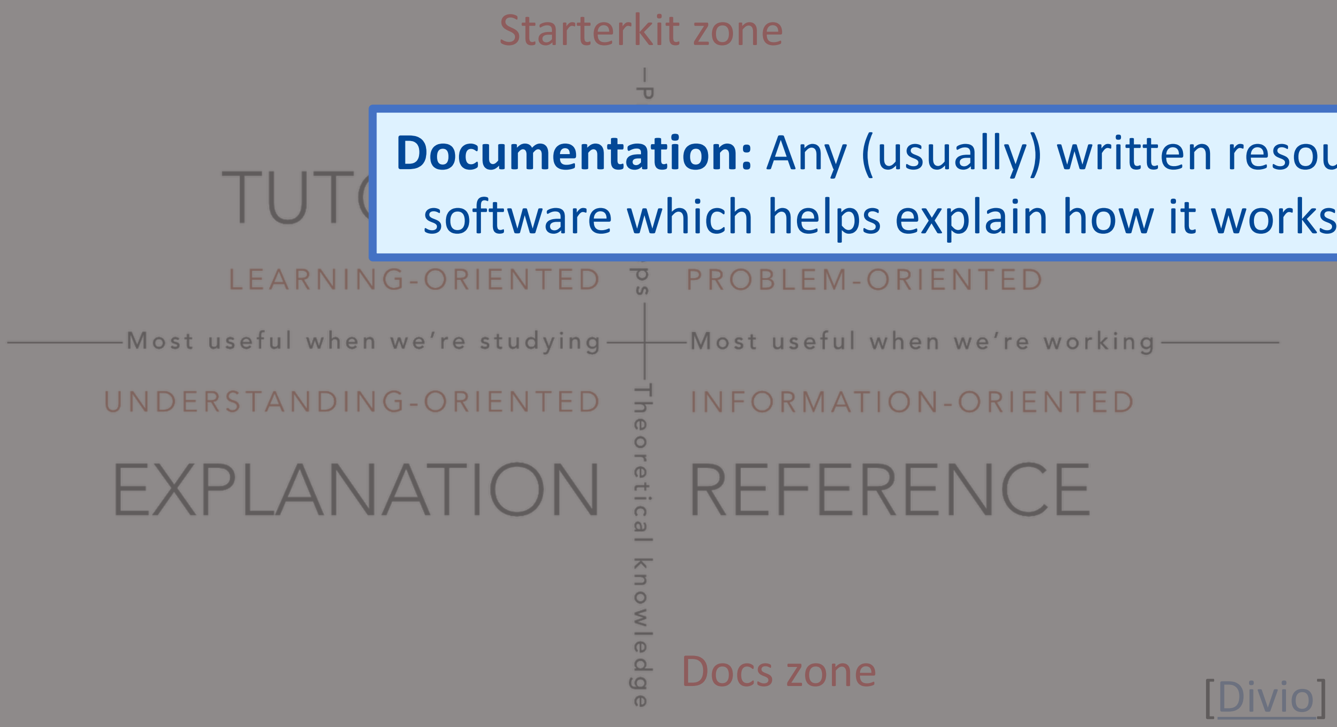
[Divio]

… But there are some common aspects I want to talk about:

- Discoverability
- Maintainability
- Detail

It’s a broad topic!

- … Even if we limit it purely to software



Documentation: Any (usually) written resource around LHCb software which helps explain how it works or how to use it

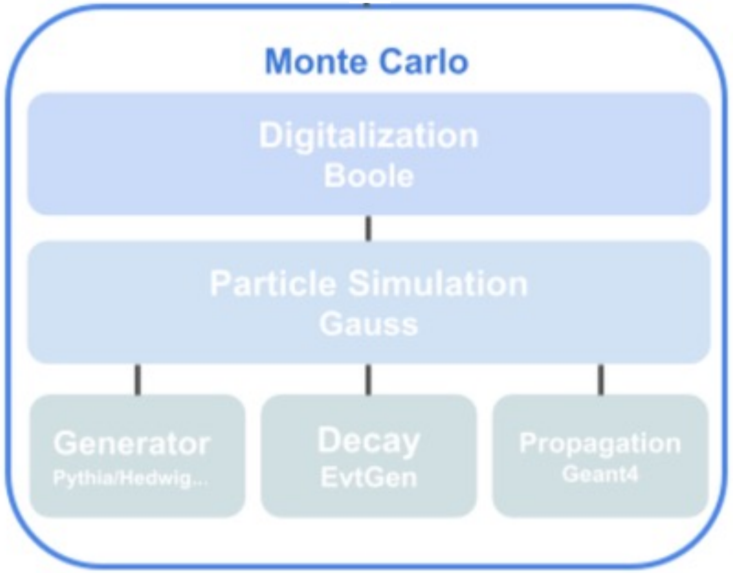
some common aspects I
ut:

- Discoverability
- Maintainability
- Detail

Discoverablility

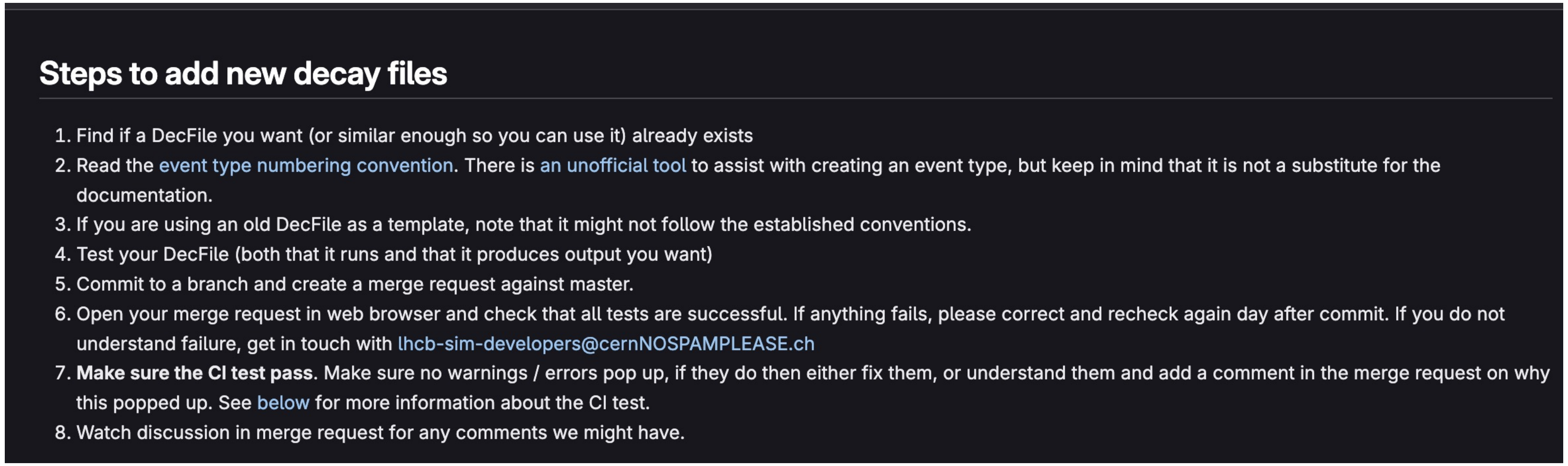
- If you write your docs and no-one can find them that’s a problem
  - LHCb documentation is spread around a lot
  - Some parts open-source, some parts closed-source – separate docs
  - Some parts of software are wrappers for other parts of software…
  - To RTFM, one must first *find* TFM

- This particularly was an issue for the Simulation group
  - People would complain
    “Simulation doesn’t have documentation”
    - This isn’t true! But finding it could be tricky (and has improved a lot)



Discoverablility

- As an example – Writing a dec file:
  - Configuration file for EvtGen
  - Detailed docs exist in the repo under [CONTRIBUTING.md]
    - Repo lives in lhcb-datapkg/Gen group – not Simulation – you have to know that!
    - People don't necessarily know what they're looking for – a paper, a twiki, a website, …
    - How did Simulation improve their discoverability here?

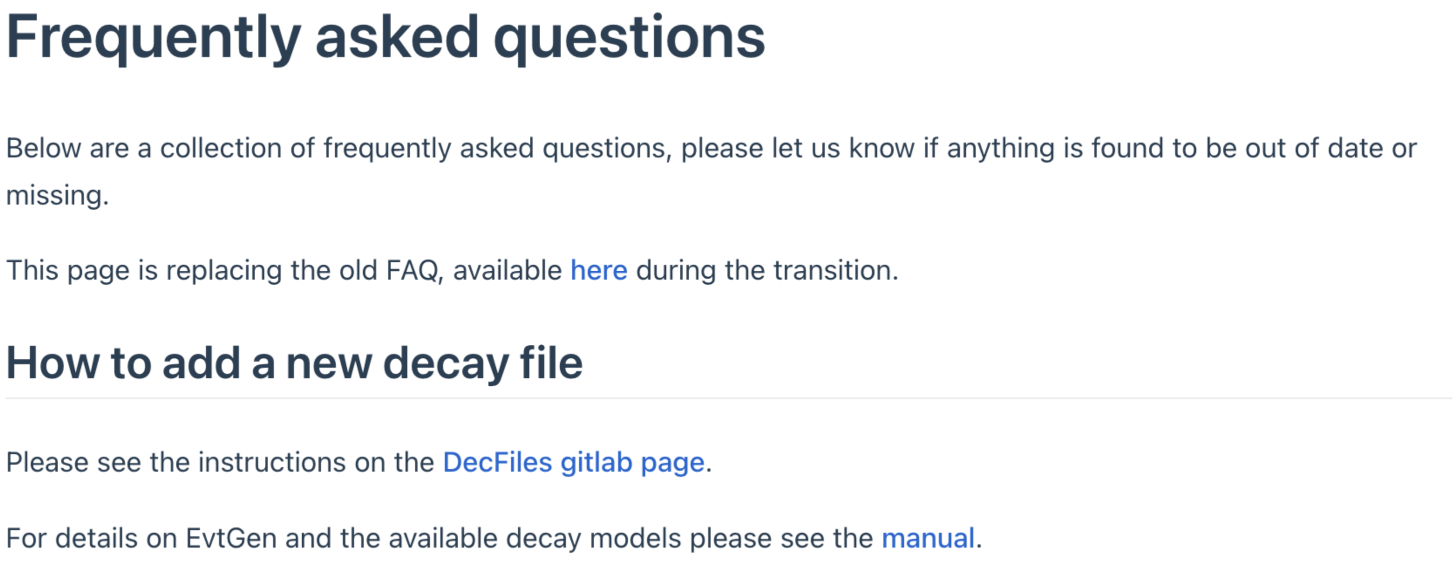


Steps to add new decay files

1. Find if a DecFile you want (or similar enough so you can use it) already exists
2. Read the event type numbering convention. There is an unofficial tool to assist with creating an event type, but keep in mind that it is not a substitute for the documentation.
3. If you are using an old DecFile as a template, note that it might not follow the established conventions.
4. Test your DecFile (both that it runs and that it produces output you want)
5. Commit to a branch and create a merge request against master.
6. Open your merge request in web browser and check that all tests are successful. If anything fails, please correct and recheck again day after commit. If you do not understand failure, get in touch with lhcb-sim-developers@cernNOSPAMPLEASE.ch
7. **Make sure the CI test pass**. Make sure no warnings / errors pop up, if they do then either fix them, or understand them and add a comment in the merge request on why this popped up. See below for more information about the CI test.
8. Watch discussion in merge request for any comments we might have.

Finding DecFiles CONTRIBUTING page

- Simultation project’s main website now has both a search bar and an FAQ!
  - Typing ‘Dec Files’ into the search bar points to the FAQ and this links straight through to CONTRIBUTING.md – perfect!



Frequently asked questions

Below are a collection of frequently asked questions, please let us know if anything is found to be out of date or missing.

This page is replacing the old FAQ, available here during the transition.

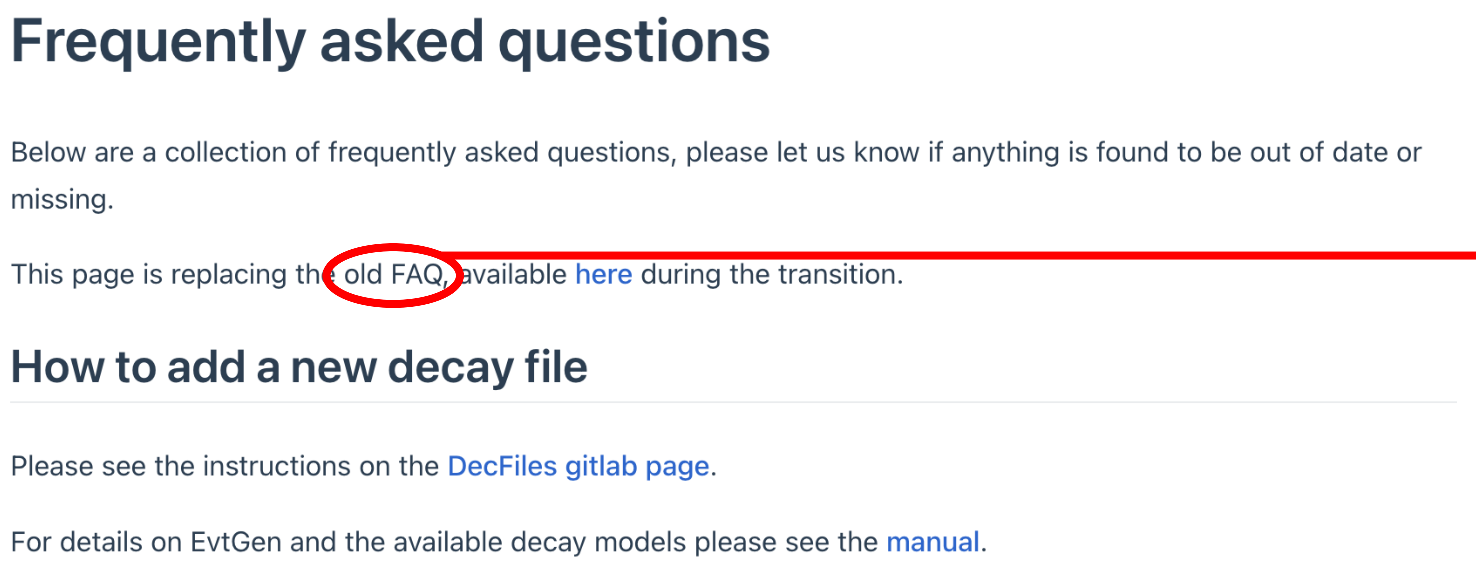
How to add a new decay file

Please see the instructions on the DecFiles gitlab page.

For details on EvtGen and the available decay models please see the manual.

Finding DecFiles CONTRIBUTING page

- Simulation project’s main website now has both a search bar and an FAQ!
  - Typing ‘Dec Files’ into the search bar points to the FAQ and this links straight through to CONTRIBUTING.md – perfect!



Frequently asked questions

Below are a collection of frequently asked questions, please let us know if anything is found to be out of date or missing.

This page is replacing the old FAQ, available here during the transition.

How to add a new decay file

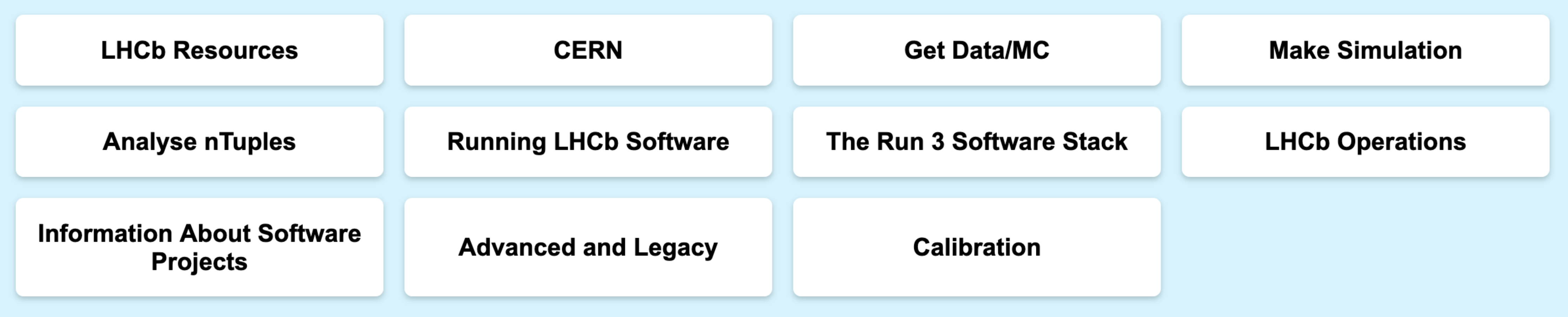
Please see the instructions on the DecFiles gitlab page.

For details on EvtGen and the available decay models please see the manual.

- A Twiki - More difficult to find
  - Also – called Gauss FAQ
- New FAQ being on the Simulation main site is more discoverable

How has WP7 tried to help

- The LHCb software landing page!



LHCb Resources

CERN

Get Data/MC

Make Simulation

Analyse nTuples

Running LHCb Software

The Run 3 Software Stack

LHCb Operations

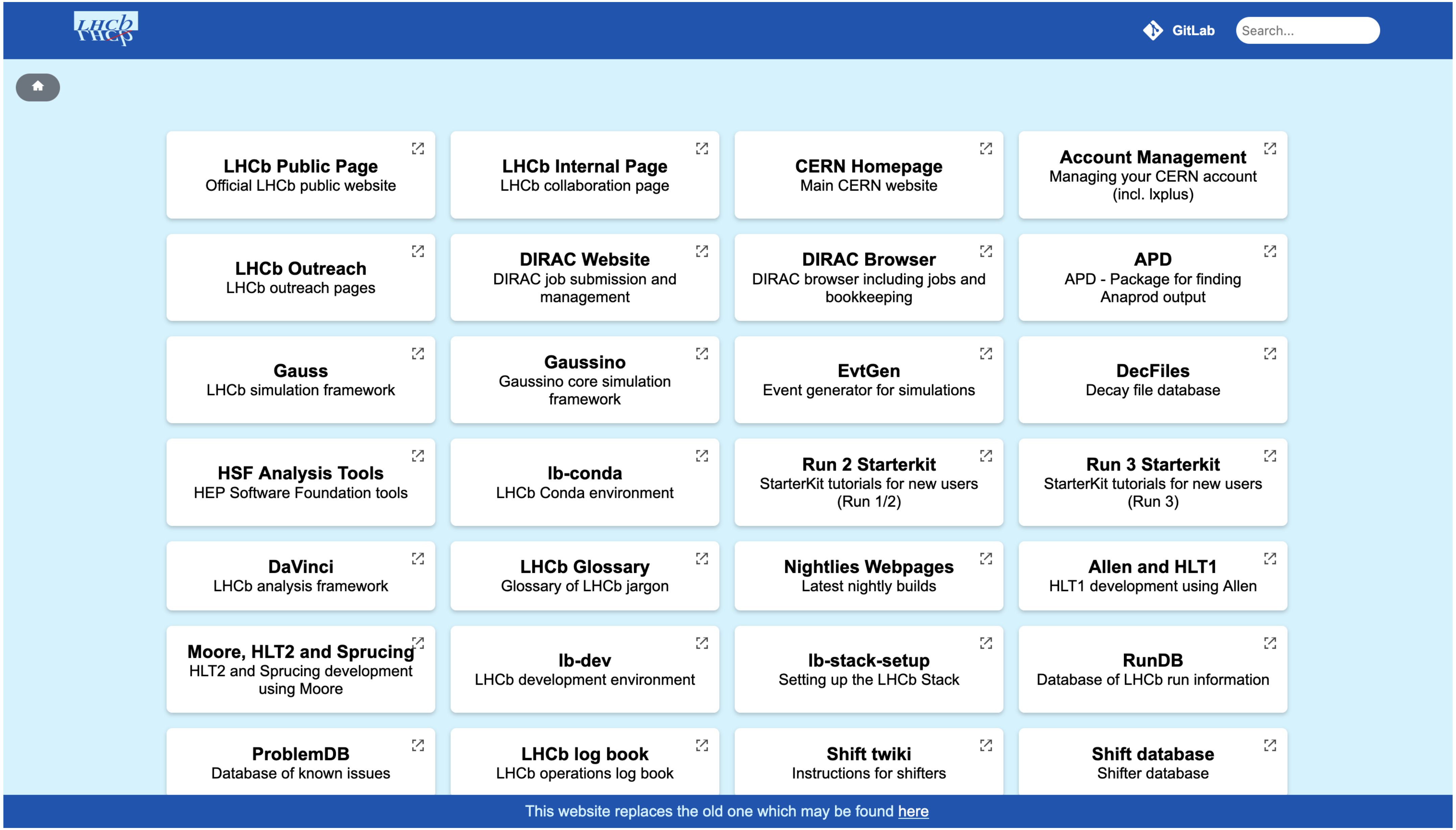
Information About Software Projects

Advanced and Legacy

Calibration

- Groups doc pages by category
- Is searchable
- Allows community contributions via Gitlab – Missing links can be added!

This work has a CC0 license – feel free to use it for your own purposes!



LHCb Public Page
Official LHCb public website

LHCb Internal Page
LHCb collaboration page

CERN Homepage
Main CERN website

Account Management
Managing your CERN account (incl. lxplus)

LHCb Outreach
LHCb outreach pages

DIRAC Website
DIRAC job submission and management

DIRAC Browser
DIRAC browser including jobs and bookkeeping

APD
APD - Package for finding Anaprod output

Gauss
LHCb simulation framework

Gaussino
Gaussino core simulation framework

EvtGen
Event generator for simulations

DecFiles
Decay file database

HSF Analysis Tools
HEP Software Foundation tools

lb-conda
LHCb Conda environment

Run 2 Starterkit
StarterKit tutorials for new users (Run 1/2)

Run 3 Starterkit
StarterKit tutorials for new users (Run 3)

DaVinci
LHCb analysis framework

LHCb Glossary
Glossary of LHCb jargon

Nightlies Webpages
Latest nightly builds

Allen and HLT1
HLT1 development using Allen

Moore, HLT2 and Sprucing
HLT2 and Sprucing development using Moore

lb-dev
LHCb development environment

lb-stack-setup
Setting up the LHCb Stack

RunDB
Database of LHCb run information

ProblemDB
Database of known issues

LHCb log book
LHCb operations log book

Shift twiki
Instructions for shifters

Shift database
Shifter database

This website replaces the old one which may be found here

Maintainability

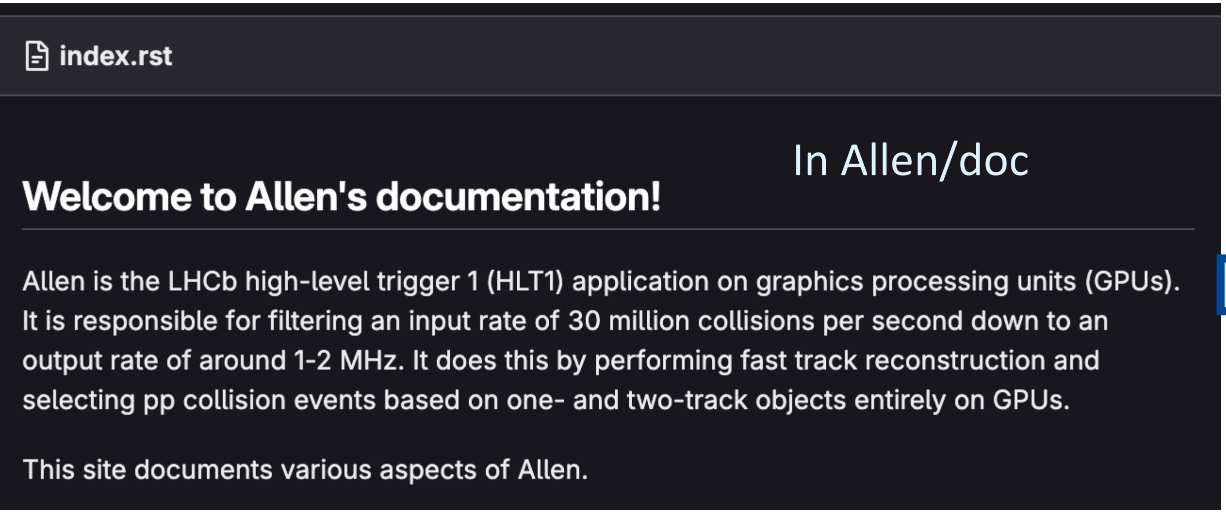
- Writing docs is great, but they go out of date fast!

- Several methods for making sure docs stay updated:

  - Keep the docs close to the code
    - Choose your medium wisely
  - Autogenerate them?
    - Not AI/LLMs – we'll come to that in a bit
  - Unit tests! Making sure your examples still run…

Keep the docs close to the code

- Putting the docs in the same repository as the code means that updates to the docs are less likely to be forgotten
  - We've seen how this works for DecFiles already, but see also Allen
    - A much more complicated software package!
- This is quite neat! But also not better per se than DecFiles' approach



index.rst

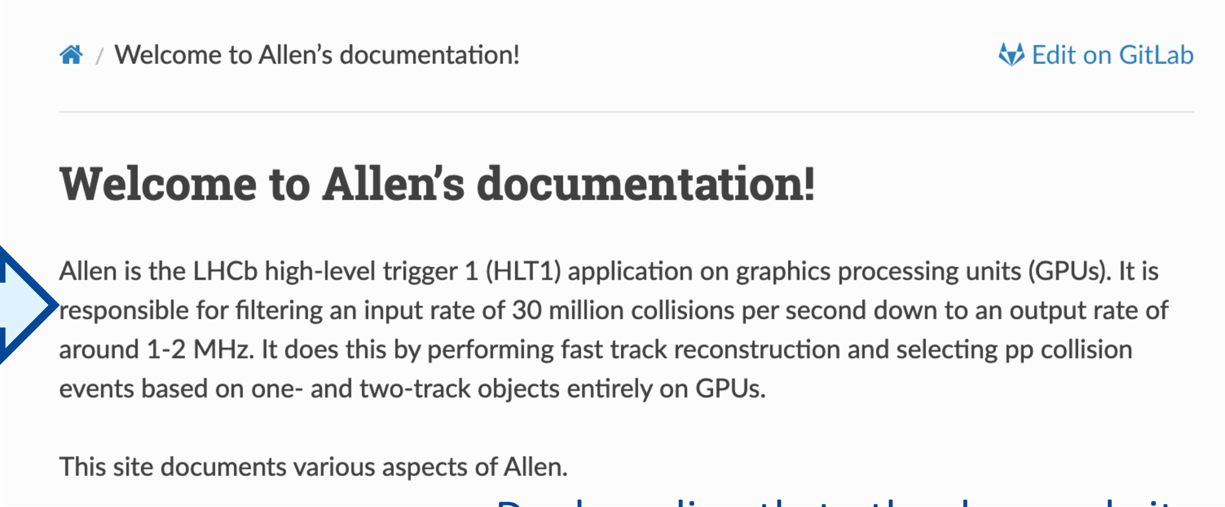
In Allen/doc

Welcome to Allen's documentation!

Allen is the LHCb high-level trigger 1 (HLT1) application on graphics processing units (GPUs). It is responsible for filtering an input rate of 30 million collisions per second down to an output rate of around 1-2 MHz. It does this by performing fast track reconstruction and selecting pp collision events based on one- and two-track objects entirely on GPUs.

This site documents various aspects of Allen.

CI



Welcome to Allen's documentation!

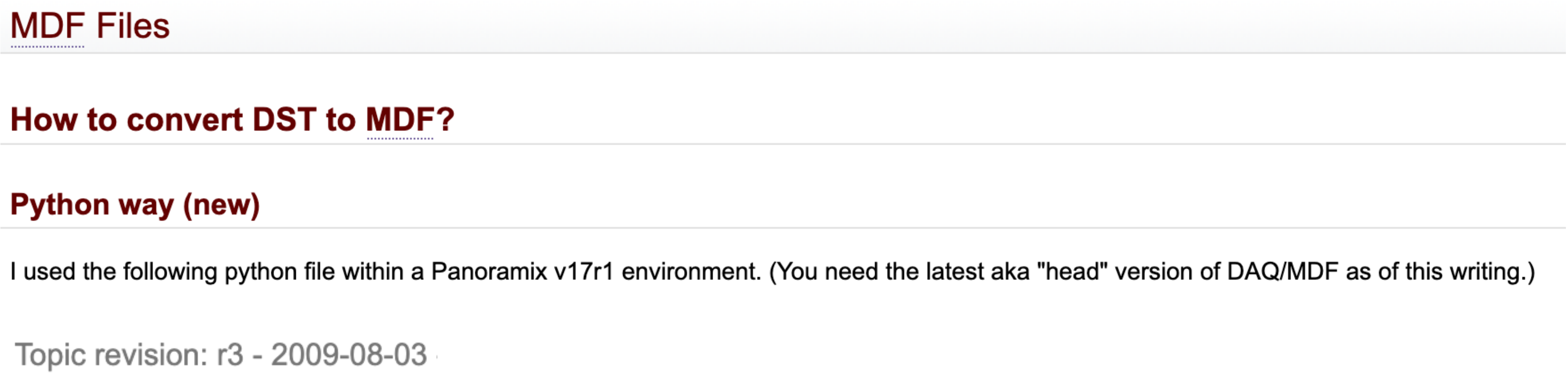
Allen is the LHCb high-level trigger 1 (HLT1) application on graphics processing units (GPUs). It is responsible for filtering an input rate of 30 million collisions per second down to an output rate of around 1-2 MHz. It does this by performing fast track reconstruction and selecting pp collision events based on one- and two-track objects entirely on GPUs.

This site documents various aspects of Allen.

Deploys directly to the docs website

Keep the docs close to the code

- On the other hand… Twiki pages…
  - More historically used – Twiki pages can and have been maintained diligently
  - In my experience they can more-often become abandoned in a way that’s more difficult if the docs are tied closely to the code repo – these can then become confusing
- As an example:



MDF Files

How to convert DST to MDF?

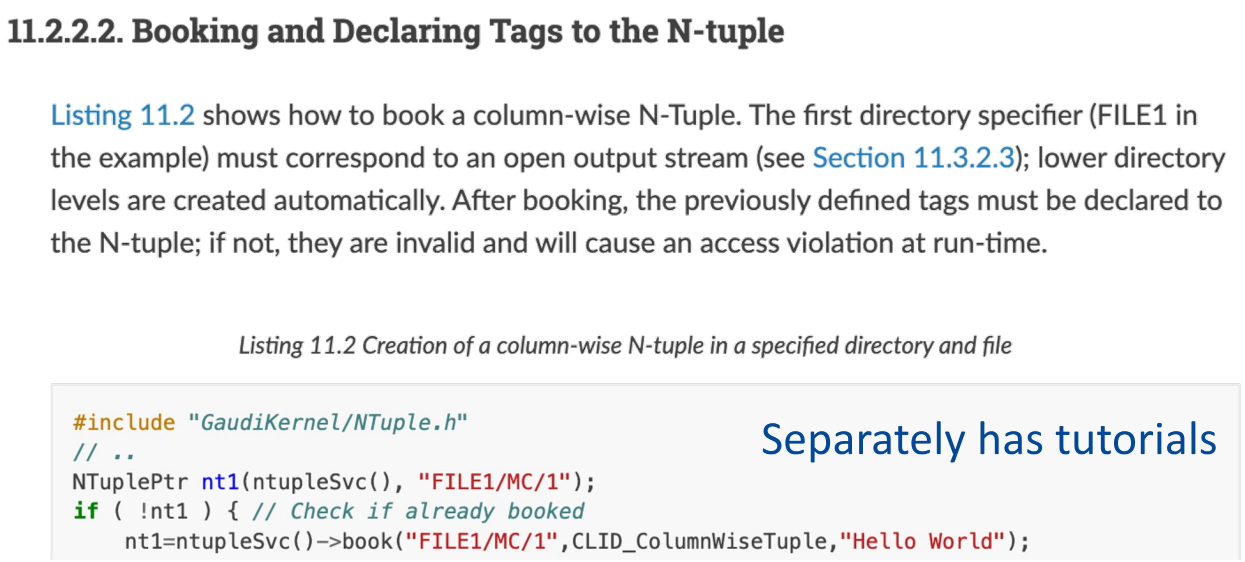
Python way (new)

I used the following python file within a Panoramix v17r1 environment. (You need the latest aka "head" version of DAQ/MDF as of this writing.)

Topic revision: r3 - 2009-08-03

Autogenerating code

- Taking your codes comments and having those build into doc pages
- These can be a great suppliment to hand-written docs but don't replace them
    - E.g. Gaudi but also ROOT



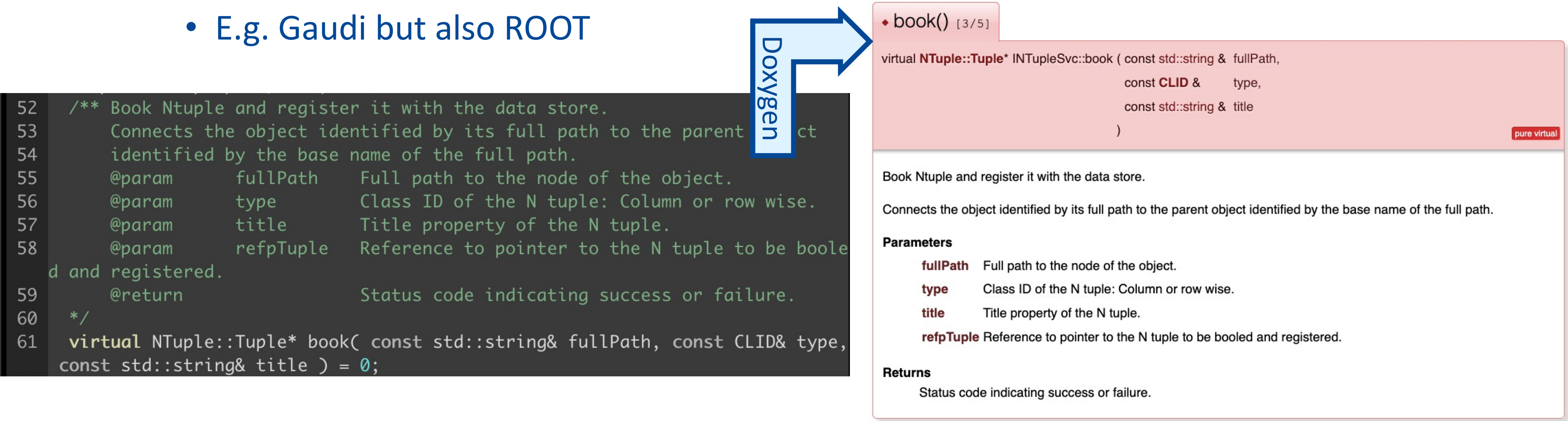
11.2.2.2. Booking and Declaring Tags to the N-tuple

Listing 11.2 shows how to book a column-wise N-Tuple. The first directory specifier (FILE1 in the example) must correspond to an open output stream (see Section 11.3.2.3); lower directory levels are created automatically. After booking, the previously defined tags must be declared to the N-tuple; if not, they are invalid and will cause an access violation at run-time.

Listing 11.2 Creation of a column-wise N-tuple in a specified directory and file

```
#include "GaudiKernel/NTuple.h"
// ..
NTuplePtr nt1(ntupleSvc(), "FILE1/MC/1");
if ( !nt1 ) { // Check if already booked
    nt1=ntupleSvc()->book("FILE1/MC/1",CLID_ColumnWiseTuple,"Hello World");
```

Separately has tutorials



```
 /** Book Ntuple and register it with the data store.
     Connects the object identified by its full path to the parent ct
     identified by the base name of the full path.
     @param      fullPath    Full path to the node of the object.
     @param      type        Class ID of the N tuple: Column or row wise.
     @param      title       Title property of the N tuple.
     @param      refpTuple   Reference to pointer to the N tuple to be booled and registered.
     @return                 Status code indicating success or failure.
 */
 virtual NTuple::Tuple* book( const std::string& fullPath, const CLID& type, const std::string& title ) = 0;
```

◆ book() [3/5]

```
virtual NTuple::Tuple* INTupleSvc::book ( const std::string &  fullPath,
                                          const CLID &         type,
                                          const std::string &  title
                                        )
```

pure virtual

Book Ntuple and register it with the data store.

Connects the object identified by its full path to the parent object identified by the base name of the full path.

Parameters

fullPath	Full path to the node of the object.
type	Class ID of the N tuple: Column or row wise.
title	Title property of the N tuple.
refpTuple	Reference to pointer to the N tuple to be booled and registered.

Returns

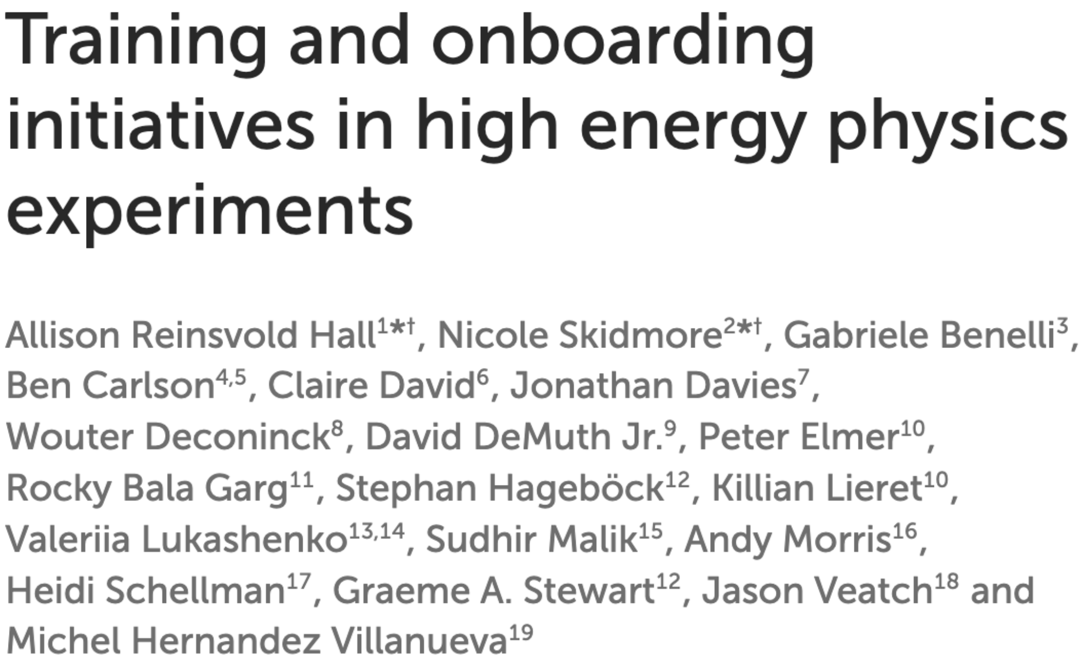
Status code indicating success or failure.

Unit tests

- Examples are great – broken examples are frustraiting
  - One thing we should aim for in LHCb is having unit tests for our examples
  - There are examples of this in LHCb but the situation isn't perfect…

- A nice example – [BASF2]

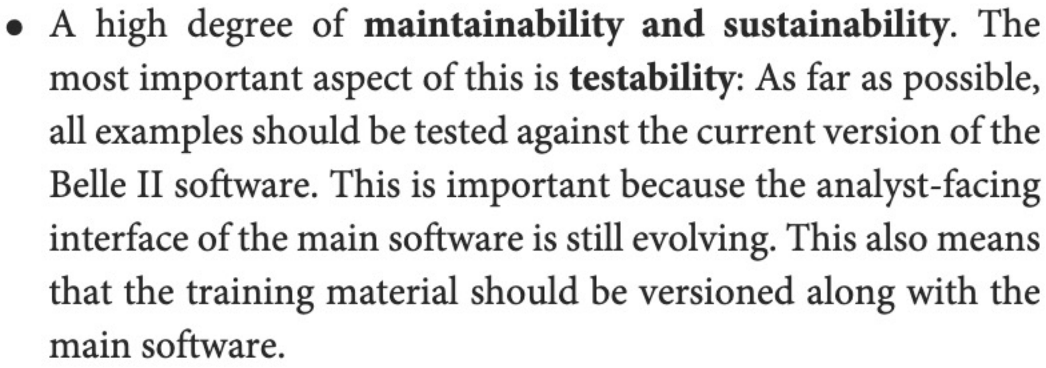
- During lockdown Belle2 their entire software framework docs to a single place with extensive testing



Training and onboarding initiatives in high energy physics experiments

Allison Reinsvold Hall[1*†], Nicole Skidmore[2*†], Gabriele Benelli[3], Ben Carlson[4,5], Claire David[6], Jonathan Davies[7], Wouter Deconinck[8], David DeMuth Jr.[9], Peter Elmer[10], Rocky Bala Garg[11], Stephan Hageböck[12], Killian Lieret[10], Valeriia Lukashenko[13,14], Sudhir Malik[15], Andy Morris[16], Heidi Schellman[17], Graeme A. Stewart[12], Jason Veatch[18] and Michel Hernandez Villanueva[19]

[Reinsvold Hall et al.]



- A high degree of **maintainability and sustainability**. The most important aspect of this is **testability**: As far as possible, all examples should be tested against the current version of the Belle II software. This is important because the analyst-facing interface of the main software is still evolving. This also means that the training material should be versioned along with the main software.

Detail

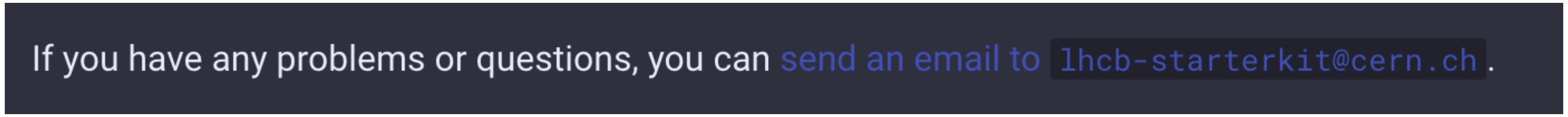
- What do people want to know when seeing your docs?

- An introduction – what is it?
- How to access it and build (if relevant)
- How to use it
  - Examples?
  - Common pitfalls/debugging
- FAQ
- References
- …

There's no one-size-fits-all answer here

Detail – The medium

- The level of detail needed should inform the choice of medium
  - We've already seen two fairly different extremes:
    - DecFiles – a single MD page
    - Gaudi – Webpages built with Sphinx and separate versioned Doxygen code
  - The Tool needs to be chosen for the job
    - A single page of md is easier to maintain – can be checked simply for correctness
    - A set of pages scales better but more difficult to maintain and deploy
      - Also more difficult to configure from an accessibility PoV (e.g. default mkdocs dark blue)



If you have any problems or questions, you can send an email to `lhcb-starterkit@cern.ch`.

Startkit lessons with the custom colours CSS removed – unreadable in dark mode

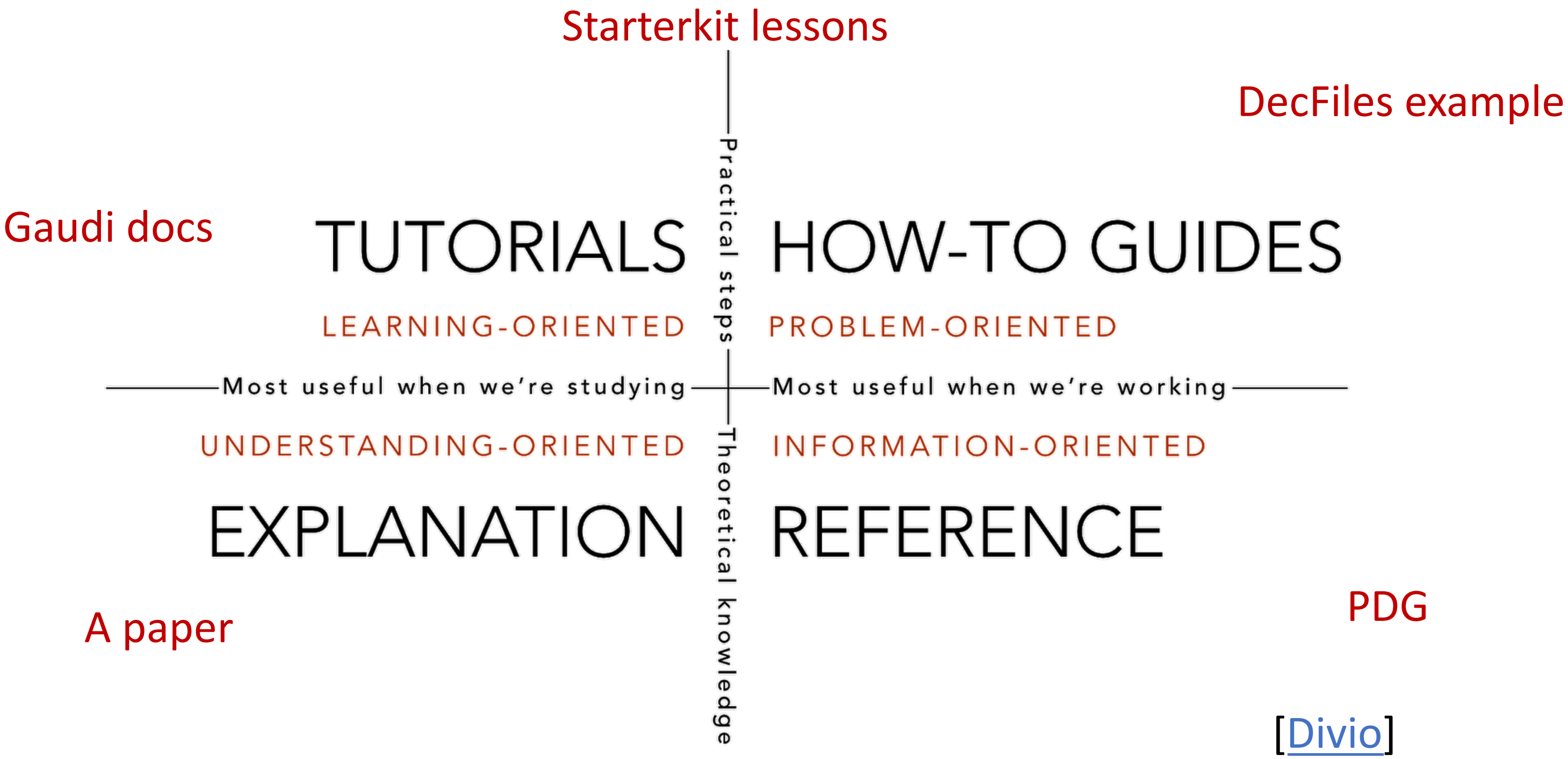
Detail – The litmus test

- As a start for docs, write what you feel you would want to see yourself as an end user – then the real test begins…

- As a maintainer of a code base people will as you questions
  - If you're getting the same question over and over – add that to the docs!
  - If the question's answer is already in the docs – people aren't finding them or they can't understand!
  - As an expert you need to remember that you're writing for non-experts

Detail – The target

- Let’s take another look at that plot – this will affect the level of detail



Starterkit lessons

DecFiles example

Gaudi docs

A paper

PDG

[Divio]

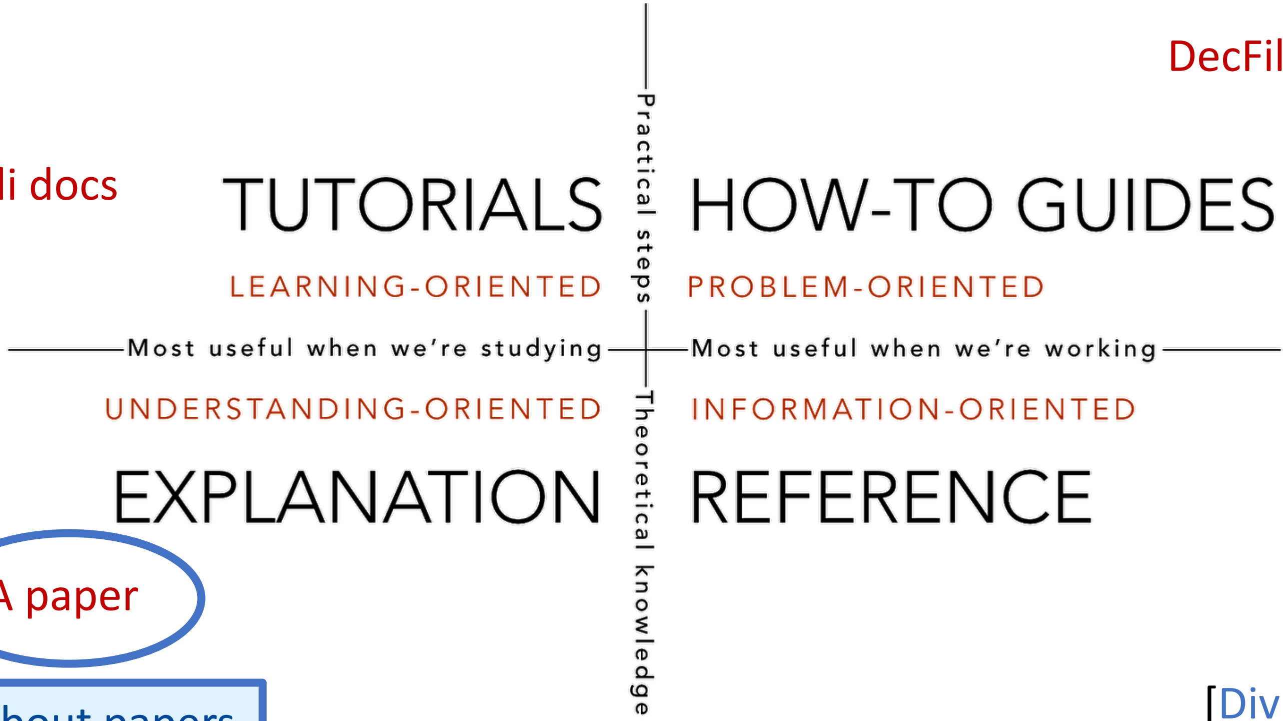
Detail – The target

- Let’s take another look at that plot – this will affect the level of detail

Starterkit lessons

DecFiles example

Gaudi docs



A paper

PDG

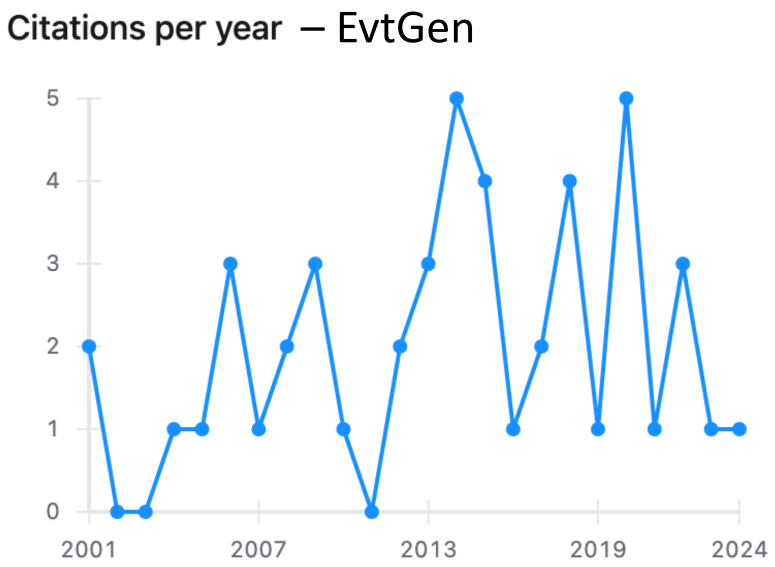
[Divio]

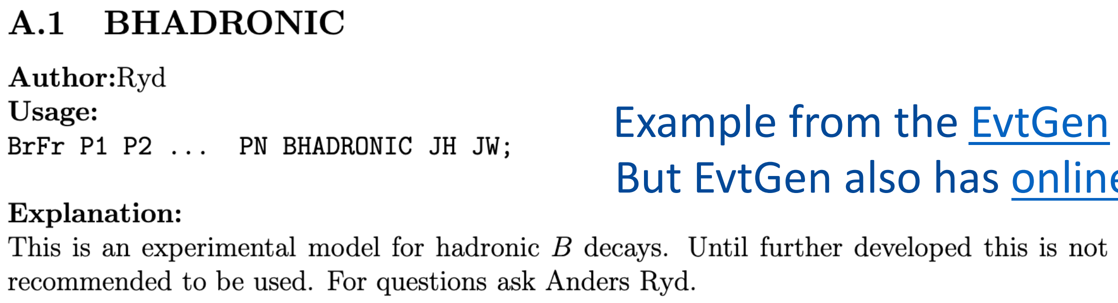
Let’s also have a quick aside about papers

A quick aside about papers

- Writing a paper for your software is an excellent idea
  - People who use your software will give you citations

- It is sometimes the case that listings are added to papers – acting more like user guides
  - This is fine as well, but if your code is still evolving it also doesn't replace traditional documentation


Citations per year – EvtGen



A.1 BHADRONIC

Author: Ryd
Usage:

```
BrFr P1 P2 ...  PN BHADRONIC JH JW;
```

Explanation:
This is an experimental model for hadronic B decays. Until further developed this is not recommended to be used. For questions ask Anders Ryd.

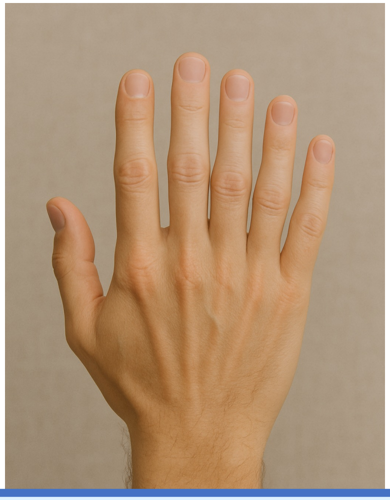
Example from the EvtGen paper
But EvtGen also has online docs

Fine… let’s talk about AI

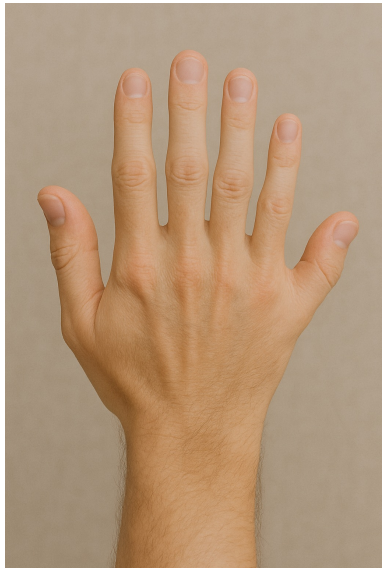
- Some applications of LLMs have genuine merit – but there are other places where I think caution is needed

Refining docs with supervision from an expert

Dumping your source code into an LLM and asking it to produce all your docs



Producing docs without expert supervision



(producing these, I made some cursed images along the way)

AI can aid in doc creation/maintanance…

- … But they do not replace an expert
  - Something produced by an AI should be checked by an expert

- E.g. The Starterkit translation
  - DPA WP7 is currently producing a translation of its docs into Chinese – the first attempt at this has been done by LLMs, which collaborators will then proofread

- E.g. Checking a written lesson reads well
  - Ask an AI to read your lesson, ask it what commands it would run to do the exercises/examples/etc… – And how obvious the steps were

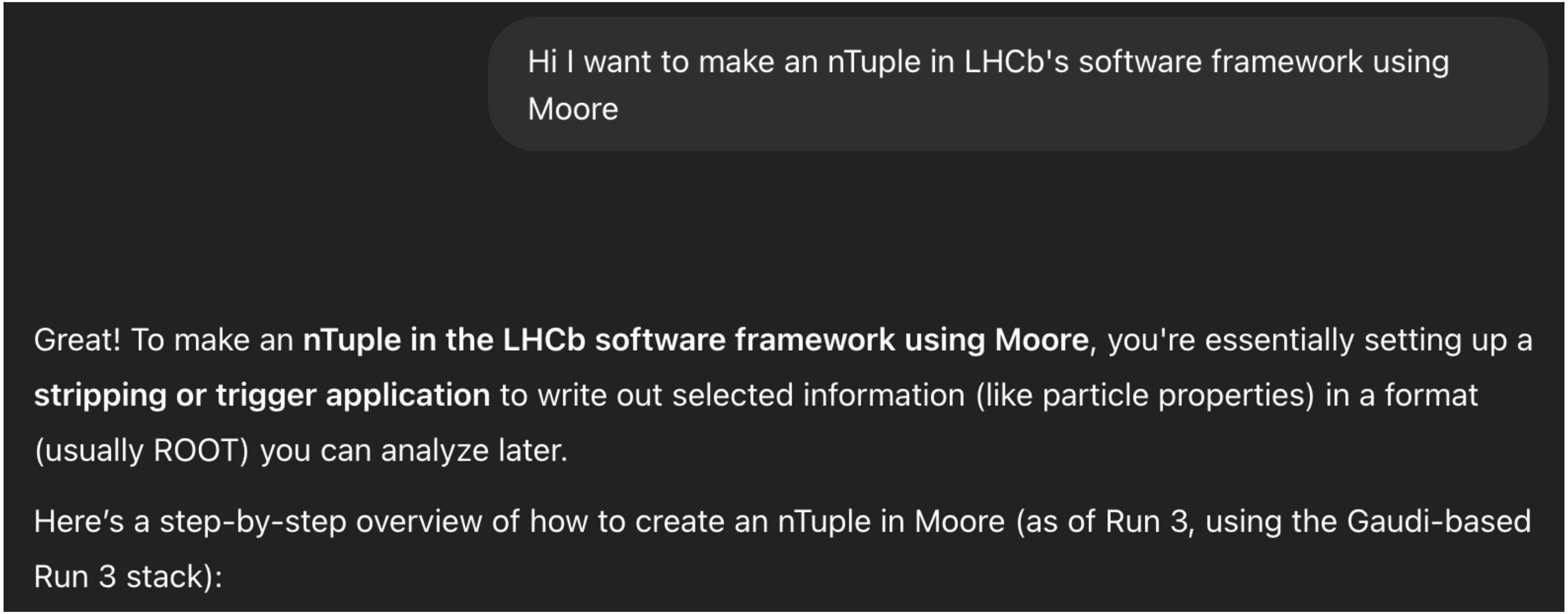
AI projects within LHCb

- There are two groups in LHCb producing LLM-powered tools trained using LHCb papers and documentation
  - A chat bot based on Llama3.1
  - A search tool to find documenation given a topic

  - I believe that these tools can suppliment traditional docs, but not replace them (at present)

- Chat bots are known to hallucinate and; more subtly, they often won’t tell you when what you’re trying to do is a bad idea
  - End users, new to the software will get confused so this is a genuine issue – A badly formed question can cause more confusion

Where caution is needed



Hi I want to make an nTuple in LHCb's software framework using Moore

Great! To make an **nTuple in the LHCb software framework using Moore**, you're essentially setting up a **stripping or trigger application** to write out selected information (like particle properties) in a format (usually ROOT) you can analyze later.

Here's a step-by-step overview of how to create an nTuple in Moore (as of Run 3, using the Gaudi-based Run 3 stack):

Any expert would first ask ”Moore configures the trigger, it doesn’t make tuples… What are you trying to do?”

What would I suggest for documenting a new package

- There is no substitution for writing traditional docs
  - Choose the medium to fit the project's size – a single MD file to a full website
  - If going for a full website – CERN suggests using mkdocs
    The Run 3 Starterkit is also under a CC4.0 license – steal code from it (css, …)!

- Make sure that the docs are findable
  - Advertise them in your project's README.md, add them to the landing page

- Including autogenerated docs are a good supliment to other code
  - However check that its output is what you want

Training

Slightly tricky to do with longevity

Training at LHCb

- There are several training initiatives at LHCb:
- Workshops:
  - The Starterkit and Impactkit
- Lecture series:
  - Startertalks and Theory talks
  - LHCb-UK Student talks – Worth mentioning despite being ‘national’ level

- These have the common thread of being organised by volunteers

The Starterkit

- A workshop hosted yearly with the goal of onboarding new arrivals to LHCb
  - Letting them get to grips with our codebases
  - Letting them meet people within the collabo

- Held in person each year at CERN but also available on Zoom and with the lessons recorded

The Starterkit

- A workshop hosted yearly with the goal of onboarding new arrivals to LHCb
  - Letting them get to grips with our codebases
  - Letting them meet people within the collabo
- Held in person each year at CERN but also available on Zoom and with the lessons recorded

Let’s talk about accessibility again

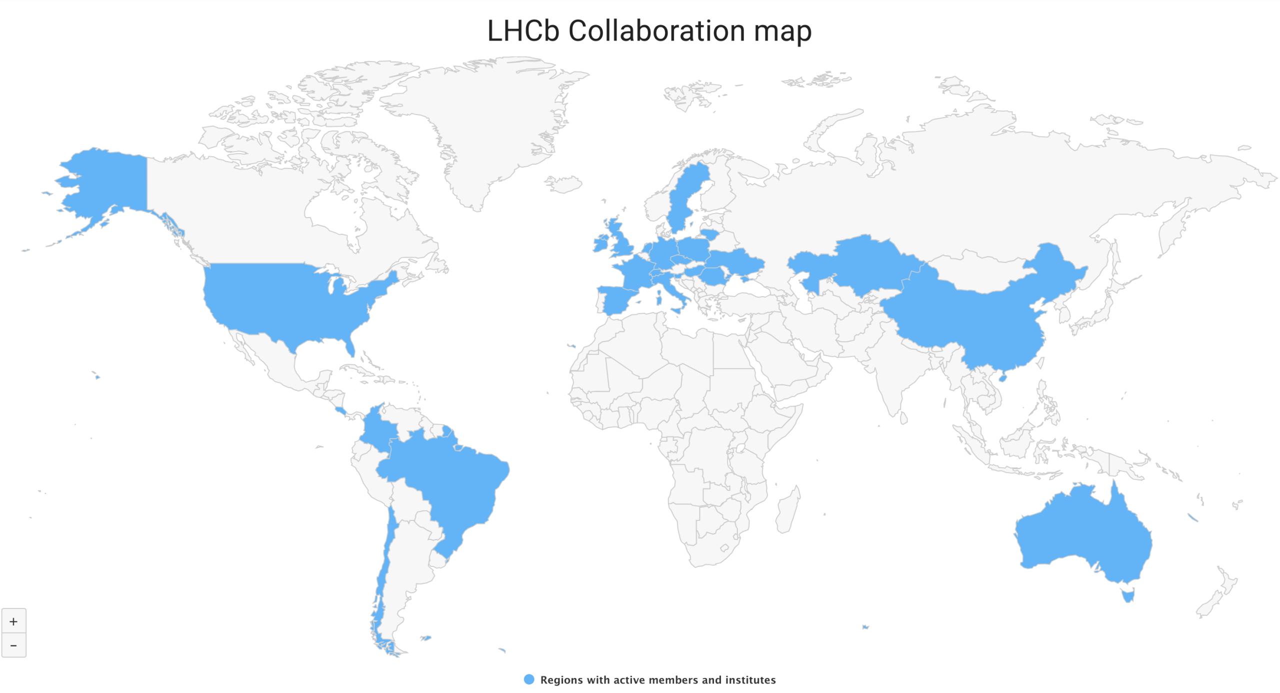
Training accessibility

- For workshops, especially the Starterkit it’s preferred for people to come in person – this isn’t always possible
- Difficulties arise for long distance travel, visas, …
- Hybrid events should therefore be organised for those who can’t make it
  - Recording sessions also help accommodate timezones
  - But can we do even better than that…?

Training accessibility – internationalisation

- With the Starterkit always being at CERN, DPA WP7 is trying to set up other international Starterkit events
  - Testing at first with China

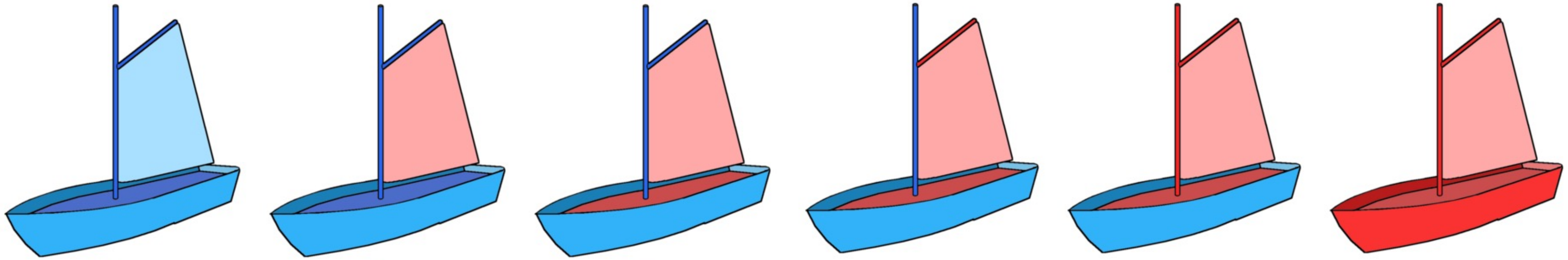
- This compliments the Starterkit translation into Chinese which is ongoing



A potential downside of organic evolution

- The cast of volunteers changes
  - Each new volunteer learns the job from the previous one – this can have the side effect of things being forgotten
  - E.g. My experience organising the Starterkit (2020), Impactkit (2021) and LHCb-UK student talks (2020)
    - For each of those the required tasks were described in meeting(s)

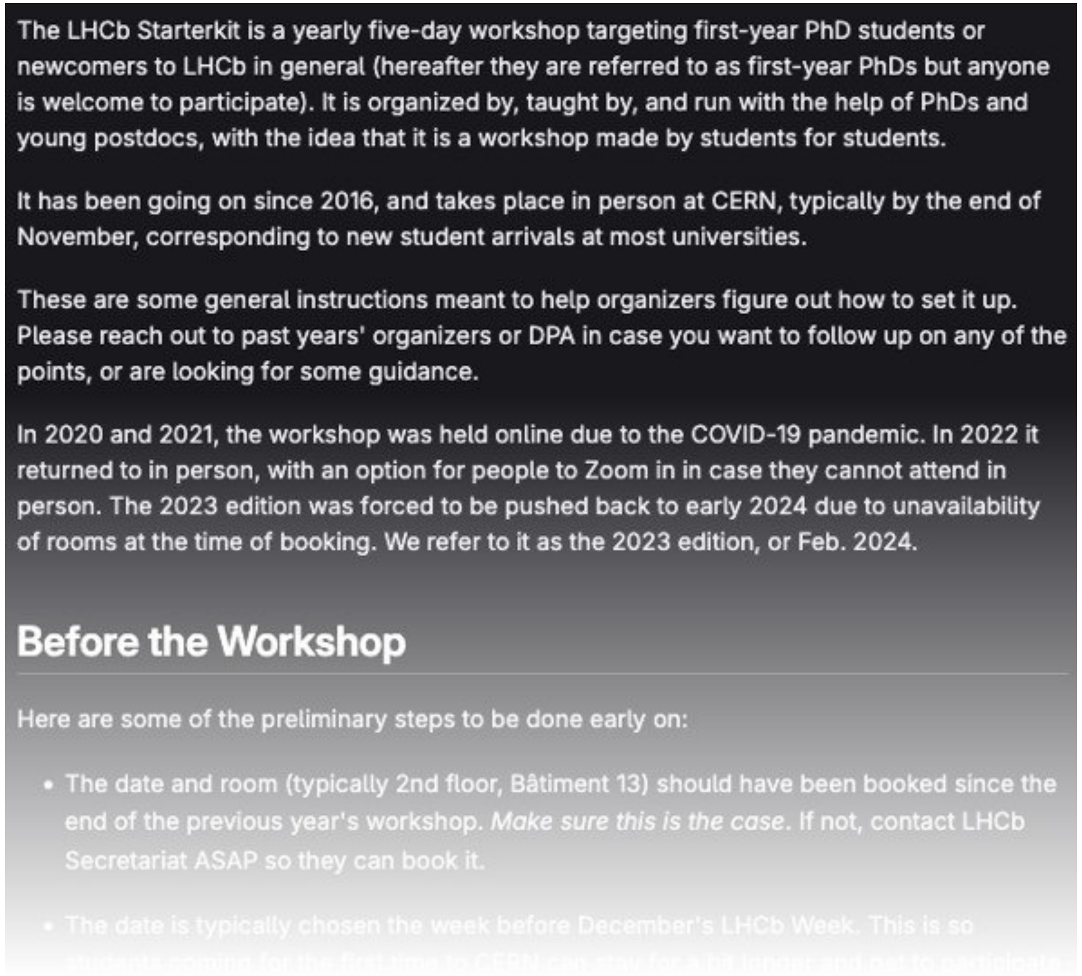
- Over time this unconciously morphs what the training looks like



Wikimedia

How ideas can be saved

- Over the years ideas used in the Starterkit changed
  - If an idea is not used one year, the organic organisation will mean it won’t even be considered in the next year
- To account for this – The Starterkit Organisation Documentation
  - Ideas are written down in such a way that they won’t be forgotten if they go unused one year
  - This then helps with the idea of having international Starterkits



The LHCb Starterkit is a yearly five-day workshop targeting first-year PhD students or newcomers to LHCb in general (hereafter they are referred to as first-year PhDs but anyone is welcome to participate). It is organized by, taught by, and run with the help of PhDs and young postdocs, with the idea that it is a workshop made by students for students.

It has been going on since 2016, and takes place in person at CERN, typically by the end of November, corresponding to new student arrivals at most universities.

These are some general instructions meant to help organizers figure out how to set it up. Please reach out to past years' organizers or DPA in case you want to follow up on any of the points, or are looking for some guidance.

In 2020 and 2021, the workshop was held online due to the COVID-19 pandemic. In 2022 it returned to in person, with an option for people to Zoom in in case they cannot attend in person. The 2023 edition was forced to be pushed back to early 2024 due to unavailability of rooms at the time of booking. We refer to it as the 2023 edition, or Feb. 2024.

Before the Workshop

Here are some of the preliminary steps to be done early on:

- The date and room (typically 2nd floor, Bâtiment 13) should have been booked since the end of the previous year's workshop. *Make sure this is the case*. If not, contact LHCb Secretariat ASAP so they can book it.
- The date is typically chosen the week before December's LHCb Week. This is so students coming for the first time to CERN can stay for a bit longer and get to participate

Training – Things to bear in mind

- Make any training you organise as accessible as possible
  - Hybrid setups with recordings are ideal

- Make sure the organisation process is written down
  - Especially if there is a high turnover for the organisers

- Collecting feedback from attendees is extremely valuable
  - Do not forget to do this – and to forward it on to the future organisers

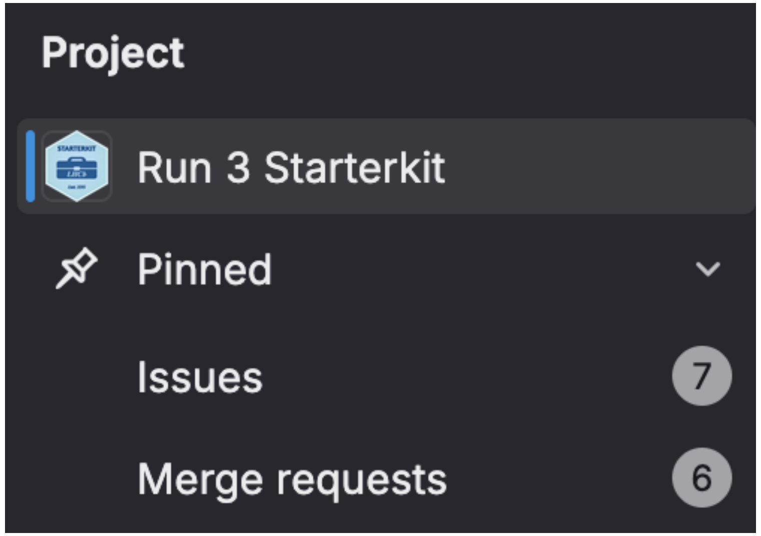
So where does this leave us?

What did LHCb’s Training and Documentation group really learn?

The lessons learned

- If you’re writing a software project – document it and make those docs visible!
  - Remember as well that documentation should be a community effort!

- Moreover: if you find some documentation that isn’t ideal – report or even contribute to it!
  - A lot of the training and documentation in LHCb relies on contributions from the community – especially the Starterkit – this is very often the case broadly



Post-scriptum – HSF

- Today – LHCb's software and documentation efforts
  - But we'd be remiss not to mention HSF
- The HEP Software Foundation has excellent tutorials and workshops for software in HEP broadly
  - Python, bash, git, snakemake, …
- Their international workshops are also the inspiration behind the internationalisation of the Starterkit docs

HSF Training Center • HSF Analysis Essentials

Thanks for listening!

Cacahuète