SAND Data Analysis status and prospective

dune-nd-sand-physics@fnal.gov
dune-nd-sand-software@fnal.gov

Status

- Full simulations w/ official beam: (FLUKA & GENIE + GEANT4)
- Fast Reconstruction
  - Beam Monitoring
  - External Background
- Studies w/ partial info from MC truth, in particular:
  - STT digit == smearing of hit (time, position)
  - Particle ID
- Performances:
  - Preliminary Pattern Reco
  - Momentum resolution w/ circular fit
  - neutrino energy resolution
- Physics Analyses: docdb-13262

Next tasks

- Define Calibration Data:
  - How do we produce it?
  - Which format?
  - How do we input in the reconstruction?

- Digitization:
  - Trigger: which time reference (t0)
  - Digit format (input from DAQ?)

- Integrate «Fast Reconstruction»

Next tasks

- Improve Reconstruction:
  - Tracking & Vertexing:
    - STT Kalman Filter
  - EM Shower:
    - ECAL Clustering
  - Particle ID:
    - e- ID in ECAL
    - μ/π separation:
      need for downstream μ catcher?
- EventSummaryBuilder $\rightarrow$ CAF

Final Goals

- Full event reconstruction
- Event classification
- Background rejection
- Physics analyses

SAND as muon spectrometer for LAr-ND

- Muon acceptance
- What about upstream muon ranger?

GRAIN review

- Re-do beam monitoring analysis

- Which information could GRAIN provide?
  - Conservative assumption:
    Selection (w/ reasonable purity) of interaction in Ar
  - Or additionally: calorimetry information

- How well can ν-LAr interactions be reconstructed w/ SAND?
  - Re-evaluate:
    - ν, μ, e, π, p momentum resolution
    - Acceptance

- What SAND w/ GRAIN can do?
  - Disentangle neutrino interaction models?

Requirements

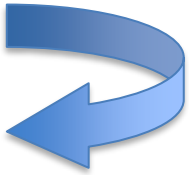
- run on CENTOS7
- complaint with interfaces
- complaint with coding convention
- C++/ROOT (cannot fix a release too early…)

- Questions:
  - High-level requirements from ND-Software?
  - Performance?
  - Functionalities, if any?
    - How to deal with competive algorithms ?
    - How to manage calibrations ?

Repositories

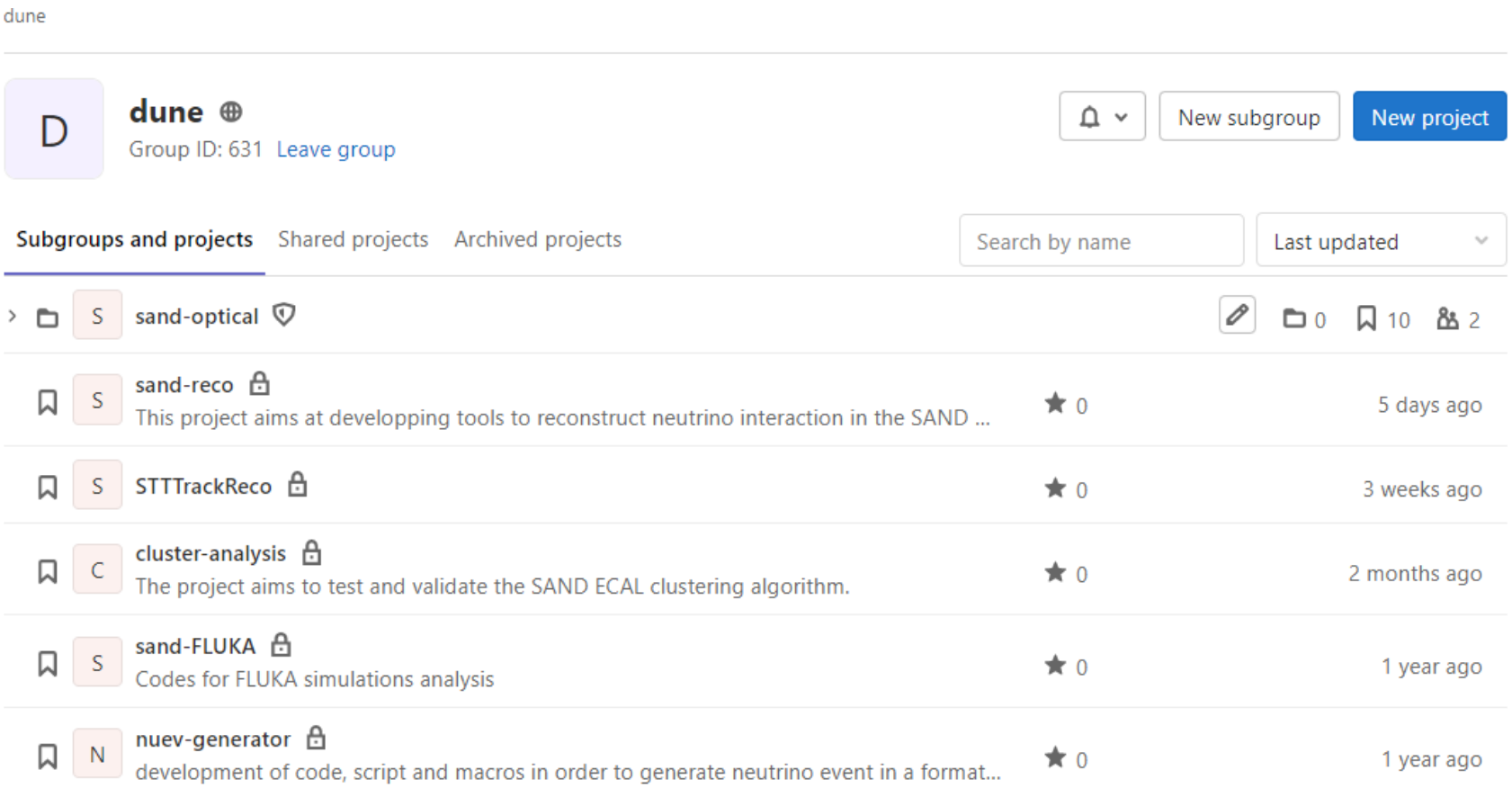
- Baltig group: **dune**

- Geometry:
  - **https://github.com/DUNE-ND-SAND/dunendggd**
  - **https://github.com/DUNE/dunendggd**



- Digitization/Reconstruction
  - **https://baltig.infn.it/dune/sand-reco**
  - Public read access
  - Exploit continuous integration resources by INFN
  - License?

Repositories



Interfaces (Data Model)

The set of external specifications conditioning this SW project.

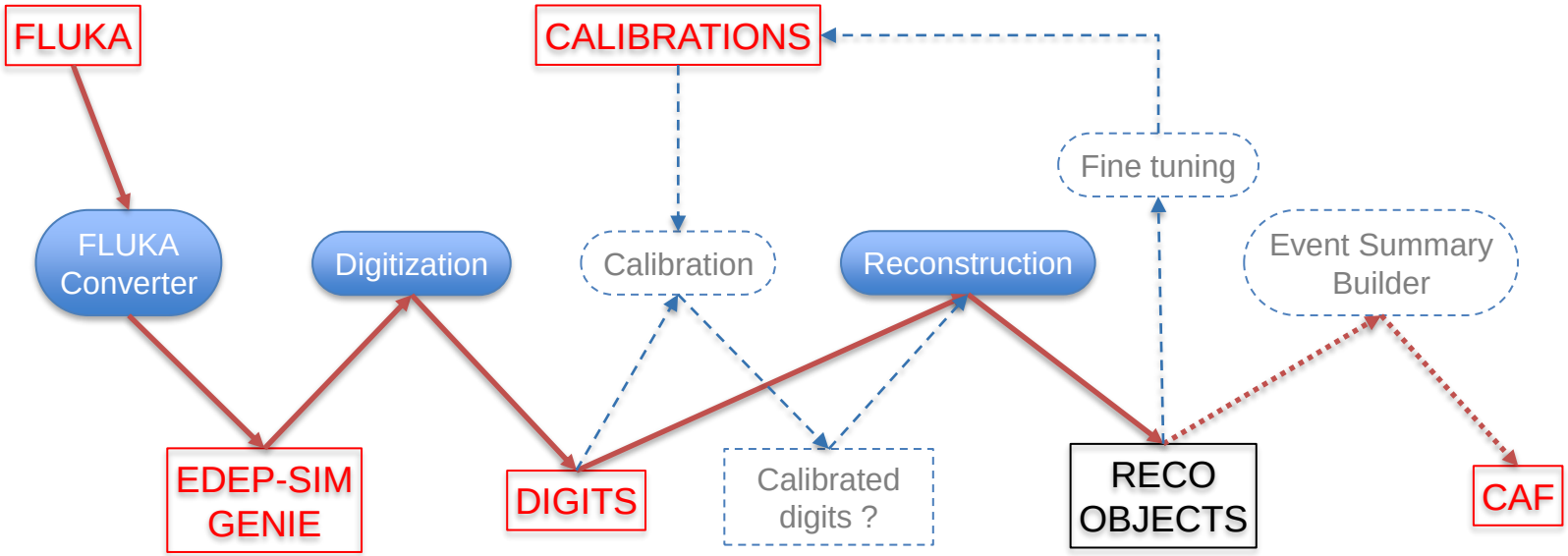
INPUT

- **[edep-sim]**
  - https://github.com/DUNE/edep-sim#output-tree-format
  - Edep-sim output (cern.ch)
- **[genie] (embedded in the edep-sim data format)**
  - GENIE Physics and User Manual (rl.ac.uk)
- **[FLUKA]**
  - /eos/user/s/salap/DUNE-IT/ntuple.spiega
- **[Digits]**
  - from DAQ
- **[Calibrations]**
  - External interface

OUTPUT

- **[CAF]**
  - https://cdcvs.fnal.gov/redmine/projects/dune-neardet-design/wiki/_CAF_ntuple_format_

SW Design (and Data Models)

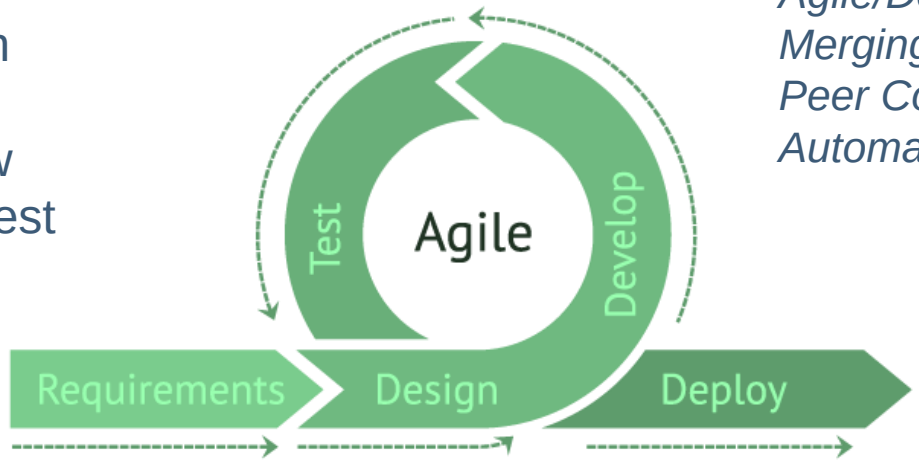


Digits & Reco Data Model: [Data Model · Wiki · dune / kloe-simu · GitLab (infn.it)]

Coding and Development Workflow

- Language: C++11
- Code Format:
  - Based on Google C++ Style Guide
  - Proposal: *clang-format -style="{BasedOnStyle: Google, BreakBeforeBraces: Linux, DerivePointerAlignment: false}"*
- Project layout: pitchfork
- Development Workflow:

1. Create Issue
2. Open Branch
3. Develop
4. Test / Review
5. Merge Request
6. Release

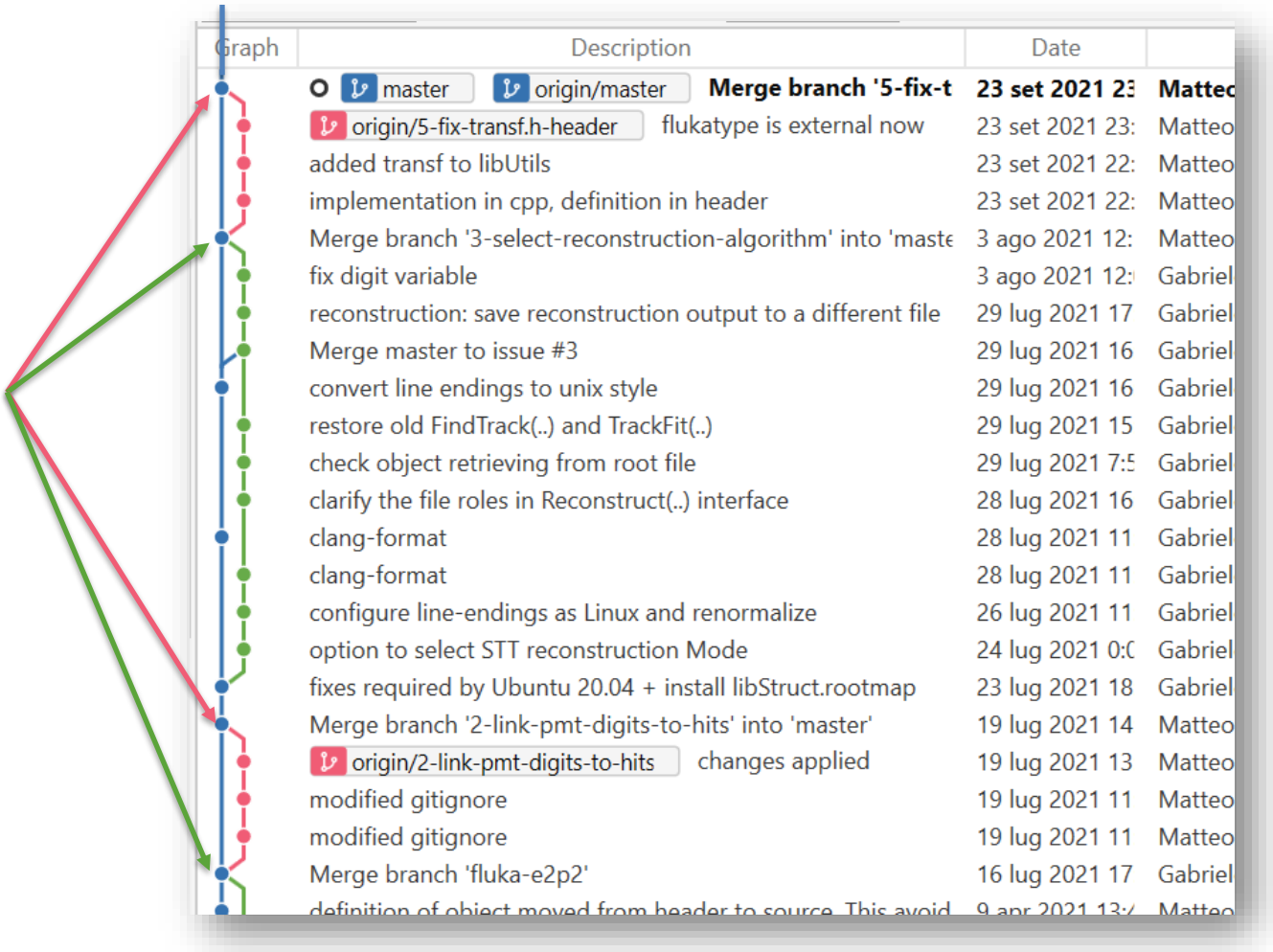


Exploits:
Agile/DevOps,
Merging by Pull Request
Peer Code Review,
Automated Tests,

Development workflow

Master branch

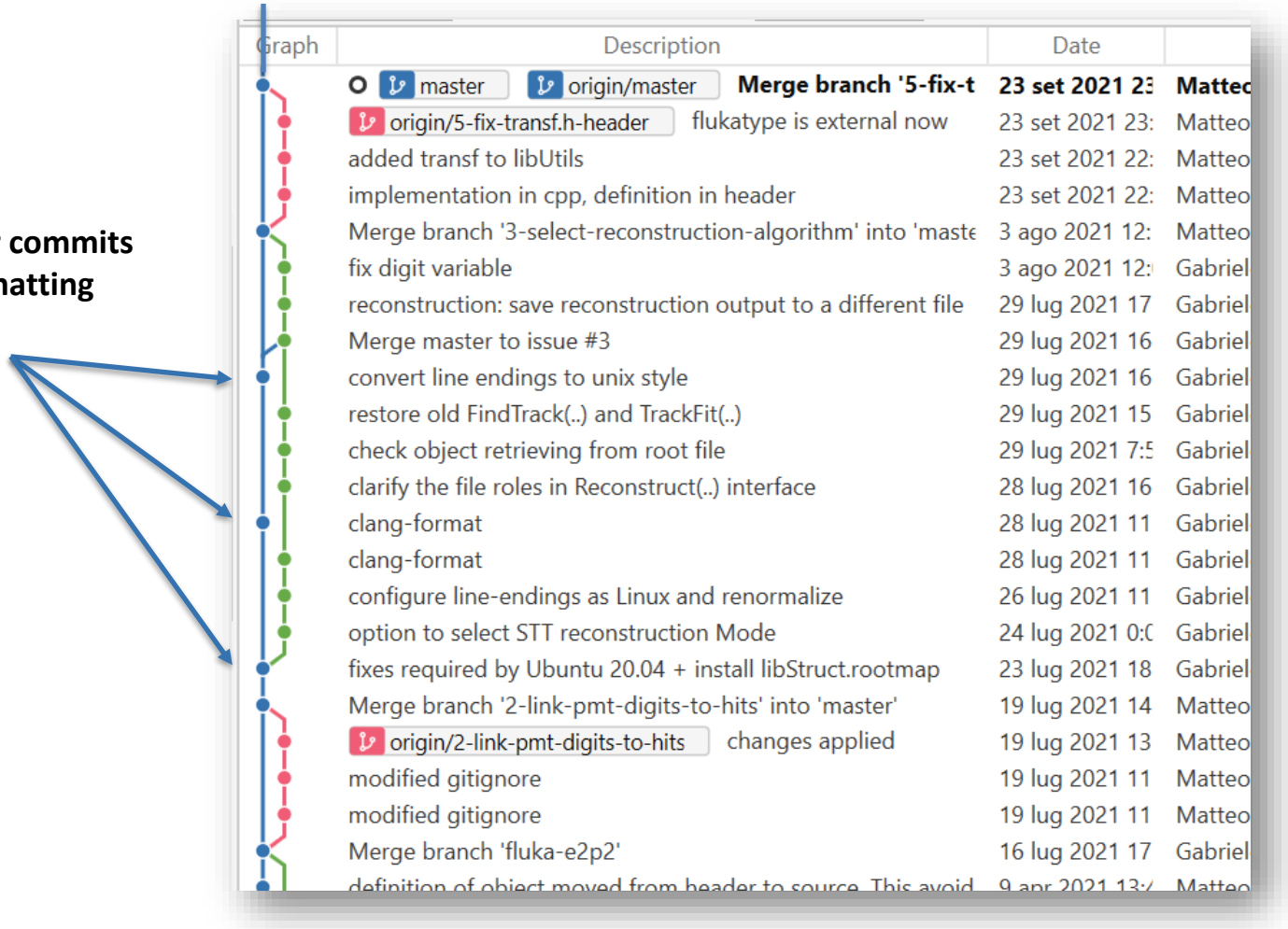
On master commit only when merge



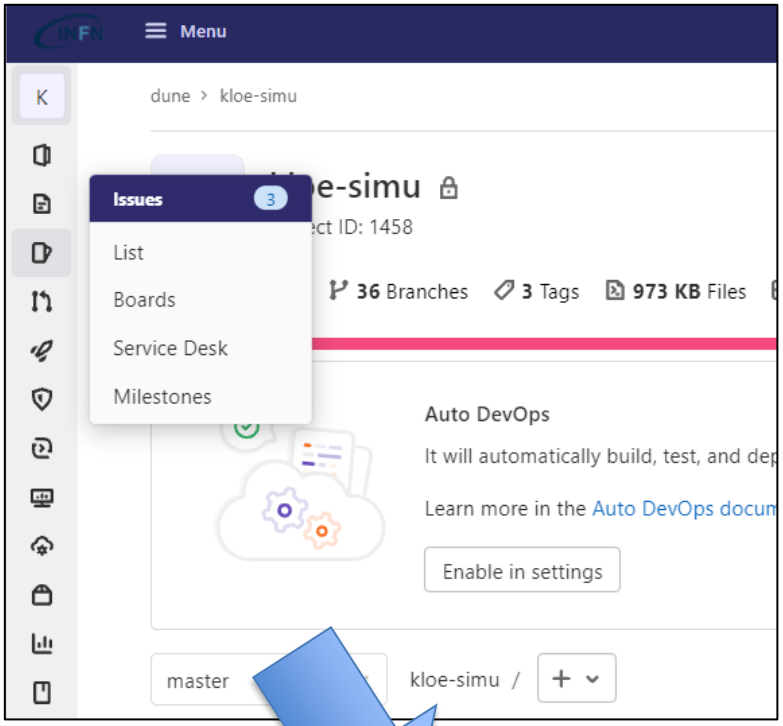
Development workflow

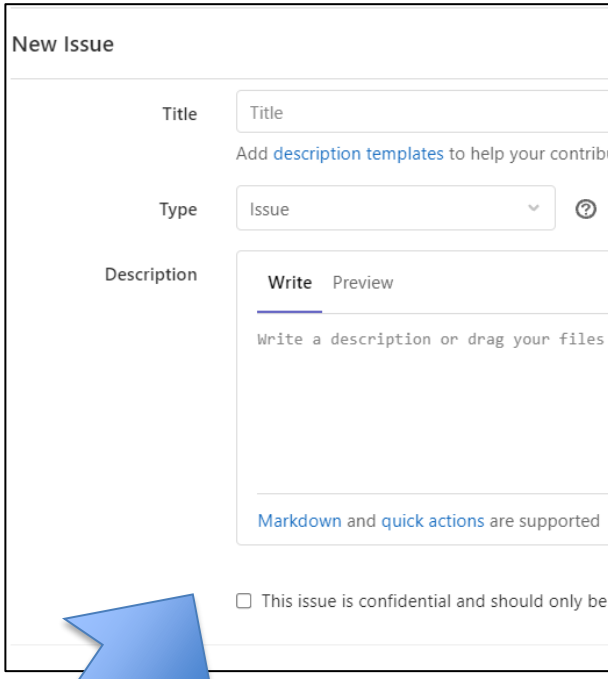
Master branch

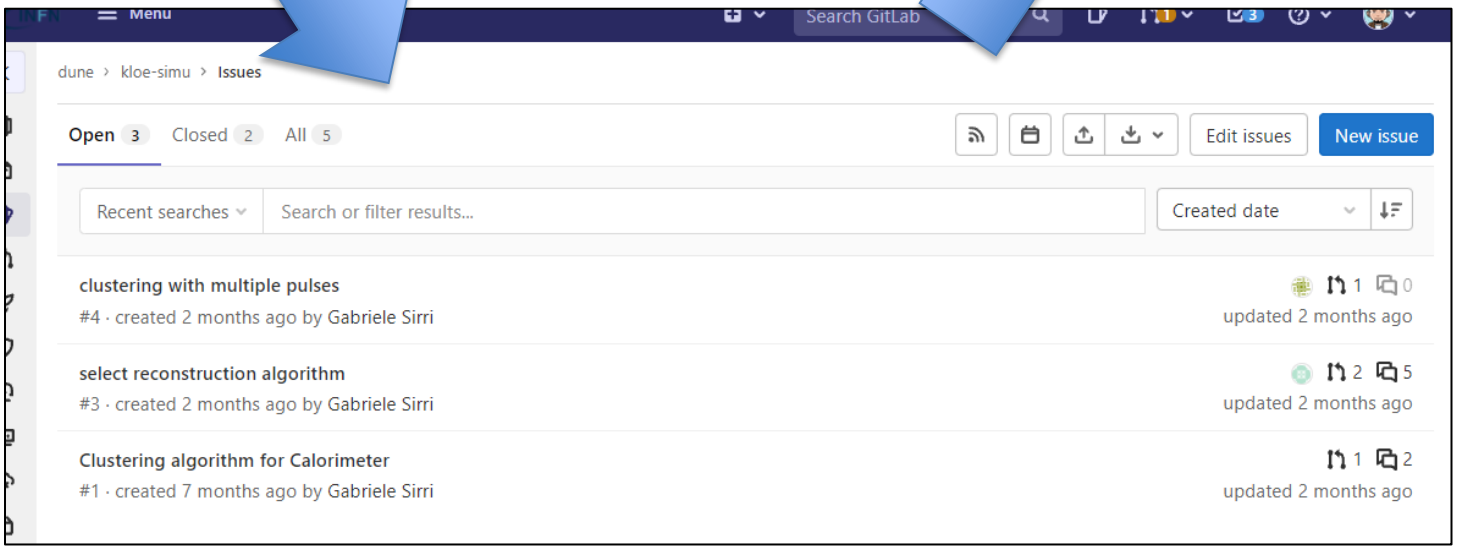
or
very minor commits
- code formatting
- bug fix



1. Create Issue
2. Open Branch
3. Develop
4. Test / Review
5. Merge Request
6. Release







1. Create Issue
2. Open Branch
3. Develop
4. Test / Review
5. Merge Request
6. Release

Testing and code review may be permorfed at each PUSH by other developers or continous integration

```
tentineutrino@neutrino-01:analysis (redesign-STTCluster)$ git checkout -b new_branch
Switched to a new branch 'new_branch'
tentineutrino@neutrino-01:analysis (new_branch)$ echo new_file > new_file
tentineutrino@neutrino-01:analysis (new_branch)*$ git add new_file
tentineutrino@neutrino-01:analysis (new_branch)*$ git commit -m "created new_file"
[new_branch 8950a83] created new_file
 1 file changed, 1 insertion(+)
 create mode 100644 new_file
tentineutrino@neutrino-01:analysis (new_branch)$ git push --set-upstream origin new_branch
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 281 bytes | 281.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote:
remote: To create a merge request for new_branch, visit:
remote:   https://baltig.infn.it/tenti/STTTrackReco/-/merge_requests/new?merge_request%5Bsource_branch%5D=new_branch
remote:
To baltig.infn.it:tenti/STTTrackReco.git
 * [new branch]      new_branch -> new_branch
Branch 'new_branch' set up to track remote branch 'new_branch' from 'origin'.
tentineutrino@neutrino-01:analysis (new_branch)$
```

1. Create Issue
2. Open Branch
3. Develop
4. Test / Review
5. Merge Request
6. Release