cern.ch/allpix-squared

Collaborative Coding

The Example of Allpix Squared

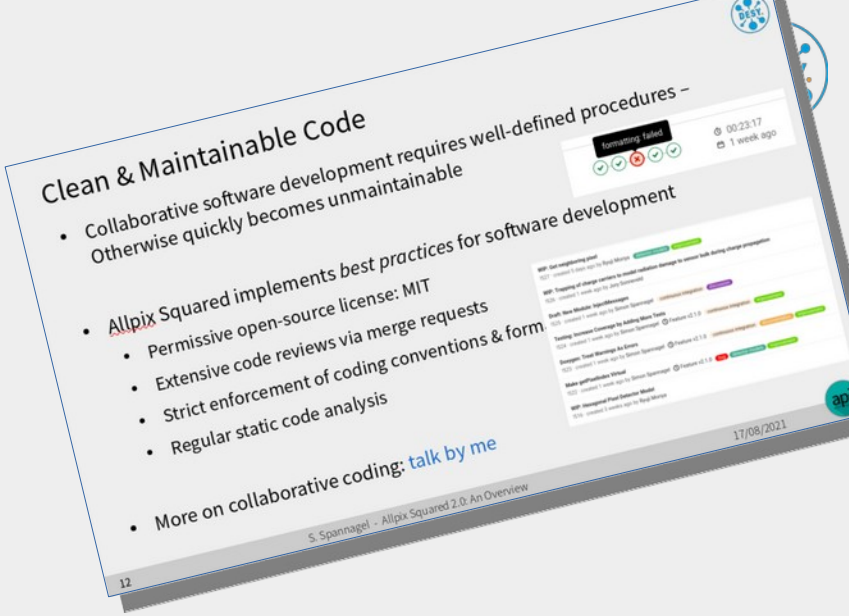
Simon Spannagel, DESY

2nd Allpix Squared User Workshop

19 August 2021

Challenges of Collaborative Coding

Scientific software thrives in collaboration –
diverse ideas, inputs, use cases, applications



Contributors have different…

- Levels of experience in the given language
- Coding styles / editor settings (e.g. indentation)
- Ideas on “what looks nice” / naming conventions
- Expectations to the software, different levels of scrutiny

...but in the end there should be one coherent “product”

What We Are Trying to Avoid

- Random indentation
- Comments of unknown relevance
- Non-functional code
- Unused code / variables
- Code that only works on specific platforms / with specific 3rd-party library versions

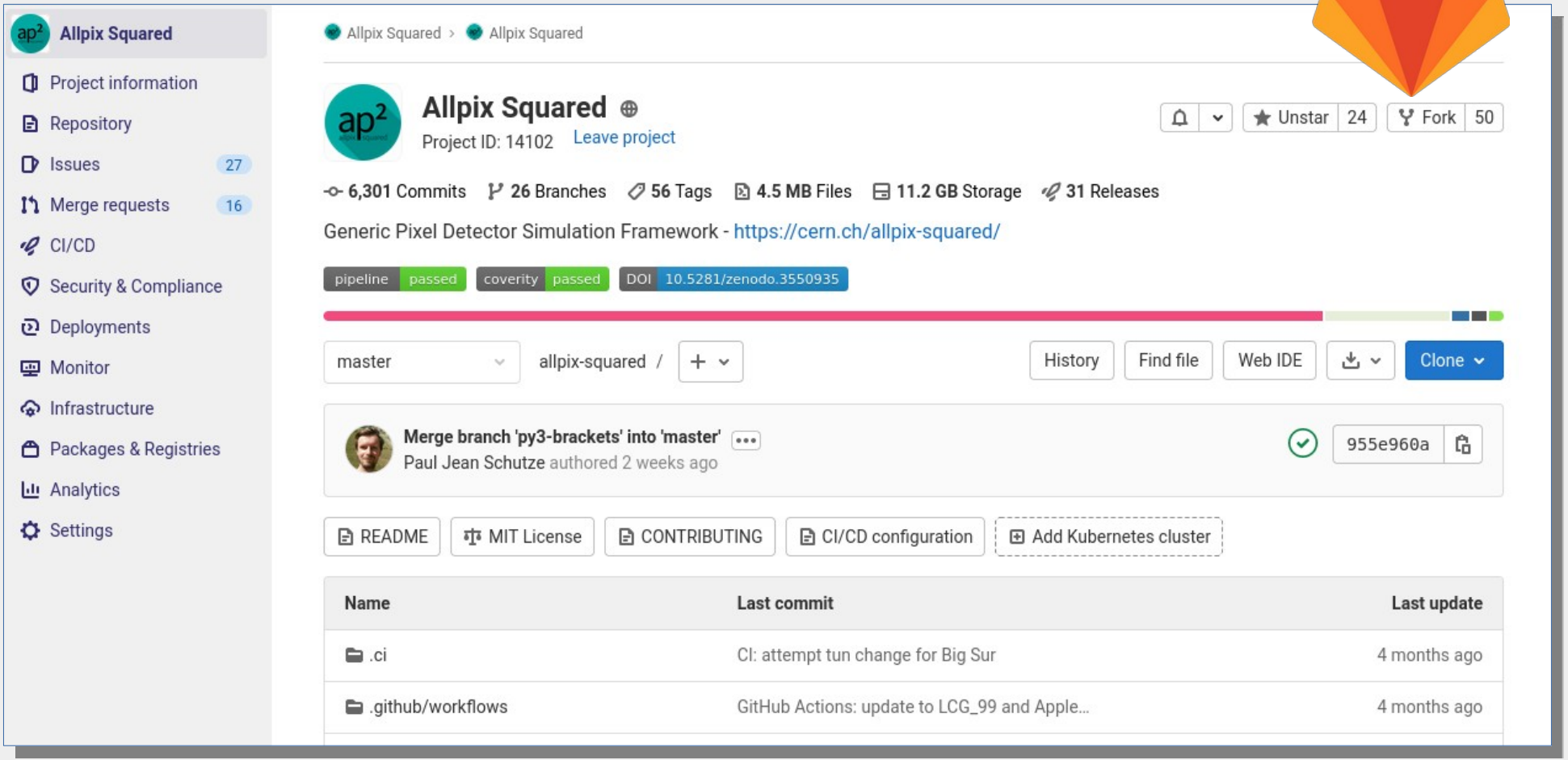
```
118             const double dr0x = (*itrAlignmentConstant).second->getXOffset();
119             const double dr0y = (*itrAlignmentConstant).second->getYOffset();
120             const double dr0z = (*itrAlignmentConstant).second->getZOffset();
121
122                         const double posLocalDiff[3] = {dr0x,dr0y,dr0z};
123                         double delta_r0[3];
124                         geo::gGeometry().local2MasterVec(sensorID,posLocalDiff, delta_r0);//Here we transform the local alignment
125                         const double posTest[3]={1,0,0};
126                         double posTestOutput[3];
127                         geo::gGeometry().local2Master(sensorID,posTest, posTestOutput);
128                         streamlog_out(MESSAGE9)<<"Here we have the test for sensor " << sensorID <<std::endl;
129                         streamlog_out(MESSAGE9)<<posTestOutput[0]<<"  "<<posTestOutput[1]<<"  "<<posTestOutput[2]<<endl;
130                         const double angleLocalDiff[3]={dalpha,dbeta,dgamma};
131                         double delta_angle[3];
132                         //IMPORTANT:Note the transformation of the angles assumes that they transform like a vector. This is not
133                         geo::gGeometry().local2MasterVec(sensorID,angleLocalDiff, delta_angle);//Here we transform the local alig
134
135
136
137 //          delta_r0 *= invR;
138
139 //#ifdef GEAR_MAJOR_VERSION
140 //#if GEAR_VERSION_GE( 17,4)
141 // ZY and ZX rotations are calculated wrongly yet, do not implement:
142 // XYZ shifts and XY rotation seems to be correct
143 //
144             geo::gGeometry().alignGlobalPos(sensorID,  xplane  + delta_r0[0], yplane  + delta_r0[1], zplane  + delta_r0[2]  ) ;
145
146 //#endif
147 //#endif
148                                         streamlog_out(MESSAGE9) << "Input and output alignment shift (translations) for sensor:
149             streamlog_out(MESSAGE4) << setw(10) << "Align Translations (Local) x,y,z  " << setw( 8) << " " ;
```

Development of Allpix Squared

- Meant as community-driven project
  - Many contributions from “core team” (+ code reviews)
  - Increasing number of external contributors (~40 by now)
- Fully GitLab-centered development
  - Issue tracking, merge requests, continuous integration
- All development is performed “in the open”
  - Full repository public, including issues
  - Subscribers receive information on all actions
- Semantic versioning:
  Major.Minor.Patch = **Framework.Features.Bugfixes**

v2.0.1	2021-07-09
v2.0	2021-06-10
v1.6.2	2021-04-01
v1.6.1	2021-01-28
v1.6	2020-10-29
v1.5.2	2020-09-14
v1.5.1	2020-07-26
v1.5	2020-04-14
v1.4.4	2020-03-10
v1.4.3	2020-01-10
v1.4.2	2019-11-26
v1.4.1	2019-09-13
v1.4	2019-07-09
v1.3.4	2019-06-07
v1.3.3	2019-04-13
v1.3.2	2019-02-21
v1.3.1	2018-12-17
v1.3	2018-11-21
v1.2.3	2018-11-13
v1.2.2	2018-09-07
v1.2.1	2018-08-02
v1.2	2018-06-13
v1.1.2	2018-04-25
v1.1.1	2018-03-08
v1.1	2018-01-11
v1.0	2017-08-[illegible]

GitLab as Nerve Center for Development



Code Review via Merge Requests

- No new code lands in *master* without review by another party
  - Using GitLab's approval feature
  - Extensive discussions about code, but also style, naming schemes
- Proven to be **very effective**
  - Several bugs found before the merge
  - New users appreciate guidance
- Proven to be **labor-intensive**
  - Read (and understanding) every change
  - Always be supportive, positive

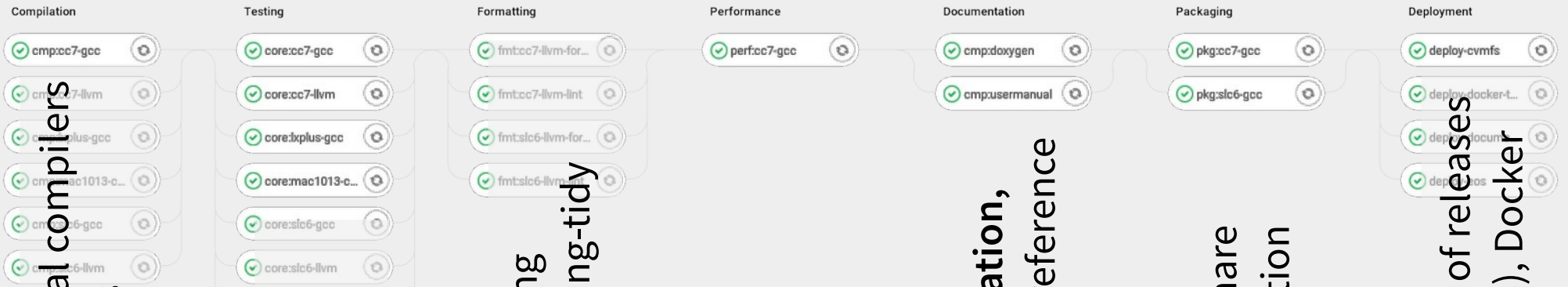
...just some of them

Python macro to read output objects TTree !191 · opened 2 months ago by Sebastien Murphy		MERGED 18 updated 1 week ago
Revamp MeshConverter: Change interpolation & improve performance !200 · opened 3 weeks ago by Simon Spannagel		MERGED 5 updated 1 week ago
Write full Proteus configuration in RCEWriter !203 · opened 2 weeks ago by Moritz Kiehn		MERGED 12 updated 1 week ago
Invert Detector Rotations !164 · opened 6 months ago by Simon Spannagel	documentation, detector models, bug	MERGED 8 updated 2 weeks ago
RCEWriter: fix Proteus geometry output !202 · opened 2 weeks ago by Moritz Kiehn		MERGED 3 updated 2 weeks ago
FieldParser: be more careful about units !201 · opened 3 weeks ago by Simon Spannagel		MERGED 1 updated 2 weeks ago
Add option for a depletion from the backplane !198 · opened 3 weeks ago by Paul Schutze	physics, improvement	MERGED 11 updated 3 weeks ago
New Field File Format APF & common FieldParser/FieldWriter !197 · opened 1 month ago by Simon Spannagel		MERGED 11 updated 3 weeks ago
New Module: DepositionPointCharge !194 · opened 1 month ago by Simon Spannagel		MERGED 12 updated 1 month ago

CI/CD

Continuous Integration & Deployment

Continuous Integration Pipeline



compilation with several compilers on 3 different platforms

many **unit tests** for modules & framework

code formatting & linting using clang-format / clang-tidy

performance tests to monitor execution time

building the **documentation,** user manual and code reference

packaging of bin, lib, share into tar ball for distribution

automatic **deployment** of releases to CVMFS, website (EOS), Docker

Automated Testing

```
[Allpix]
detectors_file = "detector_rotate_misaligned.conf"
log_level = "TRACE"
number_of_events = 0
random_seed = 0
random_seed_core = 0

[GeometryBuilderGeant4]

#PASS (DEBUG)  misaligned: (8.72466deg,171.099deg,178.504deg)
```

```
Test project /builds/allpix-squared/allpix-squared/build
      Start 53: test_core/test_01-1_globalconfig_detectors.conf
      Start 54: test_core/test_01-2_globalconfig_modelpaths.conf
      Start 55: test_core/test_01-3_globalconfig_log_format.conf
      Start 56: test_core/test_01-4_globalconfig_log_level.conf
 1/22 Test #53: test_core/test_01-1_globalconfig_detectors.conf .........   Passed    0.81 sec
      Start 57: test_core/test_01-5_globalconfig_log_file.conf
 2/22 Test #56: test_core/test_01-4_globalconfig_log_level.conf .........   Passed    2.11 sec
 3/22 Test #55: test_core/test_01-3_globalconfig_log_format.conf .........   Passed    2.11 sec
      Start 58: test_core/test_01-6_globalconfig_missing_model.conf
      Start 59: test_core/test_01-7_globalconfig_random_seed.conf
 4/22 Test #57: test_core/test_01-5_globalconfig_log_file.conf ...........   Passed    1.11 sec

[...]

100% tests passed, 0 tests failed out of 22
```

- Automatically re-run simulations with known outcomes and check them
- Framework & module tests

- Each test is a configuration file:
  - Run single event with fixed seed
  - Reproduces same output
  - Matching regular expressions
- Single change (1e difference) fails test
  → adaptation of test in case of expected change
- Invaluable for monitoring framework
  → catching issues before merging code

Warnings, Strict Formatting & Code Linting

- In order to maintain a high code quality, we
  - Enabled many compiler warnings, with **-Werror**
  - Require strict adherence to code formatting (indentation, brace positions…)
  - Clang-tidy is used to spot e.g. missing &, std::move() or NamingConventionViolations

- A bit painful for newcomers
  CI fails many times, users need to be taught how to read failed job logs

- Particularly useful: /etc/git-hooks
  - **pre-commit-clang-format-hook** – checks formatting before committing
  - **pre-push-tag-version-hook** – check for pre-release things (update version)
- CMake suggests installing hooks (suggests update on changes)

Automated Deployment of Tagged Versions

- After release, new version needs to be distributed
- CI automatically deploys new version:
  - Deployment to **website**
    Binary tarball packages placed on EOS
  - Deployment as **Docker image**
    available via GitLab's Docker repository
  - Deployment to **CVMFS** (Cern Virtual Machine File System)
    ready for use on LXPlus or HTCondor submission

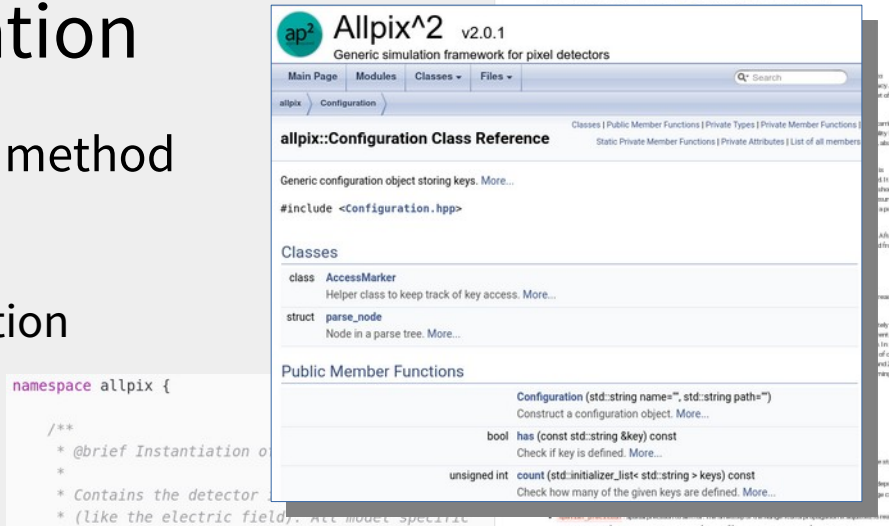
```
deploy-docker-tag:
    stage: deployment
    tags:
        - docker-image-build
    dependencies: []
    only:
        - tags
    script:
        - "echo" # unused but this line is required by GitLab CI
    variables:
        TO: gitlab-registry.cern.ch/allpix-squared/allpix-squared:${CI_COMMIT_TAG}
```

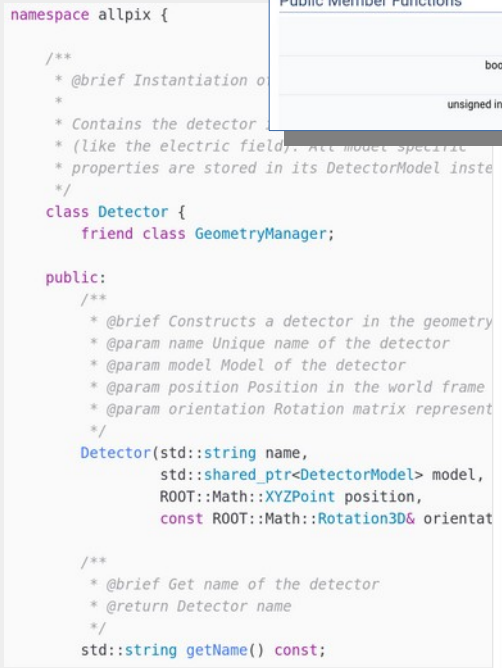
```
$ source /cvmfs/clicdp.cern.ch/software/allpix-squared/2.0.1/x86_64-centos7-gcc10-opt/setup.sh
$ allpix --version
Allpix Squared version v2.0.1
                built on 2021-07-09, 09:26:08 UTC
```

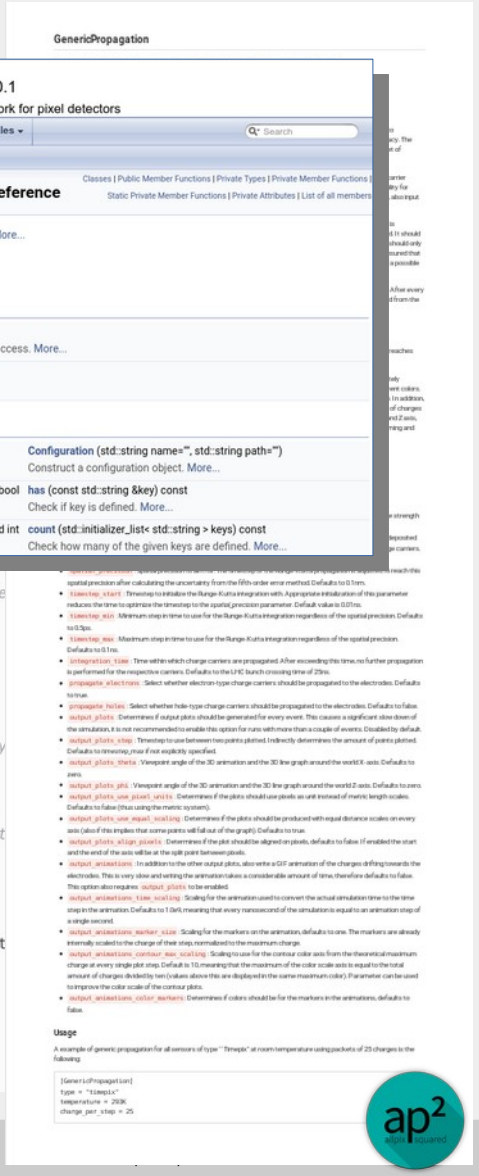
User Manual & Code Documentation

- Source code documentation for every class & method
  - Doxygen markup for code reference
  - CI checks for complete & correct documentation
  - Deployed to the website for every release

- User Manual (currently) in LaTeX
  - Module documentation as Markdown
    - Document module parameters, algorithms
    - Included in manual via Pandoc
  - Manual automatically compiled by CI
  - Deployed to the website for every release







How to Contribute

A Cookbook

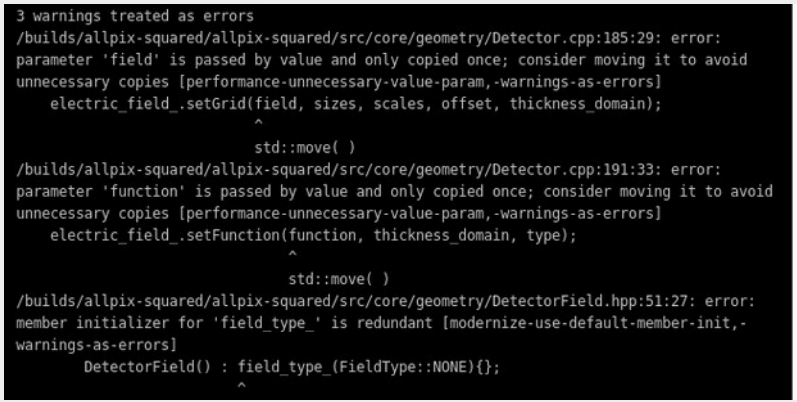
How To Contribute – A Cookbook

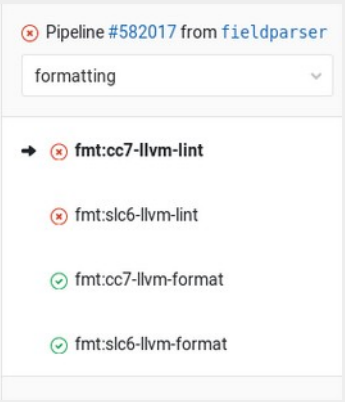
- **Get in touch** – mail, forum, issue tracker, …
  Let's discuss the idea, maybe we have input, maybe others are working on it already

- **Fork the repository**
  Creating your own copy of the code with which you can mess as much as you want

- **Start hacking**
  Implement the desired functionality, come back to us when you have doubts or questions

- **Make sure the CI passes**
  Enable the CI in your fork and publish
  your new code there – check that the CI works!
- **File a Merge Request**
  This provides us a central point to discuss and
  review all your code changes

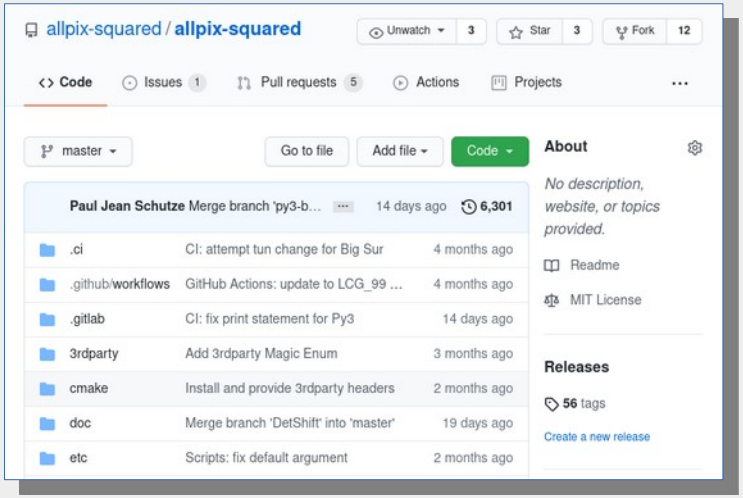


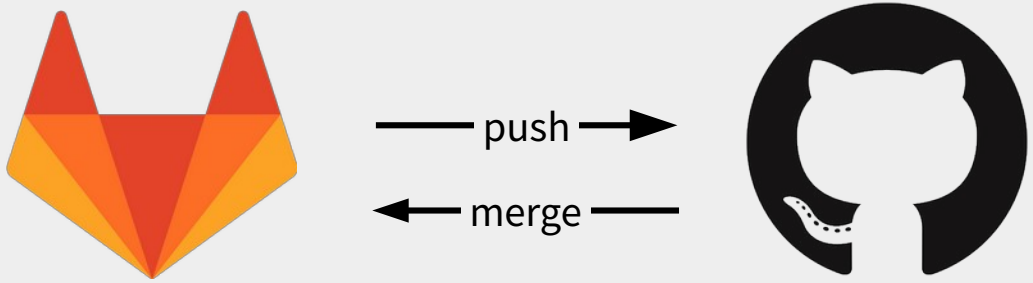


- **See your code being merged and published!**

No CERN Affiliation – no GitLab – no Problem

- We rely on CERN's GitLab CI (and our own runners)
  - Allows to run extensive CI and adjust to our needs
  - Requires full CERN account for write access
- Our solution: ping-pong mirror to GitHub
  Follow the same cookbook…



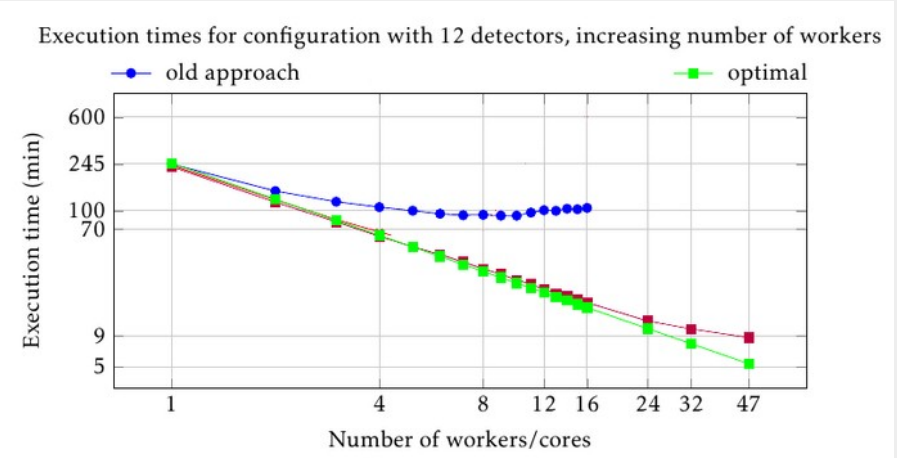


allpix-squared
the main repository

allpix-squared
the mirror repository

Google Summer of Code

- Participated twice in GSoC through HEP Software Foundation / CERN
  - Students work over summer months on FOSS project and get stipend paid by Google
  - Project: Event-based Multi-Threading for Allpix Squared
  - Quite intricate (seeding, race conditions, … – see K. Wolter's talk)
- Students were very active and motivated
  - Restructured parts of core framework
  - Implemented first working version
  - Did first benchmarks



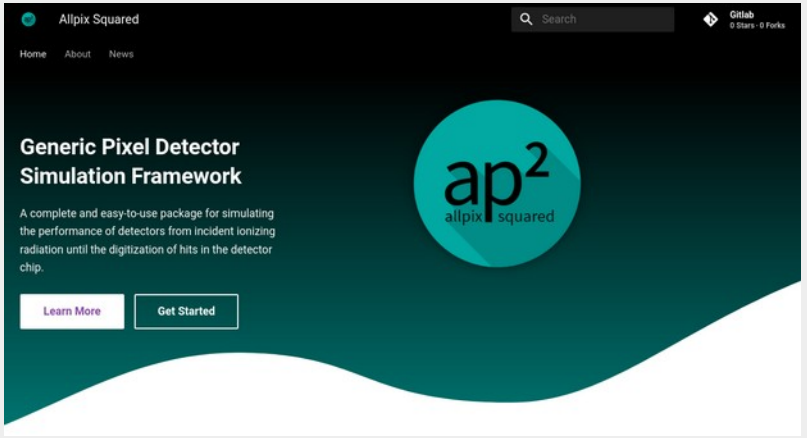
- Interesting observation:

  Students were expecting direct instructions,
  we expected scientific collaboration → worked out well after discussing!

Google Season of Docs

...is ***not*** Google Summer of Code!

- Scholarship for experienced technical writers to work on documentation of open source projects
- Allpix Squared participated through HEP Software Foundation / CERN
- Technical writer worked for three months on documentation/website
  - Goal: revision of online appearance
  - Focus on integration of online user manual
  - Improvements to tutorials/examples
- Finalization still pending,
  lined up for next feature release



Summary

- Common conventions are important for collaborative software projects

- Allpix Squared implements
  - Strict formatting and naming conventions
  - Code review through merge requests
  - Automated testing and deployment

- Continuous Integration is a powerful tool to ensure consistency & functioning

- Contributions possible & welcome through CERN GitLab or GitHub repositories!

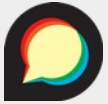
Allpix Squared Resources

Website
https://cern.ch/allpix-squared

Repository
https://gitlab.cern.ch/allpix-squared/allpix-squared

Docker Images

https://gitlab.cern.ch/allpix-squared/allpix-squared/container_registry



User Forum:

https://cern.ch/allpix-squared-forum/

Mailing Lists:

allpix-squared-users https://e-groups.cern.ch/e-groups/Egroup.do?egroupId=10262858

allpix-squared-developers https://e-groups.cern.ch/e-groups/Egroup.do?egroupId=10273730

User Manual:
https://cern.ch/allpix-squared/usermanual/allpix-manual.pdf


DESY.