

Software Quality Control at Belle II

Martin Ritter ¹, Thomas Kuhr ¹, Thomas Hauth ², Timothy Gebhard ², Michal Kristof ³,
Christian Pulvermacher ⁴ for the Belle II Software Group

¹Ludwig-Maximilians-University, Munich

²Institute for Experimental Nuclear Physics (IEKP), KIT

³Charles University, Prague

⁴High Energy Accelerator Research Organization (KEK), Japan

22nd International Conference on Computing
in High-Energy and Nuclear Physics
October 12, 2016

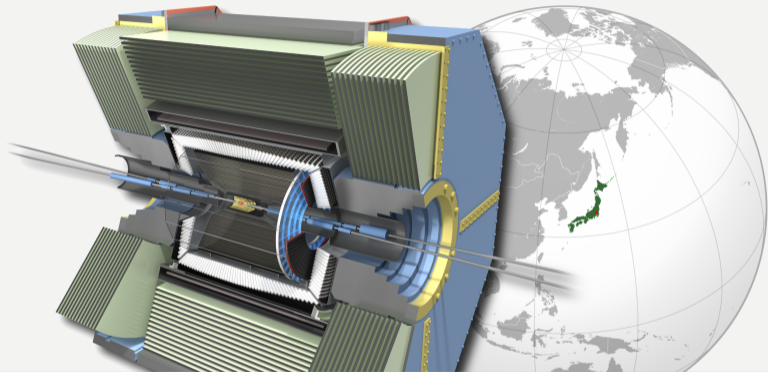


Bundesministerium
für Bildung
und Forschung

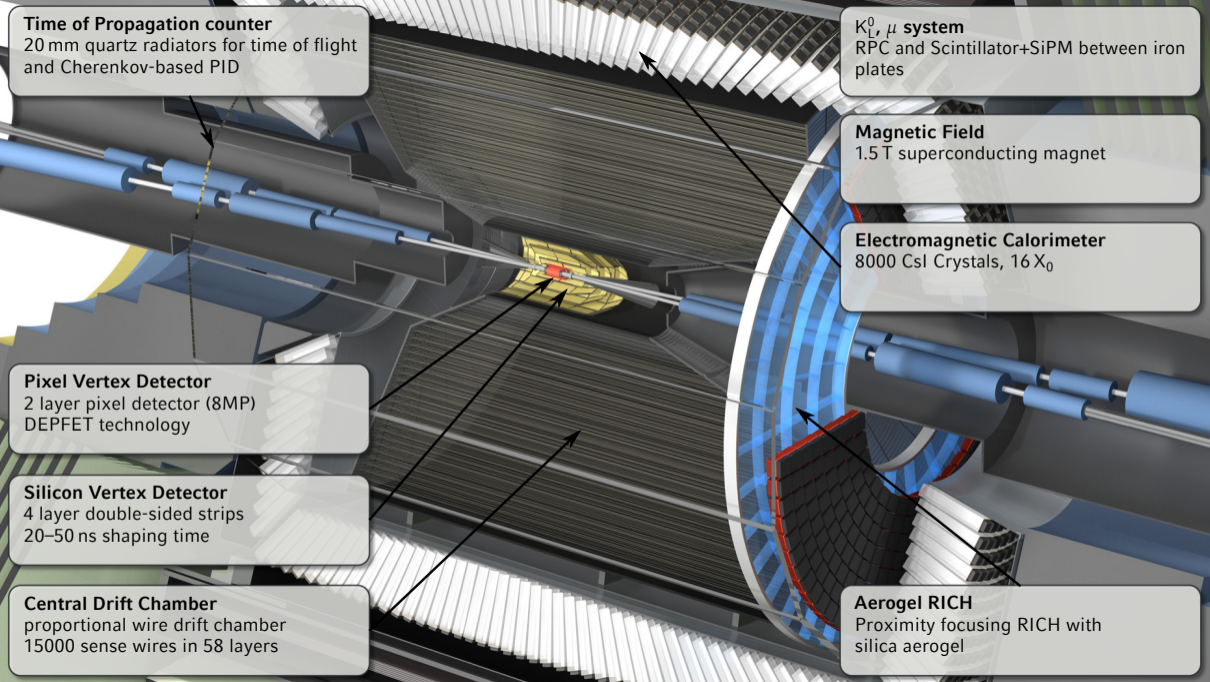


Asymmetric e^+e^-
experiment mainly at
the $\Upsilon(4S)$ resonance
(10.58 GeV)

Focus on B, charm and
 τ physics



	KEKB/Belle	SuperKEKB/Belle II
operation	1999–2010	2018–2024
peak luminosity	$2.11 \times 10^{34} \text{ cm}^{-2}\text{s}^{-1}$	$8 \times 10^{35} \text{ cm}^{-2}\text{s}^{-1}$
integrated luminosity	1023 fb^{-1} (772 million $B\bar{B}$ pairs)	50 ab^{-1}



Time of Propagation counter
20 mm quartz radiators for time of flight and Cherenkov-based PID

K_L^0, μ system
RPC and Scintillator+SiPM between iron plates

Magnetic Field
1.5 T superconducting magnet

Electromagnetic Calorimeter
8000 CsI Crystals, $16 X_0$

Pixel Vertex Detector
2 layer pixel detector (8MP)
DEPFET technology

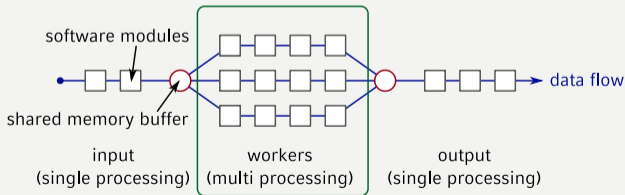
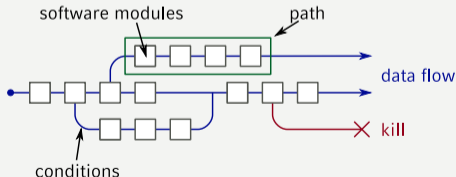
Silicon Vertex Detector
4 layer double-sided strips
20–50 ns shaping time

Central Drift Chamber
proportional wire drift chamber
15000 sense wires in 58 layers

Aerogel RICH
Proximity focusing RICH with silica aerogel

Mainly written from scratch using experiences from Belle and other experiments

- ▶ modular approach
- ▶ utilize new technologies: C++11 (GCC 5.2), ROOT 6, Geant 4.10, Python 3.5
- ▶ Python as steering/scripting language
- ▶ ROOT for input/output (also raw data)
- ▶ parallel processing support using `fork`





- ▶ approximately one million lines of code organized in 30 packages
- ▶ one librarian responsible for each package
- ▶ decides on commit permissions in package
- ▶ organized in central git repository

Git Version Control

- ▶ switched from SVN this summer
- ▶ hosted at DESY using Atlassian tools (Stash)
- ▶ works smoothly in most cases, still some tuning to be done



- ▶ librarians free to decide on development workflow
- ▶ monthly integration build to avoid divergence



Used very linear history in SVN

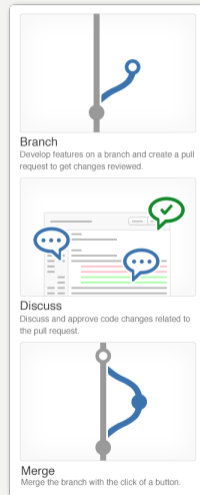
- ▶ cross-package changes required lots of coordination
- ▶ patches by email

After switching to Git/Stash

- ▶ commits to master branch still possible
- ▶ users quickly adopting to using branches
- ▶ pull-requests to simplify coordination of larger changes

Extensive Style and Permission Checks

- ▶ enforce common code style using artistic-style, pycodestyle
- ▶ check package permissions on master branch
- ▶ implemented in local pre-commit and server pre-receive hook



© Atlassian, modified

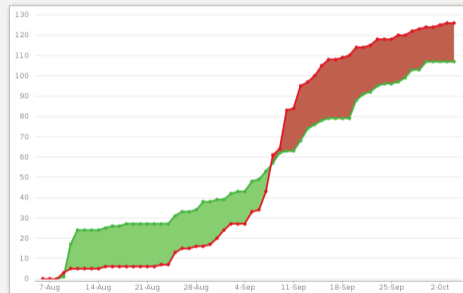


Issue tracking

- ▶ switched from Redmine to Atlassian JIRA
- ▶ old issues converted
- ▶ good user acceptance so far

Build Services

- ▶ using buildbot and Atlassian Bamboo
- ▶ incremental build on commits for all branches



The number of issues **created** vs. the number of issues **resolved** in the last 60 days.

Nightly build creates reports on

- ▶ compiler warnings (gcc, clang, icc)
- ▶ cppcheck, clang static analyzer
- ▶ missing doxygen documentation
- ▶ valgrind memcheck
- ▶ geometry overlaps
- ▶ missing/extra library dependencies
- ▶ unittest results

Results of development build						
Monday, October 03, 2016						
Revision: 54f441f						
failure All Libraries Modules Packages						
Package details						
Package	Librarian	Build Result	Intel Build Result	Clang Build Result	Cppcheck	Test Result
alignment	Tadeas Bilka	OK	OK	OK	Remarks: 4	0/0, 0/0
analysis	Anze Zupanc	OK	OK	OK	Remarks: 30	0/104, 0/15
arich	Luka Santelj	Warnings: 2	OK	OK	Warnings: 2 Remarks: 71	0/0, 0/1
b2bill	Anze Zupanc	OK	OK	OK	Error: 1 Remarks: 1	None
background	Marko Starič	OK	OK	OK	OK	None
beast	Igal Jaegle	Warnings: 4	OK	OK	Remarks: 13	None
bikm	Leo Piilonen	OK	OK	OK	OK	0/0, 0/1
calibration	Tadeas Bilka	OK	OK	OK	Remarks: 5	0/0, 1/2
cdc	Nakano Eiichi	OK	OK	Warnings: 1	OK	0/0, 0/1
daq	Ryosuke Itoh	OK	OK	OK	OK	None
decfiles	Phillip Urquijo	OK	OK	OK	OK	None
display	Christian Pulvermacher	OK	OK	OK	OK	None
eci	Torben Ferber, Miyabayashi Kenkichi	OK	OK	Warnings: 1	Warnings: 1 Remarks: 55	0/12, 0/0
ekim	Kiril Chikin	OK	OK	OK	Remarks: 57	None



Three levels of testing and validation

Unittests

- ▶ using Google Test
- ▶ cover large part of the core framework, sparse use in other packages
- ▶ limited in complexity of setup/teardown

Framework tests

- ▶ run steering files in special folders
- ▶ check exit code
- ▶ optionally check output
- ▶ allows for larger complexity of tests

Directory	Line Coverage ↕	Functions ↕
conditions/src	49.1 % 111 / 226	62.5 % 10 / 16
core/src	76.7 % 1072 / 1397	87.3 % 192 / 220
database/src	75.7 % 465 / 614	76.0 % 73 / 96
dataobjects/src	84.5 % 490 / 580	84.3 % 70 / 83
datastore/src	92.8 % 532 / 573	94.1 % 80 / 85
dbobjects/src	98.1 % 52 / 53	100.0 % 10 / 10

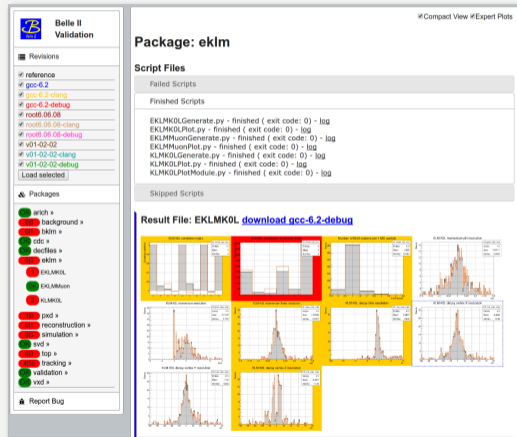
➔ run automatically after commits to master, user notification on fail

Three levels of testing and validation (cont.)

Physics Validation

- ▶ run steering files in special folders honoring dependencies
- ▶ create comparison plots from output root files
- ▶ compare between revisions and against known template
- ▶ run once per day

- ➔ all three can be run by the developers, either for selected packages or everything
- ➔ tools to simplify `git bisect` usage for finding responsible commits



Belle II Validation

Revisions

- reference
- gcc-6.2
- gcc-6.2-clang
- gcc-6.2-debug
- root6.06.08
- root6.06.08-clang
- root6.06.08-debug
- v01-02-02
- v01-02-02-clang
- v01-02-02-debug

Load selected

Packages

- arch >
- background >
- bkim >
- cdc >
- dcfiles >
- ekim >
- EKLMKOL
- EKLMMuon
- KLMKOL
- pxd >
- reconstruction >
- simulation >
- svd >
- top >
- tracking >
- validation >
- vvd >

Report Bug

Package: ekim

Script Files

Failed Scripts

Finished Scripts

- EKLMKOLGenerate.py - finished (exit code: 0) - log
- EKLMKOLPlot.py - finished (exit code: 0) - log
- EKLMMuonGenerate.py - finished (exit code: 0) - log
- EKLMMuonPlot.py - finished (exit code: 0) - log
- KLMKOLGenerate.py - finished (exit code: 0) - log
- KLMKOLPlot.py - finished (exit code: 0) - log
- KLMKOLPlotModule.py - finished (exit code: 0) - log

Skipped Scripts

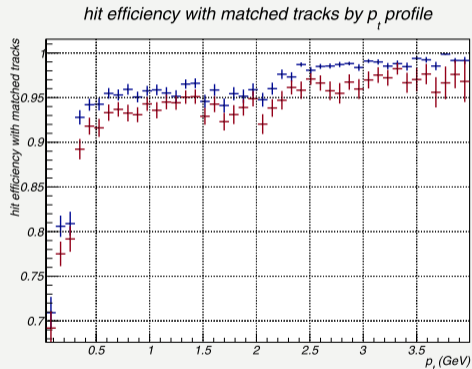
Result File: EKLMKOL download gcc-6.2-debug

Comparison plots showing distributions for various variables, including ΔE and ΔE_{rel} , comparing different simulation and reconstruction stages.



Currently migrating to GCC 6.2

- ▶ ROOT version 6.06/00 showed problems in unittests
- ▶ updated ROOT and ran full physics validation
- ▶ degradation in performance between GCC 5.2 and GCC 6.2
- ➔ erroneous `abs(double)` which gets compiled as `::abs(int)` or `std::abs(double)`
- ✓ problem could be fixed before migration





Weekly developer meetings

- ▶ informal, mostly oral
- ▶ short reports by librarians
- ▶ questions by users/developers
- ▶ quality shift report

Monthly software meeting

- ▶ more formal, fixed agenda
- ▶ presentations of new features
- ▶ discussion of release cycle

Software Quality Shifter

- ▶ weekly shifts to check for problems
- ▶ inquire about old issues, check style guides and documentation, review changes...
- ▶ counts as service task



Code Management

- ▶ switched from SVN to Git this year
- ▶ utilize branches/pull-requests to simplify collaboration
- ▶ sophisticated build service and monthly integration builds

Code Quality

- ▶ enforced unified code style
- ▶ different compilers (gcc, clang, icc)
- ▶ cppcheck, doxygen, memcheck, clang static-analyzer

Validation and Testing

- ▶ three levels of testing: Unit tests, framework tests and physics validation
- ▶ all runnable by the user

Others

- ▶ short weekly meetings, detailed monthly meetings
- ▶ software quality shift

Thank you
for your attention

