C++ Software Quality in the ATLAS experiment: Tools and Experience

Stewart Martin-Haugh (STFC - Rutherford Appleton Lab. (GB))  Stefan Kluth (Max-Planck-Institut für Physik (Werner-Heisenberg-Institut) (D))  Rolf Seuster (University of Victoria (CA))  Scott Snyder (Brookhaven National Laboratory (US))  Emil Obreshkov (University of Texas at Arlington (US))  Shaun Roe (CERN)  Peter Sherwood (University College London (UK))  Graeme Stewart (University of Glasgow (GB))

ATLAS Software:
~6 000 000 lines of Code; 140 Teams; 420 Developers; 2 Software Quality Coordinators

Information: TWikis provide education, coding tips, access to tools.
Tools: Continuous build, Static Analysis (Coverity®, cppcheck), Sanitizers, gcc plugins, Unit tests, “lizard”.

TWikis

Nightly Build

Nightly Builds are performed across different compilers (e.g. Clang, gcc49), and developers are informed of errors by email.

In addition, custom gcc plugins have been written to check for violation of internal standards (e.g. naming conventions, inheritance conventions)

The Undefined Behaviour Sanitizer (UBSan) is active in debug builds (one example: left shift of negative number).

Unit tests are performed as part of the build, including (more recently) tests in the GoogleMock framework which allow testing of complex objects.

As a final step, Run Time Tests provide physics parameters (e.g. pT or angular distributions) as a holistic test of the software for comparison against known references

Static Analysis: Coverity®, cppcheck, lizard

Coverity and cppcheck are run twice-weekly. Coverity reports are sent once per week to developers. Lizard provides cyclomatic-complexity and line count metrics which can be useful to spot rotten code. These tools all have results presented as a league table of defects per software team, intended to motivate the teams. Coverity is the most comprehensive, and has resulted in many defects being found; the progress over time is shown left. Current defect densities are 0.2 - 0.8 defects/1000-lines-of-code

The Future

ATLAS is migrating from SVN to Git, which allows a formal code review before acceptance into the repository. The tools shown above will likely form one input to the review process, but finally code quality depends on the developers motivation and the reviewers conscientiousness.