



Contribution ID: 57

Type: Oral

## Continuous software quality analysis for the ATLAS experiment

*Tuesday, 22 August 2017 15:20 (20 minutes)*

The regular application of software quality tools in large collaborative projects is required to reduce code defects to an acceptable level. If left unchecked the accumulation of defects invariably results in performance degradation at scale and problems with the long-term maintainability of the code. Although software quality tools are effective for identification there remains a non-trivial sociological challenge to resolve defects in a timely manner. This is an ongoing concern for the ATLAS software which has evolved over many years to meet the demands of Monte Carlo simulation, detector reconstruction and data analysis. At present over 3.8 million lines of C++ code (and close to 6 million total lines of code) are maintained by a community of hundreds of developers worldwide. It is therefore preferable to address code defects **before** they are introduced into a widely used software release.

Recent wholesale changes to the ATLAS software infrastructure have provided an ideal opportunity to apply software quality evaluation as an integral part of the new code review workflow. The results from static code analysis tools - such as *cppcheck* and *Coverity* - can now be inspected by a rota of review shifters as part of a continuous integration (CI) process. Social coding platforms (e.g. *Gitlab*) allow participants in a code review to consider identified defects and to decide upon any action before code changes are accepted into a production release. A complete audit trail on software quality considerations is thus provided for free as part of the review.

The methods employed to incorporate software quality tools into the ATLAS software CI process will be presented. The implementation of a container-based software quality evaluation platform designed to emulate the live infrastructure will be described with a consideration of how continuous software quality analysis can be optimised for large code bases. We will then conclude with a preview of how analytics on test coverage and code activity - useful in steering the prioritisation of defect resolution - can be incorporated into this new workflow.

**Primary author:** WASHBROOK, Andrew John (University of Edinburgh (GB))

**Presenter:** WASHBROOK, Andrew John (University of Edinburgh (GB))

**Session Classification:** Track 1: Computing Technology for Physics Research

**Track Classification:** Track 1: Computing Technology for Physics Research