Improving Code Quality of the Compact Muon Solenoid Electromagnetic Calorimeter Control Software to Increase System Maintainability

O. Holme[1], D. Di Calafiori[1], G. Dissertori[1], L. Djambazov[1], W. Lustermann[1], S. Zelepoukine[1,2]

On behalf of the CMS ECAL group

Manual Software Quality Inspection

- Time intensive
- Identifies repeated functionality implementation
- Provides qualitative feeling of code quality
- Easy to miss code problems
- No quantitative feedback of changes over time

VS.

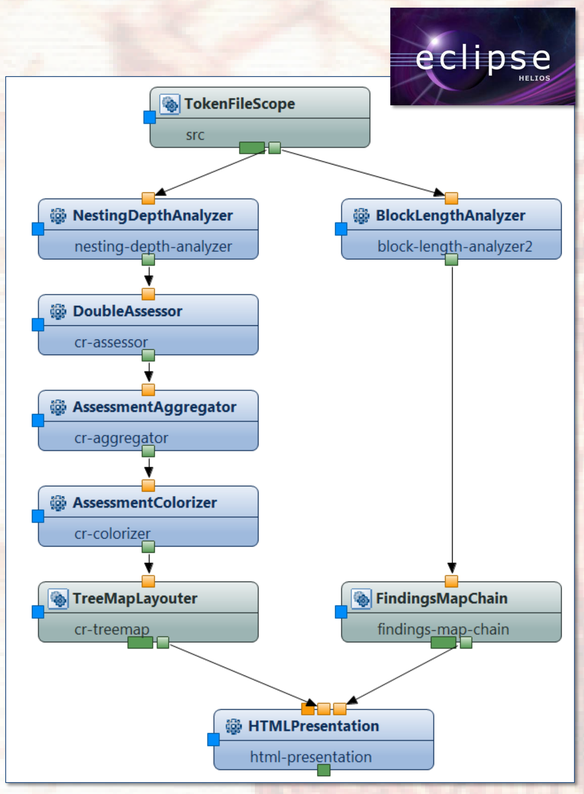
Automatic Software Quality Assessment

- Analysis performed in minutes
- Finds all code that does not meet criteria
- Quality thresholds must be manually chosen
- Some results may be false positives
- Easy to track metrics over time

Selecting ConQAT for quality control

Open-source software quality toolkit

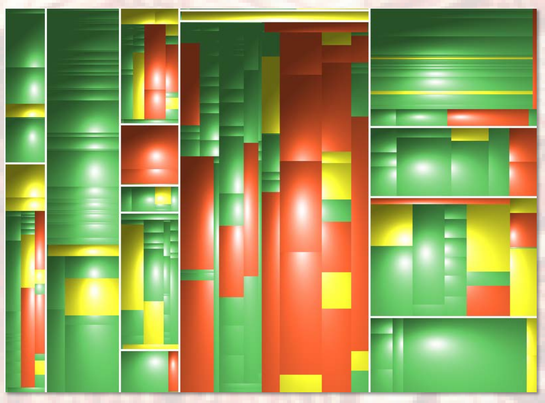
Customizable workflows, e.g.:



Choice of quality metrics, e.g.:

- Code duplication (cloning)
- Long code blocks
- Deeply nested blocks
- Code-comment ratio

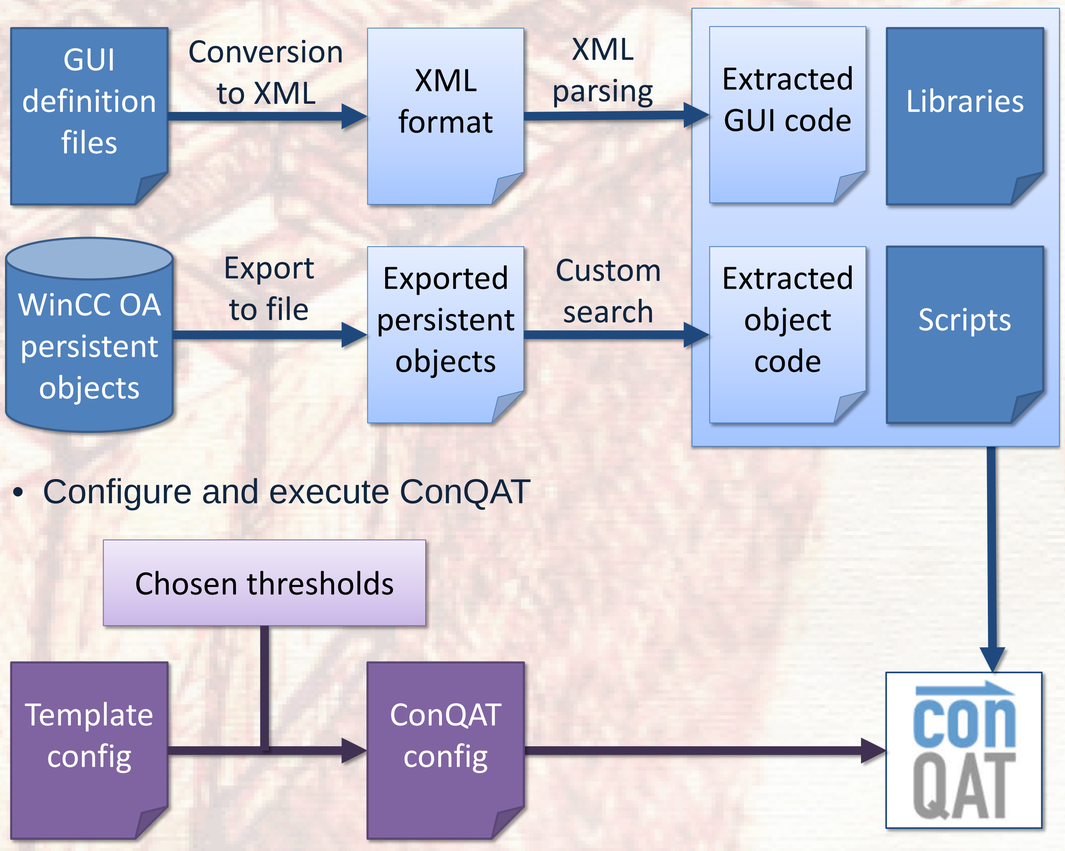
Graphical HTML output provides clear feedback to developers, e.g.:



- Rectangle size indicates file length
- Colour shows block nesting depth

Integrating ConQAT into WinCC Open Architecture (WinCC OA)

- WinCC OA is a supervisory control and data acquisition system
  - Programmed in a custom script language (CTRL)
- Extract and collect CTRL code from all sources



- Configure and execute ConQAT
- Enables manual or automatic checks of WinCC OA code
- Nightly quality assessment is triggered with Jenkins CI

Counting lines of code

LOC	Lines of code in the source files
SLOC	Lines of code excluding lines of comments or whitespace
SS	Executable source statements in the files
RFSS	Source statements remaining if code cloning is removed

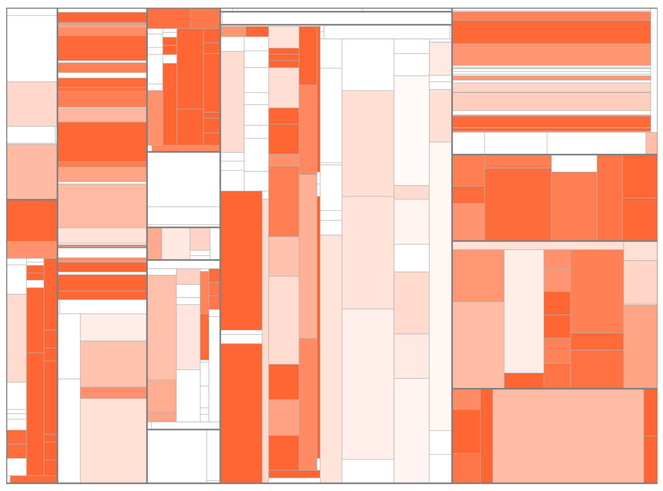
ConQAT in practice

- Code-comment ratio found not to be useful
- Thresholds for other metrics had to be defined

Metric	Threshold
Minimum clone length	15 lines
Maximum block length	60 lines
Maximum nesting depth	3 levels

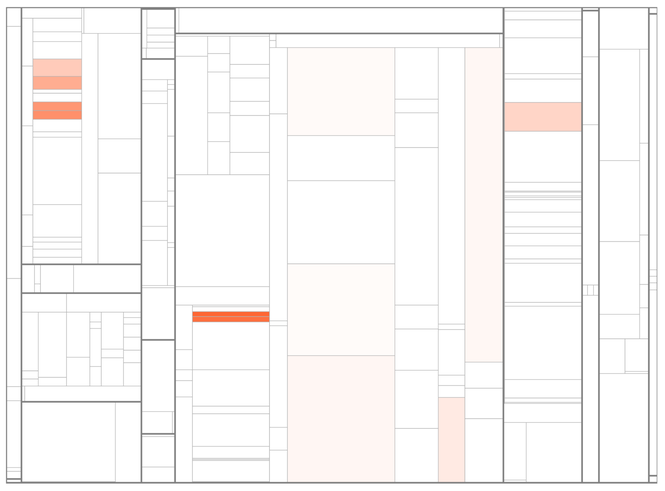
- Code cloning was chosen as the main focus
  - Clones increase code size
  - Clones make maintenance more difficult

Code Cloning – October 2011



Significant amount of cloning (marked in red)

Code Cloning – October 2013

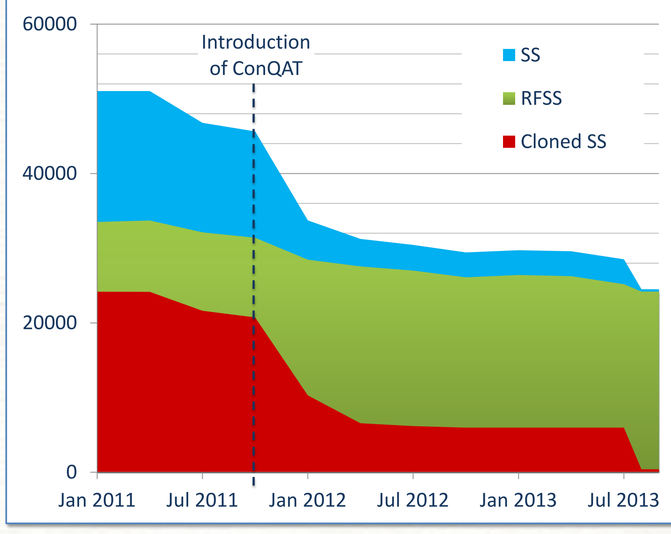


Fixing remaining issues would complicate code

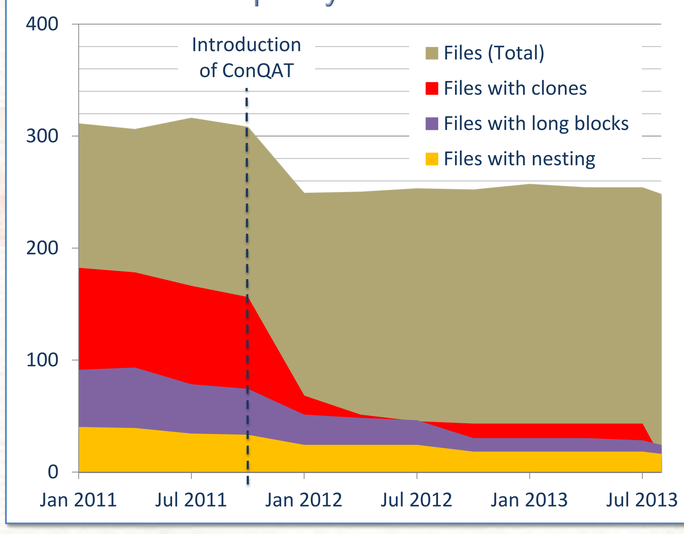
Results

- Cloned SS is minimized
- RFSS reduced due to refactoring
- Quality issues have been reduced
- New developments maintain high quality standards

Reduction of cloned code



Reduction of quality issues



ICALEPCS 2013

MOPPC088

Acknowledgements

Swiss National Science Foundation, Switzerland

1) ETH Zurich, Switzerland 2) University of Wisconsin-Madison, U.S.A.