Software Quality Control Group

Report to ATLAS Executive Board

CERN, 19 November 1999

Dario Barberis

Mandate

To evolve the software Quality Control requirements with the experience in ATLAS. Emphasis should be made on minimising overhead in effort, and good balance between documentation, coding, design and performance.

Existing structures in the detector systems and physics/performance organisation (combined performance groups) should be used for performance verification.

Mandate (cont.)

Requirements should initially be at a minimum to take into account that most collaborators are newcomers in C++/OO.

ATLAS aim to organise the software into work packages to which institutions shall commit themselves. These formal commitments will include the QC requirements.

QC Group composition

Makoto Asai, Hiroshima

Dario Barberis, CERN&Genoa, chairperson

Martine Bosman, Barcelona

Bob Jones, CERN

Jean-François Laporte, Saclay

Maya Stavrianakou, CERN

Software “Onion” Model

- kernel software in the middle
- outer layers made up by domain specific, detector specific then individual physicist’s software
- basically: the more people that share a piece of software the closer it is to the core
- software closer to the core (kernel) will have higher quality criteria to meet

Policy

Set long-term target:

- Good design → inspections/reviews
- Good code → testing procedures
- Good documentation

Relax some requirements on coding rules and design documentation for non-core software for the transition ("learning") period.

User guides mandatory!

Coding Conventions

- Take SPIDER as starting point
- Classify by importance
- Look for examples of good and bad code for each rule
- Define applicability as function of importance of rule and centrality of software
- Allow (documented) exceptions if good reasons

Software “Ownership”

- each package “belongs” to a working group (architecture, system, combined, physics)
- the WG reviews and approves software design and implementation
- the WG checks the software quality
- the QC group advises the WG’s and coordinates their work

Software Process

- In the long term: ask for design document, user guide, reference manual
- In the short term, for non-core software and for existing software: ask for user guide at least
- User guide must contain description of interfaces between the package and the outside world
- Validation by a combination of inspections, walk-throughs, reviews, tests organised by the relevant working groups