



# **SPI**

## **Software Process & Infrastructure for LCG**

---

## **Quality Assurance**

LCG Application Area Internal Review  
20-22 October 2003

Jakub T. Moscicki

# Contents

---

- Overview
- Goals, methods and tools of LCG QA
  - analysis/reporting tools
  - support tools for release process
- Practical example
- Activities and work done so far
- Outlook



# Overview of LCG QA

- The main goal of QA activity is to **help LCG projects assess and improve the quality of their software and procedures**
  - provide tools to collect useful **metrics/statistics** which help assess software quality;
  - generate **reports**;
  - verify if project setup is correct and compliant with **LCG Policies**.



# QA Tools

- Reporting tools:
  - `lcg-qa-project-report.py`
    - analyze project tree in AFS release area
    - time-based analysis (e.g. bugs reports)
    - --> generate HTML pages
  - first version ready and in use
    - announced in standard LCG environment very soon
- Release process tools:
  - e.g. include all open bug reports in the release notes automatically
  - under preparation



# The QA Focus

- **Tests/Bugs** are central for QA in our environment
  - vague/changing user requirements,
  - no “product specification” to adhere to
  - tools/procedures by agreement rather than by decision
  - sophisticated code metrics exist but these have much less importance for us -> **bug report vs test case tracability** has much more
- **LCG Policies**
  - agreed and defined by AF
  - SPI supports them in the tools and procedures and only helps to work them out



# QA Checklist

- Build the release
- Run automatic tests
- Statistics
  - Test Inventory
  - Documentation/Examples Inventory
  - Savannah Statistics
  - Code Inventory
  - Rule Checker , Logiscope
- LCG Policies
  - Configuration of a build system
  - CVS directory structure

```
scram project SEAL SEAL_1_1_0
cd SEAL_1_1_0
cvs -d :pserver:anoncvs@lcgapp.cern.ch:/cvs/SEAL co -r
SEAL_1_1_0 -d src seal
eval `scram runtime -sh`
qmtest -D ./src/config/qmtest run -o test_results.qmr
```

# QA Procedure

- Well-defined
  - clear rules and checklist of assessed items is available in advance to projects
- Transparent:
  - anybody at anytime may see project statistics and create reports themselves
- Open:
  - anybody may contribute
    - in terms of suggestions
    - in terms of tools



# Caveat

- SPI provides QA tools to spot the potential problems
- ...but SPI does not change the projects.
- Responsibility for software quality and compliance with LCG procedures/policies is within projects
- QA to be successful requires active collaboration from the projects.



# Activities: Apr-Sept 2003

- Manual / semi-automatic reports
  - POOL QA
    - QA reviews for 0.4.0, 1.0.0, 1.1.0, 1.2.0
  - SEAL QA
    - 0.3.1, 1.0.0
  - Main contribution from Massimo Lamanna
- Development / integration of automatic tools
  - SEAL\_1\_1\_0
  - tools about to be released / announced

# Activities: Apr-Sept 2003

- Evaluation of tools
  - Rule Checker
    - LCG Coding Rules vs existing activities (Atlas Rules)
  - Logiscope
    - Test coverage
  - SLOC
  - Valgrind
  - ignominy

# Outlook

---

- Activities for the end of 2003:
  - support bug/test tracability and release tools
  - investigate test coverage metrics
  - evaluate tools (logiscope, ignominy)
  
- Credits to contributors:
  - Massimo Lamanna (started the activity)
  - Ilka Antcheva (useful input and discussions)

# More information

---

<http://spi.cern.ch/qa>