LCG-SPI: SW-Testing
QMTest test framework

LCG AppArea meeting (16/07/03)

Manuel.Gallas@cern.ch

LCG/SPI

LCG Software Process & Infrastructure

Outline

SPI SW-Testing

- Test Frameworks
- QMTest
- User support

Test Frameworks

Aim: to help developers:
- to produce code for testing
- to run tests in automatic way

Requirements:

- Work with different languages (C++, Python, …)
- Allow two ways of testing: "*test the output of the code*" and "*test inside the testing-code*".
- Provide a easy way to integrate existing tests.
- Easy integration with the Nightly Building System.
- Regression testing.
- Allow the creation of dependencies among tests
- Organize the tests by components or packages.
- Provide also a graphical interface for run the tests and examine the test results.

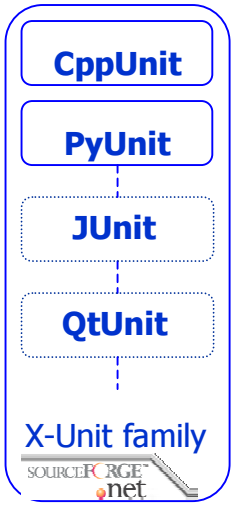
Our constrains:

- Avoid commercial software and licensing problems.
- Avoid "do it yourself solutions"
- Try to adopt commonly used open-source software.

Our inputs:

- Contacts within HEP-community.
- What is available as free open source code.

Test Frameworks: X-Unit family



A simple test:

1. Subclass the TestCase CppUnit or PyUnit class
2. Override the method runTest().
3. When you want to check a value, call the (CppUnit or PyUnit) **ASSERT**(bool) and pass in an expresion that is true if the test succeeds

```
xterm <6>

!!!FAILURES!!!
Test Results:
Run:  19   Failures: 2   Errors: 0


1) test: StringToolsTest::testToStringInt (F) line: 33 StringToolsTest.cpp
equality assertion failed
- Expected: 12345678
- Actual  : 123456789


2) test: ComplexNumberTest::testEquality (F) line: 22 ComplexNumberTest.cpp
assertion failed
- Expression: *m_10_1 == *m_1_1


PASS: test
==================
All 1 tests passed
==================
make[1]: Leaving directory `/home/user/mgallas/CppUnit-examples/newtest'
[pcitapi31] ~/CppUnit-examples/newtest >
```

- Scope: Unit –testing
- The same "assertion style" in different languages.
- Provides:- The name of the test case that failed.

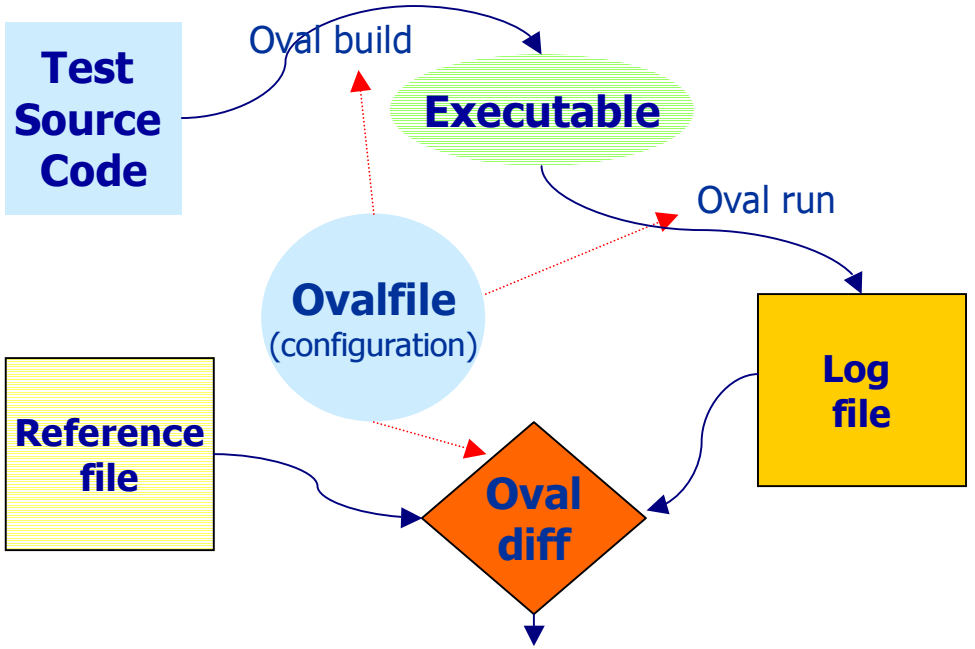
-The name of the source file that contains the test.

-The line number where the failure occurred

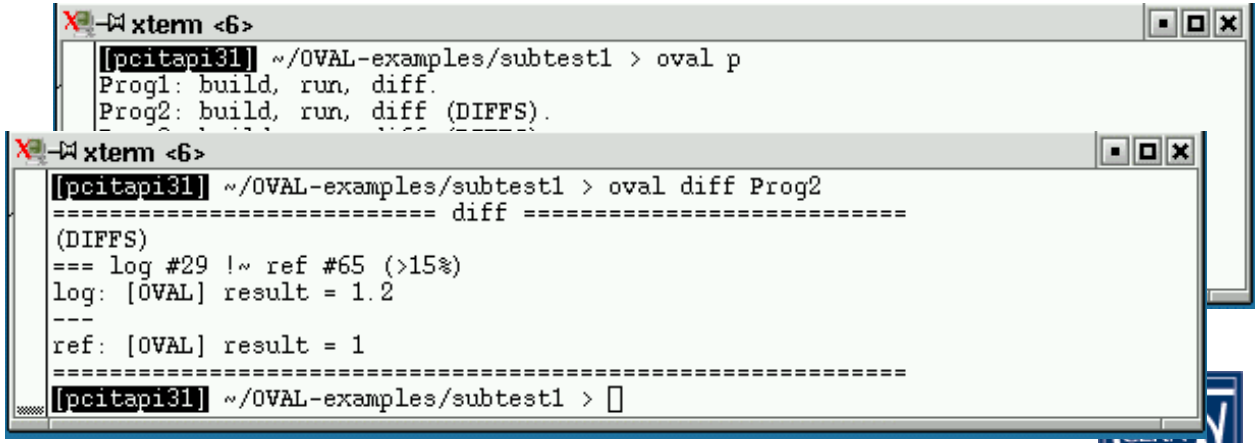
- Different platforms/compilers: (Linux/Solaris/Windows)

Test Frameworks: Oval

Oval:
- validation and regression
- used in CMS



- Can be use for Unit-test to Validation tests.
- Compare the output log file with a given reference file (Smart comparison of those lines which start with "[Oval]")
- It is possible to set different run environments.
- Can run external scripts and external binaries.

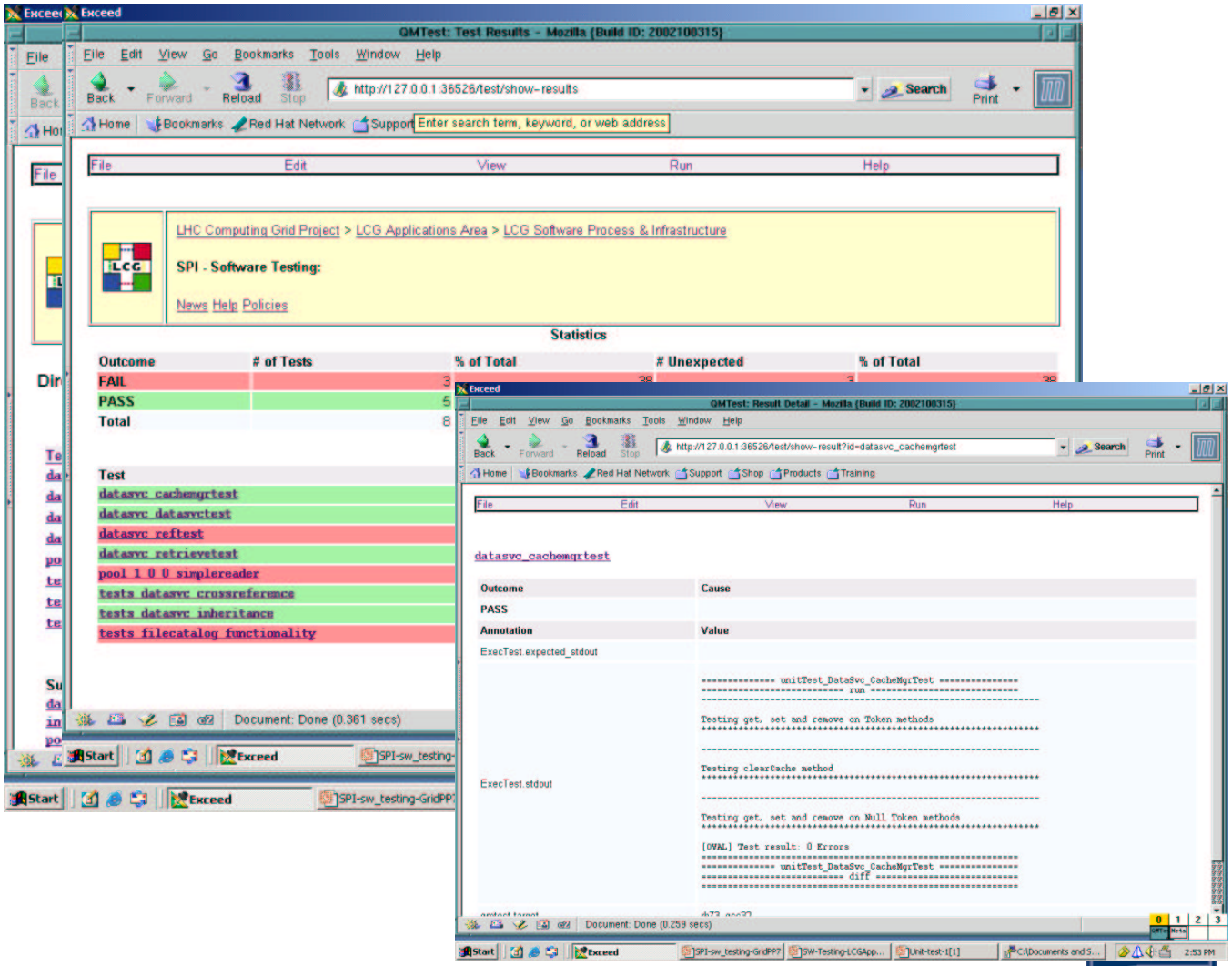


Test Frameworks: QMTest

QMTest

CodeSourcery

- Uses a graphical interface for creating and running tests
- Runs tests in parallel
- Organizes tests hierarchically
- Supports execution of a single test or many at once
- Records dependencies between tests
- Can be run in batch mode -> easy integration with the Nightly-Building systems



Test Frameworks: QMTest (LCG AA)

Use in LCG AppArea projects:

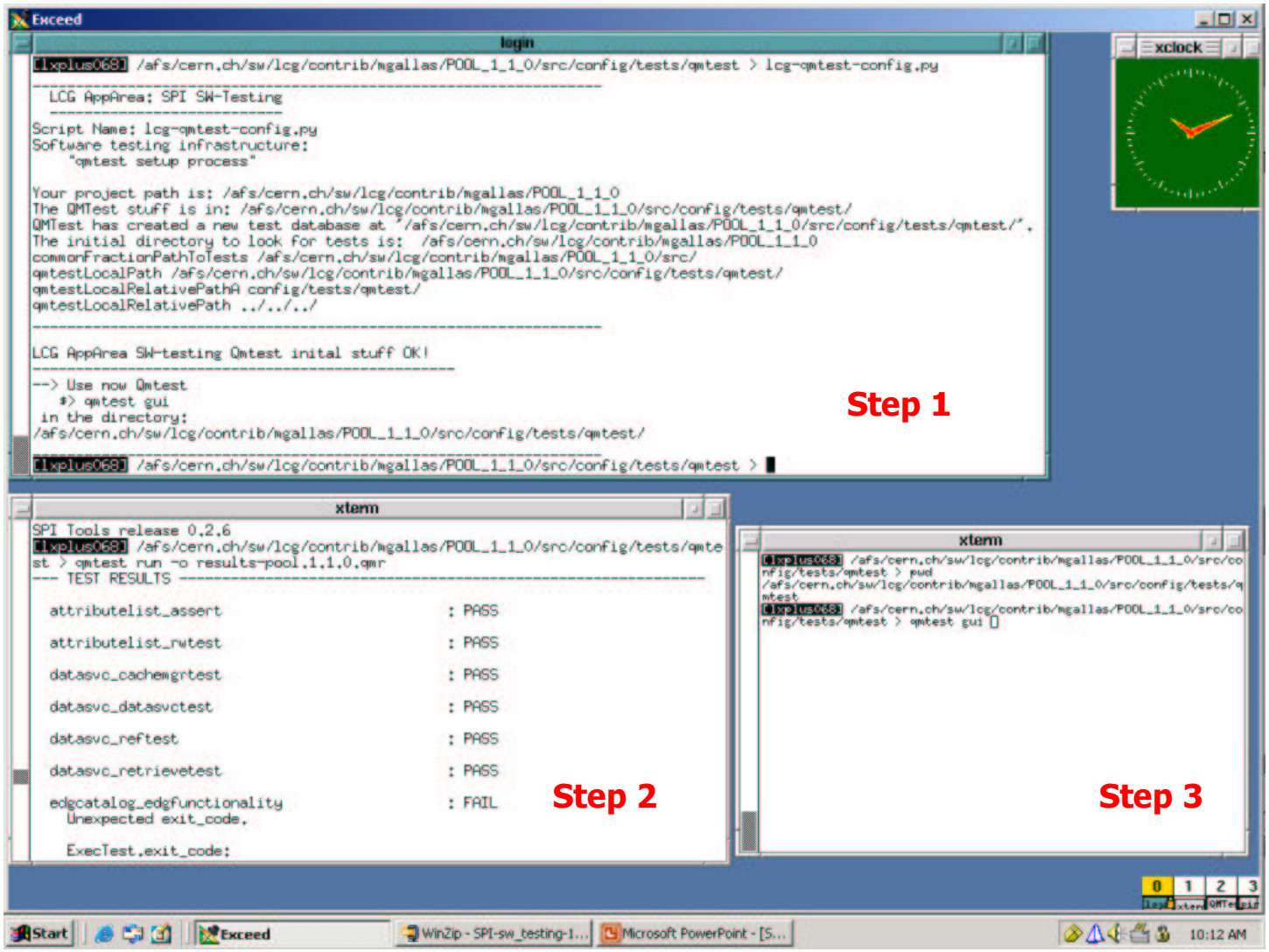
- All the QMTest configuration files will be created by a python script at once and later committed to CVS. Dependencies among tests, expected values and results could be also in CVS with the appropriate tag.
- These xml configuration files will be in → $PROJECT/src/config/tests/qmtest
- Only when a new test subdirectory is added is needed to re-run the script
- To create the xml files we use the "lcg-qmtest-config.py" script . It will scan the whole project looking for the OvalFiles. The configuration files will describe "test cases" and "test suites"
- In each project release QMTest should be run and the test results saved**:**
  $PROJECT/src/config/tests/qmtest> **qmtest run –o results_release_n_y_x.qmr**
- The test results can be browsed later using the graphical interface:
  $PROJECT/src/config/tests/qmtest> **qmtest gui**

QMTest (at LCG AppArea) requirements:

- Oval ≥ version 2_15_3
- CppUnit 1.8.0 used with the SPI test driver.
- PyUnit used with SPI test driver.
- stderr output in use only by Oval, CppUnit and PyUnit

QMTest installed in the SPI external sw-service
(Available in the SCRAM ToolBox ≥ LCG_10)

Test Frameworks: QMTest (demo I)



Test Frameworks: QMTest (demo II)

QMTEST xml configuration files:

Test suite

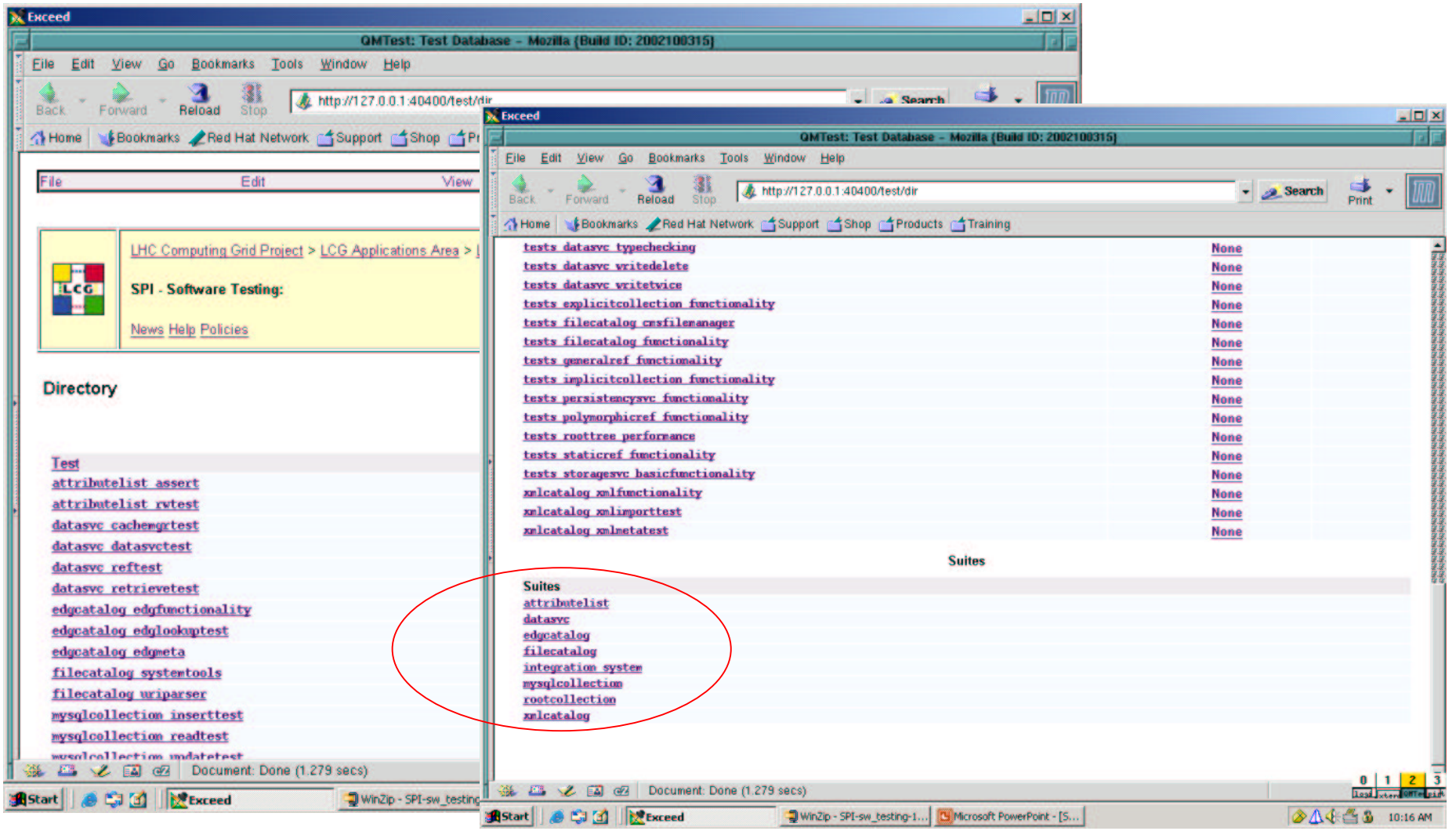
Test case

Results

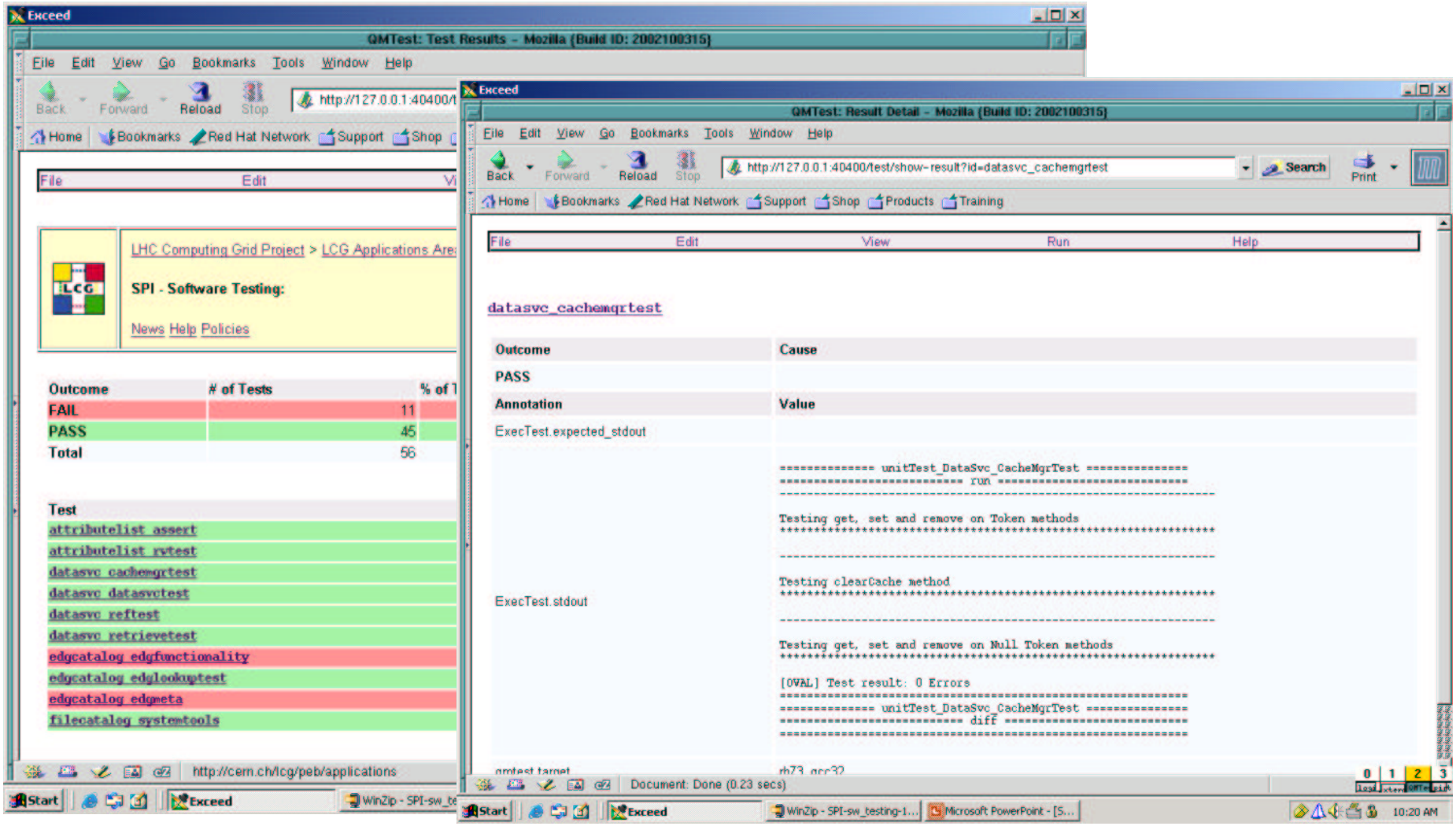
```
results-pool.1.1.0.qmr
[lxplus068] /afs/cern.ch/sw/lcg/contrib/mgallas/POOL_1_1_0/src/config/tests/qmtest > ls
QMTest                                  tests_collection_write.qmt
attributelist.qms                       tests_complexobjectstorage_functionality.qmt
attributelist_assert.qmt                tests_constref_functionality.qmt
attributelist_rwtest.qmt                tests_datasvc_appendread.qmt
datasvc.qms                             tests_datasvc_appendwrite.qmt
datasvc_cachemgrtest.qmt                tests_datasvc_crossreference.qmt
datasvc_datasvctest.qmt                 tests_datasvc_inheritance.qmt
datasvc_reftest.qmt                     tests_datasvc_objectstore.qmt
datasvc_retrievetest.qmt                tests_datasvc_ptrmember.qmt
edgcatalog.qms                          tests_datasvc_refownership.qmt
edgcatalog_edgfunctionality.qmt         tests_datasvc_readupdate.qmt
edgcatalog_edglookuptest.qmt            tests_datasvc_readwrite.qmt
edgcatalog_edgmeta.qmt                  tests_datasvc_simpleappend.qmt
filecatalog.qms                         tests_datasvc_typechecking.qmt
filecatalog_systemtools.qmt             tests_datasvc_writedelete.qmt
filecatalog_uriparser.qmt               tests_datasvc_writetwice.qmt
integration_system.qms                  tests_explicitcollection_functionality.qmt
mysqlcollection.qms                     tests_filecatalog_cmsfilemanager.qmt
mysqlcollection_inserttest.qmt          tests_filecatalog_functionality.qmt
mysqlcollection_readtest.qmt            tests_generalref_functionality.qmt
mysqlcollection_updatetest.qmt          tests_implicitcollection_functionality.qmt
results-pool.1.1.0.qmr                  tests_persistencysvc_functionality.qmt
rootcollection.qms                      tests_polymorphicref_functionality.qmt
rootcollection_inserttest.qmt           tests_roottree_performance.qmt
rootcollection_readtest.qmt             tests_staticref_functionality.qmt
rootcollection_updatetest.qmt           tests_storagesvc_basicfunctionality.qmt
tests_collection_multifileupdate.qmt    xmlcatalog.qms
tests_collection_multifilewrite.qmt     xmlcatalog_xmlfunctionality.qmt
tests_collection_read.qmt               xmlcatalog_xmlimporttest.qmt
tests_collection_update.qmt             xmlcatalog_xmlmetatest.qmt
[lxplus068] /afs/cern.ch/sw/lcg/contrib/mgallas/POOL_1_1_0/src/config/tests/qmtest >
```

lcg-qmtestconfig.py will produce these files

Test Frameworks: QMTest (demo III)



Test Frameworks: QMTest (demo IV)



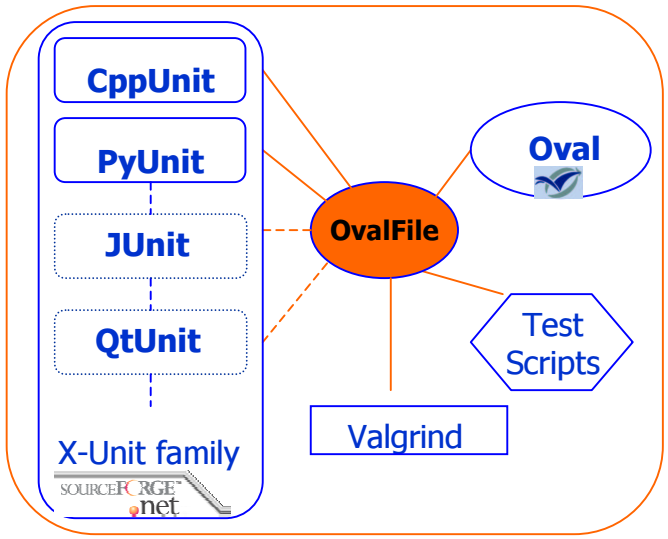
Test Frameworks: Global picture

Aim: to help developers:
- to produce code for testing
- to run tests in automatic way

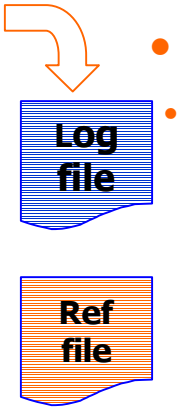
QMTest CodeSourcery

- Uses a graphical interface for creating and running tests
- Runs tests in parallel
- Organizes tests hierarchically
- Supports execution of a single test or many at once
- Records dependencies between tests

In LCG AppArea runs what is in the OvalFile and uses oval.



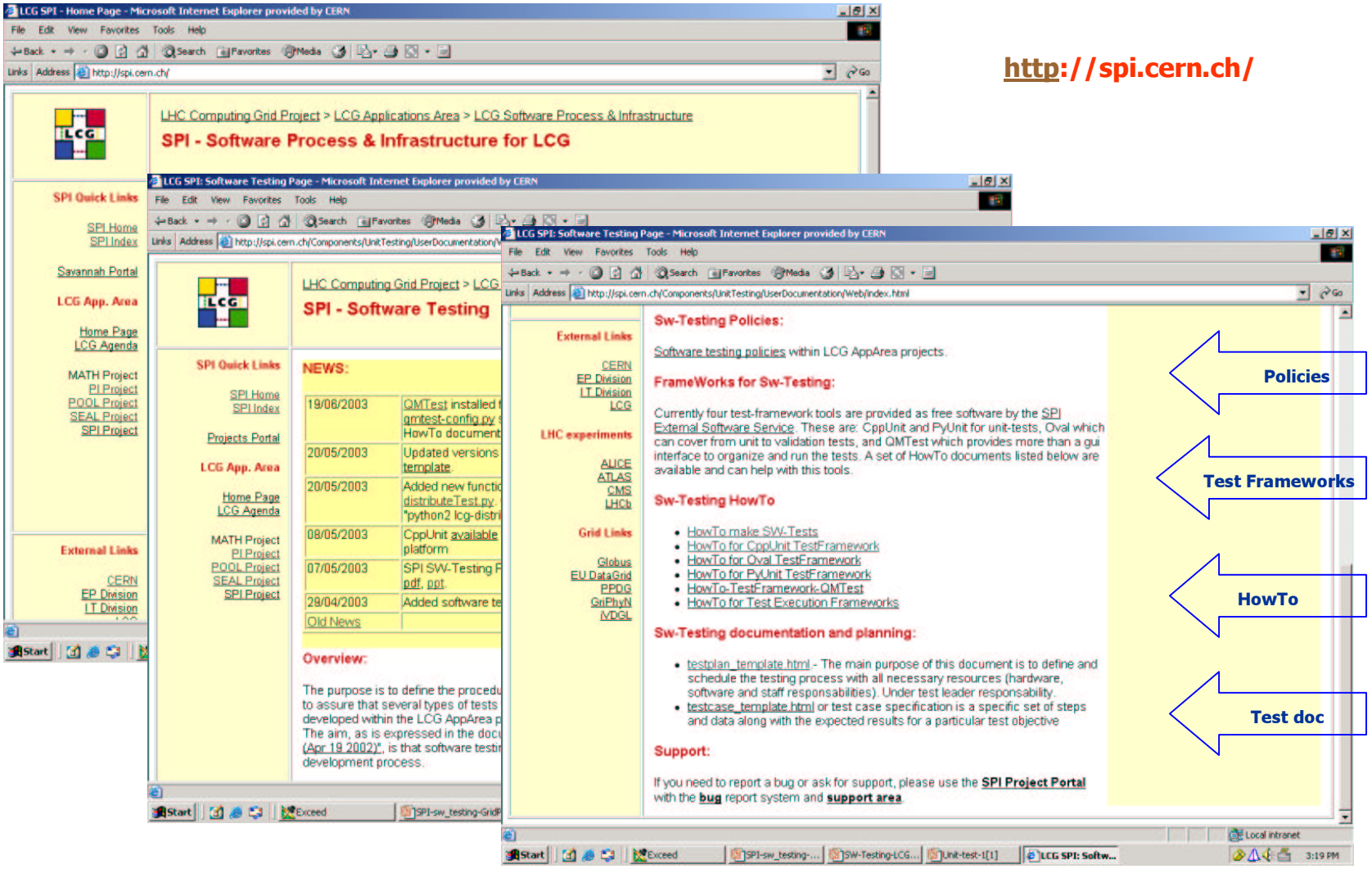
Oval



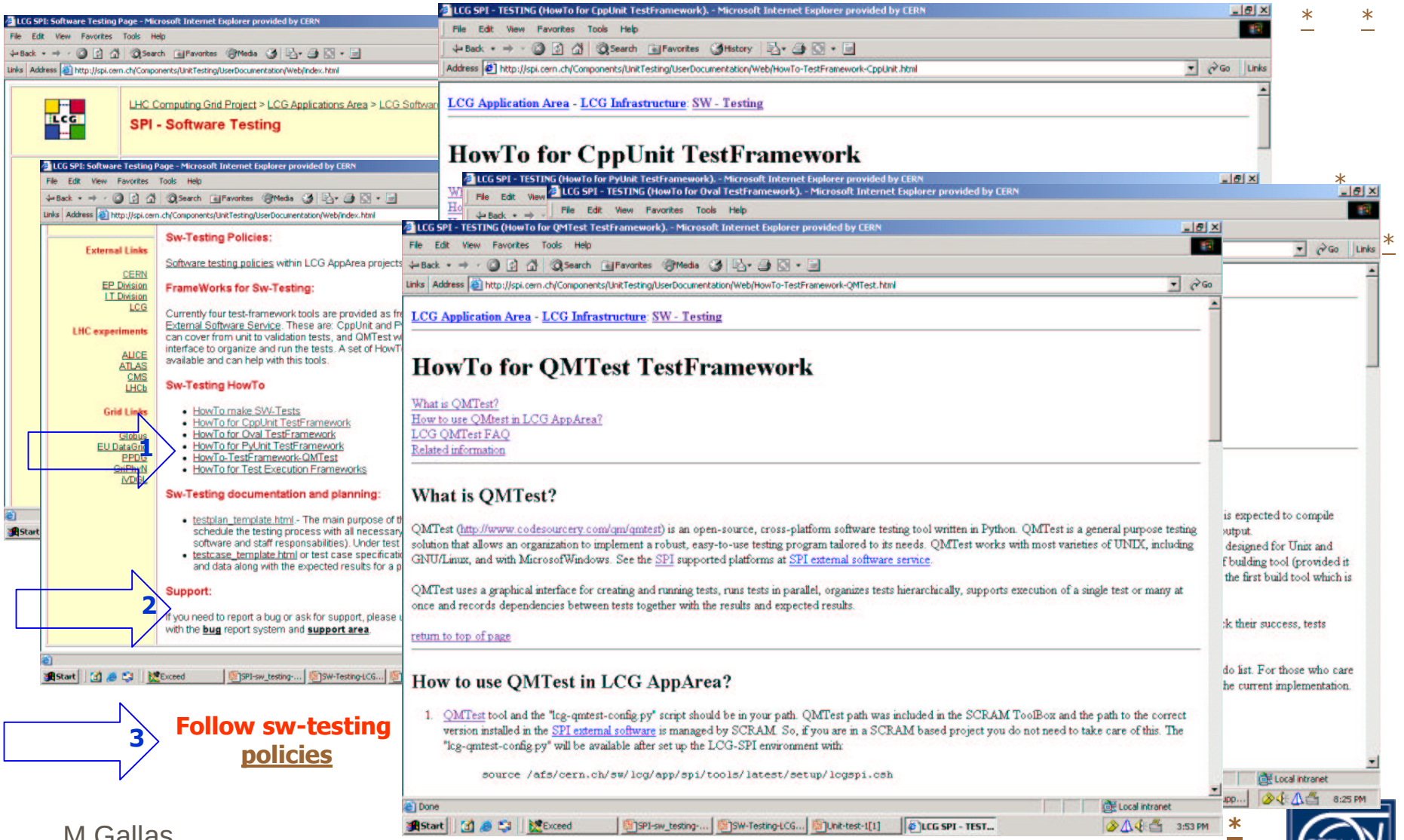
- **Automated tests**
- **Regression test**

User support: Web information

http://spi.cern.ch/



User support: How-To



SPI SW-Testing

- Test Frameworks
- User support
- Test policies
- Test documents

!!!

Thanks to:

- LCG-POOL team
- LCG-SEAL team

Feedback and interaction are always welcome!!