EUROPEAN MIDDLEWARE INITIATIVE

DJRA1.7.1 - SOFTWARE DEVELOPMENT QUALITY CONTROL REPORT

EU DELIVERABLE: D5.7.1

Document identifier:	**EMI-DJRA1.7.1-1277533-Quality_Control_Report-v1.0.doc**
Date:	**31/07/2010**
Activity:	**JRA1**
Lead Partner:	**INFN**
Document status:	**Final**
Document link:	**http://cdsweb.cern.ch/record/1277533?ln=en**

Abstract:
This document describes the status and performance of the JRA1 quality control task with details on the availability and execution of unit, functional and compliance test for EMI components.

Delivery Slip

	Name	Partner / Activity	Date	Signature
From	Andrea Ceccanti	INFN/JRA1	19/10/2010	
Reviewed by	Francesco Giacomini Maria.Alandes Pradillo Jozefs Cernak	INFN/SA1 CERN/SA2 UPJS/SA2	04/11/2010	
Approved by	PEB		24/11/2010	

Document Log

Issue	Date	Comment	Author / Partner
0.1	01/07/10	Draft TOC	Andrea / INFN
0.2	06/09/10	Draft for internal review	Andrea/ INFN
0.3	10/10/10	Integrated feedback from internal review and updated survey results	Andrea / INFN
0.4	18/10/10	Final modifications and changes	Andrea / INFN & Morris / Juelich
0.5	18/11/10	Integrated reviewers comments	Andrea / INFN

Document Change Record

Issue	Item	Reason for Change

TABLE OF CONTENTS

1. INTRODUCTION

1.1. PURPOSE

The main purpose of the periodic JRA1 Quality Control reports, as mandated by the Software Quality Assurance Plan [R 1], is to summarize the status and performance of quality control pertaining the EMI software development activities. In particular, the reports should sum up the results of the EMI software components reviews and provide details about the availability and execution of unit, functional and compliance tests for the EMI components.

Given that no new developments will be released during the first year of the EMI project, this report focuses on the description of the JRA1 Quality Control activity and its relationships with other JRA1, SA1 and SA2 tasks.

The results of a test availability survey circulated during the first months of the project are also presented.

1.2. DOCUMENT ORGANISATION

The rest of this document is organized as follows. Section 2 is the executive summary. Section 3 describes the Quality Control Activity in JRA1. Section 4 provides the results of the test availability survey circulated in the first months of the project. Section 5 concludes the report.

1.3. APPLICATION AREA

This document reports about quality control activities in the EMI JRA1 work package and affects the JRA1, SA1 and SA2 activities.

1.4. REFERENCES

R 1	DSA2.1 EMI Software Quality Assurance Plan https://twiki.cern.ch/twiki/pub/EMI/DeliverableDSA21/EMI-DSA2.1-1277599-QA_Plan-v1.2.pdf
R 2	SA2 Certification and testing guidelines https://twiki.cern.ch/twiki/bin/view/EMI/EmiSa2CertTestGuidelines
R 3	SA2 Packaging guidelines https://twiki.cern.ch/twiki/bin/view/EMI/EmiSa2PackagingReleasingGuidelines
R 4	SA2 Change management guidelines https://twiki.cern.ch/twiki/bin/view/EMI/EmiSa2ChangeManagementGuidelines
R 5	SA2 Certification and testing guidelines https://twiki.cern.ch/twiki/bin/view/EMI/EmiSa2CertTestGuidelines
R 6	DSA2.3 – KPI and metrics definition document https://twiki.cern.ch/twiki/bin/view/EMI/EmiSa2MetricsGuidelines
R 7	DSA2.2.1 - QA Tools Documentation https://twiki.cern.ch/twiki/bin/edit/EMI/DeliverableDSA221
R 8	DSA2.4 – Continuous integration and certification testbeds https://twiki.cern.ch/twiki/bin/view/EMI/DeliverableDSA24
R 9	DSA2.3.1 – Periodic QA report https://twiki.cern.ch/twiki/bin/view/EMI/DeliverableDSA231

R 10	DJRA1.5 - Standardization Work Plan and Status Report https://twiki.cern.ch/twiki/bin/view/EMI/DeliverableDJRA151
R 11	DJRA1.6 – Integration Work Plan and Status Report https://twiki.cern.ch/twiki/bin/view/EMI/DeliverableDJRA161
R 12	DNA1.3.1 – Technical Development Plan https://twiki.cern.ch/twiki/bin/view/EMI/DeliverableDNA131
R 13	JUnit - http://junit.org/
R 14	DejaGNU - http://www.gnu.org/software/dejagnu/
R 15	CPPUnit - http://sourceforge.net/projects/cppunit/
R 16	EMI Description of Work https://twiki.cern.ch/twiki/pub/EMI/EmiDocuments/EMI-Part_B_20100624-PUBLIC.pdf
R 17	EMI JRA1 Development and test plans https://twiki.cern.ch/twiki/bin/view/EMI/EmiJra1T1Coord
R 18	EGEE III SA3 Testing https://twiki.cern.ch/twiki/bin/view/EGEE/SA3Testing
R 19	EMI Test availability survey https://twiki.cern.ch/twiki/bin/view/EMI/EmiTestAvailabilitySurvey
R 20	S2 SRMv2 Compliance Test Suite http://s-2.sourceforge.net/

1.5. DOCUMENT AMENDMENT PROCEDURE

Amendments, comments and suggestions should be sent to the author and/or to the emi-jra1-quality@eu-emi.eu mailing list.

This document can be amended by the authors further to any feedback from other teams or people. Minor changes, such as spelling corrections, content formatting or minor text re-organisation not affecting the content and meaning of the document can be applied by the authors without peer review. Other changes must be submitted to peer review and to the EMI PEB for approval.

When the document is modified for any reason, its version number shall be incremented accordingly. The document version number shall follow the standard EMI conventions for document versioning.

1.6. TERMINOLOGY

APEL	Accounting Processor for Event Logs component
ARC	Advanced Resource Connector
Argus	The Argus Authorization service
BDII	Berkeley Database Information Index

BES	Basic Execution Service
CREAM	The Computing Resource Execution and Management Cream Service
JSDL	Job Submission and Description Language
PT	Product Team
QA	Quality Assurance
QC	Quality Control
SQAP	Software Quality Assurance Process
StoRM	Storage Resource Manager
VOMS	Virtual Organization Membership Service
VOMS Admin	Virtual Organization Membership Administration Service
WMS	Workload Management System

2. EXECUTIVE SUMMARY

This document is the first report on the status of the Quality Control (QC) activity in the context of the EMI JRA1 work package.

As stated in the EMI DoW [R 14], one of the JRA1 objectives is to "adopt standard software quality assurance and quality control procedures and perform them transparently to increase the confidence of the users in the software, but also the possibility of the middleware to be used or supported by commercial service providers and application developers".

The Quality Control (QC) task in JRA1 is responsible to monitor and assess the quality of middleware components developed by EMI product teams. In particular, QC has to ensure that all components developed by JRA1 satisfy certification and validation criteria established by SA2 [R 1,R 2,R 3,R 5] before they are handed over to SA1 for inclusion in EMI releases.

QC is part of the JRA1 developers' daily tasks whose objective is improving the developed software quality. This is achieved by ensuring compliance with selected code quality metrics, by enforcing that documentation respects the guidelines defined for the project and by increasingly testing the existing software, either in isolation or by checking the integration and interoperability with other EMI components. This work will be done in close collaboration with the JRA1.7 Integration and Interoperability task leveraging the testbed provided by SA2 [R 8].

Besides a description of the JRA QC activity, this report presents the results of a test availability survey [R 19] circulated among the EMI Product teams (PTs) during the first months of the project. The survey objective is to give a snapshot, at the beginning of the project, of the availability of unit, functional, regression and standard compliance tests in the existing EMI components.

The test results [R 19] show that there is great room for improvement regarding the testing of the EMI software components. The objective of the project should be to significantly improve the coverage, documentation and automation of the existing test suites so that higher software quality could be achieved.

3. QUALITY CONTROL IN JRA1

The Quality Control (QC) task in JRA1 is responsible to monitor and assess the quality of middleware components developed by EMI product teams. Quality factors, procedures and metrics are defined in the Software Quality Assurance Plan (SQAP) document [R 1] and in a set of satellite guidelines produced by SA2 [R 2,R 3,R 4,R 5].

QC is an activity that pertains to all JRA1 members and is performed as part of the daily development activities of each Product Team under the supervision of PT leaders. This work is coordinated by the JRA1 QC task leader with the support of SA2.

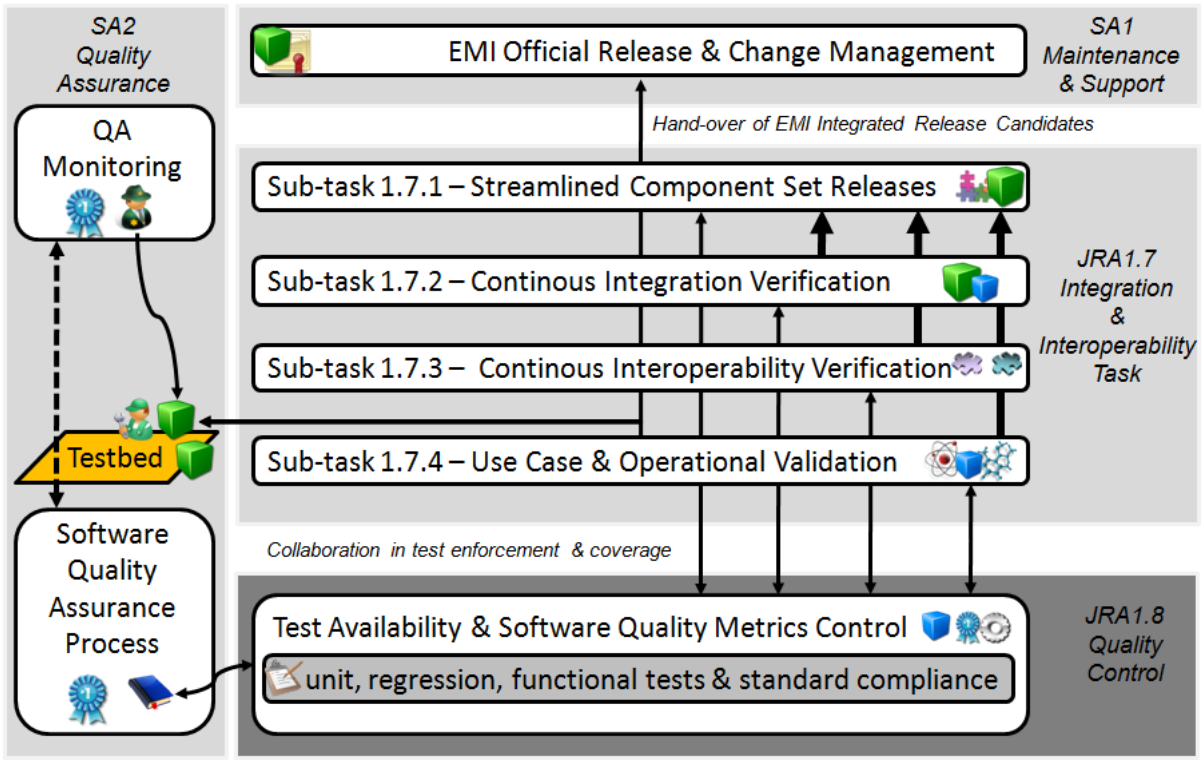

Figure 1: QC tasks in relation to other project tasks

Figure 1 shows the relationships of the JRA1 QC task with other tasks within the EMI project. QC's main responsibility is to assess that PTs are following the SQAP process defined by SA2. The quality metrics will be monitored by means of automated procedures and checks leveraging the tools provided by SA2 itself [R 7]. Any deviations from the quality objectives defined in the process will be reported by the QC task leader to the appropriate project boards so that corrective actions may be taken. JRA1.8 will closely collaborate with the JRA1.7 in the definition and monitoring of integration and interoperability test plans to ensure that high quality integrated components are handed-over to SA1 for inclusion in EMI releases. QC will also monitor the evolution of unit, regression, functional and standard compliance test availability across the middleware components so that objectives defined by SA2, which are not yet finalized, at the times of this writing, are met. The status and performance of these QC activities will be summarized yearly in periodic JRA1 Software Development QC reports.

The rest of this section describes the main QC activities as currently defined by SA2 and understood by the JRA1.8 task leader.

3.1. CODE QUALITY CHECKS

Any new code developed by JRA1 should be checked against the quality metrics defined by SA2.3 [R 6], so that project quality thresholds are met. In order to check quality metrics compliance developers will use the QA tools provided by SA2 [R 7].

3.2. DOCUMENTATION CHECKS

The SQAP defines the set of minimum required documentation for each software component developed by the EMI project [R 1]. Developers will have to check that existing components provide such documents and comply with the guidelines defined by SA2. The status of documentation will be periodically summarized in SA2 Quality Assurance reports [R 9].

3.3. COMPONENT TESTING

JRA1 is responsible for the development of the unit and functional tests required to validate the code correct behaviour, the components interfaces and the compliance with the adopted standards (where applicable).

PTs are expected to evolve and improve their software components test suites in order to meet the quality metrics defined by SA2. In particular, PTs will have to:

- provide a test plan document that complies with the guidelines defined by SA2 [R 2] (and part of the minimum required documentation cited in section 10).
- improve unit test coverage so that the thresholds defined by SA2.3 are met [R 6].
- augment functional test suite so that each component functionality is covered; if possible, measure the coverage
- integrate the test suites in the EMI continuous build and integration process so that QA tools provided by SA2 can be used to monitor the execution and coverage of the testing
- when applicable, consider using a standard compliance test suite to assess the compliance of your software with respect to the standards listed in [R 15]. This activity should be carried out under the supervision of the JRA1.6 task leader.
- test the interaction and interoperability of software components with other related EMI components, following the integration and interoperability matrixes provided by JRA1.6. This task should be part of the standard release certification activities and should be carried out leveraging the EMI distributed testbed [R 8].

The JRA1 development and test plan [R 17] considers the above requirements where appropriate. EMI technical objectives, as defined by the EMI technical development plan [R 12], are broken down in high level development and testing strategies. It is then up to each involved PT to refine these strategies into concrete tests to be included in each component test plan.

3.4. SOFTWARE COMPONENTS REVIEW

The SQAP states that all the software components should be continuously monitored to verify that the following requirements are met:

- Ensure that software components comply with the required code metrics defined by SA2 [R 6]
- Ensure that software components build successfully on the EMI supported platforms
- Ensure that software components artifacts meet naming and packaging conventions defined for the project [R 3]
- Ensure that software components successfully pass all the tests defined in their test plan
- Ensure that tickets assigned to each PT regarding development issues are being dealt with
- When applicable, ensure that security vulnerabilities have been addressed

The above checks will be done on a weekly basis by the QC task leader leveraging the build system and QA tools provided by SA2 [R 7]. Anomalies and deviations from the SQAP will be reported directly to PTs or to the appropriate boards so that corrective actions may be taken.

4. TEST AVAILABILITY SURVEY

The following types of tests have been identified as being relevant for the JRA1 QC activity:

- **Unit tests**, which are meant to test the correctness of individual units or a group of related units in a piece of software. A unit is defined as the smallest testable part of an application [R 5];
- **Regression tests,** which are meant to verify specific bug fixes and to assure that modifications to the software do not reintroduce previously fixed bugs;
- **Functional system tests,** which test the compliance of a component with specific functional requirements;
- **Standards compliance tests,** which test the compliance of a component with a specific adopted standard.

A survey [R 19] on the availability of unit, functional, regression and standard compliance tests has been circulated to the PTs. The survey objective is to understand the current status of testing in existing middleware stacks in order to develop strategies to improve quality during the course of the project. For instance, the results can be a valuable input for for SA2 to define realistic project-wide testing guidelines and quality tresholds.

The results gathered so far are summarized in the next sections. A spreadsheet with the full results can be found at the survey project home page [R 19].

4.1. TEST PLAN AVAILABILITY

As of today, ARC and UNICORE surveyed components do not provide a test plan document that describes the testing of their software. Nevertheless, these components perform tests on a regularly basis and are just not documented in a central place.

Many gLite services, on the other hand, provide test plan documents describing the functional system test suites that are typically used to verify the service behavior during certification. More details on the test plan availability for gLite components can be found in [R 18].

dCache has a test plan document that describes the testing of the software but the document is not part of the published documentation.

4.2. UNIT TESTS AVAILABILITY

Most UNICORE components provide an unified test suite that comprises unit, regression and functional test suite. The technology used is JUnit. The testing is automated and performed during the software build process.

On the other hand, most ARC and gLite components do not provide unit testing or, when they do, the code coverage of these tests is not measured. More details about individuals components follows.

Some ARC client components (ARC WS client, ARC Pre-WS client, libarclient, libarcdata) provide Unit Tests, in particular client components. The technology used is CPPUnit, and testing is performed during the software build process.

VOMS provides unit tests for its API components. DejaGNU is used as testing technology. This testing is part of the automated VOMS subsystem builds. VOMS-Admin provides some unit tests implemented in JUnit but those cannot be considered a proper test suite and coverage is not measured.

The gLite Job management product team does not have a unit test suite.

The gLite Information System has a unit test suite but do not provide it with the software. Unit tests are written in Bash and are run at development, integration and certification time. Coverage is not measured.

APEL provides some unit tests that are part of the automated build process. Coverage is currently not measured. The technology used is JUnit.

StorM provides some unit tests written in JUnit that are not part of the automated build process but are run during development. Coverage is not measured.

Logging & Bookkeeping has a unit test suite integrated in the build process. CPPUnit is used as technology. Coverage is not measured.

dCache provides a JUnit test suite integrated in the software build process. Table 3 shows the unit test coverage of components that currently provide this metric.

Product Team	Component	Unit test average coverage
UNICORE Security	XUUDB	13,7%
UNICORE WS interfaces	UNICORE BES	16,0%
dCache	dCache Client	20,0%
dCache	dCache Server	20,0%
ARC Data Libraries	libarcdata2	21,0%
ARC Compute Clients	libarcclient	25,0%
UNICORE Security	UVOS	29,3%
UNICORE Client and APIs	UNICORE Client Libs	30,0%
UNICORE Client and APIs	UCC	31,0%
UNICORE Security	UNICORE uas-authz	38,0%
UNICORE Security	UNICORE Gateway	43,0%
UNICORE Client and APIs	HiLA	49,0%
gLite Security	Trustmanager	50,0%
gLite Security	Util-Java	50,0%
UNICORE WS interfaces	UNICORE Registry	51,0%

UNICORE WS interfaces	UAS	57,0%
UNICORE Target System	XNJS	58,0%
UNICORE Security	UNICORE security libraries	58,6%
UNICORE Container	WSRF/Lite	60,0%
VOMS	VOMS server + APIs	60,0%
UNICORE Container	UNICORE/X	66,0%
UNICORE Security	UNICORE XACML Entity	71,0%

Table 3: EMI Unit test coverage

4.3. REGRESSION AND FUNCTIONAL TESTS AVAILABILITY

UNICORE components test suite does not distinguish among regression, unit and functional test, so the availability and coverage presented in section 12 apply also for functional testing.

ARC components do not provide a regression test suite with the notable exception of libarcdata that provides some regression testing using CPPUnit.

Many gLite components provide regression and functional test suites that are typically ran during release certification to verify bug fixes and to assess the components' correct behavior.

The gLite information system provides a unit test regression suite that is typically run during development. A policy states that code is not checked in unless the tests are successfully run.

VOMS is the only component (according to this survey) that measures the functional test coverage of its code.

VOMS-Admin client and server functionality is tested using a bash script test suite ran manually at certification time. Coverage is not measured.

Argus provides a functional test suite that can be automated and executed on the ETICS virtual testbed. Coverage is not measured. A regression test suite is also provided to check bug fixes during release certification.

StorM has a functional test suite that leverages existing SRM clients as well as python and bash scripts to validate the software behavior. The tests are run manually during release certification procedures.

The Logging & Bookkeeping provides a functional and regression test suite that is run regularly during certification. Coverage is not measured.

dCache provides a JUnit regression test suite that covers most of the know bug fixes. The regression test suite is ran at build and integration time. A python functional test suite is also ran by their continuous integration system. Functional test coverage is not measured.

4.4. STANDARDS COMPLIANCE TEST AVAILABILITY

The Logging & Bookkeeping PT test their software against an IPV6 compliance test suite. The testing is seldom run manually.

The gLite Information system PT tests against LDAP and GLUE standards by running insertions in an OpenLDAP database and GStat validation probes.

dCache tests their SRM implementation against the S2 SRMv2 compliance test suite [R 20].

All other PTs do not test their software against standards compliance test suites.

4.5. REQUEST FOR GUIDELINES

Almost all the PTs queried by this survey agree that the EMI project should provide clear guidelines on how the software testing process should be implemented. Besides functional standalone testing practices for services in isolation, EMI should provide an integrated testbed and clear process on how to implement Integration testing among services, in order to evaluate the full distributed chain and assess interoperability among services from different middleware stacks.

4.6. ANALYSIS OF THE SURVEY RESULTS

Documentation of the existing test suites must be improved as many components do not provide a test plan describing their test suite.

Unit test coverage must also be improved. While it could be stated that unit tests are not really needed for established production services that proved to be reliable in production for several years, newly developed code should be thoroughly unit tested yielding satisfactory coverage figures (according to the SA2 definition of the coverage metric).

Most components provide a regression and functional test suite that is used typically at release certification time. Most of these test suites, however, are not included in the automated build or integration process but are typically run manually by developers/certifiers when a new release is assembled. The integration of these test suites in the continuous build and integration process is crucial to improve the quality of the developed software.

Finally, few PTs test their software against standard compliance test suites or assess the integration and interoperability with other components in a systematic way. One of the objectives of this task, with the support of the JRA1.6 and JRA1.7 activities, is to foster and monitor the adoption of such test suites by the PTs so that integration, interoperability and standard compliance could be systematically assessed leveraging EMI QA tools.

5. CONCLUSIONS

QC is an activity that pertains to all JRA1 members and is performed as part of the daily development activities of each PT. The objective of QC is to ensure that the quality process defined by SA2 [R 1] is followed in order to produce high-quality components ready to be included in EMI releases.

This report focuses on the description of the JRA1 QC activity and its relationships with other JRA1, SA2 and SA1 activities. The results of a test availability survey circulated at the beginning of the project are also presented. These initial results give a snapshot of the current status of testing in existing middleware stacks and provide a valuable input for defining guidelines and testing strategies to be used in the EMI project.