

# EUROPEAN MIDDLEWARE INITIATIVE

## SOFTWARE DEVELOPMENT QUALITY CONTROL REPORT

### EU DELIVERABLE: DJRA1.7.1

---

Document identifier:	<b>DJRA1.7.1_Software_Development_Quality_Control Report.odt</b>
Date:	<b>10. Oct. 2010</b>
Activity:	<b>JRA1.8</b>
Lead Partner:	<b>INFN</b>
Document status:	<b>DRAFT</b>
Document link:	

---

**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.

**Copyright notice:**

Copyright © Members of the EMI Collaboration, 2010.

See [www.eu-emi.eu](http://www.eu-emi.eu) for details on the copyright holders.

EMI (“European Middleware Initiative”) is a project co-funded by the European Commission as an Integrated Infrastructure Initiative within the 7th Framework Programme. EMI began in May 2010 and will run for 36 months.

For more information on EMI, its partners and contributors please see [www.eu-emi.eu](http://www.eu-emi.eu)

You are permitted to copy and distribute, for non-profit purposes, verbatim copies of this document containing this copyright notice. This includes the right to copy this document in whole or in part, but without modification, into other documents if you attach the following reference to the copied elements: “Copyright © Members of the EMI Collaboration 2010. See [www.eu-emi.eu](http://www.eu-emi.eu) for details”.

Using this document in a way and/or for purposes not foreseen in the paragraph above requires the prior written permission of the copyright holders.

The information contained in this document represents the views of the copyright holders as of the date such views are published.

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED BY THE COPYRIGHT HOLDERS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE MEMBERS OF THE EGI\_DS COLLABORATION, INCLUDING THE COPYRIGHT HOLDERS, OR THE EUROPEAN COMMISSION BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THE INFORMATION CONTAINED IN THIS DOCUMENT, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.



## TABLE OF CONTENTS

<b>1. INTRODUCTION.....</b>	<b>5</b>
1.1. PURPOSE.....	5
1.2. DOCUMENT ORGANISATION.....	5
1.3. APPLICATION AREA.....	5
1.4. REFERENCES.....	5
1.5. DOCUMENT AMENDMENT PROCEDURE.....	6
1.6. TERMINOLOGY.....	6
<b>2. EXECUTIVE SUMMARY.....</b>	<b>7</b>
<b>3. QUALITY CONTROL IN JRA1.....</b>	<b>8</b>
3.1. CODE QUALITY CHECKS .....	8
3.2. DOCUMENTATION CHECKS.....	8
3.3. TESTING.....	8
3.4. SOFTWARE COMPONENTS REVIEW.....	9
<b>4. TEST AVAILABILITY SURVEY.....</b>	<b>10</b>
4.1. TEST PLAN AVAILABILITY.....	10
4.2. UNIT TESTS AVAILABILITY.....	10
4.3. REGRESSION AND FUNCTIONAL TESTS AVAILABILITY.....	11
4.4. STANDARDS COMPLIANCE TEST AVAILABILITY.....	12
4.5. REQUEST FOR GUIDELINES.....	12
4.6. ANALYSIS OF THE SURVEY RESULTS.....	12
<b>5. CONCLUSIONS.....</b>	<b>14</b>

# 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 and SA2 tasks and deliverables. 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 report 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 6 gives some recommendations regarding quality in JRA1. Section 7 concludes the report.

## 1.3. APPLICATION AREA

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

## 1.4. REFERENCES

**Table 1: Table of References**

<b>R 1</b>	DSA2.1 EMI Software Quality Assurance Plan <a href="https://twiki.cern.ch/twiki/pub/EMI/DeliverableDSA21/EMI-DSA2.1-1277599-QA_Plan-v1.2.pdf">https://twiki.cern.ch/twiki/pub/EMI/DeliverableDSA21/EMI-DSA2.1-1277599-QA_Plan-v1.2.pdf</a>
<b>R 2</b>	SA2 Certification and testing guidelines - <a href="https://twiki.cern.ch/twiki/bin/view/EMI/EmiSa2CertTestGuidelines">https://twiki.cern.ch/twiki/bin/view/EMI/EmiSa2CertTestGuidelines</a>
<b>R 3</b>	SA 2 Packaging guidelines - <a href="https://twiki.cern.ch/twiki/bin/view/EMI/EmiSa2PackagingReleasingGuidelines">https://twiki.cern.ch/twiki/bin/view/EMI/EmiSa2PackagingReleasingGuidelines</a>
<b>R 4</b>	DSA2.3 – KPI and metrics definition document <a href="https://twiki.cern.ch/twiki/bin/view/EMI/EmiSa2MetricsGuidelines">https://twiki.cern.ch/twiki/bin/view/EMI/EmiSa2MetricsGuidelines</a>
<b>R 5</b>	DSA2.2.1 - QA Tools Documentation <a href="https://twiki.cern.ch/twiki/bin/edit/EMI/DeliverableDSA221">https://twiki.cern.ch/twiki/bin/edit/EMI/DeliverableDSA221</a>
<b>R 6</b>	DSA2.1 – Continuous integration and certification testbeds <a href="https://twiki.cern.ch/twiki/bin/view/EMI/DeliverableDSA24">https://twiki.cern.ch/twiki/bin/view/EMI/DeliverableDSA24</a>
<b>R 7</b>	DJRA1.5 - Standardization Work Plan and Status Report <a href="https://twiki.cern.ch/twiki/bin/view/EMI/EMIJRA1T6DJRA151">https://twiki.cern.ch/twiki/bin/view/EMI/EMIJRA1T6DJRA151</a>
<b>R 8</b>	DJRA1.6 – Integration Work Plan and Status Report <a href="https://twiki.cern.ch/twiki/bin/view/EMI/EMIJRA1T7DJRA161">https://twiki.cern.ch/twiki/bin/view/EMI/EMIJRA1T7DJRA161</a>
<b>R 9</b>	JUnit - <a href="http://junit.org/">http://junit.org/</a>
<b>R 10</b>	DejaGNU - <a href="http://www.gnu.org/software/dejagnu/">http://www.gnu.org/software/dejagnu/</a>
<b>R 11</b>	CPPUnit - <a href="http://sourceforge.net/projects/cppunit/">http://sourceforge.net/projects/cppunit/</a>
<b>R 12</b>	EMI Description of Work - <a href="https://twiki.cern.ch/twiki/pub/EMI/EmiDocuments/EMI-Part_B_20100624-PUBLIC.pdf">https://twiki.cern.ch/twiki/pub/EMI/EmiDocuments/EMI-Part_B_20100624-PUBLIC.pdf</a>



## **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 10], 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 before they are handed over to SA1 for release.

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.

Besides a description of the JRA QC activity, this report presents the results of a test availability survey 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 test suite in the existing EMI middleware.

The test results show that there is great room for improvement regarding the testing of the EMI software. 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<sup>1</sup>.

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 product team leaders. This work is coordinated by the JRA1 QC task leader with the support of SA2.

It is QC task leader responsibility to assess that PTs are following the quality process defined by SA2 and report any deviations to the appropriate project boards so that corrective actions may be taken.

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 4], so that project quality thresholds are met. In order to check quality metrics compliance developers will use the QA tools provided by SA2 [R 5].

#### 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.

#### 3.3. TESTING

JRA1 is responsible for the development of the unit and functional tests required to validate the code nominal and exceptional behavior, the components interfaces and the compliance with the adopted standards.

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 3.2.).
- improve unit test coverage so that the thresholds defined by SA2.3 are met [R 4].
- 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 11]. This activity should be carried out under the supervision of the JRA1.6 task leader.
- test the interaction and interoperability of your software component 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 6].

---

<sup>1</sup> see the SQAP or the SA2 wiki <https://twiki.cern.ch/twiki/bin/view/EMI/SA2> for more details on the guidelines.



### **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 4]
- 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 weekly by the QC task leader leveraging the build system and QA tools provided by SA2.

## 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 [REF Wikipedia];
- **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 on the availability of unit, functional, regression and standard compliance tests has been circulated to the PTs. The results gathered so far are summarized in the next sections.

### 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.

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 at the following URL:

<https://twiki.cern.ch/twiki/bin/view/EGEE/SA3Testing>

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, libarcclient, 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.

Middleware	Component	Unit Test Coverage
UNICORE	OGSA-BES	16,00%
UNICORE	samly2	18,00%
dCache	dCache	20,00%
ARC	libarcdata	21,00%
ARC	libarcclient	25,00%
ARC	WS-ARC client	25,00%
ARC	Pre WS ARC client	25,00%
UNICORE	UCC	30,00%
UNICORE	HiLA	31,00%
UNICORE	UAS authz	38,00%
gLite	VOMS-API-Java	40,00%
UNICORE	XUADB server	41,00%
UNICORE	UNICORE Gateway	43,00%
UNICORE	Client API	49,00%
gLite	Trustmanager/Util-Java	50,00%
UNICORE	Registry	51,00%
UNICORE	UAS	57,00%
UNICORE	XNJS	58,00%
UNICORE	WSRF/Lite	60,00%
UNICORE	CRL-checking	63,00%
UNICORE	Xfire-secUtils	65,00%
UNICORE	UNICORE/X	66,00%
UNICORE	Xfire-secUtilsDSig	69,00%
UNICORE	XACML entity	71,00%
UNICORE	XUADB xfire voutils	71,00%
UNICORE	Security Library	78,00%
gLite	VOMS-API-C	80,00%

*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 4.2. 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 test against LDAP and GLUES standards by running insertions in an OpenLDAP database and GStat validation probes.

dCache tests their SRM implementation against the S2 SRMv2 compliance test suite.

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<sup>2</sup>. 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.

---

<sup>2</sup> the one exception considers testing an internal product team matter and as such does not need guidelines.

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**

This first QC report focuses on the description of the activity and its relationships with other JRA1 and SA2 tasks and deliverables.

The report provides the results of a test availability survey circulated at the beginning of the project and gives some recommendations on how higher quality of the software could be achieved by improving the coverage, documentation and automation of the existing test suites.