EUROPEAN MIDDLEWARE INITIATIVE

SOFTWARE DEVELOPMENT QUALITY CONTROL REPORT

EU DELIVERABLE: DJRA1.7.1

Document identifier:	**DJRA1.7.1_Software_Development_Quality_Control Report.odt**
Date:	**6. Sep. 2010**
Activity:	**JRA1.8**
Lead Partner:	**INFN**
Document status:	**DRAFT**
Document link:	

Abstract:
This document describes the status and performance of the 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 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

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 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.

Delivery Slip

	Name	Partner / Activity	Date	Signature
From	Andrea Ceccanti	INFN/JRA1.8		
Reviewed by				
Approved by				

Document Log

Issue	Date	Comment	Author / Partner
1	01/07/10	Draft TOC	Andrea / INFN
2	06/09/10	Draft for internal review	Andrea/ INFN
3			

Document Change Record

Issue	Item	Reason for Change
1		
2		
3		

TABLE OF CONTENTS

1. INTRODUCTION......5

1.1. Purpose......5

1.2. Document Organisation......5

1.3. Application Area......5

1.4. References......5

1.5. Document Amendment Procedure......5

1.6. Terminology......5

2. EXECUTIVE SUMMARY......6

3. QUALITY CONTROL IN JRA1......7

3.1. Quality control activities......7

4. TEST AVAILABILITY SURVEY......8

4.1. Survey preliminary results......8

4.2. Test plan availability......8

4.3. Unit tests availability......8

4.4. Regression tests availability......9

4.5. Functional tests availability......10

4.6. Request for guidelines......11

4.7. Summary......11

5. SOFTWARE COMPONENTS REVIEWS SUMMARY......12

6. CONCLUSIONS......13

1. INTRODUCTION

1.1. PURPOSE

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

1.2. DOCUMENT ORGANISATION

1.3. APPLICATION AREA

This document applies to the JRA1 and SA2 activities of the project.

1.4. REFERENCES

Table 1: Table of References

R 1	DSA2.1 Software Quality Assurance Plan
R 2	DSA2.3 – KPI and metrics definition document
R 3	DSA2.2.1 - QA Tools Documentation
R 4	
R 5	
R 6	

1.5. DOCUMENT AMENDMENT PROCEDURE

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

1.6. TERMINOLOGY

Table 2: Table of Definitions

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. A survey on the availability of unit, functional, regression and standards compliance testing are presented.

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[1]. QC's main role is to verify that the quality process is correctly implemented and report any deviations to the appropriate project boards so that corrective actions may be taken.

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.

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

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 this section.

4.1. SURVEY PRELIMINARY RESULTS

4.2. 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

No feedback about test plan availability has been received from dCache so far.

4.3. UNIT TESTS AVAILABILITY

Most UNICORE components provides 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 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.

No answers to the survey have been received yet from dCache developers.

Table 3shows 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%
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	XUUDB server	41,00%
UNICORE	UNICORE Gateway	43,00%
UNICORE	Client API	49,00%
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	XUUDB xfire voutils	71,00%
UNICORE	Security Library	78,00%
gLite	VOMS-API-C	80,00%

Table 3: EMI Unit test coverage

4.4. REGRESSION TESTS AVAILABILITY

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

Most UNICORE components provide an unified JUnit test suite that comprises unit, regression and functional test suite. The test suite is executed at build time.

Some gLite components provide a regression test suite that is typically used during certification to verify bug fixes. More details for gLite components are given in Table 4.

No answers to the survey have been received yet from dCache developers.

PT	Component	Regression TS	Phase	Technology	Comments
VOMS	VOMS (client, server, APIs)	yes	Build	DejaGNU	
	VOMS-Admin (client & server)	no	N/A	N/A	A functional test suite is available that can be extended to include regression testing
Logging & Bookkeeping	L&B	yes	Certification	Scripts	
gLite Information System	BDII	no	N/A	N/A	Unit tests are run during development and there is a policy that code is not checked in unless the tests are run.
APEL	APEL client	no	N/A	N/A	
StorM	StorM	no	N/A	N/A	
MPI	mpi-start	no	N/A	N/A	
Argus	Argus Java components	yes	Build	JUnit	Bug fixes normally include a JUnit test, but it's not true for every bug fix and component.
	Argus C components	yes	Manual invocation during development	Dedicated test client	

Table 4: gLite regression testing availability

4.5. FUNCTIONAL TESTS AVAILABILITY

Most gLite components provide a functional test suite used at certification time to assess the components' correct behavior.

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 suites that can be automated and executed on the ETICS virtual testbed.

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 at certification time.

The Logging & Bookkeeping also provides a functional test suite that is run regularly during certification. Coverage is not measured, however developers try to cover every aspect.

UNICORE components test suite does not distinguish among regression, unit and functional test, so the availability and coverage presented in section 4.3. apply also for functional testing.

No answers to the survey have been received yet from dCache developers.

4.6. STANDARDS COMPLIANCE TEST AVAILABILITY

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

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

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

4.7. 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[2]. 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.

[2] the one exception considers testing an internal product team matter and as such does not need guidelines.

5. SOFTWARE COMPONENTS REVIEWS SUMMARY

As no EMI releases have been delivered yet, this section is empty.

6. CONCLUSIONS