Table of Contents

EMI Testing Policy.........................................................................................................................1
1. Introduction................................................................................................................................1
2. Definitions................................................................................................................................1
3. Tests to be performed................................................................................................................1
  3.1. Static Code Analysis...........................................................................................................1
  3.2. Unit tests..........................................................................................................................2
  3.3. Deployment tests...............................................................................................................2
    3.3.1. Installation tests for EMI 1........................................................................................2
    3.3.2. Installation tests for EMI 2........................................................................................3
    3.3.3. Configuration tests........................................................................................................3
  3.4. System tests......................................................................................................................3
    3.4.1. Basic functionality tests..............................................................................................4
    3.4.2. Regression tests...........................................................................................................4
    3.4.3. Performance tests.........................................................................................................4
    3.4.4. Scalability tests..............................................................................................................4
    3.4.5. Standard compliance/conformance tests......................................................................5
    3.4.6. Inter-component tests...................................................................................................5
4. Test Plan......................................................................................................................................5
5. Test Report................................................................................................................................5
6. Testing Process........................................................................................................................6
7. Contacts......................................................................................................................................6
8. Table of References....................................................................................................................6
9. Logbook......................................................................................................................................7
  v3.2..............................................................................................................................................7
  v3.1 (Approved on 23.01.2012)......................................................................................................7
  v3.0 (approved on 22.08.2011).......................................................................................................7
  v2.0 (approved on 28.02.2011).......................................................................................................8
  v1.0 (approved on 03.12.2010).......................................................................................................8
1. Introduction

This document describes the EMI policy to be followed when testing a new version of an EMI software product. The following topics are covered by this policy:

- Tests to be performed.
- Test Plan.
- Test Report.
- Testing process.

2. Definitions

- **Product**: EMI software product as specified in DNA1.3.2 - Technical Development Plan [R1].
- **Release Task**: Savannah Release Task [R2] where the new version of the product is tracked.

3. Tests to be performed

The following sections explain in detail the type of tests, mandatory or optional, to be included in the test plan of any EMI software product.

Each section describes:

- Type of test: mandatory or optional.
- Definition of the test.
- Required coverage: how many tests must be defined.
- Release Criteria: how many tests must be executed and passed for each new product release.

3.1. Static Code Analysis

**Type of Test**: Mandatory if the necessary plugins are available in ETICS, otherwise this test is optional.

**Definition**: Static Code Analysis is the analysis of software by examining the code without executing the program. Automated tools are often used to carry out static code analysis. In the context of EMI, ETICS offers plugins for static code analysis. Currently the plugins available for static code analysis are:

- **Java**: CCCC, FindBugs, PMD, Checkstyle.
- **C/C++**: None yet. CCCC plugin is under development.
- **Python**: None yet. Pylint plugin is under development.

For more information on how to use these plugins, visit the ETICS Plugins web page [R13].

**Required Coverage**: No requirement.

**Release Criteria**: Report about the results of the Static Code Analysis in your test Report. If your code is written in a language not supported by the ETICS plugins, static code analysis is optional and can be also
3.2. Unit tests

Type of Test: Mandatory.

Definition: Unit tests are meant to test the correctness of specific sections of source code. Tools like CPPUnit, JUnit, PyUnit are generally used and they provide the necessary documentation to get started with unit tests.

Unit tests code coverage describes the degree to which the source code of a program has been tested. A high unit test code coverage not only improves the reliability of the software, but also helps to have software that will be easier to maintain.

Required Coverage: Since ETICS doesn't have a plugin to calculate unit test code coverage, there is no code coverage requirement.

Release Criteria: Unit tests must be run for any new product release. Report about which unit tests have been run and their result in your test report. If you already calculate the code coverage for your software product with any other tool, please include the code coverage % in the test report as well.

3.3. Deployment tests

Type of Test: Mandatory (see sections 3.3.1, 3.3.2. and 3.3.3 for more details).

Definition: Deployment tests verify that the product can be properly installed and configured on all the supported platforms.

Required Coverage: deployment tests must include both clean and upgrade installations and configurations.

Release Criteria: all deployment tests must be executed and successfully passed for any product release.

3.3.1. Installation tests for EMI 1

EMI 1 supported package formats are tar.gz, src.tar.gz, rpm, src.rpm as stated in the Packaging policy [R3]. Installation tests must be performed in EMI 1 for at least rpm packages.

The installation tool used in EMI 1 for rpm packages is YUM.

Installation tests must specify the necessary YUM commands and YUM repositories needed to install the software product.

- The YUM commands must be in the form of:
  - yum install/update metapackage_name, for products that have an associated metapackage. Details on how to define metapackages can be found in the Packaging Policy [R3].
  - yum install/update package_name, for the rest of the products, i.e. libraries or utilities.
- The YUM repositories needed to install the software product are:
  - ETICS YUM repository of the emi_R_X_rc project configuration, which is the project configuration containing all the new packages scheduled for this release plus the production versions of the packages that do not change.
  - Some of the following external repositories may be also needed, depending on the software product:
    ◦ EPEL repository [R4]
    ◦ SL5 OS repository [R5]
    ◦ EGI trustanchors repository [R6]
3.3.2. Installation tests for EMI 2

In EMI 2, the installation tests for EMI 1 instructions apply for SL5 and SL6. The following instructions apply to Debian 6.

Debian 6 supported package formats are tar.gz, src.tar.gz, and deb as stated in the Packaging policy [R3]. Installation tests must be performed in Debian6 for at least deb packages.

The installation tool used in EMI 2 for deb packages is APT.

Installation tests must specify the necessary APT commands and APT repositories needed to install the software product.

- The APT commands must be in the form of:
  - `apt-get install metapackage_name`, for products that have an associated metapackage. Details on how to define metapackages can be found in the Packaging Policy [R3].
  - `apt-get install package_name`, for the rest of the products, i.e. libraries or utilities.

- The APT repositories needed to install the software product are:
  - ETICS APT repository of the emi_R_X_rc project configuration, which is the project configuration containing all the new packages scheduled for this release plus the production versions of the packages that do not change.

Upgrade tests must be performed for minor releases and across major releases.

3.3.3. Configuration tests

The configuration tool depends on the specific product. Deployment tests must also define the configuration commands, configuration variables and/or configuration files, if any, needed to be able to configure the product. Depending on the product, one or more of the following have to be presented:

- A configuration command is the command that needs to be run to configure your software product. For instance, in case of using yaim: `yaim -c -s site-info.def -n metapackage_name`.
- A configuration variable is a variable that needs to be defined by the user. For instance, in case of using yaim: `BDII_HOST`.
- A configuration file or template is a file containing the minimal set of configuration items for the product to be up and running.

Configuration tests must be run after the corresponding installation test (See section 3.3.1 and 3.3.2) for both clean and upgrade installations.

3.4. System tests

System tests cover the majority of the tests for a new product version release. System tests considered in this document are:

- Basic functionality tests
- Regression tests
- Performance tests
- Scalability tests
- Standard compliance/conformance tests

The broader is the scope of system testing in these areas, the better. We encourage PTs to keep on improving their test plans throughout the EMI project lifetime by broadening and enlarging the scope of their tests as well as their level of automation.
3.4.1. Basic functionality tests

**Type of Test:** Mandatory.

**Definition:** Basic functionality tests aim at testing the core features of the product. Basic functionality tests must include the description of the functionality to be tested. When new functionality is added to the product, new tests must be written and added to the test plan of the product.

**Required Coverage:** All the core features of a product must have a corresponding basic functionality test.

**Release Criteria:** All basic functionality tests must be executed and successfully passed for any product release.

3.4.2. Regression tests

**Type of Test:** Mandatory.

**Definition:** Regression tests are tests that are meant to verify specific software defects (bugs). A regression test must be associated to the RfC where the defect has been reported in the RfC tracker. A regression test must have an identifier, ideally corresponding to the associated RfC (optional).

**Required Coverage:** All RfCs tracking a bug where the bug verification can be automatically implemented must have a corresponding regression test.

**Release Criteria:** All available regression tests must be executed and successfully passed for any product release.

3.4.3. Performance tests

**Type of Test:** Optional.

**Definition:** Performance tests are tests that aim at verifying the performance of a product, which in many cases involves the measurement of the response time for specific service requests. Performance tests should verify how well the service behaves with nominal workloads.

The execution of performance tests depends on the service under test, it may or may not be automated, and can involve the use of external tools. In some cases the execution of performance tests may require the establishment of a specific testbed, and may involve several sites and the coordination of SA2.6.

**Required Coverage:** A minimum set of performance requirements must be identified by the PT. A corresponding set of tests must be defined to check that the product is able to meet those requirements.

**Release Criteria:** If there are performance tests defined, they must be executed in every major product release where there are substantial functionality changes.

3.4.4. Scalability tests

**Type of Test:** Optional.

**Definition:** Scalability tests are meant to verify that the product behaves according to its specifications when varying one of the variables that can affect its performance. Load and stress tests are included.

The execution of scalability tests depends on the service under test, it may or may not be automated, and can involve the use of external tools. In some cases the execution of scalability tests may require the establishment
of a specific testbed, and may involve sites and the coordination of SA2.6.

**Required Coverage**: a minimum set of variables related to the performance of the product must be identified by the PT. The expected behaviour of the product with respect to the variation of those variables must be also identified. A corresponding set of tests must be defined to check that the product behaves as expected.

**Release Criteria**: If there are scalability tests defined, they must be executed in every major product release where there are substantial functionality changes.

### 3.4.5. Standard compliance/conformance tests

**Type of Test**: Optional.

**Definition**: Standard compliance/conformance tests are meant to verify that a software conforms or complies to a specific standard. In the context of EMI, some examples are Glue v.2.0 or SMRv2.

**Required Coverage**: Standard compliance/conformance tests must be defined for at least the basic functionality provided by the adoption of the standard.

**Release Criteria**: compliance/conformance tests must be executed only in those product releases where the standard is adopted for the first time.

### 3.4.6. Inter-component tests

**Work in progress. More info available at Task Force Integration Testing**

**Type of Test**: Mandatory.

**Definition**: Inter-component tests are meant to verify that a software product is able to operate with other EMI products. Inter-component tests must specify which external EMI products to the one under test are needed. Inter-component tests are written by PTs and executed in the EMI Testbed.

**Required Coverage**:

**Release Criteria**: The execution of inter-component tests is triggered by the release manager and relies on the use of the EMI testbed [R12]. Intere-component tests are the final stage before going to Production and they are done on certified software products. There will be a 2 week time window of mandatory Intercomponent testing for which test reports will be provided.

### 4. Test Plan

The Test Plan must describe the strategy that will be adopted for testing the software product. It should be written by the PTs.

The following template describes the information that must be present: Test Plan template [R7].

QC is gathering all the EMI test plans under the QC Test Plan [R8] twiki. We encourage PTs to add their Test Plans in this twiki.

### 5. Test Report

The Test Report must contain the result of the tests specified in the Test Plan [R7] that have been executed on a product release. It should be written by the PTs.
The following template must be used: Test Report template [R9].

In the summary of the report, questions applicable to more than one platform, must be answered taking into account all the applicable platforms.

This report must be attached in the corresponding release task as explained in the Change Management Policy [R10].

6. Testing Process

PTs are responsible for testing their products.

Testing a new version of a product has to be ALWAYS done for each EMI major release and supported platform. Exceptions may apply for Emergency Releases that will be discussed at the EMT. For example, imagine EMI-1 and EMI-2 are the two major EMI releases supported, and you have to do a minor release of product emi-sherpa which fixes some bugs present in both EMI-1 and EMI-2. Imagine that EMI-1 is supported in SL5 and EMI-2 is supported in SL6 and Debian 6. You will need to test:

- emi-sherpa for EMI-1 SL5
- emi-sherpa for EMI-2 SL6
- emi-sherpa for EMI-2 Debian 6

The pre-condition for starting the testing phase is that the product builds without any errors, passes all the unit-tests and has all the needed packages registered in the ETICS permanent repository. This is the repository where packages are stored once the corresponding ETICS configurations have been locked and built. See the EMI Release Checklist [R11] for more details.

The input for testing a product is the Test Plan [R7] of the product.

The output of the testing phase in all the supported platforms is the Test Report [R9]. Only one test report will be produced for all the supported platforms answering the questions taking into account the testing in every platform. This report must be attached in the corresponding release task, as explained in the Change Management Policy [R10].

7. Contacts

EMI SA2

8. Table of References

<table>
<thead>
<tr>
<th>Reference</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>R2</td>
<td>EMI Release Tracker [R2] <a href="https://savannah.cern.ch/task/?group=emi-releases">https://savannah.cern.ch/task/?group=emi-releases</a></td>
</tr>
<tr>
<td>R3</td>
<td>EMI Packaging Policy [R3] <a href="https://twiki.cern.ch/twiki/bin/view/EMI/EmiSa2PackagingPolicy">https://twiki.cern.ch/twiki/bin/view/EMI/EmiSa2PackagingPolicy</a></td>
</tr>
<tr>
<td>R4</td>
<td>EPEL repository [R4] <a href="http://fedoraproject.org/wiki/EPEL">http://fedoraproject.org/wiki/EPEL</a></td>
</tr>
<tr>
<td>R5</td>
<td>SL5 OS repository [R5] <a href="https://www.scientificlinux.org/distributions/5x/">https://www.scientificlinux.org/distributions/5x/</a></td>
</tr>
<tr>
<td>R6</td>
<td>EmiSa2TestPolicy &lt; EMI &lt; TWiki</td>
</tr>
</tbody>
</table>
9. Logbook

Details  Hide Details

v3.2

- 06.02.2012: Added upgrade tests to section 3.3.2 to clarify that there should be an upgrade path across major releases.

v3.1 (Approved on 23.01.2012)

- 16.01.2012: Added installation tests for Debian in EMI2 and reflect the changes in the test report.
- 02.11.2011: Inter-component tests changed to optional, to be aligned with the Production Release Criteria.

v3.0 (approved on 22.08.2011)

- 22.08.2011: Added feedback from ARC: improved Deployment section to differentiate from EMI 1 and EMI 2 (which will also include Debian). Given more details on configuration tests.
- 05.08.2011: Added link to ETICS plugins web page.
- 04.08.2011:
  - Added feedback from developers.
  - Review and update all sections.
  - Update Test Plan and Test Report twikis.
- 02.08.2011:
  - Added section numbers.
  - Changed component with product where applicable.
- 03.03.2011: Feedback from the training:
  - Improve section on unit tests. not clear what it is requested.
  - Improve Test Report template to clarify that ETICS configuration name refers to the subsystem.
- 02.03.2011: Changed Integration tests with Inter-component tests.
v2.0 (approved on 28.02.2011)

- **22.02.2011**: Meeting with PEB. Feedback given on unit tests (should be mandatory), improve definition of integration tests, improve test report template, other general feedback.
- **21.02.2011**: The following changes have been implemented:
  - Use **testing instead of certification** since certification is going to be used with a different meaning within EMI.
  - Use **Test Plan instead of Software Verification and Validation Plan** and **Test Report instead of Software Verification and Validation Report**.
  - Applied feedback from ARC: Added whether tests are mandatory or optional, added section for definitions, added required coverage and release criteria. Reorganised Testing process to avoid duplication.
- **14.02.2011**: The following changes have been implemented:
  - Change guidelines with policy.
  - Change release candidate with component release to harmonise with Change Management Policy.
  - Change bug with RfC to harmonise with Change Management Policy.
  - Reorganise the name of the main sections:
    - General testing guidelines to **tests to be performed to certify an EMI component release**
    - Certification on a release candidate to **component release certification process**
    - Added a section for Static Code Analysis
  - Reorganised **Tests to be performed to certify an EMI component release**
  - Reorganised **Software Verification and Validation Report** and **Software Verification and Validation Plan** after feedback collected from UNICORE. A new template is now included and both twikis have been modified to match with the new template. The relevant sections in this twiki have been updated as well to be aligned to the changes in the twikis.

v1.0 (approved on 03.12.2010)

- **01.12.2010**: Minor updates plus removed the section 'Middlewares documentation'.
- **07.09.2010**: Document updated according to feedback from meeting
- **07.09.2010**: SA2 meeting discussing the draft
- **02.09.2010**: First draft prepared