EUROPEAN MIDDLEWARE INITIATIVE

SOFTWARE MAINTENANCE QUALITY CONTROL REPORT

EU DELIVERABLE: D3.3.2

Document identifier:	**EMI-D3.3.2-Software_Maintenance_Quality_Control_Report_v1.5.odt**
Date:	**05/04/2011**
Activity:	**SA1.4**
Lead Partner:	**CINECA**
Document status:	**First Release**
Document link:	

Abstract:
This document describes the status and performance of the quality control task with details on the availability and execution of regression tests for the supported EMI components, test unit availability and coverage and various static and dynamic metrics on released components.

Delivery Slip

	Name	Partner / Activity	Date	Signature
From				
Reviewed by				
Approved by				

Document Log

Issue	Date	Comment	Author / Partner
1	28/01/2011	Table of contents	Giuseppe Fiameni/CINECA
2	21/02/2011	First release of the document	Giuseppe Fiameni/CINECA
3	23/02/2011	SA1 leader review	Giuseppe Fiameni/CINECA
4	22/03/2011	New release after reviewers' comments	Giuseppe Fiameni/CINECA
5	28/03/2011	New release after Maria Alandes Pradillo comments	Giuseppe Fiameni/CINECA
6	04/04/2011	Final release	Giuseppe Fiameni/CINECA

Document Change Record

Issue	Item	Reason for Change
1		
2		
3		

TABLE OF CONTENTS

Table of Contents

1. INTRODUCTION

1.1. PURPOSE

Quality Control (QC) verifies the application of Quality Assurance (QA) processes and procedures and, through the execution of periodic reviews, reports and measures the status and performance of the SA1 work. This document reports the results of the Quality Control activity performed until PM9. It includes an aggregated view of quality check results and performance measurements, putting in evidence which changes to internal procedures should be considered to correct anomaly or nonconformity discovered during the control process. The list of change requests will be submitted to QA team that, on the base of project's priorities and objectives, will determine which of them are indispensable to apply.

1.2. DOCUMENT ORGANIZATION

The document is organized as follows:

- Chapter 1 and 2 are the Introduction and the Executive Summary respectively;
- Chapter 3 presents the organization of the Quality Control activity and how it interacts with the Quality Assurance;
- Chapter 4 reports the results of the Quality Review scheduled for PM9;
- Chapter 5 reports the status of the EMI 1 (Kebnekaise) release and how much work is still needed to reach the deadline;
- Chapter 6 presents the status of the security assessment activity;
- Chapter 7 describes the status of the regression tests;
- Chapter 8 is the section reporting the conclusions of this deliverable.

1.3. REFERENCES

R1	**Quality Assurance Plan**, *https://twiki.cern.ch/twiki/bin/view/EMI/DeliverableDSA21*
R2	**Quality Assurance Metrics**, *https://twiki.cern.ch/twiki/bin/view/EMI/TSA23*
R3	**Quality Assurance Wiki Page**, *https://twiki.cern.ch/twiki/bin/view/EMI/SQAP*
R4	**Software Release Schedule – EMI 1**, *https://twiki.cern.ch/twiki/bin/view/EMI/EMI-1*
R5	**Software Release Plan**, *https://twiki.cern.ch/twiki/bin/view/EMI/DeliverableDSA12*
R6	**Software Maintenance and Support Plan**, *https://twiki.cern.ch/twiki/bin/view/EMI/DeliverableDSA11*
R7	**Technical Development Plan**, *https://twiki.cern.ch/twiki/bin/view/EMI/DeliverableDNA131*
R8	**Release Management Wiki Page**, *https://twiki.cern.ch/twiki/bin/view/EMI/TSA13*
R9	**Configuration and Integration Policy**,

	https://twiki.cern.ch/twiki/bin/view/EMI/EmiSa2ConfigurationIntegrationPolicy
R10	**Certification and testing policy,** *https://twiki.cern.ch/twiki/bin/view/EMI/EmiSa2CertPolicy*
R11	**Change management policy**, *https://twiki.cern.ch/twiki/bin/view/EMI/EmiSa2ChangeManagementPolicy*
R12	**DSA2.2.1 - QA Tools Documentation**, *https://twiki.cern.ch/twiki/bin/view/EMI/DeliverableDSA221*
R13	**Test Plan and Template**, *https://twiki.cern.ch/twiki/bin/view/EMI/EmiSA2TestPlan*
R14	**Quality Control Report PM6**, *https://twiki.cern.ch/twiki/bin/view/EMI/SA1QCPM6*
R15	**Software Quality Assurance Plan Documentation**, *https://twiki.cern.ch/twiki/bin/view/EMI/SQAP#SQAP_Documentation*
R16	**Firs Principles Vulnerability Assessment**, *http://www.cs.wisc.edu/mist/VA.pdf*
R17	**Review of the Software Release Plan**, *https://twiki.cern.ch/twiki/bin/view/EMI/SA1QCRSRP*
R18	**Review of the Software Release Schedule,** *https://twiki.cern.ch/twiki/bin/view/EMI/SA1QCRSRS*
R19	**Review of the Software Maintenance and Support Plan,** *https://twiki.cern.ch/twiki/bin/view/EMI/SA1QCRSMSP*
R20	**Review of the Security Assessments**, *https://twiki.cern.ch/twiki/bin/view/EMI/SA1QCRSA*
R21	**Consejo Superior de Investigaciones Cientificas**, *http://www.csic.es*
R22	**First principles vulnerability assessment,** *Proceedings of the 2010 ACM workshop on Cloud computing security workshop, James A. Kupsch, Barton P. Miller, Elisa Heymann, Eduardo César*
R23	**SA1 Quality Control Wiki Page,** *https://twiki.cern.ch/twiki/bin/view/EMI/TSA14*
R24	**Vulnerability reports,** *http://www.cs.wisc.edu/mist/includes/vuln.html*
R25	**Security Assessment Plan,** *https://twiki.cern.ch/twiki/bin/view/EMI/SA1QCSAP*
R26	**Production Release Criteria,** *https://twiki.cern.ch/twiki/bin/view/EMI/ProductionReleaseCriteria*
R27	**Security Assessment activity page,** *https://twiki.cern.ch/twiki/bin/view/EMI/SA1QCSA*

R28	**EMI Indico Digital Repository,** *http://indico.cern.ch/*
R29	**Software Release Schedule – EMI 0,** *https://twiki.cern.ch/twiki/bin/view/EMI/EMI-0*
R30	**Software Test plans list,** *https://twiki.cern.ch/twiki/bin/view/EMI/QCTestPlan*
R31	**Software Verification and Validation template,** *https://twiki.cern.ch/twiki/bin/view/EMI/SoftwareVerAndValTemplate*
R32	**gLExec Vulnerability Reports,** *http://www.cs.wisc.edu/mist/glexec/vuln_reports/*
R33	**EMT meetings participants,** *https://twiki.cern.ch/twiki/bin/view/EMI/EMTMeetingsParticipants*
R34	**EMI 1 Release Schedule tracker,** *https://savannah.cern.ch/projects/emi-releases/*
R35	**EMI development tracker,** *https://savannah.cern.ch/task/?group=emi-dev*
R36	**EMI EMT Meetings,** *http://indico.cern.ch/categoryDisplay.py?categId=3077*
R37	**GGUS- Global Grid User Support ,** *https://gus.fzk.de/pages/home.php*
R38	**EMI Release Management Process,** *https://twiki.cern.ch/twiki/bin/view/EMI/EmiSa2ReleaseManagementPolicy*

1.4. DOCUMENT AMENDMENT PROCEDURE

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. The document shall be maintained in the CERN CDS repository and be made accessible through the OpenAIRE portal.

1.5. TERMINOLOGY

ABI	Application Binary Interface
ACR	Approved Change Request
API	Application Programming Interface
CDS	CERN Document Server
CG	Change Request
CSIC	Consejo Superior de Investigaciones Cientificas
DCI	Distributed Computing Infrastructure

DMSU	Deployed Middleware Support Unit
EGI	European Grid Infrastructure
EMT	Engineering Management Team
ETICS	eInfrastructure for Testing, Integration and Configuration of Software
FPVA	First Principles Vulnerability Assessment
GGUS	Global Grid User Support
ITIL	IT Infrastructure Library
KPI	Key Performance Indicator
kSLOC	Kilo Source Lines Of Code
MCB	Middleware Coordination Board
NGI	National Grid Initiative
PEB	Project Executive Board
PTB	Project Technical Board
QA	Quality Assurance
QC	Quality Control
RfC	Request for Change
SLA	Service Level Agreement
SQAP	Software Quality Assurance Plan
SQC	Software Quality Control
SU	Support Unit
VC	Validated Change

2. EXECUTIVE SUMMARY

Performing Quality Control is an activity concerned with the monitoring of project outcomes to see whether they comply with quality standards set out in the SQAP (Software Quality Assurance Plan) [R1] or within internal procedures, such as those concerning the release and packaging of software components [R5]. Operating throughout the project, its aim is to identify unacceptable or non-conformable results and inform project executive boards (i.e. PEB) about their existence so that corrective actions can be undertaken to eliminate, or mitigate, future negative impacts on project's results. The principal goal is that all EMI components, before being included in a stable release, satisfy well-defined quality standards. In general, the adoption of quality standards must be sufficient to guarantee, to a high degree of confidence, that all EMI products (software components, documentation, etc.) meet stakeholders requirements, in terms of acceptance criteria, and do not contain defects or bring new security vulnerabilities.

As part of SA1 work-package, the QC task carries on several activities covering different aspects of the established EMI quality framework [R15]. Basically they consist of:

- **performing periodic reviews** that**,** scheduled every three months, aim to constantly control the performance of the SA1 team. The review action is performed through the adoption of control tools, such as check lists, control lists and metrics defined in the SQAP;
- **controlling the release process,** checking that release procedures and deadlines are met as well as the different stages of the release process proceed in line with the schedule;
- **controlling that all candidate components,** composing an EMI major release, satisfy Production Release Criteria [R26] , including the regression tests to guarantee that no defects have been introduced after the fixing of any bug;
- **ensuring the security assessment** of software components to ensure that most critical components do not contain vulnerabilities or security holes;
- **delivering project periodic reports** to elaborate the results of periodic control activities pointing out any nonconformity or deviation from quality standards that could introduce new defects in the future.

This document provides the outcome for each of aforementioned activities, taking in consideration that at the time of writing no EMI official releases have been released yet and that developers have not completely become familiar with important project procedures, such as the Certification and Testing guidelines (named policies in the future) [R10] due to the delay introduced to get them work together. The experience matured so far, especially during the preparation of the EMI 0 release, will surely improve generic performance and facilitate the organization of future releases.

3. THE ORGANIZATION OF THE QUALITY CONTROL

The diagram below (see Figure 1) describes how the QC and the QA entities interact and how the information flows across them. This tightly coupled interaction, and the continue exchange of information, is fundamental for the progressive improvement of project performance.

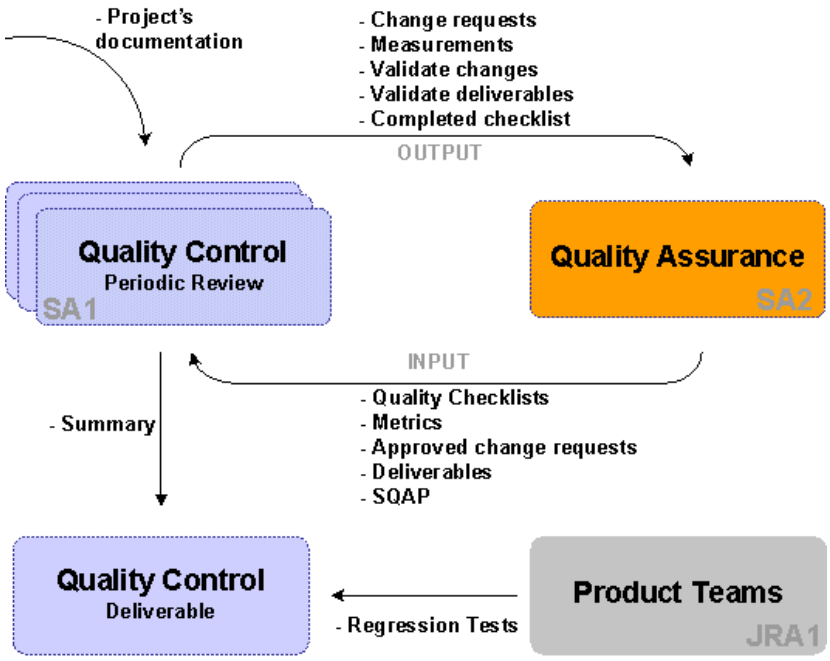


Figure 1: Quality Control Information flow

3.1. INPUTS TO REVIEW

This section presents the list of information pieces that the QC receives as input and that are indispensable to execute its controls.

Quality Assurance Plan

The SQAP (Software Quality Assurance Plan) [R1] specifies the procedures, the metrics, the standards, and the manner according which the EMI project achieves its quality goals in terms of software development.

Quality Checklists

A check-list is a structured tool used to verify whether the objectives of a process have been met or not. As each step is completed, it is checked off the list. As defined in the SQAP, the input checklists for the QC in SA1 are:

- Review of the Software Release Plan
- Review the Software Release Schedule
- Review the Software Maintenance and Support Plan
- Security Assessments

The names of the checklists are re-used later to report more details on their execution.

Quality Metrics

A quality metric is an operational definition that describes, in very specific terms, a project or product attribute and how the QC process will measure it.

The metrics defined for the QC in SA1 are:

- Review of the Software Release Plan
  - *No metric defined for this review*
- Review the Software Release Schedule
  - *Delay on the release schedule (ID: DELAYONTHERELEASE)*
- Review the Software Maintenance and Support Plan
  - *Total user incidents per user month (ID: TOTALUSERINCIDENTS)*
  - *Training and support incident per user month. (ID: TRAININGSUPPORTINCIDENTS)*
  - *Average time to deal with an incident at the 3rd level of user support (ID: AVERAGETIMEFORUSERINCIDENTS)*
- Security Assessments
  - *No metric defined for this review*

Approved Change Requests

An ACR (Approved Change Request) is a change request submitted by the QC during a previous review that, after having positively reviewed by the QA, has been granted to be applied. The list of ACRs is provided as input to the quality review in order to verify that their implementation is correct and satisfies the quality standards Approved change requests can include modifications to the work methods or to the schedule and come as a result of the change management process led by the QA in collaboration with the PEB.

Deliverables

This is the list of deliverables that the QC verifies, stating if they comply with the quality standards or not. With the term deliverable is meant any internal document (e.g. software release plan, security assessment plan, etc.) or software component produced as a result of the project and that will be delivered to a customer.

3.2. OUTPUTS FROM REVIEW

This section presents the list of information pieces that the QC returns to the QA for further elaboration, such the improvement of the quality procedures.

Change Requests

Change requests are recommended corrective or preventive actions for preventing future defects in software products. Any change request contains a request for an adjustment in a procedure or policy definition. It reports what needs to be accomplished, but leaves to the executive board how the change should be implemented.

Measurements

Measurements are the documented results of the elaboration of associated quality metrics. The purpose behind of taking measurements is to evaluate how the project is performing according to defined metrics.

Validated Changes

Validated changes refer to approved change requests that have been validated with success because their implementation satisfies quality standards. Any changed or repaired procedures or products are once again verified and could be either accepted or rejected before being considered definitive.

Validated Deliverables

Validated deliverables are deliverables, among those received in input from the QA, that have successfully passed the Quality Control review. By the term deliverable is meant any verifiable product or service that is produced within the project.

Completed Checklists

Completed checklists are output of the QC activity and become part of the project's documentation.

4. QUALITY CONTROL REVIEW – PM9

4.1. REVIEW OF THE SOFTWARE RELEASE PLAN

4.1.1 Input

Checklists

- *Checklist for the Review of the Software Release Plan [R17] .*

Metrics

- *No metrics defined for this review.*

Approved Change Requests

The list of changes requested during the previous QC report, and that have been accepted by the QA team, follows:

- *define the tolerance range of positive checks for considering the deliverable validated. It is not clear how many positive checks are needed to consider the Software Release Plan validated;*
  - this request has been accepted by the QA team and the next release of the SQAP will be modified accordingly*;*

Deliverables

- *Software Release Plan [R5].*

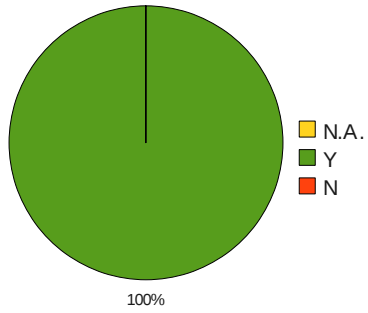
4.1.2 Output

Completed Checklist

Check Number	Question	Response
1	Does the list of supported platforms correspond to the actual set of platforms on which software components are released?	Y
	see Software Release Plan [R5]	
2	Is the installation of external dependencies well documented?	Y
	see Software Release Plan [R5]	
3	Are instructions to build the software up to date?	Y
	see Software Release Plan [R5]	
4	Is the list of supported delivery software formats up to date (source and binary packages, tarball, package lists, etc)?	Y
	see Software Release Plan [R5]	
5	Is the description of the process on how to handle changes up to date?	Y

	see Software Release Plan [R5]	
6	Are the communication channels published with updated information?	Y
	see Software Release Plan [R5]	
7	Is the process on how to deliver software to the Production Infrastructures up to date and it's aligned to what the Production Infrastructures (EGI, PRACE) are expecting?	Y
	see Software Release Plan [R5]	

Table 1: Review of the Software Release Plan (N.A. = Not Available)



- 100% of checks have been executed with success.

Measurements

There are no measurements defined for this review.

Comments

The table below (Table 2) reports specific comments on the executed checks that either have returned a non-satisfactory response (i.e. N.A. or N) or simply require further clarifications.

Check Number	Comments
1	Does the list of supported platforms correspond to the actual set of platforms on which software components are released?
	At the moment, there are no official components available but the release procedures clearly state which the reference platforms are and how to package components according to their specifications.
2	Is the installation of external dependencies well documented?
	The external dependencies are not directly described in the main document, but are maintained through EMI project wiki, facilitating their refinement as long as new packaging issues arise.
3	Are instructions to build the software up to date?

	The instructions to build the software are not directly included in the Software Release Plan, but are maintained in the Build and Configuration Policy [R9], so facilitating their refinement as long as new building issues or needs arise.
5	Is the description of the process on how to handle changes up to date?
	This process is better documented in the Software Maintenance and Support Plan [R6]
6	Are the communication channels published with updated information?
	Most of the communications among PTs take place within the EMT mailing list and all documents (e.g. minute, action lists, etc.) are maintained in the project repository [R28]. The list of participants for each EMT meeting is maintained at this link R36].

Table 2: Review of the Software Release Plan – Comments

Validated Changes

Though changes requested during the previous QC report have been approved by the QA, they have been not been reported in SQAP and therefore no further considerations can be done on them.

Validated Deliverables

Name of the deliverable under evaluation	Validated	
	YES	NO
Software Release Plan	Y	

Variations from previous report

The previous QC report reported a negative result, the Software Release plan was not available at that time and all corresponding checks failed. Since then, a significant improvement has been observed, culminating in the achievements of quality objectives as well as the release of the plan.

Change Requests

No changes are requested for this review.

4.2. REVIEW THE SOFTWARE RELEASE SCHEDULE

The *Review of the Software Release Schedule* [R18] checks that the priorities of the project are taken into account and reflected in the scheduled releases.

The *Software Release Schedule* is a document requested by the SQAP to guide PTs towards the release of EMI components. It outlines the different phases which compose the release process, which members are involved in each of them, and their deadlines. The scheduled for the EMI 1 release is available at this link [R4].

4.2.1 Input

Checklists

- *Checklist for the Review of the Software Release Schedule [R18].*

Metrics

- *Delay on the release schedule (ID: DELAYONTHERELEASE).*

Approved Change Requests

The list of changes requested during the previous QC report, and that have been accepted by the QA team, follows:

- ***define** the tolerance range of positive checks for considering the deliverable validated. It is not clear how many positive checks are needed to consider the Software Release Schedule validated;*
  - this request has been accepted by the QA team and it will be included in the next release of the SQAP.

Deliverables

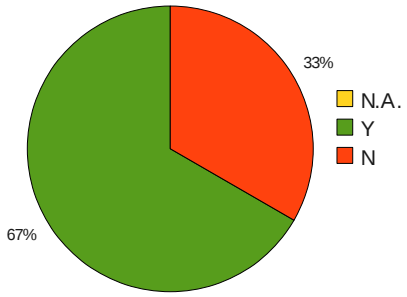
- *Software Release Schedule [R4].*

4.2.2 Output

Completed Checklist

Check Number	Question	Re-sponse
1	*Has the previous schedule been kept?*	N
	see Software Release Schedule [R1]	
2	*Does the new schedule take into account what wasn't accomplished in the previous schedule?*	Y
	see Software Release Schedule [R1]	
3	*Is the new schedule aligned to the Software Development Plan and the priorities of the project?*	Y
	see Software Release Schedule [R1]	

Table 3: Review the Software Release Schedule



- 67% of checks have been executed with success.
- 33% of checks have failed.

Measurements

In the following, the metrics defined for this review and corresponding measures are reported.

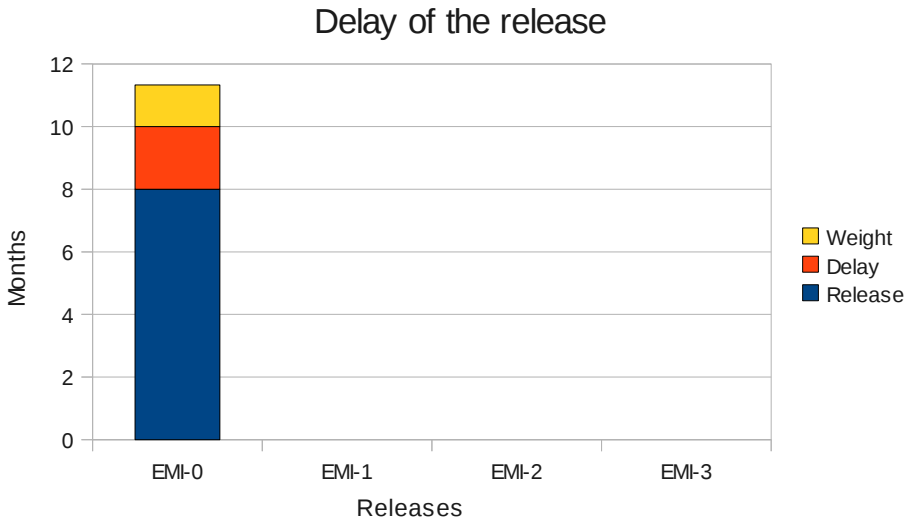
ID	**DELAYONTHERELEASE**
Name	Delay on the release schedule
Description	This metric could be provided as a histogram showing the delay time (in days) for each release, weighted using the release time
Unit	(release delay)/(release time) * 100
Measurements	
Thresholds/target value	Ideally the release deadlines should be always met, leading to 0 delays for each release. Proper thresholds have to be defined. The trend of the delays over time could provide useful hints for process optimization.
Comment	The "weight" value represents the ratio between the number of months expected to prepare a release (8) and those of delay (2). As the project progresses, the weight should gradually decrease to demonstrate that employed quality procedures are adequate for improving the release process performance and that PTs are respecting them.

Table 4: Delay on the release schedule – Metric

Comments

The table below reports specific comments, if any, on check results.

Check Number	Comments
1	*Has the previous schedule been kept?*
	The EMI 0 (Zugspitze) was release with 2 months of delay.
2	*Does the new schedule take into account what wasn't accomplished in the previous schedule?*
	The lessons learned during the preparation of the EMI 0 release have been taken in account for improving the release procedures.

Table 5: Review the Software Release Schedule - Comment

Validated Changes

No changes can be validated for this report.

Validated Deliverables

Name of the deliverable under evaluation	Validated	
	YES	NO
Software Release Schedule	Y	

Variations from previous review

The previous QC report reported a negative result, the Software Release Schedule was not available and all checks failed. Since then, a significant improvement has been observed, culminating in the achievements of quality objectives as well as the release of the schedule.

Change Requests

No changes are requested for this review.

4.3. REVIEW THE SOFTWARE MAINTENANCE AND SUPPORT PLAN

The Review of the Software Maintenance and Support Plan [R18] checks that the plan is up to date and describes the actual maintenance and support processes and that the SLAs are respected.

The Software Maintenance and Support Plan has been released and is accessible at [R5].

4.3.1 Input

Checklists

- *Checklist for the Review the Software Maintenance and Support Plan [R18].*

Metrics

- *Total user incidents per user month (ID: TOTALUSERINCIDENTS)*
- *Training and support incident per user month. (ID: TRAININGSUPPORTINCIDENTS)*
- *Average time to deal with an incident at the 3rd level of user support (ID: AVERAGETIMEFORUSERINCIDENTS)*

Approved Change Requests

The list of changes requested during the previous QC report, and that have been accepted by the QA team, follows:

- ***define** the tolerance range of positive checks for considering the deliverable validated. It is not clear how many positive checks are needed to consider the Software Maintenance and Support Plan validated;*
  - this request has been accepted by the QA team and it will be included in the next release of the SQAP;
- ***define** the metric thresholds for considering the deliverable validated;*
  - this request has been accepted by the QA team and it will be included in the next release of the SQAP;
- ***consider** to aggregate the quality metrics defined for this review with the project KPIs (KSA1.1 and KSA1.2 of the EMI DoW);*
  - this request has been accepted by the QA team and it will be included in the next release of the SQAP.

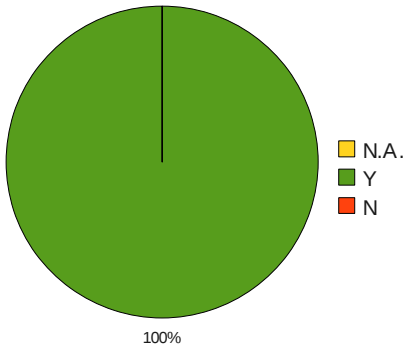
Deliverables

- *Software Maintenance and Support Plan [R5]*

Completed Checklist

Check Number	Question	Re-sponse
1	*Is the process on how to handle incidents reported by EMI users using GGUS up to date?*	Y
	see Software Maintenance and Support Plan [R5]	
2	*Is the process on how to handle requests coming from EMI users or other PTs up to date?*	Y
	see Software Maintenance and Support Plan [R5]	
3	*Is the process on how to handle problems up to date?*	Y
	see Software Maintenance and Support Plan [R5]	

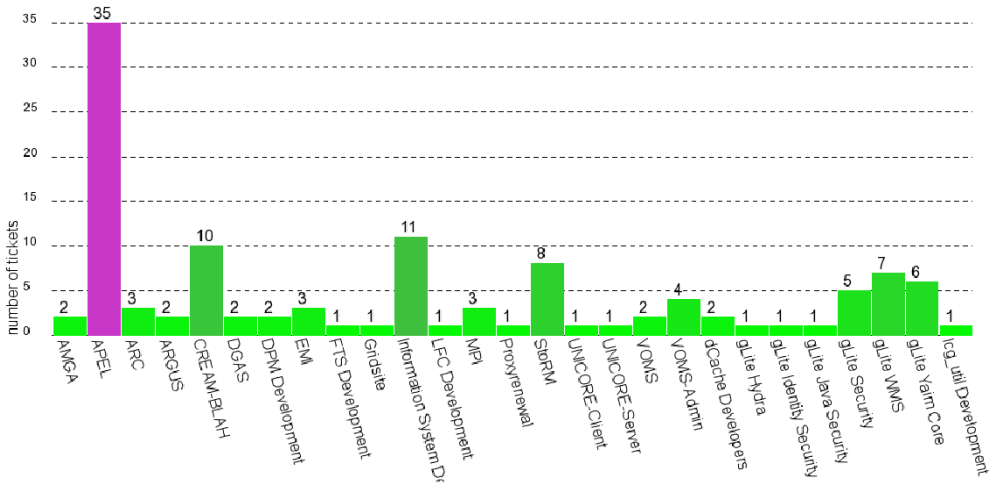
Table 6: Review the Software Maintenance and Support Plan



- 100% of the checks have been executed with success.

Measurements

The measurements on user incidents, which are reported below, concern the middle-ware releases packaged following their own release processes, not those set out in EMI. The graphics have been taken from GGUS [R37].

ID	**TOTALUSERINCIDENTS**
Name	Total user incidents per user month
Description	This metric covers defects not only in the software but also in the documentation, training and user support processes, per user month. User month means the number of users (in our case, deployed services?) per month.
Unit	GGUS tickets per user per month
Measurement	 These measurements report the number of incidents submitted to GGUS for all EMI Support Units from November 2010 to January 2011. The value reported for the APEL component is not coherent with the rest of the chart. Its definition includes the software defects and the incidents occurred during

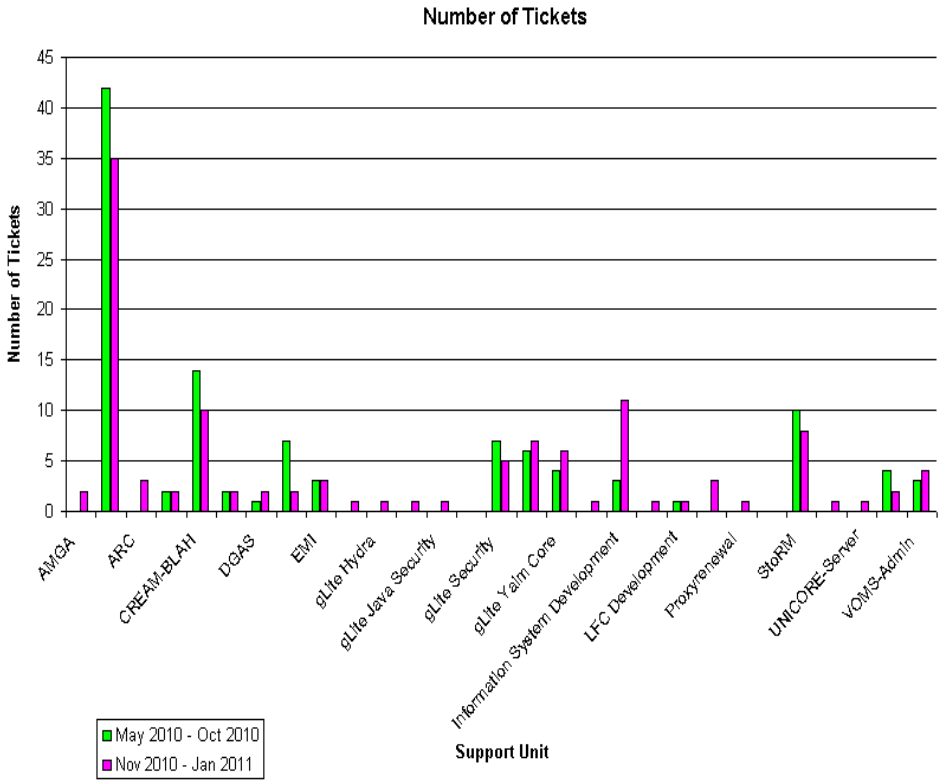
	the operation of the service, while for the other components only the formers are considered. To make the measurements of all components consistent, the APEL component will be broken down in different sub components and only the values concerning the maintenance of the software considered for this measurement.
Thresholds/target value	It is difficult to state a threshold valid for all the product teams, in general a decreasing trend would show positive results.
Comment	 This graph reports the number of tickets submitted from May to October 2010 (green bars), and those from November 2010 to January 2011 (fuchsia bars). Its aim is to show the trend of submitted tickets across the two reporting periods. Approximately, there is no significant improvements from the last control, in general the number of tickets has remained unchanged over the time. Particular attention should be deserved for the "*Information System*" component since its tickets has quadruplicated over the reporting period.

Table 7: Total user incidents per user month

ID	**TRAININGSUPPORTINCIDENTS**
Name	Training and support incident per user month.
Description	This metric covers defects in the training and user support processes, per user month. User month means the number of users per month. The training and support defects can be derived by subtracting the tickets in status unsolved (ticket that generated a bug) from the total number of opened tickets. It relies on proper bug opening from

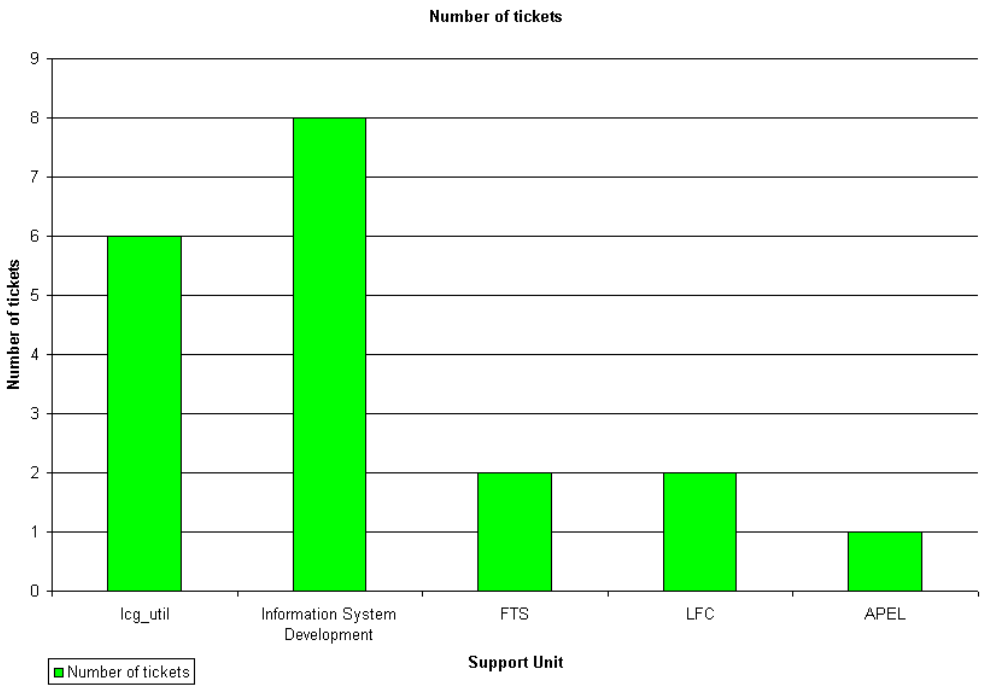
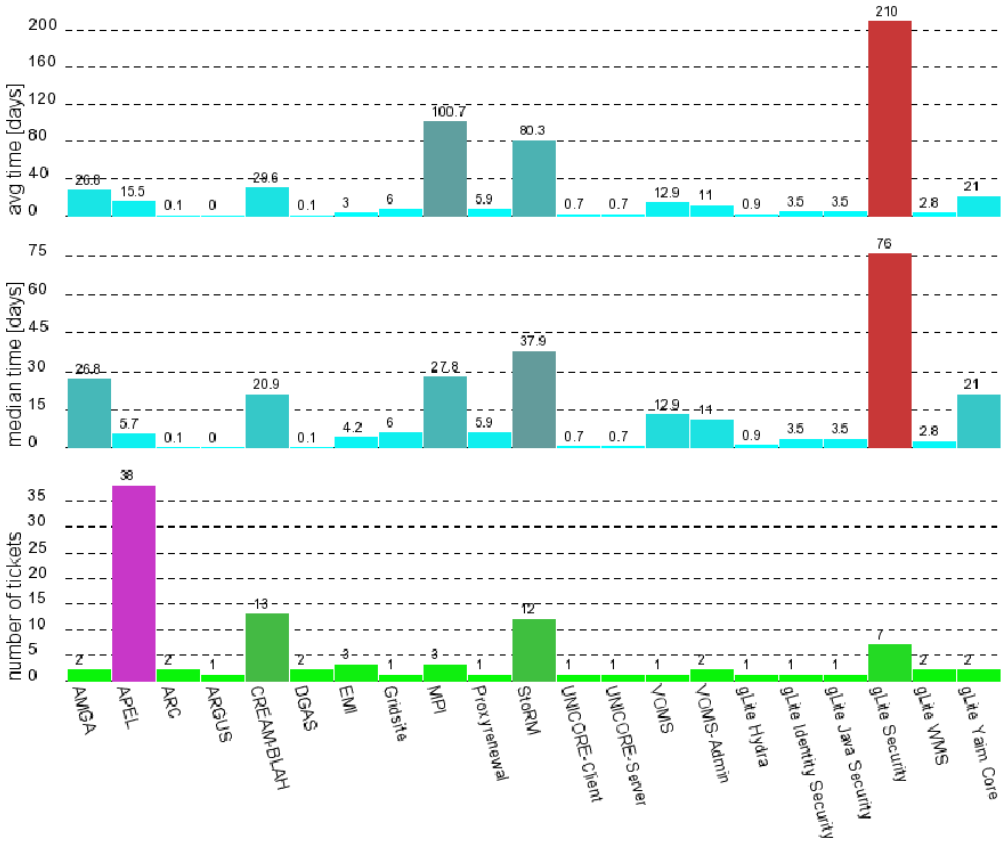
	GGUS tickets, especially for what concerns ambiguous or missing documentation.
Unit	Incident per user month
Measurement	 This graph reports the number of tickets that have been submitted in GGUS from November 2010 to January 2011 and that are still unsolved.
Thresholds/target value	Decreasing trend.
Comment	The aim of this metric would be to measure defects in training and user support processes per user month. However, obtaining a valuable measure for it is roughly complicated at the moment. According to the assumption made by the metric definition, the number of unsolved defects, as shown in the graph, should correspond to the effective number of software bugs but – unfortunately – this number is usually bigger. The adoption of different categories for tracking bugs produces inconsistent estimations. As a possible solution, a new support unit for tracking user support defects could be created and the metric definition modified accordingly.

Table 8: Training and support incident per user month – Metric

ID	**AVERAGETIMEFORUSERINCIDENTS**
Name	Average time to deal with an incident at the 3rd level of user support
Description	This metric wants to measure the effectiveness of a product team to provide 3rd level user support. The time is measured from the time the ticket reaches a PT's 3rd level support and the time the ticket is moved to the status solved or unsolved
Unit	Days
Measurement	 This graph reports the average time to solve incidents for all EMI Support Units from November 2010 to January 2011.
Thresholds/target value	Need project wide agreement.

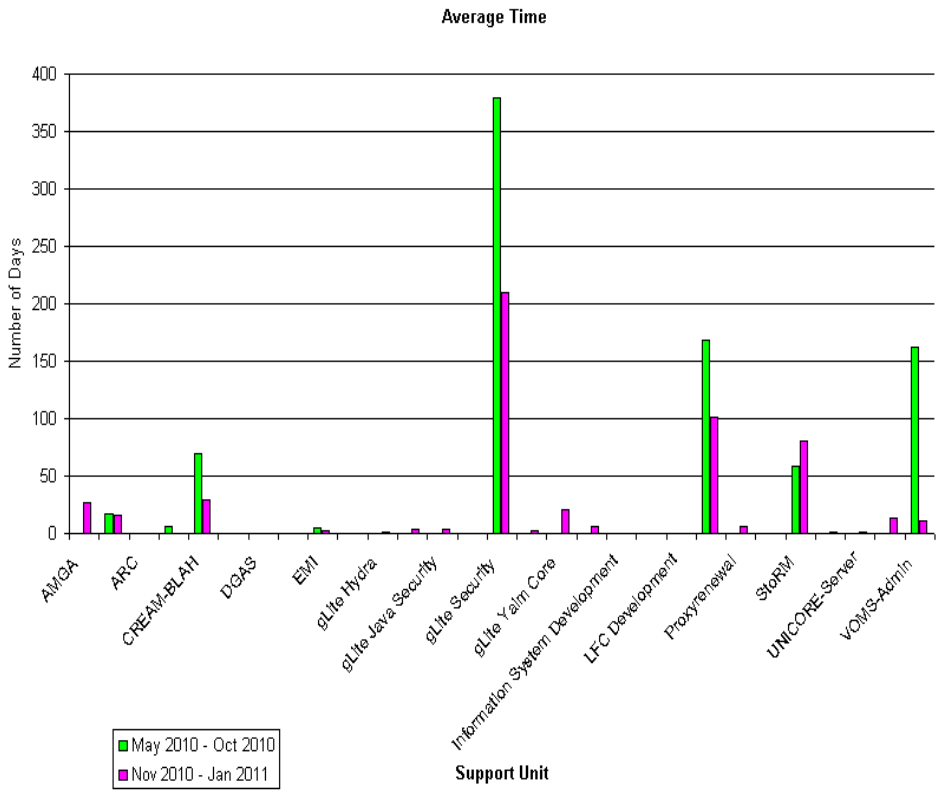
Comment	 This graph reports the comparison between the average times, in days, to solve incidents in two time periods, from May to October 2010 (green bars) and from November 2010 to January 2011 (fuchsia bars). The tickets solved in the second period could have been submitted during the first. The aim of this comparison is to understand if the average time to solve tickets is following a decreasing or increasing trend over the reporting periods. In general, it can be asserted that the time spent to resolve EMI incidents is decreasing, except, of course, for those support units that were not considered during the previous measurement taken last October 2010 (e.g. AMGA, gLite Yalm Core, etc.) as they were not properly configured in GGUS [R37].

Table 9: Average time to deal with an incident at the 3rd level of user support - Metric

Validated Changes

There are no previous change requests that could be verified for this review.

Validated Deliverables

Name of the deliverable under evaluation	Validated	
	YES	NO
Software Maintenance and Support Plan	Y	

Variations from the previous review

There are no variations to report from the previous review.

Change Requests

The change requested for this review is:

- ***clarify*** *the definition of metric "TRAININGSUPPORTINCIDENTS" and* ***define*** *the tool to gather measures for it; at the moment the information obtained from GGUS seems to be inconsistent*;

4.4. SECURITY ASSESSMENTS

The Review of the Security Assessment should check that the different stages described in the First Principles Vulnerability Assessment (FPVA) approach are being followed during the assessment of software components. More details on security assessment activity are reported in paragraph [6.].

4.4.1 Input

Quality Checklists

- *Checklist for the Review of the Security Assessment [R19].*

Quality Metrics

- *No metrics defined for this review.*

Approved Change Requests

The list of changes requested during the previous QC report, and that have been accepted by the QA team, follows:

- ***define*** *the tolerance range of positive checks for considering the deliverable validated. It is not clear how many positive checks are needed to consider the Security Assessment Plan validated;*
  - this request has been accepted by the QA team and it will be included in the next release of the SQAP.

Deliverables

- *Security Assessment Plan [R14].*

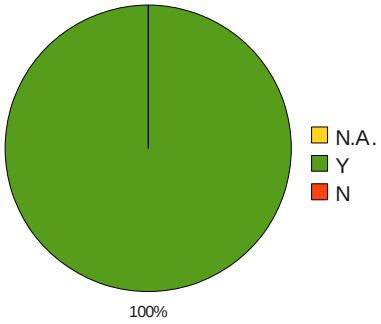
4.4.2 Output

Completed Checklist

Check Number	Question	Re-sponse
1	The Architectural Analysis has been carried out and the output contains a	Y

	diagram describing the interactions among components and end users.	
2	The Resource Identification has been carried out and the output contains the resource descriptions.	Y
3	The Trust and Privilege Analysis has been carried out and the output contains the trust levels and the delegation information for all the components and their interactions.	Y
4	The Component Evaluation has been carried out and the output contains identified vulnerabilities and their suggested fixes.	Y
5	The Dissemination of Results has been carried out.	Y

Table 10: Review of the Security Assessment Plan (N.A. = Not Available)



- 100% of checks have been executed with success.

Comments

During the review of the Security Assessment plan and its implementation, it has been verified that all stages composing the security assessment process have been performed for that component (e.g. gLExec) whose assessment has been already completed.

Currently, the progress status for the components under evaluation is as follows:

- **Argus**: its assessment is going through the *Component Evaluation* stage [R16];
- **gLExec**: its assessment has been completed and preliminary results will be disseminate soon. A preliminary list of vulnerabilities can be accessed at this link [R32].

Each intermediate stage of the security assessment process produces an artifact, namely a document describing the component internal details and its weakness points. Most of them need to be kept confidential until the entire assessment is complete; releasing information early could lead to security attacks. The only artifact that is currently possible to share is the architectural and resource diagram for the *gLExec* component (see Figure 2).

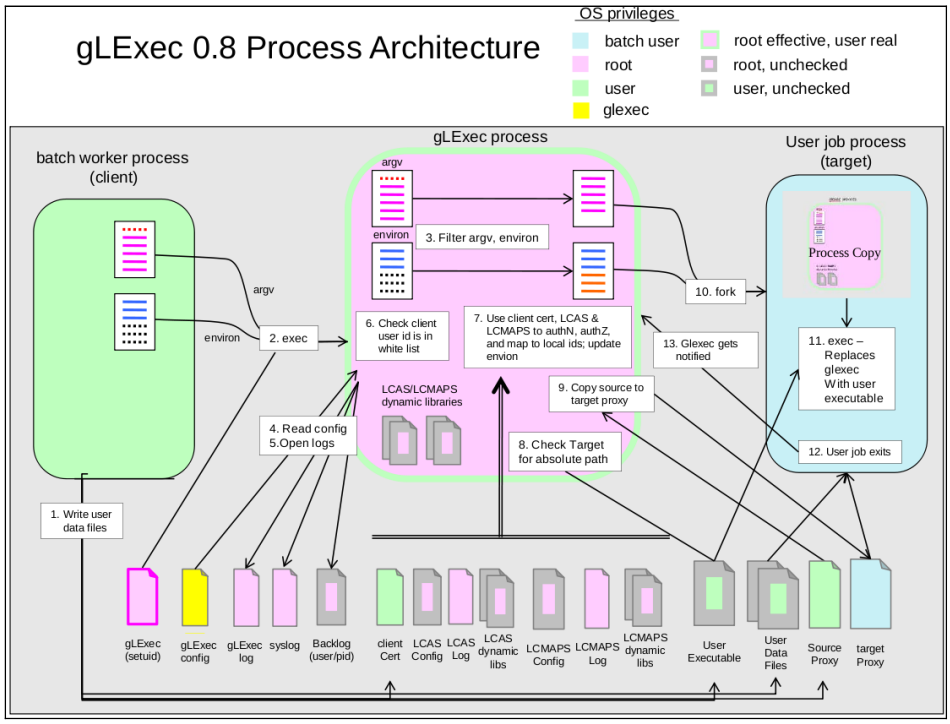


Figure 2: gLExec, architectural and resource diagram

Architectural Analysis: identify the major structural components of the system, including modules, threads, processes, and hosts. For each of these components, identify the way in which they interact, both with each other and with users. The artefact produced at this stage is a document that diagrams the structure of the system and the interactions amongst the different components and with the end users.

Resource Identification: identify the key resources accessed by each component and the operations supported on those resources. Resources include elements such as hosts, files, databases, logs, and devices. For each resource, describe its value as an end target or as an intermediate target. The artefact produced at this stage is an annotation of the architectural diagrams with resource descriptions.

Measurements

There are no measurements for this review.

Validated Changes

There are no previous change requests that require to be verified for this review.

Validated Deliverables

Name of the deliverable under evaluation	Validated	
	YES	NO
Security Assessment Plan	Y	

Variations from previous report

The previous QC report (October 2010) outlined a negative result, the Security Assessment Plan was not available at that time and all checks failed. Since then, a significant improvement has been observed, culminating in the achievements of quality objectives as well as the release of the plan.

Change Requests

No changes are requested for this review.

5. THE EMI 1 RELEASE STATUS

This section aims to give an overview of the progress status of the EMI 1 release, describing the stages of the release process, the transition from EMI 0 and EMI 1, the current status of the software development and, eventually, the degree of compliance with respect to quality acceptance criteria [R26].

5.1. THE EMI RELEASE PROCESS

According to the release plan [R5], EMI releases follow a well defined release procedure, consisting of the following steps:

1. **Release Planning** – summary of activities
   - identify requests for new features or bug fixes;
   - prioritize, plan and schedule the development and maintenance activities;
   - document and track release planning activities creating corresponding items in the User Requirements and Technical Objective Trackers and RfCs in the PTs trackers with approved priorities;
   - define the Release Schedule by creating the Components Releases items in the corresponding tracker;
2. **Release Building** – summary of activities
   - develop new features, implement required bug fixes;
   - test and certify developed components;
3. **Certification and Validation** (acceptance testing) – summary of activities
   - Component Releases are validated by the SA1 QC against the set of acceptance criteria defined by the Customers and the EMI Production Release criteria [R26];
   - components are made available for technical-preview;
   - components are deployed on the EMI test-bed for a (6 days) grace period using an automatic monitoring tool;
4. **Release Preparation and Deployment** – summary of activities
   - final packaging and release of components;

QC is mainly involved during the *Certification and Validation* stage, where real controls are performed and the compliance of software components ratified.

5.2. THE FIRST "INTERNAL" RELEASE: EMI 0

The EMI 0 (Zugspitze) packaging, making up the first EMI Reference Releases list, has been finalized at the beginning of February. Packages are now present in the EMI 0 repository [R29]. The milestone was reached late mainly due to the delay introduced to get all PTs work together and align them to new procedures and tools set out for EMI. Many PTs had to modify their packages to comply with the new way of managing external dependencies, to adapt certain components to use packages with the versions available from the OS/EPEL repositories, like *globus* and *gsoap*, to improve conformance with EMI (Fedora/Debian) packaging guidelines. Further details can be found at the dedicated "EMI 0 Activity" twiki page [R29]. That page also contains important information on the procedures, progress status, achievements, and general issues faced during the release process.

5.2.1 Lessons learned from EMI 0

The following list presents the lessons learned during the preparation of EMI 0 release. The purpose of this list is to put together any insights experienced during the preparation of EMI 0 that can be usefully applied on EMI 1. This list has been prepared with the collaboration of the Release Manager.

Lessons learned:

- PTs need to be more involved in the release process. This is a natural activity that will be necessary to carry on throughout the whole project to get PTs, each having its own background, work together;
- communications on the EMT mailing list and participation to the EMT meetings should be improved to avoid that important notifications get misinterpreted and quality procedures unattended;
- the collaboration among PTs should be improved, especially a good sharing of competencies and knowledge would be helpful during the resolution of common problems;
- QA Policy and Procedures have not been correctly acknowledged by PTs yet and are perceived as non-mandatory. As a direct consequence of this misconception, it has been already decide to replace the term guideline with the term policy;
- PTs considered the EMI 0 milestone as an "exercise" so its importance was diminished and they put poor attention in achieving it.

5.3. EMI 1

EMI 1, codename Kebnekaise, will be the first major release of the EMI middleware, established as the combination of middleware services, client and libraries. EMI 1 is being developed under the direction of the JRA1 Work Package based on the requirements and technical objectives [R7] identified by the EMI Project Technical Board (PTB) and under the general software engineering and quality assurance procedures defined by the EMI SA2 (Quality Assurance) Work Package. The following sections present the technical objectives of the EMI 1 release, the progress status for the development of the components and if the release process is in line with the schedule.

5.3.1 Progress report of component developments

The following table gives an overview of EMI 1 components, the summary of their new features, and the percentage[1] of the work performed so far to implement them. The aim is to provide a summary of changes performed and the rationale behind them. It is important to understand that part of the technical objectives are only about necessary preparation work (e.g. identification of use cases, etc.) which will be useful for later implementations.

Detailed feature list and technical objectives are contained in the overall technical workplan [R7] and tracked in the Release Schedule tracker [R34] while the developments in context of each technical objective are tracked in the development tracker [R35].

Component	Feature Summary	Status
A-REX	support of GLUE2 information providers.	75%

1 The percentages have been estimated the progress status that has been reported to JRA1 leader by each PT.

CREAM	support of GLUE2 information providers.	50%[2]
U.TSI, U.XNJS, U.UAS, WSRFLite	support of GLUE2 information providers.	90%
WMS	matchmaking module of the WMS will be enhanced to be compliant with GLUE2.	15%[3]
Libarcclient/ arc*	enhanced to work with GLUE2-compliant information.	90%
UCC	enhanced to work with GLUE2-compliant information.	100%
HILA	enhanced to work with GLUE2-compliant information.	n/a[4]
DPM	enable GLUE2.0 support support of the HTTP(S) protocol support NFS4.1 (experimental version) prototype-level support for the "file://" access protocol.	80%
dCache	enable GLUE2.0 support enable the use of HTTP(S) protocol prototype-level support for the "file://" access protocol.	95%
UAS-D	expose GLUE2.0 compliant information about storages.	90%
StoRM	enable GLUE2.0 support support of the HTTP(S) protocol prototype-level support for the "file://" access protocol.	90%

Table 11: EMI 1 Key New Component Features and progress status

5.3.2 Progress status of EMI 1 release process

The status of EMI 1 release process is presented in the table below, including the percentage of completion for each stage. These values have been calculated on the base of a subjective estimation of the work performed so far. Their update occurs on a regular basis as long as the release process proceeds. The purpose of the release process is to provide a useful 'road map' that can be used by the release manager to assist PTs in achieving project objectives with success.

ID	Name	Start	Finish	Complete
1	Release Planning	May 3 2011	Nov 1 2010	
1.1	Identify requirements for new features, bug fixes	May 3 2010	Aug 31 2010	100%

2 C**REAM**: the static and GLUE2 compliant information will be provided in EMI 1 but dynamic one might be most likely not available due to unexpected problems in its implementation.

3 Component is likely to fail in delivering a solution to satisfy this required key feature

4 **Change to DNA1.3.1** – HILA is in harmonization of client libraries process; no need to add GLUE2

1.2	Prioritize, plan and schedule the development activities	Sep 1 2010	Sep 30 2010	100%
1.3	Feature Submission Deadline	Oct 1 2010	Oct 1 2010	
1.4	Fill User Requirements, Technical Objectives & RfCs trackers	Oct 1 2010	Oct 4 2010	75%
1.5	Define Release Schedule	Oct 5 2010	Nov 1 2010	95%
2	Release Building	Oct 1 2010	Feb 28 2011	
2.1	Develop features, implement bug fixes	Oct 1 2010	Dec 15 2010	75%
2.2	Test & certify developed components	Dec 16 2010	Feb 28 2011	0%
2.3	Feature Freeze	Dec 31 2010	Dec 31 2010	
2.4	Feature Complete	Feb 1 2011	Feb 1 2011	85 %[5]
2.5	Final Change Deadline (code freeze)	Feb 28 2011	Feb 28 2011	90 %[6]
3	Certification & Validation (acceptance testing)	Mar 1 2011	Apr 19 2011	
3.1	Release Candidate	Mar 1 2011	Mar 1 2011	
3.2	CR validation	Mar 1 2011	Mar 28 2011	0%
3.3	Software components available for Tech. Preview	Mar 29 2011	Apr 11 2011	0%
3.4	CR deployment on EMI-testbed	Apr 12 2011	Apr 19 2011	0%
4	Release Preparation & Deployment	Mar 29 2011	Apr 29 2011	
4.1	Final packaging & signing of components	Apr 11 2011	Apr 15 2011	0%
4.2	Prepare & publish release documentation	Mar 29 2011	Apr 29 2011	0%
4.3	Components uploaded in the official EMI Software Repository	Apr 18 2011	Apr 19 2011	0%
4.4	Announce the Release to Customers	Apr 29 2011	Apr 29 2011	

Table 12: progress status of EMI 1 release

It is worth mentioning that this report takes a snapshot when the project is transitioning from phase 2.4 to 2.5 (highlighted in green in the table above) and its status will surely be different at the time of the release of this document.

5 **Estimation**: overall percentage and status is good; value influenced by two components not progressing well (CREAM and WMS).
6 **Estimation:** majority of code is written for features; major code writings still for porting (EPEL) and tests.

5.4. VERIFICATION OF THE SCHEDULE

The progress status for EMI 1 release (see 5.3.2) presents a roughly negative deviation from the schedule [R4], actually the reality differs a little bit from the planned baseline. The extent to which this deviation occurs is difficult to measure. Precisely, if no improvements happen during the next weeks, the probability that software components will not meet quality standards is high.. Activities (1.4, 1.5, 2.1, 2.2) are late with regards to the schedule and result still open.

5.5. VERIFICATION OF THE COMPLIANCE WITH THE RELEASE PLAN PROCEDURES

As reported in the release process (see 5.1.), all user and technical objectives should be reported in form of RfC in the EMI tracking tool (Savannah). Even if the number of technical objectives and developments is relevant (see R7), the few RfCs currently linked in Savannah seem to not reflect the real status of the development activities and the impact that the technical objectives are having on software components.

The table below reports the list of components that at the time of this report comply with the release management policy [R38]; basically the list of RfCs is available for each of them and the corresponding description in maintained on a specific trackers. This verification was taken at end of January 2011.

Item ID	Summary	Assigned to
#18734	UNICORE security libraries release v.2.0.0	emi-rel-sched-unicore
#18733	UNICORE AIP release v.2.0.0	emi-rel-sched-unicore
#18732	UNICORE XACML PDP release v.2.0.0	emi-rel-sched-unicore
#18731	UVOS release v.1.4.0	emi-rel-sched-unicore
#18729	UNICORE Gateway release v.6.4.0	emi-rel-sched-unicore
#18728	UNICORE/X release v.6.4.0	emi-rel-sched-unicore
#18685	jobwrapper release v.3.3	emi-rel-sched-jobman
#18684	CEMon release v.1.13	emi-rel-sched-jobman
#18683	BLAH release v.1.16	emi-rel-sched-jobman
#18587	VOMS release v.2.0.0	emi-rel-sched-voms
#18575	UNICORE XNJS release v.1.4.0	emi-rel-sched-unicore
#18568	WMS release v.3.3.0	emi-rel-sched-jobman
#18534	CREAM release v.1.13	emi-rel-sched-jobman

Table 13: components accompanied with RfCs

Among 89 components, only 13 of them contain the list of corresponding RfCs.

5.6. *EMI 1 RELEASE DATA FORECAST*

At this moment the prevision is that the release date (29 April) will be met. For this reason, in this period, the PTs should concentrate all their efforts on components packaging, testing and certification, in order to be able to provide to SA1 release candidates that meet the production release criteria [R26]

6. STATUS OF THE SECURITY ASSESSMENT ACTIVITY

The middleware security and testing groups of the University of Wisconsin (UWM) and Universitat Autònoma de Barcelona (UAB) have developed and are continuing to develop the First Principles Vulnerability Assessment (FPVA) methodology for assessing software for critical vulnerabilities. FPVA is a primarily analyst-centric (manual) approach to assessment whose aim is to focus the analyst's attention on the parts of the software system and its resources that are mostly likely to contain vulnerabilities. FPVA is designed to find new threats to a system. It's not dependent on a list of known threats.

Assessments of several major middleware systems have been carried out, significant vulnerabilities found in many of them, and the developers helped with remediation strategies. FPVA is being applied to various security related middleware packages supplied by EMI as part of the SA1 Quality Control process.

A Security Assessment Plan was discussed and agreed with the EMI software providers. As FPVA is a manual methodology, it is slow and we can only estimate how much time we will need to assess a software package. And those estimations may experience changes depending on the length and complexity of the piece of software to assess. Nevertheless we defined a scheduling for the assessments to be carried out for EMI. It included mainly 2.5 years and 6 pieces of software, and the remaining 6 months were left for re-assessing the same software after the reported vulnerabilities will be fixed. The last version of the Security Assessment plan is available at this link [R25], while the reference activity page is available here [R27].

Of the EMI components, so far VOMS Admin 2.0.15 has been assessed using FPVA. Serious vulnerabilities were found and reported to the development team, together with possible fixes. The development team is currently working on fixing the vulnerabilities found. The vulnerabilities are not disclosed yet, but will be after they are fixed and the different user groups have had time to update to the new security release.

The picture below presents one of the artifacts produced during the assessment of VOMS Admin component. More precisely, it gives an overview of the system architecture, describing how various components interact together and which are their privileges.

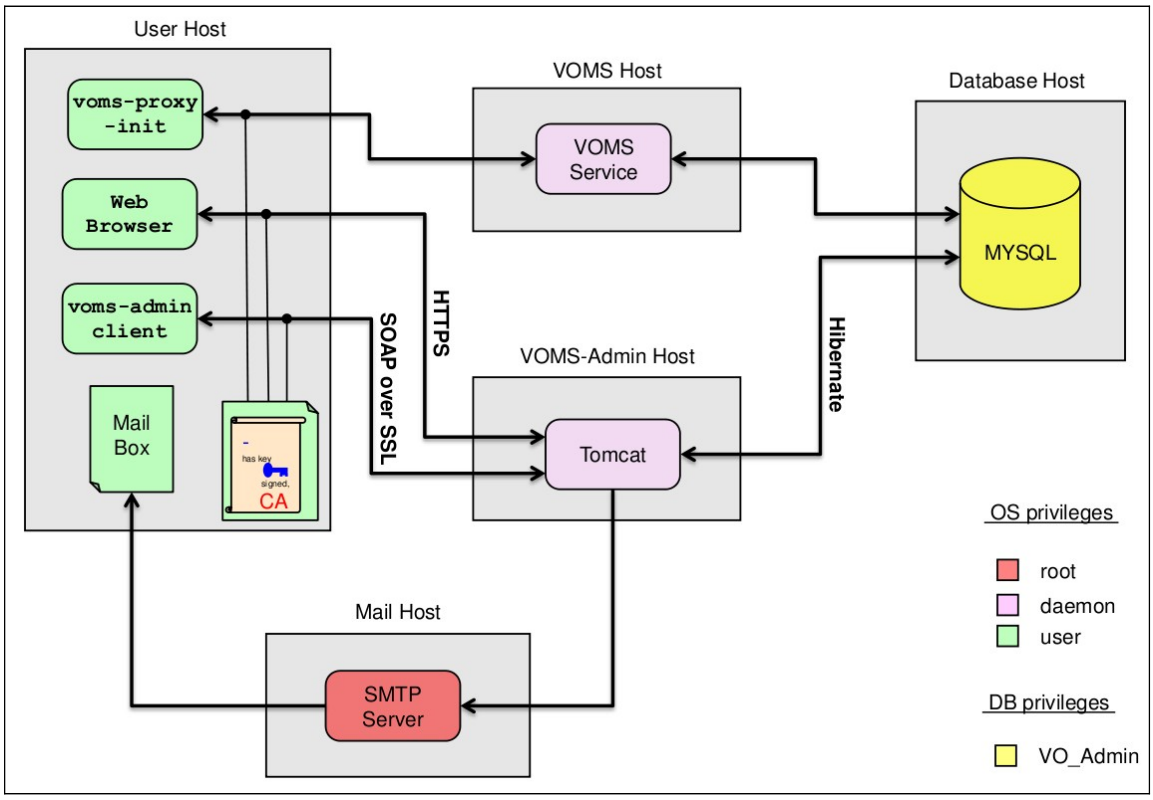


Figure 3: VOMS Admin – system architecture and privileges analysis

Currently Argus[7] 1.2 and gLExec[8] 0.8 are being assessed. These assessments are expected to be finished at the end of March 2011. Argus is being assessed by Manuel Brugnoli from the UAB and gLExec by Daniel Crowell from the UWM. An independent assessment is a key issue in security, so we keep an absolute discipline, which means that no details on the current assessment can be leaked. There is one way communication between the assessment and development teams. That means that the assessment team asks question to the development team, but does not provide information about the assessment process. The outcome of the assessment will be vulnerability reports to be delivered to the person responsible for the piece of software being assessed. Some vulnerability reports for already assessed components can be accessed here [R24].

7 **Argus** is the gLite Authorization Service. It is intended to provide consistent authorization decisions for distributed services (e.g. compute elements, portals). Argus consists of 3 main components: (a) the Policy Administration Point (PAP) to manage policies; (b) the Policy Decision Point (PDP) to evaluate policies; and (c) the Policy Enforcement Point (PEP) to enforce policy decisions.

8 **gLExec** provides and identity mapping service. It has been assessed in the past and has since undergone a rewrite, mainly to address some of the problems found by these assessments. It is necessary to re-assess the new version, and this is already in progress.

7. STATUS OF THE TEST

This paragraph reports the status of testing activity, including the availability of test plans for software components and regression tests for fixed bugs. According to the software release plan [R5], the test plans should have been provided 4 months before the release of the software. Being the release of EMI 1 planned for the end of April 2011, the deadline for providing test plans was December 2010. The list of test plans is available at this link [R30]. PTs have been informed about this deadline and encouraged to link their plans through announcements made during EMT meetings.

7.1. TEST PLANS

Currently, not all the components have been associated to a test plan as it is shown on the corresponding Wiki page set out for collecting this information [R30]. On that page only 37 entries are present, while, according to the release tracker, Savannah, they should be 89, namely one for each component. Among those 37 entries, only 29 are really linked to a test plan, but none of them have been reported in the release tracker (Savannah), failing to comply with what is stated in the software release plan. In the following table, the status of test plans availability is reported.

Component name	Test plan	Component name	Test plan
AMGA	Yes	UNICORE TSI	Yes
Apel	Yes	UNICORE XNJS	Yes
Argus	Yes	UNICORE UAS (Job Management)	Yes
BDII	Yes	UNICORE Service Registry	Yes
BLAH	Yes	UNICORE UAS (Data Management)	Yes
CREAM and CEMon	Yes	UNICORE Gateway	Yes
DPM	Yes	UVOS	Yes
FTS	Yes	XUUDB	Yes
Hydra	Yes	UNICORE PDP (XACML entity)	Yes
LB	Yes	UNICORE AIP (authz data providers)	Yes
LFC	Yes	UNICORE Security Libraries	Yes
VOMS	Yes	dCache	Yes
WMS	Yes	DGAS	NO
A-Rex	NO	MPI	Yes
ARC Compute Elements	NO	SAGA	Yes
ARC Data Clientlibs	NO	StoRM	NO
ARC Infosys	NO	GridSite	Yes
ARC Security utils	NO	ProxyRenewal	Yes
ARC Container	NO		

Table 14: components accompanied with test plans

If a component is marked with “Yes”, it means that the corresponding test plan is available and complies with the EMI Test Template defined by QA [R13].

7.2. REGRESSION TESTS

Regression tests are useful to retest a previously tested program following modification to ensure that faults have not been introduced as a result of the changes made especially for fixing bugs. When a new regression test is implemented, its output must be documented and included in a report. As set out in the Certification and Testing policy [R6], all regression test reports should be linked in the Release Tracking Tool [R34] and maintained in respective middleware trackers.

The certification and testing activity (see R5 for further details) officially ends at the end of February 2011 and official information will be only made available at that date.

The table below reports a preliminary list of components which some testing trials have been performed with. Regression tests for affecting bugs have been executed with success and sample reports including their results linked to the respective task in the release tracker [R34]. The reports are maintained on middleware specific trackers and public available.

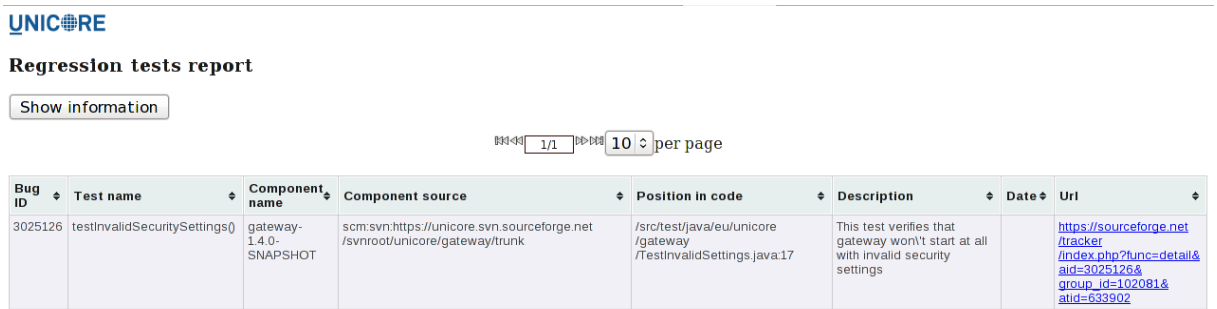
Item ID	Summary	Assigned to
#18734	UNICORE security libraries release v.2.0.0	emi-rel-sched-unicore
#18733	UNICORE AIP release v.2.0.0	emi-rel-sched-unicore
#18732	UNICORE XACML PDP release v.2.0.0	emi-rel-sched-unicore
#18731	UVOS release v.1.4.0	emi-rel-sched-unicore
#18730	XUUDB release v.1.3.1	emi-rel-sched-unicore
#18729	UNICORE Gateway release v.6.4.0	emi-rel-sched-unicore

Table 15: components accompanied with test reports on January 2011

Inside each test report there is a section on regression tests. According to the QA policies, that section should contain a small set of data:

- RfC unique ID;
- description of the test that will prove that the RfC has been properly implemented and that the problem it fixes is indeed not present any more in the component;
- input needed for the test;
- criteria for considering the test successful (PASS) or failed (FAIL).

As example of generated report, in the following a test report for the UNICORE Gateway component is presented:


UNICORE

Regression tests report

Show information

1/1 10 per page

Bug ID	Test name	Component name	Component source	Position in code	Description	Date	Url
3025126	testInvalidSecuritySettings()	gateway-1.4.0-SNAPSHOT	scm:svn:https://unicore.svn.sourceforge.net/svnroot/unicore/gateway/trunk	/src/test/java/eu/unicore/gateway/TestInvalidSettings.java:17	This test verifies that gateway won\'t start at all with invalid security settings		https://sourceforge.net/tracker/index.php?func=detail&aid=3025126&group_id=102081&atid=633902

Figure 4: sample regression test report for the UNICORE Gateway

In this case the input for the test is included in the Java test class code ("Position in the code" field) and the successful criteria is the successful execution of the test, only reports for successful tests are generated. In the near future also other middleware components will be tested and listed.

8. CONCLUSIONS

This document reports the organization of the QC activity in SA1 and the results of quality activities performed after nine months of project work.

While the execution of periodic reviews (see chapter 4.) has reported notable results, as well as the security assessment activity is progressing with success, the release process is performing slightly under the expected level and acceptance criteria are not completely met at the moment. Although final considerations could not be taken now, the release process is still progressing and improvements are still possible, the number of PTs satisfying release procedures could not be considered satisfactory. Since there is a decisive distinction from who is fully complaint and who is totally not, one could conclude that, on one hand, the quality procedures are feasible to apply, on the other, that PTs can reach the compliance level only if start-up conditions are present. For instance, if a PT was adopting a formalised release procedure, strict quality standards or similar procedures before joining EMI, the possibility for it to be still complaint would be much higher with respect to its colleagues.

According to results reported in this document, it is evident that many PTs are not prepared to comply with quality procedures, or did not correctly estimate the effort needed to implement them with the consequence of being late in the release process. Most of them were not used to have formal quality procedures and this aspect was underestimated during the planning phase of the project. However the goal is to have quality procedures fully followed across the three years of the project and not already on the first year.

On the QA side, effort is still needed to finalize quality policies, get them approved by PEB and set up appropriate tools for helping QC in performing checks. An in-reach training event has been organized to cover topics of interest to members of the EMI. During the event, scheduled at the beginning of March 2011, EMI policies will be deeply explained to PTs and real user-scenario analyzed. In the meantime, with the collaboration of the Release Manager, the QC task will keep monitoring the progress of the release process and report any-nonconformity during the upcoming Certification and Validation stage (see paragraph 5.1.). The QC will not enforce any component to comply with policy for being included in the final release; only reports and measurements will be reported to project's boards. Whether discovered non-conformity is blocking or not, it is the PEB to take the final decision. However, PTs will be asked to explain why they do not follow the policy and by when they will comply.