EUROPEAN MIDDLEWARE INITIATIVE

e-infrastructure

EUROPEAN MIDDLEWARE INITIATIVE

SOFTWARE MAINTENANCE AND SUPPORT PLAN

EU DELIVERABLE: DSA1.1

Document identifier:	**EMI-DSA1.1-1277556-SoftwareMaintenanceAndSupportPlan-v0.9.odt**
Date:	
Activity:	**SA1**
Lead Partner:	**INFN**
Document status:	**DRAFT**
Document link:	**http://cdsweb.cern.ch/record/1277556**

Abstract:
This document describes the Software Maintenance and Support processes, the roles and responsibilities and the main metrics to be used for the Service Level Agreements.

Delivery Slip

	Name	Partner / Activity	Date	Signature
From				
Reviewed by				
Approved by				

Document Log

Issue	Date	Comment	Author / Partner
1	2010-10-06	Ready for Review	Francesco Giacomini/INFN
2	2010-10-20	Addressed comments by Mathilde Romberg and Morris Riedel	Francesco Giacomini/INFN
3	2010-11-28	Addressed comments by Maria Alandes Pradillo and Jozef Cernak	Francesco Giacomini/INFN
4	2011-01-27	Addressed comments by Alberto Aimar	Francesco Giacomini/INFN

Document Change Record

Issue	Item	Reason for Change
1		
2		
3		

TABLE OF CONTENTS

Table of Contents

1. INTRODUCTION

1.1. PURPOSE

The purpose of this document is to present the fundamental principles that will guide the Software Maintenance and the User Support tasks within the project.

1.2. DOCUMENT ORGANISATION

Besides this Introduction, the Executive Summary and the final Conclusions, the document includes two main sections.

Section 3, Software Maintenance, introduces the approach that the EMI project adopts for the maintenance of the software components that will be included in the EMI distribution, with a strong focus on the preservation of the stability of what is deployed in a production environment. The different types of releases are described and the management of Requests for Change is explained.

Section 4, User Support, introduces the approach that the EMI project, in collaboration with EGI, adopts for the support of users of the software components that will be included in the EMI distribution.

For both Software Maintenance and User Support, relevant Key Performance Indicators (KPI) are presented, together with operative hints on how the information needed to compute them should be collected.

1.3. REFERENCES

R 1	Information Technology Infrastructure Library (ITIL), http://www.itil-officialsite.com/home/home.asp
R 2	EMI Requirements tracker, https://savannah.cern.ch/projects/emi-req/
R 3	ARC RfC tracker, http://bugzilla.nordugrid.org/
R 4	gLite RfC tracker, https://savannah.cern.ch/projects/jra1mdw/
R 5	UNICORE RfC tracker, http://sourceforge.net/projects/unicore/
R 6	Technical Development Plan, DNA1.3.1, http://cdsweb.cern.ch/record/1277540
R 7	EMI Description of Work, https://twiki.cern.ch/twiki/pub/EMI/EmiDocuments/EMI-Part_B_20100624-PUBLIC.pdf
R 8	Service Level Agreement Template, DNA1.2.1, http://cdsweb.cern.ch/record/1277517
R 9	GGUS, https://gus.fzk.de/pages/home.php
R 10	GGUS documentation, https://gus.fzk.de/pages/docu.php
R 11	EMI Software Quality Assurance Plan, DSA2.1, http://cdsweb.cern.ch/record/1277599

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.

1.4. TERMINOLOGY

ABI	Application Binary Interface
API	Application Programming Interface
CDS	CERN Document Server
DCI	Distributed Computing Infrastructure
DMSU	Deployed Middleware Support Unit
DoW	Description of Work
EGI	European Grid Infrastructure
EMT	Engineering Management Team
ETICS	eInfrastructure for Testing, Integration and Configuration of Software
GGUS	Global Grid User Support
ITIL	IT Infrastructure Library
KPI	Key Performance Indicator
kSLOC	Kilo Source Lines Of Code
TCB	Technical Coordination Board
NGI	National Grid Initiative
PEB	Project Executive Board
PTB	Project Technical Board
RfC	Request for Change
SLA	Service Level Agreement
SU	Support Unit
TPM	Ticket Processing Management
VCS	Version Control System
WSDL	Web Service Description Language

2. EXECUTIVE SUMMARY

This document presents the fundamental principles that will guide the Software Maintenance and the User Support tasks within the project.

The Software Maintenance task is responsible to coordinate the continuous maintenance of the middleware components developed within the project and included in an EMI distribution, preserving at the same time their stability in terms of interface and behavior, so that higher-level frameworks and applications can rely on them.

Once a year the project will produce a major release of the EMI distribution, characterized by a well-defined interface and behavior for each of its components. Interface and behavior are allowed to change within a major release of the distribution only in a backwards-compatible way. Component releases are classified in major, minor, revision and emergency, based on the impact of the changes on the component interface and behavior.

Requests for Change will be managed adopting a priority-driven approach, so that the risk to compromise the stability of the software deployed in a production environment is minimized. Requests for Change will also be properly monitored across the different trackers adopted by Product Teams.

The User Support task is responsible to coordinate the support, together with EGI, to users of the middleware components developed within the project and included in an EMI distribution. The User Support is organized in three levels, of which only the third one is within the EMI project and provides the most specialized knowledge needed to investigate a reported incident. Many Support Units, corresponding approximately to the products delivered by EMI, are established and registered on the reference support portal (GGUS).

Both the Software Maintenance and the User Support tasks are monitored through Key Performance Indicators, that are presented at the end of the respective sections.

3. SOFTWARE MAINTENANCE

The Software Maintenance task in SA1 is responsible to coordinate the continuous maintenance of the middleware components developed within the project and included in an EMI distribution, preserving at the same time their stability in terms of interface and behavior, so that higher-level frameworks and applications can rely on them.

The term interface is intended in a broad sense. Interfaces include, but are not limited to, APIs, ABIs, WSDLs, DataBase schemas, network protocols, authentication and authorization mechanisms, logging formats, packaging and other deployment characteristics of a component.

A strong constraint of the software maintenance activity is that backwards-incompatible changes can be introduced in a production environment only in a strictly controlled way, according to the project guidelines as in the EMI Description of Work.

3.1. EMI RELEASES

The EMI distribution will be organized in periodic major releases, tentatively delivered once a year (corresponding to milestones MSA1.2.2, MSA1.2.2 and MSA1.2.4 - “EMI Reference Releases”, due respectively in April 2011, April 2012 and April 2013), providing a good balance between the conflicting requirements of stability and innovation.

An EMI major release is characterized by well-defined interfaces, behavior and dependencies for all included components, available on a predefined set of platforms. What is included in a new EMI major release is defined by the PTB and included in the yearly Technical Development Plan (corresponding to deliverables DNA1.3.1, DNA1.3.2 and DNA1.3.3) and the implementation of the plan is coordinated by JRA1 (corresponding to milestones MJRA1.19.1, MJRA1.19.2 and MJRA1.19.3, “Integrated EMI Major Release Candidates”).

Backward-incompatible changes to the interface or to the behavior of a component that is part of the EMI distribution can be introduced only in a new EMI major release. Changes to interfaces that are visible outside the node where the component runs (e.g. a WSDL or a network protocol) need to be preserved even across major releases, according to end-of-life policies to be defined on a case-by-case basis.

The availability of a new major release of EMI does not automatically obsolete the previous ones and multiple major releases may be supported at the same time according to their negotiated end-of-life policies (see Section 4.5).

3.2. COMPONENT RELEASES

An EMI distribution includes all the components that are developed within the project and that have reached production quality. Within an EMI major release, only one version of a given component is maintained.

Four types of releases have been identified for a given component:

- **Major Release**

  A major release for a component is characterized by a well-defined interface and behaviour, potentially incompatible with the interface or behaviour of a previous release.

  New major releases of a component can be introduced only in a new major release of EMI.

  The contents of a new major release are endorsed by the PTB and included in the project Technical Development Plan. The implementation is coordinated by JRA1 and described in the JRA1 deliverables.

- **Minor Release**

  A minor release of a component includes significant interface or behaviour changes that are backwards-compatible with those of the corresponding major release.

  New minor releases of a component can be introduced in an existing major release of EMI.

  The contents of a new minor release are endorsed by the PTB and included in the project Technical Development Plan. The implementation is coordinated by JRA1. If the release is going to be introduced in an existing major release of EMI, the implementation is also supervised by the Release Manager (within SA1) in order to guarantee that the production quality and the backwards-compatibility are preserved.

- **Revision Releases**

  A revision release of a component includes changes fixing specific defects found in production and represents the typical kind of release of a component during the lifetime of an EMI major release.

- **Emergency Releases**

  An emergency release of a component includes changes fixing only Immediate-priority defects found in production, typically security-related.

The type of release is reflected in the version of the corresponding package(s) and will be described in the Software Release Plan, DSA2.1.

3.3. INCIDENTS, PROBLEMS, CHANGES

SA1 is in general responsible for *corrective* and *adaptive* maintenance to address defects, potential defects and minor improvements found in running services in the production environments, based on requests for changes (RfC) in the code of EMI software components.

ITIL [R 1] defines a **change** as the addition, modification or removal of authorized, planned or supported service or service component and its associated documentation.

One of the main sources of RfCs are the incidents reported by users through the support channels, notably GGUS. ITIL defines an **incident** as an unplanned interruption to an IT service or reduction in the quality of an IT service. Failure of a component that has not yet affected service is also an incident.

After investigation by the different levels of support (see Section 4.1), an incident may be traced to an actual problem in the code. ITIL defines a **problem** as the cause of one or more incidents .

If that is the case, the problem is recorded into an RfC tracker (e.g. Savannah, Bugzilla, RT, etc.) and further processed by the team responsible for the affected component, usually leading to changes in the code.

Another major source of RfCs is the continuous stream of user requirements, notably from the EGI TCB and similar bodies. Requirements are collected and processed by the PTB and addressed by JRA1. They are maintained in a dedicated tracker [R 2]

Finally, RfCs, both to fix defects and to introduce improvements, can also be generated internally in the project or even in the Product Team itself in charge of a given component.

3.4. PRIORITY-DRIVEN DEVELOPMENT

Since SA1 addresses only corrective and adaptive maintenance, it is necessary to define how to select RfCs that qualify as corrective or adaptive.

The criterion for such classification is the priority of the RfC, where the priority is the result of the composition of a number of factors:

- **severity**: a measure of the degradation of the quality of service of the affected component;
- **impact**: a measure of the effect of the degradation of the quality of service of the affected component;
- **urgency**: a measure of how long it will be until the quality of service of the affected component is not significantly degraded;
- **cost**: a measure of the resources needed for the management of the change, including the risks associated to the degradation of the quality of the affected component in case the change is not fully successful.

The evaluation of the priority of an RfC results in one of four possible logical values. Each level implies a very specific behavior for the management of that RfC.

The four priority levels and the corresponding behaviors are:

- **Immediate**

  The RfC needs to be addressed as soon as possible, in all affected EMI major releases.

  A release containing Immediate-priority changes can contain only Immediate-priority changes. Multiple Immediate-priority changes can be included in the same release, provided that any change does not delay the release significantly.

  This constraint minimizes the risk of introducing new defects in the new release of the affected component and allows the adoption of special accelerated procedures for its release. Moreover it will not force site administrators to deploy lower-priority changes during an emergency update.

- **High**

  The RfC will be addressed in a next release of the affected component, in all affected EMI major releases.

- **Medium**

  The RfC will be addressed in the release of the affected component that will be shipped with the next EMI major release.

- **Low**

  There is no target date for addressing the RfC.

Each logical value needs to be mapped to a specific value in the tracker used by each PT.

The priority of an RfC can be suggested by the proponent, but the PTB is the ultimate responsible for setting it.

3.5. TRACKING OF CHANGE REQUESTS

Each RfC needs to be tracked with an appropriate tool. Each PT is free to choose the one to use for the product or products it delivers, provided that the tool satisfies the constraints described in this section. Currently used trackers include Bugzilla (by ARC [R 3]), Savannah (by most of gLite [R 4]) and Sourceforge (by UNICORE [R 5]).

For each RfC the following information should be available or recorded:

- a URL pointing to a description of the RfC. If the RfC concerns a security vulnerability the URL should point to a private page. The URL acts also as a unique identifier for the RfC;
- if applicable, one or more URLs pointing to GGUS tickets (or equivalent incident reports) that have caused the opening of this RfC;
- the affected component (the list of components is included in the Technical Development Plan, DNA1.3.1 [R 6]);
- the EMI major release(s) including a version of the component that is affected;
- whether the RfC is platform-specific and, if so, which are the affected platforms;
- the priority, according to the classification in Section 3.4;
- whether the RfC is to fix a defect or to introduce a new feature;
- the state of the RfC and the date the RfC has entered each state. The minimal set of logical states is: Open (just submitted), Accepted (assessment has been done and RfC accepted), Fixed (change committed to the VCS), Closed;
- for Closed RfC, the resolution (solved/won't fix/unreproducible/invalid/etc.);
- the detection area, that is the context in which the version of the component affected by the change is available. Possible values are “production”, “testing” and “development”;
- whether the change has been verified during testing. If it has been, the report for the test should be included in the documentation that accompanies the release of the affected component. Likewise the lack of verification should be justified in the documentation;
- the target component release where the change will be available;
- the target EMI major release (or ARC, dCache, gLite or UNICORE major releases in the transition period) where the change will be available;
- whether the RfC has already been approved.

If the same RfC is going to be applied to different component releases or to different EMI major releases, then an RfC is created for each release.

An Open RfC should be assessed within two weeks and as a result it should either be Accepted or Closed with a resolution different from solved;

An Accepted RfC whose priority is Immediate or High should be immediately associated to a next release of the affected component.

The above information will be collected periodically, at least once a week, to allow the Software Maintenance task leader to monitor the progress of the task. Any anomaly will be brought to the attention of the SA1 leader and if necessary of the PEB. Each PT is responsible to report RfC information according to the format defined by SA2.

3.6. ROLES

The assessment and the approval of all the RfCs concerning EMI components are a responsibility of the *Change Advisory Board*, formed by the PTB members and by the SA1 leader. The assessment includes specifically the determination of the priority of an RfC and its association with the EMI major release(s) where it will be implemented. The PTB can delegate the decisions concerning corrective and adaptive maintenance to the EMT. No RfC can be introduced in production releases without approval.

The role of *Change Manager*, i.e. following the process of controlling the life-cycle of approved changes, is taken either by the SA1 Maintenance task leader or by the JRA1 leader, depending on whether that RfC is for a defect or for a new feature.

3.7. KEY PERFORMANCE INDICATORS

The KPIs mentioned in the Description of Work [R 7] and relevant for the SA1 Maintenance Task are:

- **KSA1.3 - Number of Problems**

  Number and trends of problems (defects) submitted in the Defect Tracker(s) (in total and per category) as absolute value and as density over kSLOC .

  Information on the number and categories of problems is extracted from the RfC reports. Information on kSLOC of components is computed using the *sloccount* tool, which is already integrated in ETICS. For each component multiple codebases could be considered if that component is available in multiple EMI major releases.

- **KSA1.4 - Number of Urgent Changes**

  Number of changes (defects or enhancements) with priority Immediate.

  Information on the number, type and priority of RfCs is extracted from the RfC reports.

- **KSA1.5 - Change Application Time**

  Average time, from incident submission to release, for applying a change (possibly per category and priority).

  The submission time of an incident causing the opening of an RfC is available in the incident report (typically a GGUS ticket). A reference to the GGUS ticket and the rest of the information required to compute this metric - the time the RfC is closed, categories and priority - are available in the RfC reports.

The KPIs are continuously collected and reported by the TSA2.3 task on Metrics and KPIs Definition and Reporting . They will be monitored by the Software Maintenance task leader and any anomaly will be brought to the attention of the SA1 leader and if necessary of the PEB.

4. USER SUPPORT

The User Support task in SA1 (TSA1.5) is responsible to coordinate the support, together with EGI, to users of the middleware components developed within the project and included in an EMI distribution. Support activities will be regulated by Service Level Agreements with EMI customers. A template to be used to negotiate specific SLAs with external customers is included in DNA1.2.1 [R 8].

Although support requests can concern anything from documentation to configuration, from receiving advice to asking for a new feature, the following description will concentrate mainly on incidents, because that is the type of request that will involve the EMI support task.

4.1. SUPPORT MODEL

The EMI support model integrates in the overall support structure adopted in EGI, which foresees an organization in three levels:

1. The EGI Helpdesk represents the main contact point for a user where to get support. Within the Helpdesk the Ticket Processing Management (TPM) is responsible for the monitoring and routing of all active tickets to the appropriate support units (SUs).

   In EGI the Helpdesk is a distributed infrastructure consisting of a central Helpdesk interconnected with a collection of local NGI Helpdesks.

   If the Helpdesk is unable to resolve the incident, this is escalated for further investigation to a 2nd-level support unit.

2. The Deployed Middleware Support Unit (DMSU) ensures the availability of more specialized skills than those offered by the Helpdesk in the investigation and resolution of incidents. The DMSU includes people that together can cover all middleware areas: job and compute management, data management, security, information systems, accounting, etc.

   The DMSU is an integral part of EGI.

   If the DMSU is unable to resolve the incident, this is escalated for further investigation to a 3rd-level SU.

3. 3rd-level SUs offer the most specialized skills needed for the investigation and resolution of an incident and are typically represented by the developers of the affected software component.

   3rd-level SUs are not normally part of EGI but are integrated in the organization of the software providers, such as EMI.

This industry-standard model provides the most effective use of resources, for it involves the ultimate technical experts only when their detailed knowledge is indispensable for the investigation of an incident.

Support tickets should not normally flow from the Helpdesk directly to the EMI SUs, unless it is evident that the incident is caused by a software problem.

The tool adopted by EGI to track support requests is GGUS [R 9]. Incidents occurring to users on the production infrastructure, even if initially reported through other means, typically mailing lists, should always be reported through GGUS and their processing tracked through GGUS tickets. This would allow to compute user-oriented metrics completely from GGUS data.

Different support models for other DCIs adopting EMI components will be evaluated on a case-by-case basis.

4.2. EMI SUPPORT UNITS

Within EMI many SUs are established. Their exact number and scope can change during the course of the project, but approximately an SU is registered in GGUS for each software product that is visible to the 1st- and 2nd-level support units in EGI.

Since multiple software products can be under the responsibility of the same Product Team, it may happen that the memberships to two or more SUs are in fact overlapping or even coincident.

Upon registration in GGUS, an SU should provide:

- an e-mail address that will be notified when tickets are escalated to 3rd-level support;
- a FAQ describing the SU, according to the template provided by GGUS [R 10].

The internal organization of an SU is left to the responsibility of the corresponding PT, provided it is adequate to satisfy the SLAs established between EMI and EGI.

The following metrics will be computed for every EMI SU:

M1) Number of incidents per week

M2) Incident resolution time

The flow of tickets and the above metrics will be monitored by the User Support task leader and any anomaly will be brought to the attention of the SA1 leader and if necessary of the PEB.

4.3. THE GENERIC SUPPORT UNIT

Aside the product-specific SUs, a generic EMI SU is also established, whose purpose is to intercept and quickly re-assign all GGUS tickets for which the EGI support units are not able to properly identify a specific EMI SU.

The organization of the generic SU is under the responsibility of the SA1 User Support task. Considering that a) historically the number of GGUS tickets that require the intervention of the 3rd level is relatively low and b) the assignment of a ticket to the generic SU should be in turn an exceptional situation, the organization of the generic SU should be lightweight. Initially it will simply consist of a mailing list (emi-support@eu-emi.eu) that will re-direct all ticket notifications to the specific support mailing list of ARC, dCache, gLite and UNICORE. It is then expected that subscribers to those lists will process the tickets and re-assign them to the specific SUs. If needed, particularly complex issues can be discussed within the EMT.

The tickets arriving at the generic SU will be properly monitored to guarantee that they are promptly re-assigned. If not, the problematic ticket should be brought to the attention of the EMT.

The following metrics will be computed for the generic SU:

A) Number of incidents per week

B) Re-assignment time

The flow of tickets and the above metrics will be monitored by the User Support task leader and any anomaly will be brought to the attention of the SA1 leader and if necessary of the PEB.

If, contrary to the expectations, A) or B) are consistently high, the organization of the generic SU and the relationship between the EGI DMSU and the EMI specific SUs will be reviewed.

4.4. INCIDENT RESOLUTION

In case an incident is reported, the goal of the user support activity is to restore normal service operation as quickly as possible, thus ensuring that the best possible levels of service quality and

availability are maintained. What "normal service operation" means is defined in the Service Level Agreement (SLA) established between EMI and the infrastructure deploying EMI components.

An incident may or may not be caused by a problem. If it is, a corresponding entry shall be created by the SU in the RfC tracking tool specific to the affected software product and the two should be cross-linked. In a GGUS ticket this is done using the "Related issue" field.

The incident should stay open until a satisfactory (for the user) solution is found. The solution may not necessarily require that the causing problem is fully fixed, for example an acceptable (according to the SLA) workaround may exist. If the incident resolution instead does require that fix, then the ticket should stay open until the RfC corresponding to the fix is Closed, i.e. it becomes part of a new product release.

4.5. SUPPORT TIMELINE

It is foreseen that only the latest two EMI major releases are supported at a time. Within an EMI major release only the latest version of a component is supported. More extensive coverage will be evaluated on a case-by-case basis together with the users requesting it.

4.6. KEY PERFORMANCE INDICATORS

The KPIs mentioned in the Description of Work and relevant for the SA1 User Support Task are:

- **KSA1.1 – Number of Incidents**

  Number and trends of incidents registered by the EMI Support Units (in total and per category).

  Periodic information on the number, categories and submission time of the incidents escalated to the EMI Support Units, including the generic Support Unit, are extracted from GGUS.

- **KSA1.2 – Incident Resolution Time**

  Average time for resolving an incident by the 3rd-level support (possibly by category).

  Information on the number, categories and submission and resolution times of the incidents escalated to the EMI Support Units are extracted from GGUS.

The KPIs will be continuously collected and reported by the TSA2.3 task on Metrics and KPIs Definition and Reporting. They will be monitored by the User Support task leader and any anomaly will be brought to the attention of the SA1 leader and if necessary of the PEB.

5. CONCLUSIONS

This document presents the fundamental principles that will guide the Software Maintenance and the User Support tasks within the project. The focus is on providing users with a positive experience with the software delivered by EMI, both in terms of stability and feature evolution.

The basic principles will be expanded into practical guidelines and procedures in collaboration with SA2 and in conformity to the Software Quality Assurance Plan (DSA2.1) [R 11].

The DoW defines some KPIs to measure the performance of the Software Maintenance and User Support tasks. Any anomaly will be investigated and the results fed into the continuous improvement process described in the Software Quality Assurance Plan.