EGI-InSPIRE

SOFTWARE PROVISIONING PROCESS

EU DELIVERABLE: MS503

Document identifier:	EGI-MS503-final
Date:	**05/11/2010**
Activity:	**SA2.3**
Lead Partner:	**CESGA**
Document Status:	**FINAL**
Dissemination Level:	**PUBLIC**
Document Link:	https://documents.egi.eu/document/68

Abstract

This document describes the process by which components will be deposited in the EGI Software Repository, primarily by external software providers, processed and released for deployment into production. It describes the assessment process and some of the criteria that will be applied to all software components and outlines some of the component specific tests that may be applied as part of the software validation process.

I. COPYRIGHT NOTICE

II. DELIVERY SLIP

	Name	Partner/Activity	Date
From	Carlos Fernández	CESGA	3/09/2010
Reviewed by	**Moderator:** Steven Newhouse **Reviewers:** Daniele Cesini	EGI.eu SA2	30/07/2010
Approved by	**AMB & PMB**		13/9/2010

III. DOCUMENT LOG

Issue	Date	Comment	Author/Partner
0.5	June 14th 2010	First draft	Carlos Fernández / CESGA
0.6	June 28th 2010	Second draft (comments from Mario David and Enol Fernández)	Mario David/LIP, Enol Fernández/CSIC
0.7	July 1st 2010	Third version (comments from Steven Newhouse)	Steven Newhouse/EGI.eu
4.0	July 19th 2010	Review and comments from Steven Newhouse	Carlos Fernández / CESGA
5.0	July 26th 2010	Review and comments from Daniele Cesini	Carlos Fernández / CESGA
6.0	July 30th 2010	New input from Enol Fernández	Carlos Fernández / CESGA
7.0	July 30th 2010	Review and comments from Daniele Cesini and Steven Newhouse,	Carlos Fernández / CESGA
8.0	August 10th 2010	Additional input and comments from Steven Newhouse	Steven Newhouse/EGI.eu
9.0	September 3rd 2010	Comments from Daniele Cesini	Carlos Fernández / CESGA

IV. APPLICATION AREA

This document is a formal deliverable for the European Commission, applicable to all members of the EGI-InSPIRE project, beneficiaries and Joint Research Unit members, as well as its collaborating projects.

V. DOCUMENT AMENDMENT PROCEDURE

Amendments, comments and suggestions should be sent to the authors. The procedures documented in the EGI-InSPIRE "Document Management Procedure" will be followed: https://wiki.egi.eu/wiki/Procedures

VI. TERMINOLOGY

A complete project glossary is provided in the EGI-InSPIRE glossary: http://www.egi.eu/results/glossary/.

VII. PROJECT SUMMARY

To support science and innovation, a lasting operational model for e-Science is needed – both for coordinating the infrastructure and for delivering integrated services that cross national borders.

The EGI-InSPIRE project will support the transition from a project-based system to a sustainable pan-European e-Infrastructure, by supporting 'grids' of high-performance computing (HPC) and high-throughput computing (HTC) resources. EGI-InSPIRE will also be ideally placed to integrate new Distributed Computing Infrastructures (DCIs) such as clouds, supercomputing networks and desktop grids, to benefit user communities within the European Research Area.

EGI-InSPIRE will collect user requirements and provide support for the current and potential new user communities, for example within the ESFRI projects. Additional support will also be given to the current heavy users of the infrastructure, such as high energy physics, computational chemistry and life sciences, as they move their critical services and tools from a centralised support model to one driven by their own individual communities.

The objectives of the project are:

1. The continued operation and expansion of today's production infrastructure by transitioning to a governance model and operational infrastructure that can be increasingly sustained outside of specific project funding.
2. The continued support of researchers within Europe and their international collaborators that are using the current production infrastructure.
3. The support for current heavy users of the infrastructure in earth science, astronomy and astrophysics, fusion, computational chemistry and materials science technology, life sciences and high energy physics as they move to sustainable support models for their own communities.
4. Interfaces that expand access to new user communities including new potential heavy users of the infrastructure from the ESFRI projects.
5. Mechanisms to integrate existing infrastructure providers in Europe and around the world into the production infrastructure, so as to provide transparent access to all authorised users.
6. Establish processes and procedures to allow the integration of new DCI technologies (e.g. clouds, volunteer desktop grids) and heterogeneous resources (e.g. HTC and HPC) into a seamless production infrastructure as they mature and demonstrate value to the EGI community.

The EGI community is a federation of independent national and community resource providers, whose resources support specific research communities and international collaborators both within Europe and worldwide. EGI.eu, coordinator of EGI-InSPIRE, brings together partner institutions established within the community to provide a set of essential human and technical services that enable secure integrated access to distributed resources on behalf of the community.

The production infrastructure supports Virtual Research Communities (VRCs) – structured international user communities – that are grouped into specific research domains. VRCs are formally represented within EGI at both a technical and strategic level.

VIII. EXECUTIVE SUMMARY

EGI-InSPIRE will initially use software components provided by the European Middleware Initiative project (EMI), by the Initiative for Globus in Europe (IGE) project, and other external sources called "Community Contributions". These software components will form the Unified Middleware Distribution (UMD). A given UMD release will be composed of software (services, libraries, tools, etc.) provided by Product Teams (PT) that build on top of a given base release. Services from the gLite, ARC, UNICORE, and Globus middleware stacks will be included in the UMD. Globus components, previously provided in EGEE by the Virtual Data Toolkit (VDT), might instead be provided directly by the IGE project.

Further components are contributed by the community. For example; the batch systems support in the middleware, the package containing the IGTF approved Certification Authorities (CA) will come from sources other than the IGE and EMI projects. The production infrastructure also depends on the Operational Tools such as the "Operations Portal", the "Accounting Portal" (account for their usage) and Nagios monitoring tools (needed for a reliable and stable operation of the infrastructure as well as to monitor its resources). Frequently, these will be deployed at a national or regional level and increasingly these software components will also have to be installed outside the environment they were developed in.

From here on we will define "Software Providers" as any entity providing any piece of software which falls into the previous description. The software provisioning process described in this document is designed to ensure that these software components can be installed and will work reliability in the environments and loads that they have been designed for.

We give below a summary of the major categories of software in use or expected to be used by EGI-InSPIRE:

- The EMI release containing components from gLite, ARC and UNICORE middleware stacks.
- The Globus middleware stack provided in Europe by IGE.
- Community Contributions, such as the Certification Authority packages or the batch system integration into the several middleware stacks.
- Operational Tools provided inside the EGI-InSPIRE: Operations Portal, Nagios monitoring tools, etc.

All of these software categories and all of their releases (e.g. major, minor or patch release) will undergo the Software Validation (SV) procedure, though the time-lines and depth of the SV may vary with the software. The sole exception is an Emergency release, for which, under exceptional circumstances to be evaluated in a case by case basis, may skip the SV. The objectives of the SV process will be to check that the individual software component has been validated against the generic and component specific conformance criteria. Verification of each component will be summarised in an acceptance report, available in the component repository. The detailed procedure of the SV is described in Section 4.

EGI-InSPIRE will accept only certified and validated updates provided by the Software Providers. The validated components will undergo the Staged Rollout (SR) procedure managed by SA1 [R2], and if successful can then be widely deployed in the production infrastructure. The SR will ensure that new

software releases will be deployed safely and reliably without any degradation of the service to the production infrastructure, through staged deployment for all the software.

In the SV phase, if bugs or issues are found in a given component for which some solution or workaround is proposed, the fix(es) should be communicated and implemented by the Software Provider. Software components with workarounds to bugs or issues should be avoided in production.

The EGI.eu Technology Unit (TU) will define how individual services will be integrated together.

TABLE OF CONTENTS

1 INTRODUCTION

This document describes the process by which components will be deposited in the repository by the software providers, processed to ensure they meet the defined acceptance criteria, and made available to the staged rollout process before deployment into production.

The major source of software components that will be integrated by EGI-InSPIRE into the Unified Middleware Distribution (UMD) are:

- The EMI release containing components from gLite, ARC and UNICORE middleware stacks.
- The Globus middleware stack provided in Europe by IGE.
- Community Contributions, such as the Certification Authority packages or the batch system integration into the several middleware stacks.
- Operational Tools provided inside the EGI-InSPIRE: Operations Portal, Nagios monitoring tools, etc.

2 SOFTWARE PROVISIONING IN EGI

2.1 SOFTWARE PROVIDERS

EGI is driven by its user community as concerns the production infrastructure that it delivers. As the production infrastructure, and therefore the software used to integrate these resources, will change over time there is no guarantee that, even if the partners currently involved in EGI (primarily NGIs) were able to deliver a particular software solution now, they would provide the best solution in the future. Strategically therefore, EGI will not therefore develop its own middleware solution, but instead it will source the required components from external software providers. This provides EGI with great flexibility – it is able to select the best available solution that meets its user's requirements rather than being locked into any particular solution due to the involvement in EGI of any particular partner. It can remain completely technology neutral.

2.2 Technology Coordination Board (TCB)

The TCB is an advisory body which develops strategy and technical priorities concerning the maintenance, support and evolution, of the technologies (including grid middleware) adopted for production use in the EGI e-Infrastructure. The TCB is composed of representatives of the following areas:

- Technical and managerial representatives from within EGI.eu
- The main software providers engaged with EGI;
- The operational requirements of the production infrastructure through the Chief Operations Officer;
- User communities affiliated with EGI represented through the Chief Community Officer.
- Representatives from the USAG and OTAG

The role of the TCB is to collect and prioritise high-level requirements following the requests from users and operational staff, and to endorse (or to eventually reject) updates to the UMD Roadmap as they relate to the provision of EGI's production infrastructure. As the software will be sourced from outside EGI formal agreements must be established with the relevant software providers, notably the proposed EMI project. It is through these activities that the TCB advises the EGI.eu Director on strategic and technical issues concerning the technology requirements for the EGI's production infrastructure. It has no involvement in the day to day management of the activities within the middleware unit (SA2). It is expected that detailed technical discussion and alignment between the software providers contributing middleware for deployment on the production infrastructure will take place outside this body.

2.3 The Unified Middleware DiStribution (UMD)

The Unified Middleware Distribution (UMD) can therefore be considered to be:

- A set of functional specifications, and performance and quality requirements
- A set of software components meeting the functional, performance and quality requirements registered in the EGI Software Repository
- A set of integrated components, taken from those in the EGI Software Repository which meets the established criteria, released as an integrated distribution for installation

To source these components it will be necessary to establish close but formal relationships with the providers of the key software components within the UMD Release. The relationship, defined in a Service Level Agreement (SLA), will include the agreed release schedule and expected support and maintenance of the software components. During the project this SLA model is expected to evolve towards a sustainability model which may include agreements negotiated with commercial software providers, as well as open source contributions etc. Managing these relationships and agreements will be the responsibility of the SA2 Activity Manager. It is expected that a very strong collaboration will be established with the proposed EMI and IGE projects as they will provide many of these key software components.

The UMD Roadmap will indicate when the contributed components will be included in UMD Releases. The roadmap will provide important information for both operations and users about upcoming new functionality and the phasing out of existing functionality, as well as for software providers to know about requirements for new functionality. The UMD Interfaces will evolve continuously, reflecting new infrastructure and the requirements of new users. Entries in the UMD Roadmap will contain, for each major and minor release of a software component:

- Functionality description, including links to the requirements addressed by this release
- Expected release date
- Expected level of maintenance and its duration
- Component-specific acceptance criteria
- Dependencies with other components
- Any associated risks (security, privacy, etc.)

Possible conflicts in the UMD Roadmap will be detected and resolved through discussion with the relevant software providers and refinement of the UMD Roadmap. In general, the UMD Roadmap must ensure that components used in production are supported at an appropriate level. If a component is planned for replacement or phase out, a transition plan must be provided. Component use will be monitored through feedback from the NGIs through their service logs and feedback from the user community. Sparsely used or unused components downgraded in support or removed from the distribution entirely. SA2 through the Activity Manager will be responsible for continuous maintenance of the UMD Roadmap. Updated roadmaps will be submitted to TCB every 6 months for approval and the approved version published. Draft versions of the roadmap will be available for community comment and feedback.

2.4 Software versioning scheme

A new version of any software component can be categorized as follows:

- **Emergency release** (when needed): it fixes critical functionality problems and/or serious security vulnerabilities. It is backwards compatible.
- **Revision release** (at most once every two weeks): it provides bug fixes and is backwards compatible.
- **Minor release** (at most once per month): it provides new functionality and is backwards compatible.
- **Major release** (at most once per year): it offers new functionality, not necessarily backwards compatible and may also include new services.

A given UMD major release will contain baseline major versions of a set of components. These major versions are subject to the agreement between EGI and the software providers, and will be detailed in the UMD Roadmap document.

Any middleware component can be updated only up to a minor release within any major UMD release. The major releases of any given component may only be included in the next major UMD release, depending on the roadmap.

It is foreseen that all categories of component updates will undergo the staged-rollout process, but the time-lines and the extensiveness of the staged-rollout will vary according to the category. The sole possible exception is an emergency release, which may skip staged-rollout under exceptional well document circumstances that are evaluated on a case by case basis.

The components classified as “Community Contributions” and “Operational Tools” will follow a similar procedure.

EGI-InSPIRE will accept only certified and validated updates provided by the software providers. The validated components will undergo the staged-rollout procedure. If successful, they can then be widely deployed into the production infrastructure.

If bugs or issues are found during the staged-rollout phase in a given component for which some solution or workaround is proposed, the fix(es) should be communicated and implemented by the respective software provider. Middleware components with workarounds to bugs or issues should be avoided in production.

Each middleware stack is in general composed of several capabilities. As such, it is the responsibility of the EGI.eu Technology Unit, to provide requirements about any given capability. For example, the Compute capability should be have integration to several Local Resource Management Systems (LRMS), with several parallel programming environments, etc.

2.5 Operational tools

The stability, reliability, monitoring, accounting and user support in the EGI production infrastructure relies on several operational tools which were developed in the EGEE project, and have its continuation, both further development and maintenance in the EGI-InSPIRE JRA1 task.

A more detailed description can be found in [R3].

While any given software stack is or can be deployed by any site participating in the Grid infrastructures, thus having a fairly large number of deployments, most of the operational tools will be deployed and operated by a smaller number of sites, which committed to provide such services for National or Regional Grid Initiatives, or even for the whole EGI.

Nonetheless, the workflow to rollout new versions of such components into the production infrastructure should follow as close as possible the same path as the software components described in the previous section.

2.6 Tools used in the SOFTWARE VALIDATION process

There are several tools already setup by EGI and which will be used in several step of the SV process. A list will be given below, with some details of its use:

1. **EGI RT** (https://rt.egi.eu/) [R7]: A "Request Tracker" to follow all the Software validation process from the moment it is uploaded in the repository and made available by the Software Providers until it is released into the production infrastructure.
2. **EGI WIKI** (https://wiki.egi.eu/) [R8]: to hold more dynamic information such as documentation of all releases with deployment advisories, with links to release notes, certification and validation of software components provided by the Software Providers.
3. **EGI Repositories [R9]:** these provide access to the software packages that are part of the UMD distributions, during the various stages of the software lifecycle. Details are given in [R4]
4. **EGI Single Sign On (EGI-SSO) [R6]: contains user accounts and LDAP groups, such as the Early Adopter group of users**

3 QUALITY CRITERIA

The software validation will verify that all the software included in the Unified Middleware Distribution (UMD) meets a set of Quality Criteria defined by EGI. The Quality Criteria can be classified into generic criteria, i.e. criteria which should hold for any component of the UMD, and specific criteria, i.e. criteria valid for a particular component only.

3.1 Verification Process

In order to be verified, the quality criteria are specified as a set of tests. Those tests must ensure the correctness, completeness and security of each service. Software providers must include with each component a complete test plan that covers all the quality criteria. The test plan is composed by:

- General description of the test plan.
- A Test Suite with documentation for each of the test cases (objective of the test, how to run it, expected output, possible errors, pass/fail criteria) included in:
  - Generic criteria.
  - Specific criteria applicable to the component.
- Tests results for all the specified tests.

In the case of revision releases, the test plan must cover bugs fixed in the release.

3.2 Specific Acceptance CrIteria

The specific acceptance criteria of the UMD are classified according to the following preliminary areas which will be aligned to the capabilities being defined in the UMD Roadmap:

- Security Services
- Computing Services
- Data Services
- Information Services

In this document we present only the detailed information for the generic acceptance criteria. The detailed specific criteria is available in [R1]

3.3 Generic acceptance criteria

Documentation

Services in UMD must include a comprehensive documentation written in a uniform and clear style, which reflects all of the following items:

- Functional description of the software.
- User documentation, including complete man pages of the commands and user guides.
- Complete API documentation (if there is an API)
- Administrator documentation that includes the installation procedure; detailed configuration of service; starting, stopping and querying service procedures; ports (or port ranges) used and expected connections to those ports; cron jobs needed for the service)
- List of processes that are expected to run, giving a typical load of the service. List of how state information is managed and debugging information (e.g.: list of log files, any files or databases containing service information).
- Notes on the testing procedure and expected tests results.

Verification: existence of the documentation with all the required items.

Source Code Quality and Availability

The source code of each component of the UMD should follow a coherent and clear programming style that helps in the readability of the code and eases maintenance, testing, debugging, fixing, modification and portability of the software. Open source components must publicly offer their source code and the license with the binaries.

Verification: for Open Source components, availability of the code and license. Source code quality metrics are desirable.

Management, Monitoring and Traceability

All the services must include tools related to:

- Starting, stopping, suspending, listing and querying the status of all the service daemons.
- Checking the responsiveness of all the service components or daemons
- Checking the correctness of the service components behaviour (expected actions after a request are taken)
- Tracing all the user actions in the system (e.g. by generating logs)

Ideally, these tools should be also available remotely, allowing operators to react timely to problems in the infrastructure. A uniform interface for remote management and monitoring should be followed by all the services. These services must also be easily monitored using existing systems such as Nagios.

Verification: Test suite must include tests cases for:

- start, stop, suspend, and query status of service
- check responsiveness of service (expected ports open and expected answer to commands received)
- check correctness of service behaviour (expected actions after a request are taken)
- track of user actions in the system (generation of logs and accounting information)

Configuration

Tools for the automatic or semi-automatic configuration of the services must be provided with the software. These tools should allow the unassisted configuration of the services for the most common use cases while being customizable for advanced user. Complete manual configuration must be always allowed.

Verification: test suite must include the configuration mechanisms and tools. YAIM is considered as the preferred tool.

4 OVERVIEW OF THE SOFTWARE VALIDATION WORKFLOW

The basic workflow for a component release is shown in Figure 1. Providers are required to deliver a component release versioned according to the conventional —major.minor.revision— scheme, where increment of the revision number means that only bug(s) have been fixed and no new functionality has been added, increment of the minor number brings new functionality while preserving backward compatibility of interface and functionality, and increment of the major number means large revision, possibly breaking the backward compatibility. Baseline releases of UMD are defined by specific versions of all included components. In a given baseline release, backwards compatibility of interfaces of all components (i.e. major version number) is assured. New UMD Releases (baselines) will occur at a time and frequency determined by the Technology Coordination Board (TCB) in consultation with the community. The baselines are complemented with updates, consisting mostly of fixes for individual critical bugs and cumulative bug fixes within specific components. Minor, backward compatible functionality additions can occur with these updates if the functionality is urgently needed, however the addition of new functionality will normally be postponed until the next baseline release.

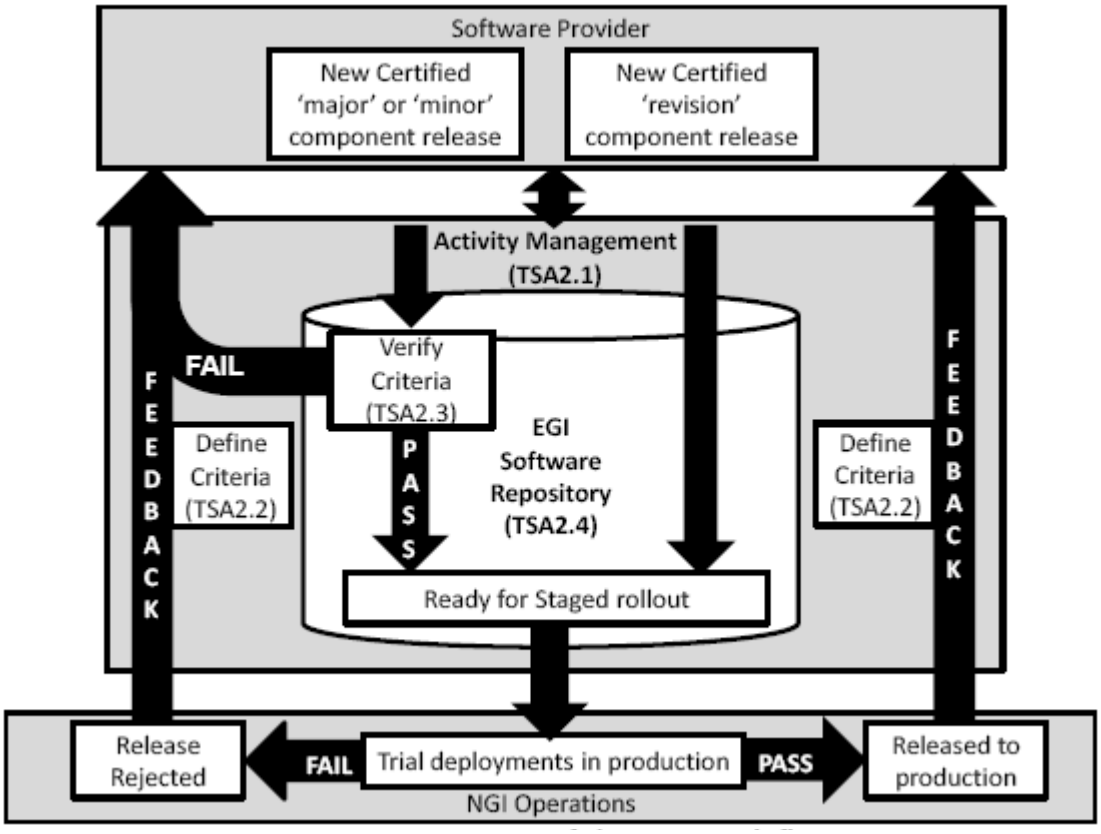

Figure 1- Overview of the middleware rollout process

For each release, the software provider will have to provide, apart from the software packages, the following documentation or a link to the relevant item:

- Release notes.
- Change log.
- Certification report(s) relating to the agreed quality assurance documentation and tests.
- Documentation: Users' Manual, Admin Manual, etc. The documentation should be updated if applicable (for example if the new version introduces new functionality).
- Installation scripts and procedures.
- A test plan for the component with:
    - A general description of the test plan
    - A test suite that covers all the test cases included in the generic and specific criteria applicable to the component
    - Each test case must have a report with documentation on the objective of the test, how to run it, expected output, possible errors and pass/fail criteria.
    - The report must contain enough details as to be able for SA2 to repeat the procedures if needed. Back-out plan

The approach taken by the SV process when verifying a release contributed from a software provider into the EGI Software Repository is dependent on the type of release from that software provider (see **Error! Reference source not found.**).

The verification process will be in charge of checking the documentation and the reports of the software provider and, with that information accept the release or reject it. In the first case, the SV will provide the following documentation:

1. Verified installation procedures
2. Tested components
3. Known errors and bugs
4. Test results
5. Administration and support documentation
6. Affected components and systems
7. Operation instructions and diagnostic tools
8. Disaster recovery plans and back-out plans approved
9. Signed validation reports

In the second case, it will report the reasons for the release not being accepted and giver a period of time (no more than 2 weeks) to the software provider to resolve the reported issues.

Under minor releases, the SV process will randomly test some of the functionalities to check that the report of the software provider corresponds with the test results.

For the major releases, the SV process will, in coordination with staged rollout team, check the new functionalities and repeat some of the procedures described in the report provided in the software.

For new components, or components including **major new functionality, complete testing**, ensuring that released software meets all specified criteria, will be done by the **software provider** in environments representative of those found in production. The software provider may cooperate with users and operations in providing these environments and test cases. SA2 will be involved as an observer in the testing process to ensure that the defined criteria accurately reflect the expected use cases. Therefore the final (after the component is uploaded by the software provider into the EGI Software Repository) independent verification that the component meets the defined criteria will be a lightweight process, based on results of these tests. The verification will be summarised into a publicly available acceptance report.

Upon component acceptance, the terms of SLA for the component as well as duration of the support are negotiated between EGI and the provider. For long-term sustainability of EGI and its community, an open environment is essential in order to promote competition and innovation to achieve high-quality software. Thus EGI will objectively assess which components meet the specified criteria and record these in the software repository. A subset of the components in the software repository meeting the specified criteria will be made available in a UMD Release.

Action	Major Release	Minor Release	Revision Release
Pre-release testing by the software provider	All alpha and beta testing takes place under the control of the external software provider. The software provider may choose to include NGIs, EIROs and specific user communities in testing their releases. SA2 staff may be observers in this process helping to clarify issues relating to the assessment criteria		
Submission of Release Candidate to the EGI Software Repository	A release candidate is uploaded to the repository with documentation, release notes, etc. relating to the completed testing		
Verification of Conformance Criteria	Verification by SA2 of new functionality or interface changes against UMD quality criteria (R1). The verification process is based around manual testing and the development of automated test suites involving SA2 and other stakeholders (i.e. EA, operations, the users community and the software provider, see R2) wherever possible	The availability of test suites and a test report from a trusted provider will allow a streamlined verification process. SA2 may still perform some manual testing	Self-certification by the software provider that all bugs have been fixed and functional interfaces and behaviours remain unchanged. There is no direct testing by SA2. The quality assurance process from the software provider is relied upon
Staged roll-out onto production resources	The release candidate is deployed onto selected production resources by SA1 for release validation in a production environment. SA2 and the external software provider observe the process. SA2 for refinement of the criteria. The external software provider to provide early feedback on any issues raised by its early production use. Typically this period last 1-2 weeks.		
Released to wide-scale deployment	The release candidate is marked in the Software Repository as being ready for wide-scale deployment		

Table 1: SA2's verification activity with a major, minor or revision release

5 DETAILED PROCEDURE OF THE SOFTWARE VALIDATION PROCESS

This section details the workflow of new software component versions from the time when they are released by the software provider to the time when the component is distributed for deployment in the EGI production infrastructure, including the software validation procedure.

1. Actions performed by the software provider
    1. A new version of a component has been produced and certified by the software provider. At this stage or earlier, the software provider creates a new ticket in the "**staged-rollout**" RT queue in the state "**Certified**". This ensures that the software provider is notified on any change in state of the ticket. This ensures a close and direct contact or a quick action if, at any step, there is a problem or issue with the component.
    2. The software provider has to provide the material, or a link, to the following information in the RT ticket:
        1. The software release notes, or the advisory written by the Software Vulnerability Group (SVG), in the case of a software vulnerability.
        2. Changelog.
        3. Certification report(s) from the agreed quality assurance documentation and tests.
        4. Documentation: users manual, system administration manual, etc. The documentation should be updated if applicable, for example if the release introduces new functionality.
        5. Links to all bugs, issues, features in this new release.
    3. The ticket is then assigned to the EGI Software Repository Manager.
2. Actions performed by the EGI Software Repository manager:
    1. The EGI Software Repository Manager pulls the packages (rpm, deb, tar, etc.) into the EGI repository called "**Unverified**", and after automatic checksum verification they set the relevant ticket in RT to "**Unverified**" If this step fails, the process is repeated.
    2. The EGI Software Repository team assigns the ticket to the EGI Technology Unit group, after step 2.1 is successfully accomplished.
3. Actions performed by the EGI Technology Unit (TU):
    1. Verifies the new version. This includes the verification of all information provided according to step 1.2.
    2. If the verification is successful the packages are moved into the "**Staged Rollout**" repository. If there are problems or issues, the Software Provider is notified immediately. After discussion a countermeasure will be agreed upon to solve the issue at hand. This measure can include the rejection of the component which case the RT ticket is set to "**Rejected**".

3. The URL of the release in the EGI Software Repository or the URL of all the packages in the release is set in the RT ticket. This will be used later by the early adopters to perform the staged-rollout.
4. Sets the status of the RT ticket to "V**erified**". At this point the ticket will be assigned to "**staged-rollout**" group which is thus notified that new packages are ready for the staged-rollout.

The staged rollout team works to deploy the software with the early adopters as described in [R2].

Figure 2 shows the state diagram for a ticket in the RT queue "staged-rollout".

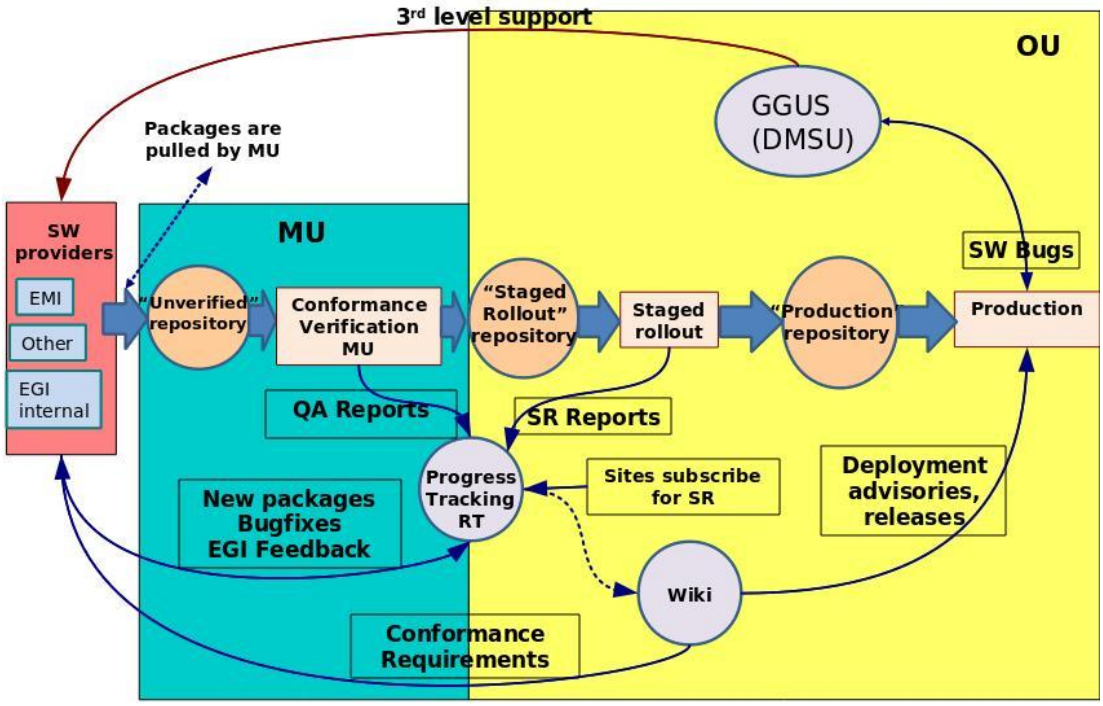

Figure 2: Middleware rollout process, from the software provider until deployment into the production infrastructure. The red square corresponds to actions from the software providers, while the blue to the EGI Technology Unit and the yellow one to the "Deployed Middleware Support Unit" (DMSU) [R5]

6 SERVICE LEVEL AGREEMENTS

Service Level Agreements (SLAs) will be signed for all software providers delivering software to EGI. This includes SLAs between EGI and EMI for the gLite, UNICORE and ARC middleware stacks, and between EGI and IGE for Globus.

The SLAs will define the level of commitment between third-party Software Providers and EGI, and should drive the stability and robustness of the software through a high level of trust.

The SLA terms and definitions are planned for a later stage of the project. More details will be documented in milestone MS505: “Service Level Agreements with Software Providers”.

7 METRICS

The software validation process will provide some reports to validate the quality of the process and also of the software providers. These will include, among others:

1. Number of new versions provided by the software provider and percentage of these releases approved or rejected by the TU. If the percentage of rejected releases for a particular SP exceeds 20%, the TCB will be notified in order to take the necessary actions.
2. Number of back-outs needed of a software component once it has entered into production and their justification, and percentage for each software provider.
3. Number of incidents related to the new versions of software when in production use, and percentage of them related with a software provider.
4. Number of software incidents found in production that result in changes to quality criteria (M.SA2.3).
5. Number of new releases validated against defined criteria (M. SA2.4).
6. Mean time taken to validate a release (M. SA2.5).
7. Number of releases failing validation. (M. SA2.6).

8 CONCLUSION

Most of the testing of the quality criteria is done by the software provider. The verification process SA2.3 will be in charge of checking the documentation and the reports of the software provider and, with that information accept the release or deny it. Under minor releases, SA2.3 will randomly test some of the functionalities to check that the report of the software provider corresponds with the tested results. For the major releases, SA2.3 will, in coordination with SA1.3, check the new functionalities and repeat some of the procedures described in the report provided in the software.

This approach balances the professionalism and trust placed in the software provider to deliver high-quality releases by having an established and rigorous quality assurance and testing process, with the time and resources needed by SA2.3 to validate a release. This approach focuses effort on releases which introduce significant functionality changes as these are likely to have a greater risk of undiscovered defects. It relies greatly on the software provider having a professional approach to delivering their software and this will be one of the major assessment points in selecting a software provider. A software provider, who repeatedly delivers faulty or low-quality software components the relationship that EGI has with the provider will be re-evaluated by the TCB and may be terminated early.

9 REFERENCES

R 1	https://wiki.egi.eu/wiki/EGI-InSPIRE:UMDQualityCriteria UMD Quality Criteria
R 2	MS402: Deploying Software into the EGI Production Infrastructure
R 3	MS702: Establishing the Operational Tool product teams
R 4	MS501: Establishment of the EGI Software Repository and associated support tools
R 5	MS502: Deployed Middleware Support Unit Operations Procedures
R 6	EGI SSO: https://www.egi.eu/sso/
R 7	EGI RT: https://rt.egi.eu/rt/index.html
R 8	EGI Wiki https://wiki.egi.eu/
R 9	EGI Repositories http://repository.egi.eu