EGI-InSPIRE

SOFTWARE PROVISIONING PROCESS

EU DELIVERABLE: MS508

Document identifier:	EGI-MS508-final
Date:	**15/07/11**
Activity:	**SA2**
Lead Partner:	**CESGA**
Document Status:	**FINAL**
Dissemination Level:	**PUBLIC**
Document Link:	https://documents.egi.eu/document/505

Abstract

This document describes the process by which components will be deposited in the EGI Software Repository, mainly by external software providers, processed and released for deployment into production. This document is a revision of the MS503 milestone. It describes the assessment process and the criteria that will be applied to 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	Alvaro Simon	CESGA	13/07/11
Reviewed by	**Moderator:** Daniele Cesini **Reviewers:** Miroslav Ruda	INFN CESNET	13/07/11
Approved by	**AMB & PMB**		15/7/2011

III. DOCUMENT LOG

Issue	Date	Comment	Author/Partner
1	16/05/2011	First draft	Alvaro Simon, EGI.eu
2	8/06/2011	Comments (M. Drescher) and changes	Alvaro Simon, EGI.eu
	14/06/2011	Included StagedRollout process (M. David comments)	Alvaro Simon, EGI.eu
3	11/07/2011	Included reviewer comments and changes	Alvaro Simon, EGI.eu

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.

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 from the software providers will be deposited in the EGI UMD repository [R9] , processed to ensure they meet the defined acceptance criteria, and made available to the staged rollout process before deployment into production.

The major sources 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 supported in Europe by IGE.
- Community Contributions, such as the Certification Authority packages or the Batch system plugins for integration into the available middleware stacks.
- Operational Tools provided inside the EGI-InSPIRE JRA1 activity: 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 production infrastructure, and therefore the software used to integrate these resources, will change over time. Thus, there is no guarantee that, even if current EGI partners (primarily NGIs) where able to deliver a optional solution, this could be not true in the future. Therefore, the strategy will be for EGI do not develop its own middleware, but instead it will source the required components from external software providers. This will provide EGI with great flexibility – being able to select the best available solution that meets its user's requirements rather than being locked into any particular solution . EGI 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:

- Technical and managerial representatives from within EGI.eu
- The main software providers engaged with EGI;
- The production infrastructure through the Chief Operations Officer;
- User communities affiliated with EGI represented by 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 EGI's production infrastructure. It has no involvement in the day by 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) [R21] can therefore be considered:

- A set of functional specifications, 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 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 architecture 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 will be 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:

- **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 functionalities, not necessarily backwards compatible and may also include new services.
- **Emergency release**: This is a special release and should be evaluated in a case by case basis. Emergency releases are only scheduled when a new middleware critical bug or security issue is found. These releases should go straight to production. Emergency releases are not verified to avoid bottlenecks or time constraints.

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.

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 integrated with 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 during the EGEE project and have its continuation, both further development and maintenance in the EGI-InSPIRE JRA1 activity.

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 steps of the SV process. A list will be given below, with some details about its use:

- **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.
- **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.
- **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]
- **EGI Single Sign On (EGI-SSO) [R6]:** Contains user accounts and LDAP groups, such as the Early Adopter and sw-rel-qc group of users.
- **EGI DocumentDB [R17]:** Verifiers will use DocDB spaces to store their verification reports.
- **GGUS [R23]:** The EGI Helpdesk to track Technology Providers feedback.
- **OpenNebula [R18]:** OpenNebula cloud frameworks is used for SA2 Verification tesbed.

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 [R10]. 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 documentation that covers all the quality criteria. This documentation must include:

- Functional description of the component.
- Release notes.
- User and Administrator documentation.
- Service reference card.

In the case of minor releases, the provided documentation 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 are 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: The test suite executed by the SA2 verification team 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: The test suite executed by the SA2 verification teammust 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 **Error! Reference source not found.**. 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.

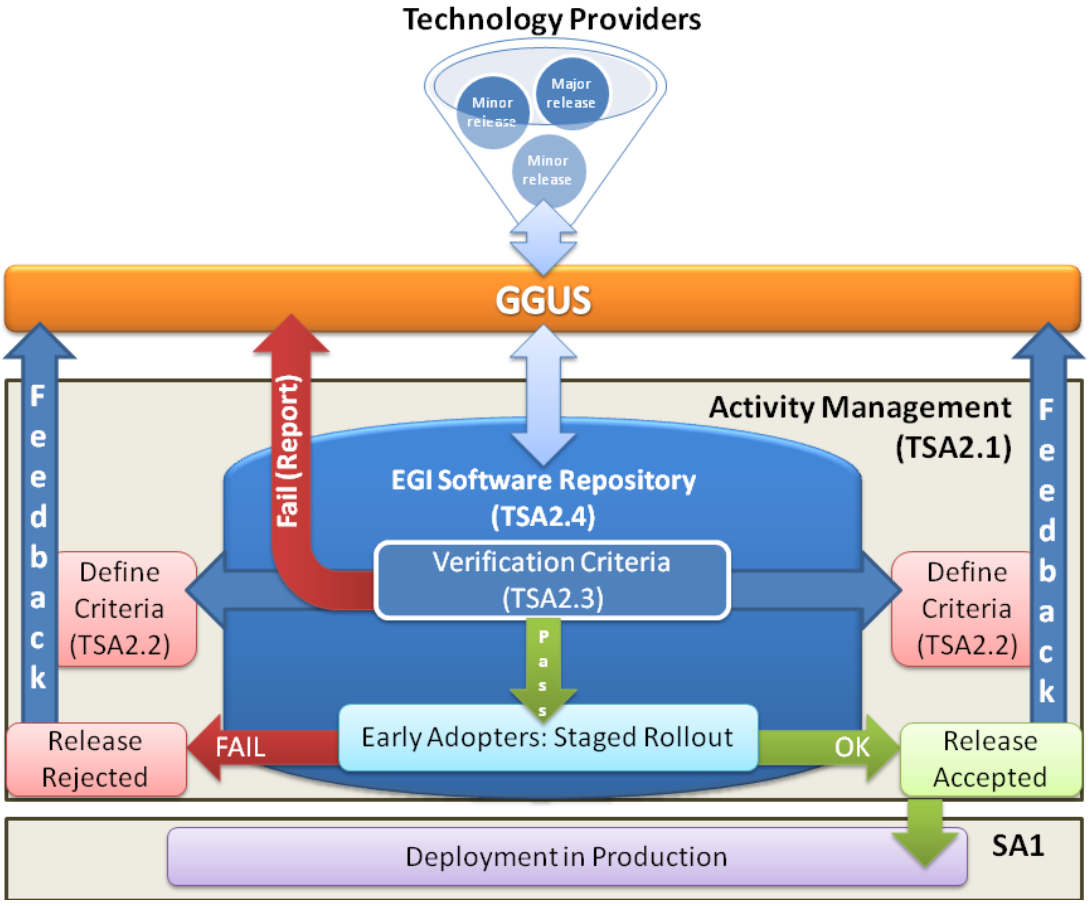

Figure 1: Overview of the middleware rollout progress

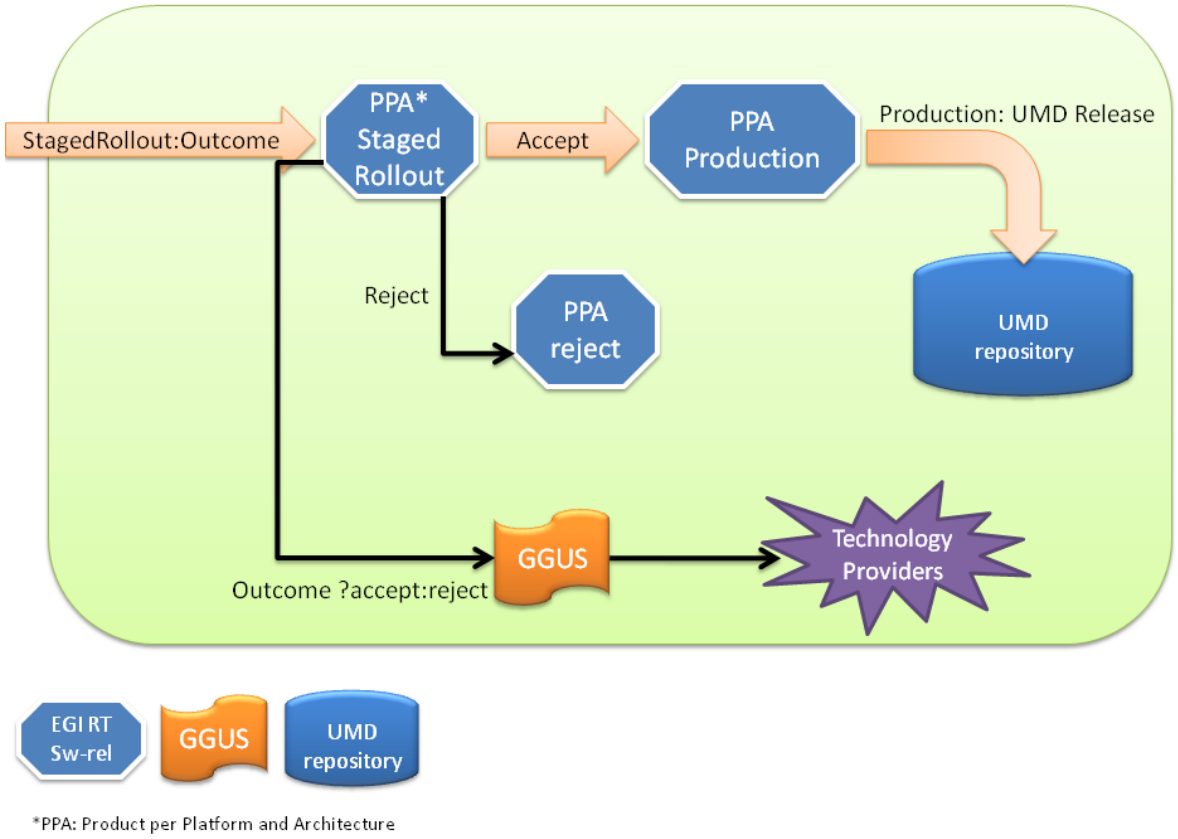

Figure 2: The StagedRollout workflow

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.
- Changelog.
- Documentation: User 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 list of known issues.

As example the release notes for EMI1 Kebnekaise are available here [R22].

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

- Verified installation procedures.
- Tested components.
- Known errors and bugs (list of opened bugs and tickets during QC verification process).
- Test results.
- Administration and support documentation.

- Affected components and systems.
- Operation instructions and diagnostic tools.
- Signed validation reports and executive summaries.

In the second case, it will report the reasons for the release not being accepted and give a period of time to the software provider to resolve the reported issues.

If the product is rejected, the verifier writes a complete report about the rejection reasons (executive summary and Quality Criteria report). RT ticket is set as rejected and the feedback is submitted through GGUS to the Technology Provider.
If the new product is verified and accepted the product RT ticket status is changed to StageRollout.At this moment TSA1.3 takes care of the installation and test of the new middleware. The complete Staged Rollout process is described in MS409 [R19]. Staged Rollout workflow after verification is shown in Figure 2. In this phase Early Adopter sites install and test the new verified middleware provided by the UMD repository.

Unlike the verification testbed, the Early Adopters test and update in a production environment that is more heterogeneous than what is possible during the certification and verification phases. Early Adopters may accept or reject a new product release:

- **REJECTED:** The new product fails the deployment into a production environment. The Early Adopter change RT status to rejected and fills and submits the StageRollout report.
- **ACCEPTED:** The new product is accreted and pass the Staged Rollout process. The new middleware is ready to be included into the UMD production repository and to be used by the community.

EGI Production Release

When the new product pass the Staged Rollout process, and if the Early Adopters and Technology Providers don't detect any issue, the new middleware is released into the production infrastructure using the EGI repository. The UMD release manager moves the new middleware from an intermediated UMD repository (a staging area used in Verification and Staged Rollout processes) to the final UMD production repository.

The production release is finally published in the EGI Software Repository [R9], the portal repository is automatically updated with the new releases. The information is gathered through RSS feeds. UMD repository maintains a RSS feed and adds a new RSS channel when a new release is processed. The technical information is available in MS506 Milestone [R21], the new release post should include this information:

- **Product name and version.**
- **Short description**: A description about the new product and its capabilities.
- **Documentation links**: Installation, configuration and middleware manuals.
- **Release notes**: Changelog, known issues and fixes.
- **Repository URL**: Used by the EGI user community to install the new software from UMD repository

When the new post is created the announcement of the new production release is done through the operations portal broadcast tool to the following groups:

1. Production Site Administrators (taken from the GOCDB).
2. WLCG Tier-1 contacts (static CERN maintained list).
3. inspire-sa2(at)mailman.egi.eu: EGI SA2 team including the staff responsibe of defining the middleware quality criteria, the repository administrators, the DMSU support unit and the staff responsible of the verification process.
4. NGI Operations Centre managers: noc-managers(at)mailman.egi.eu .

Finally the generated RT tickets by the bouncer are closed and the new software is ready to be used by the EGI community. Due to the spiral lifecycle model followed by the development, the process does not end here (See Figure 3).

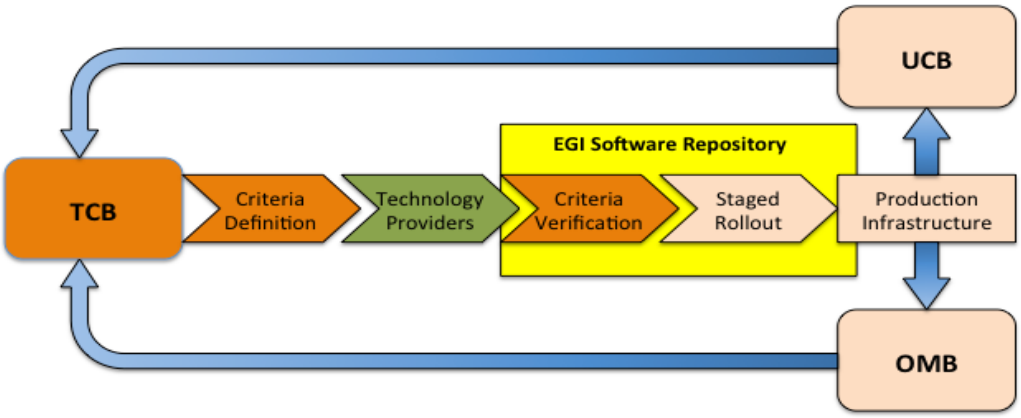

Figure 3: The EGI Software Provisioning workflow

The behaviour of the new software included into production is reviewed by the User Community Board (UCB) and the Operations Management Board (OMB). The main tasks assigned to these groups are to Identify, prioritize and resolve issues related to users or Virtual Organizations and identify current and future problem areas and propose corrective actions. If any issue is found in production UCB and OMB submits their feedback to take appropriate remedial actions. These actions are assigned to TCB and the lifecycle starts over again.

5 DETAILED PROCEDURE OF THE QC VERIFICATION 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.

When a new product is available, the TP has to follow the Software Provisioning Process [R13]. Once the software is correctly uploaded to the repository, the release enters into the verification phase. The requirements are that the TP has to provide all the necessary information to the Quality Criteria Verifier (QCV) so that the QCV can assess that the TP has tested in advance the quality of the software. Depending on the type of release, different actions will be taken by the QCV, Verification process is described in detail in the EGI Verifier Guideline [R12].

5.1 Verification Pre-Conditions

The verification process starts when the following pre-conditions are met, verifiers must check these RT tickets fields:

1. RT ticket is in state **Open**. This state is set by a bouncer script wich proccessses the information submitted by the TP. A complete description of the EGI Software Repository Architecture is available here [R21].
2. The RolloutProgress is set to **Unverified**.
3. CommunicationStatus is **OK**. (also set by the bouncer script)
4. Owner is set to **nobody**.

5.2 Verification Process Starts

Once the verification ticket meets the preconditions described above, the QCV must perform the following steps:

1. Set RT ticket **Owner** with the current QCV name.
2. Set **UMDRelease** to the appropriate UMD Release and save the state.
3. Changes RolloutProgress to **In Verification** to start the actual verification.

5.3 Verification Report Templates

Each product has a specific template that includes all QC that the product must comply with. Report and Executive Summary Templates are available at [R14] and are created according to the QC products mapping [R15]. These documents are updated if new UMD Quality Criteria is released.

The fields to fill in the template are:

- **Accepted**:
  - Y, when the product meets the criteria
  - N, when the product does not meet the criteria,
  - NA, when the criteria is not applicable for the product (e.g. API documentation for products without a public API)
- **Tested**:
  - TP, when the criteria was tested by the Technology Provider and the validator trusts the results of the tests.
  - or VLD, when the criteria was tested by the validation team
- **Comments**: include here any relevant comments or links to more information for the specified criteria.

5.4 Level of testing

All the products in verification **must** be installed and tested in the EGI Verification Testbed [R16], however the verification process is different for UMD **Minor** or **Major** releases (Emergency releases are not verified and go straight to production):

- **Major releases** (may not be backwards compatible):
  - Verifier **MUST** actively assess all assigned QCs (executing specific tests, reading available documentation, etc).
  - Product installation from scratch (or upgrade if It's supported by the product).
- **Minor releases** (backwards compatible):
  - Verifiers only check QCs affected by update changes.
  - Package update installation and verification. Verifiers must update the new packages (using the new umd unverified repository) in an previously installed machine.
  - Product installation from scratch. Verifiers must install from scratch the new product using the umd unverified repository.

If during the verification process, arise any problems that need to be solved interacting with the Technology Provider (i.e. missing documentation), the Verifier should change the RT ticket **RolloutProgress** to the *Waiting for Response* status. The verifier must open a GGUS ticket, and include all the created GGUS tickets as reply into the RT ticket to track all the issues found**.** Once the problem is solved, the **RolloutProgress** must be changed again to *In verification.*

5.5 RT Comments

There are two kind of RT comments or replies:

- **Comment**: A background communication (not visible to watchers).

- **Reply**: A public response (visible to watchers, TP and RT sw-rel queue members). GGUS and TP opened tickets MUST be included always as Reply.

5.6 Product Acceptance

QCs tests are **Mandatory** (M) or **Optional** (O).

- A product is **REJECTED** if it fails the installation or configuration process.
- A product is **REJECTED** if it fails **ANY** Mandatory QC.
- A product is **VERIFIED** if it pass **ALL** assigned QCs.
- A product is **VERIFIED** if it fails **ANY** Optional QC.

5.7 Verification Reports and Executive Summaries

When the Verification template is finished an Executive Summary is created, following the template available at [R14]. This document includes:

- A summary of tests failed and passed (Critical and non critical).
- A list of comments for other teams involved in the Software Provisioning process (SR, QC,).

In order to store the verification reports, a new space must be created in the EGI DocumentDB [R17]. The new document must comply with the following information:

- **Title**: QC Verification Report: <PRODUCT_NAME>-<VERSION>.
- Files to include:
  - First file field is used to upload Executive Summary doc. Executive Summary file names should have this nomenclature: QC_Verification--Executive_Summary_<PRODUCT_NAME>_<VERSION>.doc.
  - Second file field is used to upload QC Verification report. Verification report file names should have this nomenclature: QC_Verification--Report_<PRUDUCT_NAME>_<VERSION>.doc .
- **Keywords** field: Space separated keyword names, must include Quality, Criteria, Verification, etc.
- **Media type:** must be set to Document.
- **Status:** must be set to FINAL.
- **View:** must be set to public.
- **Modify:** should be set to inspire-SA2.
- **TOPICS:** space field must be set to Software Provisioning Verification Reports and WP5.

The generated document DB link must be included into the **QualityCriteriaVerificationReport** field of

the RT ticket of the verification.

5.8 End of Verification

Once the reports are stored in the doc DB and linked in the RT ticket, the result of the verification must be set by the verifiers changing the *RolloutProgress* field of the ticket. If the product met is **accepted** then change this field to the "*StageRollout*" value. This will cause the Rollout teams to continue with the software provisioning process. If the product is **not accepted**, then change the value to *"Rejected"*, causing the process to stop. When a RT ticket is rejected, a script automatically moves the software to a rejected repository, the TP is automatically notified by the RT ticket system (TP should be always a watcher) and through GGUS. A detailed explanation of the TP and GGUS interaction is described in the “EGI Repository Architecture” [R21].

Finally if the new ticket pass the verification process and the Staged Rollout in production without problems, the Staged Rollout coordinator changes the ticket field to *Production* . The Bouncer script automatically moves the new middleware to the UMD production repository so that the ticket can be closed by Staged Rollout coordinator [R19]. The Technology Provider is notified by the RT ticket system when the new middleware is released into production.

6 EGI VERIFICATION TESTBED

Due to the imminent arrival of EMI1, one of the highest priorities for the SA2.3 group was the creation of a SA2 verification testbed. The new testbed acts as a secure platform where verifiers may install, configure and verify the new incoming middleware. This testbed is implemented using the OpenNebula toolkit [R18] and was configured and installed using CESGA cloud computing resources. New verifiers only have to request a new virtual machine instance, and in a few minutes a new virtual machine is created and it is ready to be used for the verification process.

6.1 VMs Request

Each released product **must** be installed in different machines. Since each product is installed and integrated in the verification testbed, the verifiers can check the interoperability between different products. SA2 Verifiers may request new virtual machines for verification submitting an email to grid-admin(at)cesga.es. This email should include:

- **Subject**: *SA2 Verification Tesbed request.*
- **Body**:
  - *Verifier Name.*
  - *Product name to verify.*
  - *OS and hardware requirements (if necessary)*
  - *Verifier SSH public key (generated id_dsa.pub)*

The verification team will submit a confirmation email with the details needed to connect to the new machine (public IP, hostname, etc). New VMs have a limited lifetime of 1 week by default, after this period the VM is destroyed. The verifier must contact to admin(at)cesga.es if he/she needs more time to finish the verification process.

7 METRICS

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

1. Number of new releases validated against defined criteria (M. SA2.4).
2. Mean time taken to validate a release (M. SA2.5).
3. Number of releases failing validation. (M. SA2.6).

These metrics are gathered directly from EGI RT [R7] for each quarterly report, for the next Q6 it's planed to be used the new EGI metrics portal to store SA2.3 metrics. These values are a good measure of the sanity of the software provisioning and verification processes to know how many products are verified and how much time is spent in the process.

If the mean time taken to validate a release (MSA2.5) increases significantly over time then SA2 task leaders should take corrective actions as:

- Increase the number of verifiers for each UMD release.
- Decrease the number of products included in each release.
- Identify which products (if any) are more complex to install to find possible bottlenecks in the verification process.

The metrics of the latest Quarterly Reports (only CAs and SAM updates were verified) are displayed in Figure 4.

Metric ID	Q1	Q2	Q3	Q4
MSA2.4	0	1	1	3
MSA2.5	0	4h	8h	4h
MSA2.6	0	0	0	0

Figure 1- Software validation process metrics

8 CONCLUSIONS AND LESSONS LEARNT

The verification process SA2.3 will be tasked with checking the documentation and the assessment of the QC and, with that information, accept the release or deny it. Under minor releases, SA2.3 will only test the QCs affected by the new functionalities or changes. For the major releases, SA2.3 will, in coordination with SA1.3, check all the new functionalities and repeat some of the procedures described in the report provided in the software.

The Software Provisioning Process has experienced significant changes in the last year. Now the interaction with external TPs are coordinated by the GGUS ticket system [R21]. When the external TP releases new software, the TP creates a GGUS tickets and GGUS automatically triggers a new RT ticket into sw-rel queue . The software migration to UMD production repository and release.xml parsing are handled automatically by a bouncer and composer script. Verifiers and Early Adopters in production only have to change the ticket status when their duties are finished. This workflow avoids human error and simplifies SA2 team tasks.

The process is highly automated, but this does not prevent future issues. As example, in the latests CA update was detected a critical problem by the TP before production release. The TP rejected its own ticket, this is not the normal behaviour and a new condition was created in the software provisioning process. Sw-rel RT tickets can only be rejected by the software provisioning members not by the TPs. The Technology Providers must contact directly to Verification or Staged Rollout coordinator explaining the reason for the rejection. At this point the verifier or the Early Adopter can reject the product based on the TP reasons.

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. The Verification process is needed after TP certification and tests to ensure and asses the EGI Quality Criteria. EGI QC assessment is a critical threshold to include new software into UMD repository and is the aim objective of the verification workflow.

9 REFERENCES

R 1	UMD Quality Criteria: https://documents.egi.eu/public/ShowDocument?docid=240
R 2	MS402: Deploying Software into the EGI Production Infrastructure https://documents.egi.eu/document/53
R 3	MS702: Establishing the Operational Tool product teams https://documents.egi.eu/document/52
R 4	MS501: Establishment of the EGI Software Repository and associated support tools https://documents.egi.eu/document/46
R 5	MS502: Deployed Middleware Support Unit Operations Procedures https://documents.egi.eu/document/504
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 Software Repository: http://repository.egi.eu
R 10	Quality Criteria Definition: https://wiki.egi.eu/wiki/EGI_Quality_Criteria_Definition
R 11	NSRW Implementation RT: https://wiki.egi.eu/wiki/NSRW_IMPLEMENTATION_RT
R 12	QC Verifier Guideline: https://wiki.egi.eu/wiki/EGI_Verifier_Guideline
R 13	Software Provisioning Process: https://wiki.egi.eu/wiki/Software_Provisioning_Process
R 14	Verification Reports and Executive Summary Templates: https://documents.egi.eu/document/417
R 15	QC Products mapping: https://documents.egi.eu/document/418
R 16	EGI Verification Testbed: https://wiki.egi.eu/wiki/EGI_Verification_Testbed
R 17	EGI DocumentDB: https://documents.egi.eu/secure/DocumentAddForm?mode=add
R 18	OpenNebula Toolkit: http://www.opennebula.org/
R 19	MS409 Deploying software into the EGI Production Infrastructure: https://documents.egi.eu/document/478

R 21	MS506 EGI Software Repository:https://documents.egi.eu/document/503
R 22	EMI1 kebnekaise relese notes: http://www.eu-emi.eu/kebnekaise-products
R 23	GGUS: https://ggus.eu/pages/home.php