EGI-InSPIRE

Annual Report on the status of Software Provisioning activity and the work of DMSU

EU DELIVERABLE: D5.3

Document identifier:	EGI-D5.3-375-v10.doc
Date:	**16/05/2011**
Activity:	**SA2**
Lead Partner:	**EGI.eu**
Document Status:	**FINAL**
Dissemination Level:	**PUBLIC**
Document Link:	https://documents.egi.eu/document/375

Abstract

This document is the annual report of the Software Provisioning activity in EGI-InSPIRE. It provides a record of progress of all activity related to Software Provisioning, the Distributed Middleware Support Unit, and the general EGI.eu IT services providing the general IT infrastructure for EGI.eu and EGI-InSPIRE.

I. COPYRIGHT NOTICE

II. DELIVERY SLIP

	Name	Partner/Activity	Date
From	Michel Drescher	EGI.eu	19/02/2011
Reviewed by	**Moderator:** Mario David **Reviewers:** Stuart Kenny Alexander Papaspyrou	LIP TCD TU	26/04/2011
Approved by	**AMB & PMB**		14/05/2011

III. DOCUMENT LOG

Issue	Date	Comment	Author/Partner
1	19-02-2011	First draft	Michel Drescher, EGI.eu
2	20-02-2011	TOC, work assignments	Michel Drescher, EGI.eu
3	25-02-2011	Revised TOC	Michel Drescher, EGI.eu
4	15-03-2011	Contributions (except TSA2.5) integrated	Michel Drescher, EGI.eu
5	17-03-2011	All contributions (plus TSA2.5) integrated	Michel Drescher, EGI.eu
6	22-03-2011	Draft ready for internal review.	Michel Drescher, EGI.eu
7	23-03-2011	Draft ready for external review	Michel Drescher, EGI.eu
8	23-03-2011	Last minute changes for review integrity	Michel Drescher, EGI.eu
9	20-04-2011	Incorporated suggestions from reviewers	Michel Drescher, EGI.eu
10	27-04-2011	Included correction claimed for v9	Michel Drescher, 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 at the following page: http://www.egi.eu/about/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

The Annual Report on the status of the Software Provisioning activity and the work of the DMSU (this document) summarises activities in the first year of the EGI-InSPIRE project in WP5/SA2 “Provisioning the Software Infrastructure”. It covers general EGI.eu (and EGI-InSPIRE) IT support; activities that have taken place to implement, and to enact the core of the Software Provisioning; providing and populating a central EGI software repository; and second level support for EGI users, provided by the DMSU.

TABLE OF CONTENTS

1 INTRODUCTION

The Annual Report on the status of the Software Provisioning activity and the work of the DMSU provides a retrospective summary of all activities that have taken place to:

- Provide IT support to EGI.eu and EGI-InSPIRE
- Design and implement the necessary infrastructure to efficiently execute the necessary processes to provision software taken from external Technology Providers and make it available in a central EGI Software Repository
- Define and integrate the service processes that in total constitute the Software Provisioning activity
- Define and integrate the DMSU processes for incident and problem management
- Integrate and harmonise all processes within the activity to enable efficient information flow between the participating sub-processes.

This document covers activities that took place from and including May 2010, the first month of the EGI-InSPIRE project, up to and including the first two weeks of March 2011 (project month 11).

Chapter 2 gives an overview of the general EGI.eu IT support activities and the provided software tools enabling EGI.eu and EGI-InSPIRE to efficiently use the available IT resources to accomplish their respective goals.

Chapter 3 describes the architecture of the Software Provisioning activity in a nutshell. It provides a description of each technical component that is part of the Software Provisioning infrastructure, including, as the final container of software provisioned for deployment in the EGI, the EGI Software Repository.. This chapter also provides an overview of the software that has been assessed using the developed processes and infrastructure.

Chapter 4 provides an analysis of the software that is deployed in the EGI from the perspective of incident management in the DMSU.

Chapter 5 gives an outlook on necessary changes to further improve and tune the processes for the Software Provisioning activity and the DMSU.

This document ends with conclusions taken from the information provided in previous chapters.

2 EGI IT SERVICES

IT services supporting the day-to-day work of the project were established, continuously supported, and customized when necessary. Those consist of:

Single Sign On (SSO) system - SSO is used to hold elementary user identification data (name, email, affiliation etc.). The LDAP backend provides authentication facilities to all other services mentioned below. SSO also provides management of user groups, which are used for authorization within the services. At the time of reporting there are 1112 user entries and 155 groups.

Web space www.egi.eu and related content management system - OpenCMS [R 12] is used to run the main EGI.eu website

Wiki at wiki.egi.eu - The Wiki is implemented using a dedicated instance with access restrictions used by security groups (such as the SVG). MediaWiki [R 13] was chosen as the backend Wiki software since it is one of the most commonly used systems, and several plugins (e.g. OpenOffice publisher) were deployed. Currently there are 216 content pages.

Document server at documents.egi.eu - The document server [R 14] was deployed, fully integrated with EGI SSO, and several extensions (e.g. introducing document status, improved public/private area transitions) were provided. The document server is used routinely; currently there are 409 distinct documents in more than 2000 revisions.

Meeting and conference agenda server (Indico) - CDS Indico [R 15] was deployed as the meeting agenda server. During the first year the system also underwent a major upgrade when new stable version 0.97 was released. So far, Indico has served over 450 distinct meetings, ranging from small group teleconferences to very large events, such as the EGI Technical Forum 2010.

Tracker for generic use (issue and task tracking) in the project at rt.egi.eu - The out-of-box RT [R 16] was heavily extended to support diverse requirements of different groups. The extensions range from adding queue-specific custom fields to provision of rather complex application logics implemented as overlay code on top of core RT. These extensions allow RT to be used as a tracker system for virtually any project-internal processes. Currently we have 35 active queues, and about 1500 tickets were handled so far.

Mailing lists server at mailman.egi.eu - The mailing list server runs GNU mailman [R 17] integrated with EGI SSO for both user and administrative interface. Currently the server handles almost 100 mailing lists. Most of them are managed through EGI SSO.

All the services are run on redundant servers, allowing fast recovery in case of hardware failure, and are covered with standard backup and monitoring infrastructure. A support team with three members provide fast response and management times to reported issues.

This list of IT infrastructure is run and maintained within the EGI-InSPIRE project. It is complemented by groupware (Email, calendar, CRM, URL shortening service, etc.) by a series of applications based on the Google Apps platform. A collaborative data cloud is hosted by Humyo.com.

3 SOFTWARE PROVISIONING

In the first year TSA2.4 focused on the design and implementation of the first iteration of the New Software Release Workflow (NSRW), with regular reviews provided by TSA2.2, TSA2.3 and TSA1.3. This first iteration has been implemented using existing EGI tools, such as RT and the EGI Software Repository. Important supportive phases in the workflow were identified as targets for automation with the potential to streamline and scale the overall process that spans several tasks and activities. Intensive work has been performed on defining the contents of a release, its structure, and the implications on the layout and setup of the EGI Software Repository. Together with pioneer Technology Providers within EGI-InSPIRE, JRA1 as provider for Operational Tools, and EuGridPMA for providing baseline trust anchors for the EGI Production Infrastructure, several executions of the Software Provisioning Workflow, both manual and automated, were conducted.

A series of meetings with external technology providers such as EMI and IGE identified several issues in the design and layout of the Software Rollout Workflow causing an adaptation of the process and the associated tool integration to be product oriented instead of release oriented.

The following paragraphs describe how each module of the Repository and support tools contribute to NSRW. More technical information about the current implementation of the NSRW may be found in MS504 [R 18].

The Backend system of the EGI software repository handles the business layer of the New Software Release workflow. It is responsible for sanitising the input coming from RT and for storing new software releases in its data store so that it can be passed through the workflow according to instructions given by RT. It supports concurrent provisioning of more than one major distribution version at a time, and even more than one minor or revision release of the same major distribution release. The workflow, and the backend process emergency releases that fixes critical security vulnerabilities, in a similar, yet time-sensitive manner. Based on the information a Technology Provider specifies in a formal release notification description the backend automatically downloads the software the Technology Provider declared in that notification, and provides temporary repositories for Quality Criteria and StagedRollout.

The portal for EGI Software Repository is deployed as a Wordpress site [R 19], available at http://repository.egi.eu. The site shares the same look and feel with the main EGI website, http://www.egi.eu, and offers general information about the repository. In contrast to the repository backend the frontend is designed as a news portal serving dynamic content such as software release announcements, release schedule updates etc. to the wider EGI communities. This news portal nature of the frontend combined with existing expertise with the available frameworks made Wordpress a natural choice. The portal serves as an entry point providing a user interface where users can browse through the releases of popular grid middleware software that have reached the StageRollout and Production stages of the EGI workflow as well as preliminary installation guides.

Both the repository and frontend portal are mirrored and synchronised in a frequent basis to three different web-server installations. Each of these can work independently serving packages and providing portal access. A round-robin mechanism is applied in order to create a failover/load balancing service. Access logging information is gathered using two tools: Google Analytics [R 20] and awstats [R 21]. Awstats is deployed on each individual server, collecting access log entries from all three portal web servers. The logging information is statically analysed and post-processed for graphic presentation by the awstats tool, allowing for repository popularity and usage analysis.

The EGI Request Tracker is the core support system driving the software release process. A dedicated queue "sw-rel" holds tickets corresponding to each release, and the "staged-rollout" queue to follow the final step -- the staged rollout on several sites. Each of these tickets records the progress of the release of a given software product. Human actions (e.g. completion of the verification step) are recorded through the RT web interface, and they are reflected in a change of the ticket custom status in general. Such actions trigger calls from RT to the software repository to perform the next step in the workflow (e.g. download software packages from the provider). On completing such an action, the repository calls back RT to push the ticket further, typically towards another human action. Technically the functionality on the RT side is implemented using specific "custom fields" on tickets in the sw-rel queue, for example to hold references to the permanent records of verification reports, and StagedRollout reports. Several custom scripts deployed to this dedicated queue implement the application logic of the Software Provisioning workflow. These scripts are triggered by actions on the ticket, whether automatic actions or performed by humans.

Dedicated EGI SSO user groups are used to control access to the RT queue and tickets as well as access to provisioning reports in the document database. However, reports are at all times configured for public read access so that any interested party may review and comment on them.

3.1 EGI Trust Anchors

The EGI Trust Anchors provide the basis for authentication for all users and resources of the EGI infrastructure. In that sense, the EGI Trust Anchors component does not represent software as such, rather a configuration item used to drive further establishment of common trust decisions across the EGI. In order to verify the release, Quality Criteria were specifically defined in the egi.eu wiki to ensure the correctness and validity of the CA certificates included in the package.

The EGI-InSPIRE internal provider of those trust anchors felt the need to assure a certain level of quality checks before the trust anchors were installed in the systems since, other infrastructural software components introduced tight dependencies to those CA certificates. To prevent disruption of the infrastructure, the EGI Software Provisioning process was chosen as quality assurance effort. At the same time the EGI Trust Anchors packages was well understood by all participants to start using the process, learn from mistakes and train staff on using the workflow and its associated tools. The following sections provide summarised records of the formal Quality Criteria Verification phase of the full Software Provisioning Workflow.

3.1.1 Summary of verification activities

The verification process of EGI Trust Anchors 1.38-1 was recorded in RT [R 1] and passed without major issues and produced as result the verification report that is available as an attachment to the ticket and in EGI's document database [R 2] as part of the Stage Rollout report.

The EGI Trust Anchors 1.38-1 release activities were recorded in RT [R 3]. This release was performed after the first release of the Quality Criteria documents. These documents, available in the document database [R 4], replaced all previous Quality Criteria drafts available at the EGI wiki[1] and include a section devoted to the CA distribution. However, the Quality Criteria wiki pages were still available and both verification team and Technology Provider used them initially for checking the correctness of the release instead of the normative Quality Criteria documents [R 4]. This confusion on the authoritative Quality Criteria sources produced some misunderstanding in the verification process as recorded in the RT ticket. In order to clarify which are the Quality Criteria documents, all the older wiki pages with drafts of the criteria have been set as deleted in the wiki.

The final verification report and executive summary of this release are permanently available in the document database [R 5], although initially they were only available as attachments to the ticket.

[1] https://wiki.egi.eu/w/index.php?title=EGI-InSPIRE:UMDQualityCriteria:CA

3.2 System Availability Monitor (SAM)

The SAM is an infrastructural component that is deployed in many instances across the EGI. Its main purpose across the EGI federation is metering the availability of systems over time. This information is used as an assessment criterion for sites to be either taken into the EGI federation or, in the unfortunate case of slacking availability figures, to be expelled from it.

The SAM was the first real software product that went through the verification process. The preparation of the verification started with the development and definition of Quality Criteria prior to the actual release with the collaboration of monitoring experts from the Operations Community, the Technology Provider (in this case JRA1 of EGI-InSPIRE) and the Quality Criteria definition team. As with the Trust Anchors, the criteria were initially made available in the egi.eu wiki[2] and later as part of the first Quality Criteria release.

3.2.1 Summary of verification activities

The verification of SAM update 6 was tracked in RT [R 6]. As both the Verification team and JRA1 were new to the process and its tools, several attempts of starting the Software Provisioning process were necessary to actually apply it to this SAM release. However, once started satisfactory, the QC Verification of this release concluded without major issues. During the verification, some deficiencies were detected in the definition of the Quality Criteria that resulted in updated criteria for next releases. At that time the process for tracking changes of Quality Criteria was insufficient, and the report templates recording the activity of the verifiers did not provide space for suggestions for changes of Quality Criteria. Specific Quality Criteria for the Compute Element probes that were shipped with the SAM update were changed to cover error and warning messages emitted by those and similar probes. The verification documents are available in EGI’s document database [R 7].

SAM update 7 was tracked in RT [R 8].. This was a release with minor changes and the verification process was fast since most of the checks done for update 6 were still applicable. Verification documents are available as ticket attachments.

While most of the verification of SAM updates 6 and 7 relied on the tests performed by Technology Provider at their systems, the verification of SAM update 9 for the first time included independent installation and verification of the software. The verification of update 9 used the recently released Quality Criteria documents from the EGI Document Database, and was performed without any major issues. As with all verification the resulting reports are permanently available in the document database [R 10].

[2] https://wiki.egi.eu/w/index.php?title=EGI-InSPIRE:UMDQualityCriteria:Monitoring

4 MIDDLEWARE SUPPORT

The Deployed Middleware Support Unit (DMSU) provides support for the middleware used in EGI. The GGUS system is used for providing the support. Users or operators, who are experiencing middleware issues create a ticket in GGUS describing the issue. Initially, the Ticket Process Management unit (TPM) screens the created tickets; if applicable, the TPM unit passes middleware related tickets on to the DMSU. If possible, the ticket is resolved by DMSU, otherwise the ticket is assigned to a specific support unit, which has expertise for the middleware in question. To ensure progress on tickets, a weekly meeting is held with the DMSU assigners and the DMSU task leader. At the meetings unassigned and open tickets are processed, e.g., assigning them or inquiring for more information.

4.1 Middleware support

Initially the number of tickets assigned to DMSU was rather low. The ticket count has however increased steadily, and from late 2010, the ticket count is now more substantial with 40-50 tickets a month.

Table 1: Number of tickets assigned to DMSU since the start of EGI-InSPIRE until 28 February 2011

#	GGUS Support Unit
1	APEL-EMI
1	ETICS
1	gLite_Hydra
1	gLite_Identity
1	gLite_Java
1	gLite_Release
1	gLite_SGE
1	Gridsite
1	Proxyrenewal
1	SAM/Nagios
1	TPM
1	UNICORE-Client
1	UNICORE-Server
2	AMGA
2	ARGUS
2	dCache
2	DGAS
3	ARC
3	EMI
3	MPI
3	VOMS
6	FTS
6	gLite L&B
6	InformationSystem / GIP / BDII
7	VOMS Admin
9	gLite Security
9	lcg_util
9	LFC
10	StoRM
12	gLite YAIM
14	CREAM-BLAH
14	DPM
17	Information System
21	DMSU
21	gLite WMS
41	APEL

Table 1 shows an uneven distribution over the different products. Furthermore the tickets are focussed around gLite middleware components. A metric that is not directly measured is the nature of the supports issues. A non-exhaustive analysis revealed that the overwhelming majority of the tickets are configuration related, and only relatively few tickets concern actual middleware bugs. The time to respond from support units has not been measured, but is experienced as sufficiently good in general. It is often a much bigger problem to get specific details from ticket submitters. With the current reporting mechanisms in GGUS the ticket overview presented in Table 1 is the level of metrics that is feasible to create.

The reason for having a high number of gLite tickets is probably two-fold. The gLite Middleware is deployed on more sites than ARC, dCache, and Unicore, leading to more tickets. The large share of configuration related tickets supports this. Furthermore some of the middleware already has existing well-functioning support infrastructure, and it is likely that support happens by these or other out-of-band channels. The reasons for this can be that users and operators are used to the existing systems, and are happy using them. While such tickets are not measurable, it is not overly important, assuming users and operators get proper support. Though clearly not desirable from an EGI point of view, mitigation of this situation is limited, if not impossible at all: A user of the production infrastructure (whether an end-user or a site administrator) cannot be forced to use the offered support infrastructure when operating in a federated, non-exclusive environment. If not, using GGUS and DMSU for reporting the ticket can act as a political incentive, as the stakeholders know that metrics are being extracted about the tickets. Furthermore DMSU provides a single point of entry for support, which can be of great value of new users and operators.

The uneven distribution of tickets per support unit indicates that a small number of middleware components cause a rather large portion of the reported tickets. As most of the tickets are configuration issues, the high number of tickets is not due to software defects as such, but instead more likely to be confusing configuration or lack of documentation. The DMSU communicated this pattern to the TCB which endorsed this as a requirement to the affected Technology Provider to include into the software roadmap.

The uneven distribution of tickets also mean that some assigners have significantly more tickets to handle than others. This is not a problem in itself, but assigners handling none or very few tickets are often missing from the weekly meeting, most likely due to lack of perceived importance, which is not an unfair point of view.

4.2 Tracking Estimated Time of Availability (ETA)

When a ticket is identified as a software issue, the technology provider responsible for the software must provide a fix ETA and version for which the issue will be fixed. A common case is that the issue has already been solved, but either a release has not been created, or the user is running an old version of the software. For the remaining cases, technology providers have usually responded in good manner with an ETA and version for when the issue will be solved. So far, a common reply has been "next EMI release". While next release may sound good, EMI releases are currently so infrequent and are planned to continue that way. This makes the ETA responses less useful.

This analysis is admittedly biased towards EMI as the currently major Technology Provider. One major factor in this is that EMI worked towards is first major release of integrated middleware stacks (EMI 1.0). In parallel, development and deployment of gLite 3.1 and 3.2 took place. So any reported bug was fixed for both EMI 1.0 and parallel gLite releases. While the release date of EMI 1.0 is fixed t30 April 2011, gLite released several updates in the meantime. However, the recording and monitoring of such ETAs completely bypassed the EGI Software Provisioning and monitoring infrastructure simply because the deployment of gLite intentionally did not undergo the EGI Software Provisioning process.

The second reason for this bias lies in the chosen release model of EMI. It closely follows the release models of Linux distributions, with few integrated major releases (such as Scientific Linux 5, 6, etc, Ubuntu 10.04, 10.10 etc), and independent component updates in between. Until the time of writing EMI was not able to deliver any specific plan of component updates in between its yearly major release cycles. Therefore reporting sensible ETAs other than the EMI major releases were not possible.

5 MEASURING THE SUCCESS OF THE ACTIVITY

The EGI-InSPIRE project has established a series of metrics to gauge the success and progress of the project as a whole. A subset of those metrics was set up to measure the progress of the SA2 activity, and shown for the first three project quarters below:

Table 2: Metrics for the SA2 activity

Metric ID	Metric	Task	Q1	Q2	Q3
M.SA2-1	Number of software components recorded in the UMD Roadmap	TSA2.1	0	0	30
M.SA2-2	Number of UMD Roadmap Capabilities defined through validation criteria	TSA2.2	0	6	17
M.SA2-3	Number of software incidents found in production that result in changes to quality criteria	TSA2.2	0	1	0
M.SA2-4	Number of new releases validated against defined criteria	TSA2.3	0	1	1
M.SA2-5	Mean time taken to validate a release	TSA2.3	n/a	4h	8h
M.SA2-6	Number of releases failing validation	TSA2.3	n/a	0	0
M.SA2-7	Number of new releases contributed into the Software Repository from all types of software providers	TSA2.4	0	1	3
M.SA2-8	Number of unique visitors to the Software Repository	TSA2.4	0	507	412
M.SA2-9	Number of releases downloaded from the Software Repository	TSA2.4	0	0	0
M.SA2-10	Number of tickets assigned to DMSU	TSA2.5	n/a	8	144
M.SA2-11	Mean time to resolve DMSU tickets	TSA2.5	n/a	2d	n/a

The following sections reflect on each metric, its original intention and the applicability in the project.

M.SA2-1: Number of software components recorded in the UMD Roadmap

The UMD Roadmap describes EGI.eu's vision of the development and evolution of software technology deployed in the EGI. The number of software components that are associated with UMD Capabilities in the UMD Roadmap reflects the pool of software provided by Technology Providers. The more components are registered in the UMD Roadmap, the higher the choice of alternative software components for Resource Infrastructure Providers to install.

Even though the level of understanding of the concept of a UMD Capability and the subsequent definitions varies greatly across the Technology Providers represented in the Technology Coordination board, this metric provides a good indication of the solidity and variety of the software pool EGI may draw the contents of the UMD versions from.

The metric is relatively easy to collect during the regular review efforts of the UMD Roadmap itself and will be kept for future activity tracking.

M.SA2-2: Number of UMD Roadmap Capabilities defined through validation criteria

This metric tracks the coverage of UMD Capabilities with Quality Criteria. The goal is that for each UMD Capability a number of Quality Criteria are recorded that together define the quality of software contributed against that UMD Capability. The metric itself is recorded globally, i.e. across all Capabilities defined in the UMD Roadmap.

To understand the implications of any given value of this metric, one needs to know how many UMD Capabilities are defined in the UMD Roadmap. If given as an absolute value, the metric does not capture the case of new UMD Capabilities defined while the number of UMD Capabilities covered by existing Quality Criteria stays the same: the coverage would drop to a lower level, indicating an increased risk of software faults remaining undetected in the Verification step of the Software Provisioning process. However, if the metric were recorded as a relation of UMD Capabilities covered by Quality Criteria against the total number of UMD Capabilities defined in the UMD Roadmap, the described scenario would cause the collected metric to change to a lower value.

Therefore future recordings of this metric will be given as a percentage representing the relation of UMD Capabilities covered by Quality Criteria against the total number of UMD Capabilities defined in the UMD Roadmap.

M.SA2-3: Number of software incidents found in production that result in changes to quality criteria

This metric intends to measure the quality of the Quality Criteria. Incidents found in the EGI should result in a change of the Quality Criteria, which then will be reflected in an adaptation of the Quality Assurance of the Technology Providers.

While it is relatively easy to collect this metric it is difficult to manage or assess the implications of an analysis. Taking into account that no software will ever be totally bug free, the forces that influence this metric are manifold: Software may grow in popularity, exposing it to more usage, hence implicit software testing (i.e. testing by use) which implicitly raises the number of bugs found. Another possible source of influence is the effectiveness or diligence of the Verification and the StagedRollout phases of the EGI Software Provisioning process – no process is perfect, so this metric's accuracy will always be tainted to a certain degree by the diligence exposed by those processes. There are already services and processes installed that take care of the day-to-day management of software bugs and vulnerabilities – the EGI Service Desk providing first-line and second-line support (i.e. TPM and DMSU) for software bugs, and the SVG and RAT for vulnerabilities that are discovered in the deployed software. However, incidents disrupting the production infrastructure, whether only parts or at large, after rolling out the software that successfully mastered Quality Criteria Verification and StagedRollout, require a post-mortem investigation revealing what exactly has happened, including a root cause analysis. The outcome of such post-mortem reports may indicate that relevant Quality Criteria that were applicable to the offending part of the software might not exist, were ill defined or not tested. Those incidents are one of the identified sources that may cause a change of the Quality Criteria.

Therefore this metric will be clarified to take only those incidents into account that were investigated in post-mortem reports, and that indeed caused a change in the Quality Criteria. The change management process for Quality Criteria includes naming the source of the change, including incidents collected in this metric.

M.SA2-4: Number of new releases validated against defined criteria

This metric intends to measure the workload of TSA2.3. Originally, the Software Provisioning workflow was designed to verify full releases of Technology Providers (in a distribution model). The second iteration of the Provisioning workflow however is scoped towards releases of Products (which may be grouped in coordinated releases of a distribution) – which would dramatically increase the value of the metric as opposed to its original meaning.

On the other hand, since no Technology Provider actually has provided any official release as to date, the reported validation effort reflects releases of internal Technology Providers only, which in scope quite exactly match the semantics of the new Software Provisioning workflow.

Therefore this metric should be re-named as “Number of new Product releases validated against defined criteria.” and used and reported as is.

M.SA2-5: Mean time taken to validate a release

This metric measures the responsiveness of the task TSA2.3 and is taken in the units of hours.

Effectively, TSA2.3 uses generic fields on the RT tickets used to track the provisioning progress to record how many hours have been worked on the respective (Product) release.

This metric is not accurately reflecting the different levels of effort necessary to validate a revision release, a minor release or a major release (with increasing levels of effort). The metric was defined when it was not clear as to which release model Technology Providers would follow. It was assumed that Technology Providers would follow the common model of releasing integrated bundles of software updates. In the meantime this is not true for all Technology Providers. Therefore a different model of metering the time spent on Verification is necessary. A fair and reasonably accurate metering model across Technology Providers takes into account that integrated releases of software are almost always a bundle of components, or Products, that are made available for integration into the UMD. As Technology Providers determine the level of integration and product isolation, the chosen frequency and differentiation of software updates directly influences the effort that has to be spent on the Verification. For example, highly integrated and well isolated Products may be released in less frequent intervals, and the Verification effort is well predictable, whereas a software architecture and deployment model exposing a large amount of lower level components as individual updates to EGI.eu will almost always incur higher costs of Verification of said components. Using this approach, the cost of Verification of any software release is much more predictable, allowing the calculation of Verification costs per individual Technology Provider.

Therefore this metric is changed to meter the costs of Verification on the Product/component level. It will be renamed to “Mean time taken to validate a product release” and used and reported as is.

M.SA2-6 Number of releases failing validation

This metric indicates the quality of the Technology Provider’s QA process. It is an indication only as it is inherently impossible to provide total coverage in a verification effort of any given Product under assessment. Therefore bugs that would be caught in the verification of a specific Quality Criterion may slip from a reasonable Technology Provider QA process *as well* as in a reasonable QA process conducted by EGI. However, this metric, in conjunction with metric M.SA2-3 allow for a reasonable analysis of software Quality Assurance process of the Technology Providers.

M.SA2-7: Number of new releases contributed into the Software Repository from all types of software providers

This metric measures how often (Product) releases are contributed to the EGI Software Repository. The current status of implementation of the EGI Software Repository takes into account only software sourced from external Technology Providers (EMI and IGE), and from internal Technology Providers (currently JRA1). It does not take into account releases that were contributed as community efforts that are otherwise unsupported by EGI, simply because this part of the infrastructure is not well defined in terms of implications, responsibility and required access interfaces.

However, this metric is considered useful and will be kept for future performance measurement.

M.SA2-8: Number of unique visitors to the Software Repository

This metric aspires to measure the visibility of the EGI Software Repository in the EGI community.

The description of the metric does not accurately define the scope of the measurement, as to which parts of the EGI Software Repository must be covered in the measurement.

Therefore this metric will be re-scoped to cover only the front-facing part of the EGI Repository, i.e. http://repository.egi.eu, and leaves the back-end serving the actual software packages out of scope.

M.SA2-9: Number of releases downloaded from the Software Repository

This metric measures the popularity of the EGI Software repository by tracking the numbers of releases downloaded.

The predominant means of access of the software package serving part of the EGI Repository is using YUM for downloading RPM packages. However, a Product release consists of more than one package, and one specific package is used in more than one Product. So based on YUM requests into the repository one cannot accurately determine the number of Product releases that are actually downloaded.

Therefore this metric is re-scoped to the "Number of unique visits to the Repository backend" to reflect this metric measuring access to the package serving backend only.

M.SA2-10: Number of tickets assigned to DMSU

This metric measures the use of the DMSU within the EGI community. The scope of this metric is unclear in that it does not clarify whether only newly assigned tickets should be recorded, the total number of tickets that were assigned to DMSU in that period, or the number of tickets that were active and worked on by the DMSU in that period of time.

Although the DMSU reported an imbalance of tickets across components that are deployed in the infrastructure, flexible grouping and reporting mechanisms for further analysis is lacking in the infrastructure. This is a known limitation of the GGUS system, and is taken up by the GGUS development team, including suggestions made by the DMSU.

This metric is fairly easy to generate, and will be subject to an automated collection in the future.

M.SA2-11: Mean time to resolve DMSU tickets

This metric demonstrates the effectiveness of the DMSU for resolving the tickets assigned to the DMSU.

With the growing use of DMSU to report incidents in the EGI, extracting this metric manually from GGUS is infeasible with the current means available. In general, it has been noted that the interface and reporting capabilities of GGUS need to be expanded with a reporting interface to extract not only metrics, but particularly for DMSU an extensive set of reports indicating trends and types of tickets beyond temporal ticket processing data.

Until this feature of GGUS is not present this metric is not feasible to collect manually.

6 PLANS FOR THE NEXT YEAR

6.1 Quality Assurance: Definition of the UMD Quality Criteria

The Quality Assurance team produces the major artefact used in the Software Provisioning Process. The Quality Criteria (QC) documents define all those quality criteria that are applicable for software implementing UMD Capabilities. Consequently the QC documents cover all UMD Capabilities defined in a given UMD Roadmap version. During the first year, the Quality Assurance task has established the lifecycle and roadmap of the Quality Criteria documents. These documents will be released every 6 months in coordination with the UMD Roadmap releases and will be used for verification of the software releases that reach the Quality Control task. A first release of the Quality Criteria documents was made in February 2011, which covered 70% of the identified UMD Capabilities.

The completeness of criteria for UMD Capabilities will be one of the main tasks for the Quality Assurance Team for next year. The missing capabilities are identified and groups of experts in the different areas of middleware have already been contacted in order to produce criteria that meet the requirements of operations and user communities.

Improving the overall quality of the defined criteria will be assured by the periodic communication of changes in the criteria documents and the feedback collection from the Technology Providers through the task force on Software Quality Assurance, consisting of Quality Assurance and Quality Control teams from EGI, and all of its formally appointed Technology Providers (at the time of writing EMI and IGE).

With the deployment of the first software releases in the EGI infrastructure, operations and users community will raise new requirements that will lead to the definition of new criteria or the modification of the previously existing criteria. The Quality Assurance team will impose a control of changes procedure in the criteria definition phase in order to track the source of every modification and hence provide a reliable and solid change management.

6.2 Quality Control: Verification of Conformance Criteria

Verification reports were not included in EGIs document database in the first releases. This issue was partly solved in the latest updates but the verification process must be fully documented in the EGI.eu wiki, and related EGI-InSPIRE milestone documents, for example [R 11] for further reference. To fill this documentation gap the Quality Control task has created a complete set of report templates for each product, but these documents are not public at this moment. The document templates will be made available in the document database and referenced in appropriate places (milestone documents Wiki, etc.). As with the Quality Criteria documents the verification templates will have to undergo regular reviews closely synchronised with the publication cycle of the UMD Roadmap. A minor issue of discussion was the ticket ownership for the Software Provisioning process. This detail controls the responsibility for updating the progress of the provisioning of the software under verification in the corresponding RT ticket. This point should be clarified in the QC verification process to know how and when ticket ownership and SSO group should be changed.

The verification test-bed will be installed and tested before the first EMI release is published, to improve the verification process. This fully virtualised test-bed, based on OpenNebula, will allow the installation and verification of Technology Provider Products in a short period of time. The new cluster includes 46 HP ProLiant SL165z G7 servers providing 1200 cores (2,2GHz) and 2400GB of memory in total.

6.3 Provision of a software repository and support tools

During the first year of the project several meetings and reviews of the implemented Software Provisioning workflow lead to the decision to make a dramatic shift from a Technology Provider Release oriented system to an independent release provisioning system that will be able to integrate products into a Unified Middleware Distribution (UMD).

In order to realise the aforementioned decision the NSRW needs to be adapted to handle Software Products. Two new modules need to be implemented, and integrated into the existing EGI Repository. The first component serves as the workflow entry component, splitting the Technology Provider oriented release declaration into independent descriptions of the Software Components contained in that release. This component takes into account that in the future, Software Products may be available on more than one platform in the same distribution release. The second component will serve as the Provisioning workflow's final phase. Providing a staging area for Software Products that successfully passed Quality Criteria verification and StagedRollout, thus qualifying for inclusion into the UMD. A web interface to that component will allow to create or modify which Software Products will be included in a given UMD update (whether major or minor).

6.4 Distributed Middleware Support Unit

The DMSU is investigating how GGUS may be expanded for better metrics collection and ticket classification. From the perspective of DMSU GGUS is mostly focused on ticket processing and workflows hence of limited usability for data analysis on tickets themselves. Reporting in the current state of GGUS is mostly time-consuming and a manual effort.

To allow for more in-depth analysis of tickets processed by the DMSU other than qualitative indications, suggestions for flexible tagging of tickets assigned to the DMSU were submitted to the GGUS development team and will be monitored to ensure acceptable quantitative reporting tools for the DMSU in the future.

7 CONCLUSIONS

This document provided a summary of activities that took place in the first year of the EGI-InSPIRE project.

The most profound challenge in the first project year was the shift from a computing infrastructure commissioned and maintained for, and within the clear scope of, a funded project with a limited time span, to an infrastructure that is not gauged towards the needs of one specific project or customer base, but must accommodate a broad diversity of users. Along with this fundamental shift, the sourcing and provisioning of software that is deployed on the EGI is designed to change profoundly once the transition from gLite based software delivery to an EGI based delivery if the UMD. Whereas software used to be written, maintained and used for a specific family of requirements, it now must satisfy a whole new set of requirements, grouped around the fundamental question of sustainability of software.

With this, issues such as TCO, ROI and operating margin of software become important to meet the challenges of a sustainable infrastructure. To answer those questions that are investigated in great detail in the EGI Sustainability Plan [R 22], a distinct infrastructure is necessary, however large or small, that enables the maintainers of the infrastructure to extract the figures necessary to underpin the answers to said questions. It is not the selection of a set of processes (such as ITIL, as a whole of only partially) that fits best the needs of the communities and infrastructure. It is the application of the chosen processes onto the existing resources that proves to be the challenge:

- The processes need to be implemented with supportive infrastructure, either off the shelf or, if necessary, custom made.
- Teams need to be educated in the processes to the point that the process enactment (what to do, and which persistent artefacts to produce) becomes a routine activity
- More importantly, teams need to understand why certain processes are implemented, and the respective goals, in order to add value to the mere execution of the process (for the sake of the process).

It is these three issues that prove to be the challenge of providing an integrated and sustainable activity to ensure constant improvement of Software Quality.

On that background, the first project year has seen the start of the activity as a collection of relatively independent tasks. Partially this was due to staffing issue at EGI.eu and project partners, which were resolved only after half of the year already had passed (EGI.eu), or near the end of the first project year (CESGA). With acquiring experience in the first implementation of the Software Provisioning processes came the insight that processes had to change. Being shared and supported across all activities in SA2, the need to change processes initiated and drove cohesion between the tasks that ultimately contribute to an integrated Software Provisioning service to the EGI community. The DMSU, operating at the seam of EGI Production Infrastructure and Technology Providers, is in the process of being integrated into the overarching goal of Software Quality Assurance. However, GGUS as the main means for DMSU operations is currently not in the place of providing the necessary tools (e.g. a Knowledge Base of past and current issues, including sophisticates search facilities) for the DMSU to efficiently provide feedback for the relevant stakeholders (such as the TCB, the TPM, the Technology Providers as 3rd level support) in this process while all the way staying within the budgeted efforts.

8 REFERENCES

R 1	https://rt.egi.eu/rt/Ticket/Display.html?id=460
R 2	https://documents.egi.eu/secure/ShowDocument?docid=263
R 3	https://rt.egi.eu/rt/Ticket/Display.html?id=1125
R 4	https://documents.egi.eu/document/240
R 5	https://documents.egi.eu/document/338
R 6	https://rt.egi.eu/rt/Ticket/Display.html?id=490
R 7	https://documents.egi.eu/document/208
R 8	https://rt.egi.eu/rt/Ticket/Display.html?id=626
R 9	https://rt.egi.eu/rt/Ticket/Display.html?id=1281
R 10	https://documents.egi.eu/document/371
R 11	MS508: Software Provisioning Process, to be published in June 2011
R 12	OpenCMS, http://www.opencms.org/en/
R 13	MediaWiki, http://www.mediawiki.org/wiki/MediaWiki
R 14	DocDB, http://docdb-v.sourceforge.net/
R 15	CDS Indico, http://indico-software.org/
R 16	RT, http://bestpractical.com/rt/
R 17	GNU Mailman, http://www.gnu.org/software/mailman/index.html
R 18	MS 504: EGI Software Repository Architecture and Plans, https://documents.egi.eu/document/89
R 19	Wordpress, http://wordpress.org/
R 20	Google Analytics, http://www.google.com/analytics/
R 21	Awstats, http://awstats.sourceforge.net/
R 22	EGI Sustainability Plan, D2.7: https://documents.egi.eu/document/313