

...

The reviewers would also like to highlight a number of areas in which the project has made particular progress compared with the status at the end of EGEE-II:

- Interoperability with other grid infrastructures, notably with the Open Science Grid and the Nordic DataGrid Facility.
- Significant progress in IPv6 compliance, with numerous middleware components adapted and the dual-stack operation of gLite being demonstrated in April 2009.
- Active participation by the project in relevant standards activities, especially those related to production grids and interoperability, and the implementation of OGF standards within the middleware.
- Closer interworking between the middleware engineering and testing activities (JRA1 and SA3), including the co-location of many module teams. This has also led to a better balance in the rate of building and releasing fixes.
- The availability of the “roll-back feature” in the deployment of new grid middleware releases, thus promoting higher levels of reliability.

...

Whilst the reviewers are aware that highest priorities in year 2 of the project will naturally be associated with the preparations for the transition to EGI/NGI, they would like to highlight a number of areas of concern where improvements in the current activities should be sought:

- Software engineering. There is evidence of inadequacies in the current processes, such as the persistent high number of outstanding change requests (including high severity ones), the time to fix bugs, and the inconsistencies in software documentation.
- Metrics. The metrics shown to the reviewers for availability and reliability, and for bug-fix times, give a misleading picture of what the user will typically experience.
- ...
- MPI. There is evidence of a latent user demand for MPI but this feature of gLite has not been implemented in many, if not most, of the sites.

...

[*In the comments to NA1*] Good progress has been made in moving towards a metrics controlled quality assurance culture. The reporting of the metrics from quality assurance is good but there still needs to be substantial progress made to make the metrics chosen reflect in a meaningful way the end user experience of the availability of the service and the quality of the support. As it stands here mid-way there are too many serious inconsistencies between the metrics used by the different WP managers to run their shops.

...

[*In the comments to NA4*] [...] a lack of MPI support within EGEE has been reported by several SDCs and this issue should be resolved before the project ends, to prevent user communities from leaving EGEE.[...] an analysis of application fragility to EGEE middleware upgrades (gLite in particular) ought to be carried out in order to limit the impact of upgrades to user applications and make progress towards achieving EGEE middleware back-compatibility.

...

[*In the comments to SA1*] Monitoring and metrics for evaluating the service have advanced with the key GGUS tool. In order for this effort to give meaningful information to the end-user, the reporting metrics have to be coupled with the part of the Savannah tool which reports on bugs in the distributed middleware.

...

[*Comments to SA3*] The main objective of the SA3 activity is to take gLite middleware code produced by JRA1 and generate documented, well-tested, stable and deployable software releases and distributions from it. Additional goals include improving the multi-platform support of gLite and increasing its interoperability with other grid infrastructures.

Solid progress has been made during the review period resulting in gLite releases 3.0, 3.1 & 3.2, supporting Scientific Linux 3, 4 and 5 respectively. At the same time updates for gLite 3.1/SL4, released on average every two weeks, were extensively deployed across the infrastructure. These addressed numerous change requests and were intended to improve reliability and stability as well as enable a gradual introduction of all services. By and large, the purpose of the releases has changed from introducing major functional changes to implementing bug fixes and stability enhancements. Even though many aspects of software development process have been reviewed and streamlined, a suitable, coherent and comprehensive software life cycle model, implementing a sustainable pace of handling change requests, is yet to be fully adopted. Aside from constituting a good practice in and by itself, such an approach is crucial considering the intrinsic complexity of the grid environment as well as the forthcoming transition to EGI.

As SA3 and JRA1 are tightly interwoven the comments made about JRA1 below regarding clear delineation and interface definition among user support, incident management, change request handling and development priorities apply to SA3 as well, especially with regard to timely resolution of critical severity items.

Likewise, comments made regarding porting to multiple platforms in JRA1 are equally applicable to SA3. Several additional factors hindering the provision of new platforms are of non-technical nature, such as the lack of MAC-OS development systems, lack of manpower or specific expertise, etc.

[*Comments to JRA1*] The JRA1 work package aims to maintain the middleware and evolve the key

services needed for running the infrastructure and focusing on production needs. The work is predominantly focused on support, bug fixes and small enhancements. The introduction of new features is limited to development of a crucial new authorisation service, replacing the existing various ad-hoc mechanisms with a coherent framework, and some work on interoperability with other grid infrastructures through the adoption of established standards when available. Both, the security and interoperability aspects constitute important targets of the project especially in view of the forthcoming transition to a more sustainable structure.

The reviewers would like to commend the significant progress made so far in JRA1. The advancements were made possible through competent and critical management and clear focus on vital aspects and issues by relying on resourceful use of clusters of competence.

On the other hand, a great deal still remains to be accomplished. The transition to EGI will require significant changes to the current approach, such as more effective organisation as well as streamlined processes and procedures, in order to be ready when the EGEE-III project ends. The current model is not sustainable; in the long term the support service offered by developers need to be based on better guarantees than best-effort. The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, broadly referred to as software engineering, is crucial. Product management, software development and configuration management procedures, processes and standards remain to be defined and consistently enforced.

Although much progress has been made, the difficulty of implementing the software process with the available tools is still present according to the report in deliverable DJRA1.1. While the ETICS tools that are now operational have made it possible to discern and express software dependencies much more rigorously than in the past, there are still serious gaps. Even though progress has been made in the porting to multiple platforms and a reference gLite distribution running on Scientific Linux 5 is now available, the methodology for rapid and effective porting to other platforms has not been fully established yet, let alone mastered. Clear delineation and interface definition among user support and incident management (GGUS); ticket and task management (Savannah); and software development, bug management, testing, integration and configuration tools (ETICS) is not apparent. Moreover, the management of requested feature and/or reported problem priorities or severities while taking into account specific user requirements or availability of workarounds is rather rudimentary, requiring better granularity and effective procedures for continuous reassessment and/or reclassification. Furthermore, as the JRA1 management team is well aware, the management of software product requirements is not adequate for a software project of the size and distribution of gLite (a suitable product management framework would cover the product lifecycle from user requirements to release). These are the challenges that the JRA1 management team and TMB need to deal with during the final year of the project, in addition to producing a stable version of gLite for at least one reference platform with a significant reduction of outstanding change requests.