Certification of production-quality gLite Job Management components

P. Andreetto\(^1\), S. Bertocco\(^1\), F. Capannini\(^2\), M. Cecchi\(^2\), A. Dorigo\(^1\), E. Frizziero\(^1\), F. Giacomini\(^2\), A. Gianelle\(^1\), M. Mezzadri\(^3\), E. Molinari\(^3\), S. Monforte\(^1\), F. Prelz\(^3\), D. Rebatto\(^3\), M. Sgaravatto\(^1\), L. Zangrando\(^1\)

E-mail: glite-jobmgmt-devel@lists.infn.it

\(^1\) INFN Padova, Via Marzolo 8, I-35131 Padova, Italy
\(^2\) INFN-CNAF, Viale Berti Pichat 6/2, I-40127 Bologna Italy
\(^3\) INFN Milano, Via Celoria 16, I-20133 Milano, Italy
\(^4\) INFN Catania, Via Santa Sofia 64, I-95123, Italy

Abstract. With the advent of the recent European Union (EU) funded projects aimed at achieving an open, coordinated and proactive collaboration among the European communities that provide distributed computing services, more strict requirements and quality standards will be asked to middleware providers. Such a highly competitive and dynamic environment, organized to comply a business-oriented model, has already started pursuing quality criteria, thus requiring to formally define rigorous procedures, interfaces and roles for each step of the software life-cycle. This will ensure quality-certified releases and updates of the Grid middleware. In the European Middleware Initiative (EMI), the release management for one or more components will be organized into Product Team (PT) units, fully responsible for delivering production ready, quality-certified software and for coordinating each other to contribute to the EMI release as a whole. This paper presents the certification process, with respect to integration, installation, configuration and testing, adopted at INFN by the Product Team responsible for the gLite Web-Service based Computing Element (CREAM CE) and for the Workload Management System (WMS). The used resources, the testbeds layout, the integration and deployment methods, the certification steps to provide feedback to developers and to grant quality results are described.

1. Introduction

Grid middleware distributions are often large software artifacts, which include a set of components each providing a basic functionality, e.g. data storage, authentication and authorization, resource monitoring, and job management. EMI [1] is a collaboration of the three major middleware providers in Europe (Advanced Resource Connector (ARC) [2], Lightweight middleware for Grid computing (gLite) [3] and Uniform Interface to Computing Resources (UNICORE) [4]) and other consortia which aims to deliver a consolidated set of middleware components for deployment in existing distributed computing infrastructures strengthening the reliability of the services and establishing a sustainable model to maintain and evolve the middleware. To release software components satisfying these requirements, the PT model [10] was chosen. The PTs are usually small teams of software engineers responsible for the release of a specific software product fulfilling both the technical requirements of robustness, reliability, integration and the user communities requirements. The gLite Job Management PT is
responsible for the gLite Computing Resource Execution and Management (CREAM CE) [5, 6] and for the gLite WMS [7, 9]. The CREAM CE is a job management web-service used to submit, manage and monitor computational jobs to a Local Resource Management System (LRMS); the gLite WMS is a higher level service meant to provide reliable and efficient distribution and management of end-user jobs on the resources available in the Grid. The job submission from WMS to CREAM is managed by the Interface to Cream Environment (ICE) component [7], which is part of the WMS. To efficiently release CREAM CE and WMS, the job management PT has defined precise processes and procedures and dedicated ad hoc testbeds for the testing and certification chain.

2. Software Quality control flow
The gLite middleware software components certification process occurs through three main steps: certification, staged roll-out and production. Each PT is then in charge of the certification process (described in details in section 3 referring to the gLite Job Management PT). When a software component is declared completely certified, the PT releases it, in conjunction with a certification report, to the gLite Release Team. This team is in charge to formally verify the certification report and to prepare and distribute a new update of the software component. The staged roll-out phase then follows. This phase consists in upgrading services relevant for the release candidate software component in a set of small production sites to proof its functionality and reliability directly in a production environment. These sites, called early adopter sites, report on update and re-configurations and after a period of production usage without troubles, the new release of the software component can be rolled-out to all the production services.

This staged roll-out phase is useful to detect problems that can be seen only in production environments or that are site specific issues, and therefore they could not be detected during the certification phase.

If some installation, configuration or functionality problems are found during the staged roll-out phase, the relevant software component is rejected and it returns to the relevant PT to be fixed.

This paper is focused on the first step of the just described release process: the certification for the gLite Job Management Product Team.

3. Main steps in the certification process
The release of a good quality software component able to satisfy the users from both functionality and robustness points of view requires well defined testing and certification procedures. The gLite Job Management PT certifies its software components following three testing stages.

The first stage is to test (i) installation from scratch, (ii) update of an already installed and running production version, (iii) configuration in both cases (from scratch and after an update) of the software component under test. gLite components are installed/updated using metapackages. A metapackage is a package used to just specify dependencies on the packages that are needed for that software component. Installing a metapackage, all the dependent packages (with well specified versions) are installed. Metapackages allow users to install a complete product, without the need to determine the whole set of dependencies. Metapackages are built using the ETICS (E-infrastructure for Testing, Integration and Configuration of Software) integration tool used for gLite distributions.

Installation and update tests are needed to verify the correctness of the metapackage. The configuration of each gLite software component is done using YAIM (Yet Another Installation Manager), a tool which basically consists of a set of shell scripts used to configure the middleware.

Configuration tests, both for a new installation and for an updated production version of the same component, are needed to check the correctness of the relevant YAIM modules.
The second certification stage consists in a careful automatic and manual check of new functionalities and issues supposed to be addressed in the new version of the component being certified. Test description is in section 5. If in this phase some problems are found, the issues are addressed by the relevant developers, and the process is restarted.

After this second stage, the software is ready to be tested in a simulated production environment.

The third stage of certification consists in running high load and stress tests.

The software component is released in production when the third stage is successfully completed.

4. Testbed layout

To accomplish the described software quality control flow and particularly the certification process, a testbed infrastructure was implemented, which is very flexible from the hardware, software configuration and usage policies points of view.

This testbed is logically composed of two sets of machines, called “testbed A” and “testbed B”. The testbed A is used like an early adopter: when a release candidate of a software component is ready, it is installed, configured and then basic integration and functionality tests are performed. After this first step, the automatic tests to check functionality and backward compatibility are performed. These are the steps one and two of the certification process described above in the section 3. The testbed B is instead used to simulate a production environment: it is geographically distributed, it includes 32- and 64-bits hardware, it contains the main software components included in a grid site all configured to support multiple users and Virtual Organizations (VOs) [11]. The testbed B is used to test robustness of the software under high load conditions sustained for long time, to reproduce production-related situations. This is the step three of the certification process.

![Testbed A and B resources](image)

**Figure 1.** Testbed A: Used for preliminary tests on a release candidate. Update/installation and configuration procedures are tested. Integration, functionality, backward compatibility and regression tests are performed.

Testbed A and B resources are depicted in figure 1 and 2. Both testbeds are geographically distributed in two sites: INFN (Istituto Nazionale di Fisica Nucleare - National Institute of Nuclear Physics) Padova section and INFN-CNAF (INFN Centro Nazionale per la Ricerca e Sviluppo nelle Tecnologie Informatiche e Telematiche - INFN Italian National Center for...
Figure 2. Testbed B: Used to simulate a production environment. Used to test robustness of the software under high load sustained for long time.

Research and Development about Information and Data transmission Technologies) in Bologna, Italy. The batch systems used are LSF (Load Sharing Facility is a computer software job scheduler) and Torque (an open source resource manager providing control over batch jobs and distributed compute nodes, a community effort based on the original project PBS - Portable Batch System, a computer software job scheduler). At INFN Padova both the clusters are reserved to the testing tasks. Torque is divided in small clusters, each with its own batch master (which is at the same time a Computing Element) and few Worker Nodes (from one to three). Torque small clusters are assigned to testbed A or B. LSF is a unique cluster which shares all available LSF Worker Nodes among the two testbeds A and B with a machine dedicated as batch master. LSF configuration features are exploited to split the Worker Nodes in two sets dedicated to the two testbeds. At INFN-CNAF in Bologna there is only a LSF Batch System and the Computing Elements dedicated to the gLite Job Management Product Team testing procedure share the Worker Nodes with the production environment. The service machines (User Interface, Workload Management System and BDII) are for the most part located at INFN-CNAF in Bologna. Only one old SL4 User Interface is located at INFN Padova.

This complex layout is used to have, when needed, the ability to simulate the production environment in all its aspects: the variety of batch systems, the presence of different operative systems, the geographical distribution, the number of users and Virtual Organizations involved, the load and the resource exploitation.

The usage policies of both the testbeds are flexible: during the phases of the certification process described on section 3 the testbed machines are reserved to the certification procedure but out of this period the machines are devoted also to other testing tasks. Specific developer’s tests are performed to verify well defined conditions (e.g. problems of memory management related to high load conditions, proxy renewal procedures in presence of high number of users and jobs, etc.). Sometimes a sub cluster is dedicated, through an ad hoc batch system configuration, to integration tests in conjunction with other (non job-management) product teams. Specific user oriented activities are carried out to test features required by users. In these cases it can happen that some machines are ad hoc installed and configured. All these activities are traced by a system administrator who is in charge to maintain the testbeds.

5. Tests description
There are different kinds of tests used to carry out the certification process. To describe all the involved testing scenarios, the CREAM CE software component certification is a good example.
The tests used to certify the CREAM CE component are:

- Installation/upgrade and configuration tests. They implement the first preliminary step of the certification process already described in section 3.
- Bug fix verification. This process is performed mainly manually. Each bug fix supposed to be included in the version under test is checked following a “verification recipe”: the developers provide a description of the condition that re-produces the bug and the certifiers verify that, in the same condition, the bug is no more appearing.
- Functionality and backward compatibility tests. They are performed running a test suite composed by a set of python scripts that use the CREAM CE command line client tools. Tests are executed both with correct input to verify the normal work flow, and with erroneous input, to verify that proper error messages are returned. Tests are configurable either by options to the script or a configuration file.
- Performance tests. They are run submitting simple jobs with a specific environment configuration. This bunch of tests involves also the Workload Management System and particularly its ICE component since job submission is performed also through this component. In this way the whole job management system is tested as already mentioned, different scenarios are checked with different environment configurations. Following the description of an example test to simulate the production environment to test the QueryEvent operation, an operation used by the ICE to detect CREAM job status changes:

  * 14400 collections each of 20 jobs
  * One collection every 30 seconds
  * Four users
  * max_ice_threads = 10
  * We use these CEs located at INFN Padova and INFN-CNAF:
    - 6 CEs SL5/64b with cream version 1.12 (2 lsf + 4 torque)
    - 4 CEs SL4 with cream version 1.11 (2 lsf + 2 torque)
    - 11 CEs SL4 with cream version 1.12 (5 lsf + 6 torque)
  * Use automatic-delegation
  * The job is a "sleep random(7200)"
  * Resubmission is enabled
  * Use proxy renewal service (myproxy.cern.ch)
  * Lease mechanism is not used

6. Future work
The plan for the future is to consider a Nagios [13] monitoring system for multiple purposes: to monitor machines during tests; to detect optimal tuning of the software and hardware configurations (to be able to provide better guidelines on the topic) and to run functionality tests. This will allow increasing the automation level of the product team certification process.

7. Acknowledgments
The research results of the EMI project are co-funded by the European Commission under the FP7 Collaborative Projects Grant Agreement Nr. 261611

References
[1] EMI project home page, http://www.eu-emi.eu/
[2] ARC home page, http://www.nordugrid.org/middleware/
[3] gLite home page, http://glite.web.cern.ch/glite
[4] unicores home page, http://www.unicore.eu/
[5] CREAM home page, http://grid.pd.infn.it/cream
[6] C. Aiftimiei et al, Design and Implementation of the gLite CREAM Job Management Service, Future Generation Computer Systems, Volume 26, Issue 4, April 2010, pp. 654-667, doi: 10.1016/j.future.2009.12.006.
[7] P. Andreetto et al, The gLite Workload Management System, Journal of Physics, Conference Series, vol 119, Issue 6, 2008, pp. 062007, ISSN 1742-6596
[8] EGEE The Enabling Grids for E-sciencE project http://www.eu-egee.org/
[9] P. Andreetto et al, Practical approaches to grid workload and resource management in the EGEE project, Proc. of CHEP'04 conference, Interlaken, Switzerland, Sept 27 - Oct 1 2004.
[10] Transition to the EGI model for the release process in EGEE year 2 https://edms.cern.ch/file/1041827/1/TransitionToEGIreleasev1.0.pdf
[11] E. Laure, S. M. Fisher, A. Frohner, C. Grandi, P. Kunszt, A. Krenek, O. Mulmo, F. Pacini, F. Prelz, J. White, M. Barros, P. Bunic, F. Hemmer, A. Di Meglio, A. Edlund Programming the GRID with gLite, Computational Methods in Science and Technology 12(1), 33-45 (2006)
[12] CREAM Test work plan https://twiki.cnaf.infn.it/twiki/bin/view/EgeeJra1It/CreamWorkPlan
[13] Nagios Open Source Monitoring http://www.nagios.org