Future perspectives on the middleware release process

Francesco Giacomini
EGEE JRA1

EGEE'09, Barcelona, 23rd September 2009

www.eu-egee.org

Information Society and Media

PTs in EMI

- **Definition**
  - the services/components implementation teams […] responsible to deliver software releases and all associated material. They perform the required technical tasks from design to release through implementation, testing and certification [...]. A Product Team usually has a responsible person [...] to transform the project objectives into concrete software releases
    - Includes third-level support

PTs in EMI /2

- Every PT is responsible to execute its own testing
- All testing is done publicly and transparently, the tests and the test results are stored in the [...] test repository (ETICS)
- PTs using another PT products have a fixed grace period to validate new release candidates before they are released
- If the agreed tests/criteria are not passed, the release is rejected (can happen at various stages)

Release Management in EMI

- **Major releases**
  - once or twice per year, may contain non-backward compatible changes
- **Minor releases**
  - a few times per year, fully backward compatible, may contain new functionality
- **Revisions**
  - every week or two weeks, only bug fixes
- **Emergency**
  - as needed, only specific bug fixes, use emergency release procedures

How to be ready for EMI?

Implementing PTs in gLite

- **Draft proposal available at https://edms.cern.ch/document/1019948**
- **Very similar to the EMI approach**
  - with some more details

gLite versioning

- **major.minor.revision(-age)**
  - Major releases are periodic
- **Major: well identified set of supported platforms and components, internal and external, defining the interfaces**
  - All changes are strictly backwards compatible
- **Minor # increment: a new version of a component with an interface change**
- **Revision # increment: bug fix**
- **gLite ETICS project configurations are locked and versioned**

A new version of a component

- **Configuration locked and built against a project config**
  - Either an old gLite major.minor with a “sufficient” interface...
  - … or the latest, but using constrained dependencies
    - Probably the best option
- **Testing and certification is done by the responsible PT, based on that component testplan**
- **If certification is successful the product goes to production**
  - Almost: Release candidates, grace period, staged rollout, …
  - In general no node-type certification, apart maybe for major releases

Pre-requisites

- **Discipline in release management**
  - Changes (when and where) are priority-driven
    - See proposal on “Problem management and change management in gLite”, https://edms.cern.ch/document/1019911
- **Effective testing**
  - Unit testing
  - System testing
  - Deployment testing
  - Integration testing
  - Interoperability testing
  - Scalability testing
  - ...
- **Re-allocation of resources**
  - Clusters-of-competence → Product Teams

Effective testing

- **The test plan for a product should be**
  - As complete as possible, to cover all the uses of that product
  - Public, to establish trust on that component and on that team
  - As automatic as possible
- **Test beds are critical**
  - ETICS testing
  - Reference testbed: a set of services maintained by PTs, with a production + RC installation
    - Mainly for functional testing
  - Experimental services
    - Alpha testing
  - Pilot services
    - Beta testing

Post-mortem of a recent bug

- **How to re-build a bunch of affected RPMs as quickly as possible?**
  - The Release Manager should be able to do it alone

1. **Identify the affected RPMs and the relevant configurations**
   - Should be possible with ETICS
2. **Clone those configurations, increment the age, lock and build**
   - Lock against which project config?
     - Either the previously-locked-against project config is kept in a locked component config and that one is used...
     - ...or use constrained versions in dyn deps and lock against the latest project config

- **A script can do most (all) of this**

What about starting gLite 4?