Task-Tracking Tools for EGI MU

Aleš Křenek, CESNET

EGEE to EGI Middleware Unit: Services and Interactions, CERN,
April 6–7,2010

Tracked entities

- user wishes
  - informal description, evolving in time
  - map to feature specifications
- feature specifications
  - map to service releases
- software defects
  - tracked externally (sw providers), links to releases desirable
- service release
  - planned in advance
  - made available by the provider
  - major/minor: spawn verification tasks
  - revision: automated tests only
- UMD release/update
  - load-balance MU work
- stage rollout actions
  - per EA site, spawned for each UMD release
- miscellaneous actions

Existing processes

- ARC
  - system architects + diverse developers (including students)
  - technical coordination group (NTCG)
  - binarians: team responsible for release build
  - independent test infrastructure and team
  - "testing campaigns" on release candidates
- gLite
  - high-level coordination in EGEE
  - implemented Product Teams
  - heavy use of Savannah and Etics
  - per-node (deployment type) repositories extensively discussed
  - provision of deployment configuration (node)

Existing processes (2)

- UNICORE
  - no response yet :-(
- VDT
  - integration of components from diverse sources
  - central build from source
  - three-level testing
    - internal
    - validation team (a few sites)
    - end users (more production sites)
  - provision of deployment configuration

Tool choice

- dozens available
  - `List_of_project_management_software` and `Comparison_of_issue_tracking_systems` at Wikipedia
- "why not RT" rather than "choose the best one"
  - long experience and skilled administrator at CESNET
  - integration with GGUS available
  - Is there any important feature available elsewhere and missing in RT?
- shortlisting (open-source only)
  - RT: generic task tracker
  - MantisBT, Trac: mainly bug tracking, explicit support for roadmap
  - Savane: all-in-one, lacks programmatic interface
  - GGUS: strongly user oriented
  - hosted environments (Launchpad, Google Code, Source Forge): not flexible, risk of discontinuation
- all more or less equivalent in functionality, references, support

Tentative conclusions and plans

- go with RT
  - If there are objections and/or alternatives, raise them NOW.
- sandbox installation available shortly (this week?)
- define the release process in terms of specific RT usage
- exercise of hypothetic UMD release to define the process?

- more details and links at `https://wiki.egi.eu/wiki/Middleware:Release_Process`