Committee feedback to SPI

Management and planning

- Need to identify or establish **coordination body for interface to middleware/fabric** for testing and standardization
  - Concern about heterogeneity among computing centers
- Work-plan reasonable (see concerns on SCRAM to CMT migration) but **grave concerns as to manpower** situation which seems critical
  - Some descoping proposed in context of build and release process and documentation BUT
  - Cannot afford further manpower decrease

CERNLIB situation

- Need to review what is still needed, by whom (e.g. generators), as well as by whom it is to be maintained, packaged and distributed

Development and bug tracking

- **Savannah**
  - A success story…
  - Replace cookbook by Twiki page (de facto standard for HowTo documents)?
    - query and link savannah items directly from Wiki documents
    - useful for writing release notes
  - Provide regular (WWW-based?) project tracking reports
    - May help project managers to identify problematic areas
  - No need for G4 migration to Savannah
    - Collaboration agreement with KEK guaranteeing support of current system
    - BUT still need for some gateway mechanism to allow single-entry point for end users
- **Bug prioritization**
  - Additional input from experiments?
    - Use Savannah “voting” mechanism? Will need policy as to who “votes”…

Configuration management

- **SCRAM to CMT migration**
  - Clarify LCGCMT status; extend support agreement?
  - Clarify “internal” vs “experiment-visible” configuration with CMT
    - will CMT users see packages as CMT projects?
  - Dependency handling and granularity
    - ability to build and distribute ONLY what one needs and declare dependencies at such granularity that no global recompilation is triggered
  - CMS concerns about timing
    - gradual migration anyway…

Proposal

- migration to be handled in close collaboration with Architects Forum
- taking into account limitations due to descoping

Release, packaging and installation

- **Further steps to speed up procedure essential**; various proposals
  - Outsource package installation to developers
    - Must however ensure uniformity
  - Centralize and automate re-build system
  - Deploy nightly builds
  - Deploy a continuous build system that follows package dependencies, carries out unit testing, packages build products, produces reports (and alerts in case of failures)
  - Automate test result checking
  - Revisit/rationalize platform support and “retirement”
  - Consider use of auxiliary release management procedures and tools (open vs closed releases, tag collectors etc)
- **Clarify situation wrt external packages**
  - Dependency handling
  - Automated build procedure?
  - Maintain version publication for all external packages; important for experiment integration as well as several Grid middleware tools

Quality assurance, documentation, training

- Automate build WWW page updates
- Clarify situation wrt QMTest
  - Concerns about automation capabilities?
  - Plan to further advocate its use? Maintenance and support?
- Launch automated testing facility that would constantly run high level functional tests (some of which could be provided by the experiments)
  - To a large extent project-specific; may not be feasible under SPI given descoping…
- ROOT doxygen documentation
  - Port existing doc or provide links to native ROOT documentation by generation of doxygen compatible TAG files by the ROOT system
- Reinstate popular Python course in collaboration with outside (experiment) experts