



# Project Management Summary

Castor Development Team

*German Cancio, Giuseppe Lo Presti, Sebastien Ponce*  
CERN / IT

Castor Readiness Review – June 2006



# Outline



- ❖ Introduction
- ❖ Current status and next steps
  - Task planning and tracking
  - Change management
  - Testing and SW quality
  - Build and release process
  - Documentation



# Introduction



- ❖ Project management: one of the weaker areas in CASTOR-dev
- ❖ New “Fabric Development” section in IT/FIO since Q4 2005
  - Bringing together ELFms, Remedy and Castor developments
- ❖ Advantages:
  - Flexibility in reallocating development tasks (pool of developers)
  - Minimise risks of knowledge concentration (technical and project mgmt)
  - Share experience
  - Common tools, methodologies -> improved software process
- ❖ Section-wide task force for addressing SW project management and engineering issues
- ❖ For Castor-dev, tight manpower situation (see Monday’s slides) has so far not allowed much room for preparing and introducing large changes
  - Pragmatic approach for 2006
  - Advanced project status -> improvements in specific areas only
  - Try to leverage as much as possible from existing tools and infrastructures (SPI, ETICS+EGEE)



# Task planning and tracking



## ❖ Previously:

- Tasks defined in multiple places:
  - Savannah task tracker
  - Deployment and morning meeting to-do lists
  - Informal agreements, e-mails, etc.

## ❖ Now:

- New development planning guideline for FIO/FD
  - Work Breakdown Structure based:
    - Task name, description, time estimation, manpower, progress, dependencies, risks+alternatives
  - Regularly examined and updated
  - Weekly written progress reports discussed in section meeting
  - Gantt chart
  - [Put in place](#) for CASTOR-dvp in Q2 2006



# Change Management

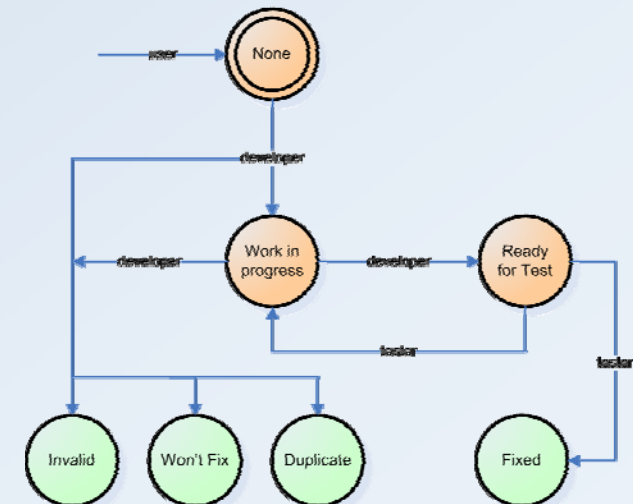


## ❖ Previously:

- Savannah portal available for bug tracking
  - but rarely used
- Not well-defined and unused bug states + transitions

## ❖ Now:

- Enforce Savannah usage for bug reporting
  - Investigating how to link Remedy with Savannah
- Use Savannah also for change requests (RFE's, API changes, etc)
- Defined common bug flow in CASTOR/ELFms
  - Common states and transitions
  - Common definitions for priority, severity





# Build and Release Process

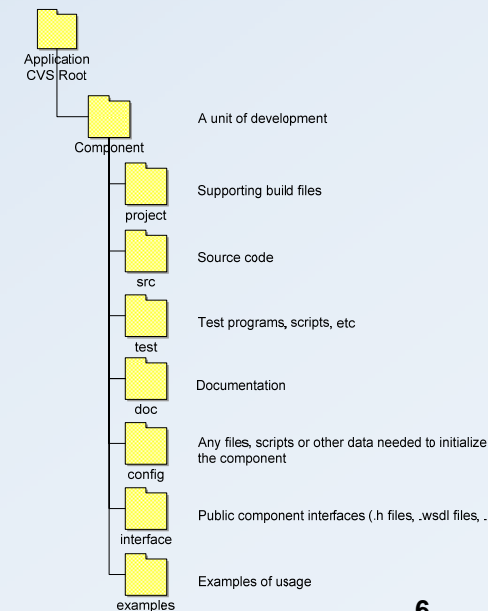


## ❖ Now:

- Manual and time-consuming process
- Complex – only two developers know how to perform a complete build
- Monolithic – independent components need to be build together
- Repetitive – for each OS x platform combination
- Weak control on builds

## ❖ Plans:

- Common CVS structure and build tools for IT/FIO SW (CASTOR, Quattor, Lemon,..)
  - Homogeneous, per-module structure and CVS tags
  - Common make targets for configuring, building, testing, packaging, documentation
  - Evaluate/extend Quattor build tools
- Possibility to build individual modules (tape, clients, etc) taking into account dependencies
- ETICS Integration (see Giuseppe's talk)
  - Currently gathering experience with integration of Quattor into ETICS
  - Enforce testing
  - Nightly HEAD and last tagged autobuilds
  - Integration and release builds
  - Releases based on autobuild (RPM) packages, themselves based on CVS baselines
- Formalize, and automate integration with software repositories
  - CERN-CC (Quattor), Scientific Linux (APT), LCG-AA (AFS)





# Testing and SW Quality



## ❖ Now:

- No formal test plan nor test procedures
- Many components lack test suites; no uniformity
- No software quality metrics used

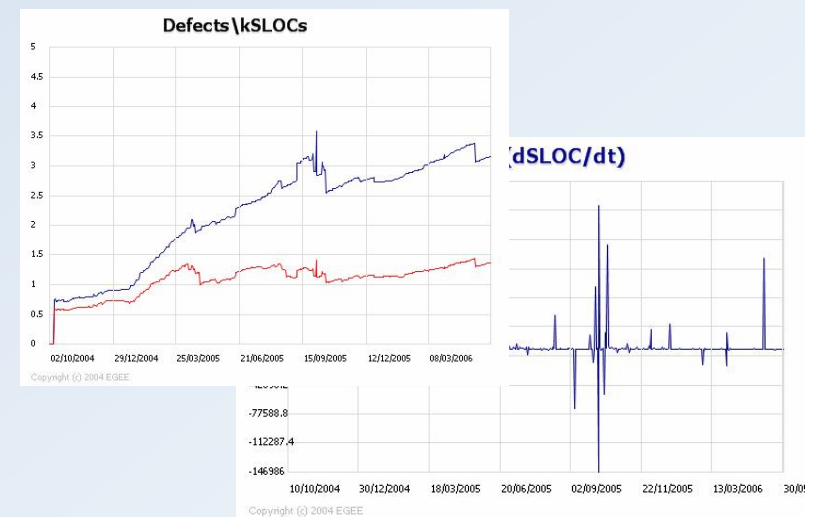
## ❖ Plans:

### ➤ Testing

- Unit tests (XUnit – CppUnit, PyUnit, etc), increasing coverage (found bug->add test)
- Integration and regression tests
- Coding conventions
- Web page with results; e-mail notifications in case of errors
- Automation (via ETICS)

### ➤ SW Quality Metrics: Use Savannah, CVS and ETICS as input for reports and historical trends, e.g.

- Lines of source code (total, per component)
- Code stability (dSLOC/dt)
- Coding convention tests
- Total open/closed bugs
- Bugs by status, severity, category
- Bugs per KSLOC
- Bugs per component and developer
- Percentage of passed/failed unit tests per build





# Documentation



## ❖ Previously:

- CASTOR web site was out of date
- Redesigned used common structure and look&feel for ELFms/CASTOR, cleaned up

## ❖ Plans:

- Coherent templates for design documents, user manuals, install guides, release notes
  - Define strategy on when to use what format (TWiki, web, LaTeX, etc)
- Autogenerated documentation published on the web as part of the build process
  - Code documentation, API's using Doxygen