

Table of Contents

Sustainable Software Lifecycle Management for Grid Middleware: Moving from central control to the open source paradigms.....	1
---	----------

Sustainable Software Lifecycle Management for Grid Middleware: Moving from central control to the open source paradigms

- Presented by Alberto AIMAR
- Type: Poster presentation
- Session: Poster presentations
- Track: Software Engineering, Parallelism & Multi-Core

In the recent years, with the end of the EU Grid projects such as EGEE and EMI in sight, the management of software development, packaging and distribution has moved from a centrally organised approach to a collaborative one, across several development teams. While selecting their tools and technologies, the different teams and services have gone through several trends and fashion of product and techniques. In general the software tools and technologies can be divided in three categories: (1) the in-house development of tools and services, (2) the participation and use of open source solutions and (3) the adoption and purchase of commercial tools and technologies.

The European projects spanned a period of more than a decade. Since the initial choices have been made new tools and paradigms have emerged. The initially adopted more central approach to the overall organization helped the process of a coherent migration to adequate tool chains.

The contribution will show, with concrete examples (mock/koji, JIRA, Bamboo, etc.), how we moved from centralized in-house services and repositories to distributed open source and commercial solutions. We will compare, based on experience, the benefits, shortcomings, costs and the risks of the approaches and the lessons learned from the different solutions.

In addition to the change of technologies also a more loosely coupled development of the different software components has emerged. In this case the adoption of popular public repositories (EPEL, Debian, Maven) together with their policies and standards for packaging and release management allows a less tight synchronization of the release of different packages. Each component can be updated at its own pace because it has to be compliant to the widely- adopted and enforced procedures for building testing and releasing into these repositories.

This presentation will provide some examples and insight on how a change of the core paradigms lead to the move to new tool chains and a loosely coupled collaboration with the focus on the impact on the development, build, release and deployment activities.

This topic: LCG > SwLifeCycle
Topic revision: r1 - 2013-09-25 - AlbertoAimar



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback