AGILE DEVELOPMENT AND DEPENDENCY MANAGEMENT FOR INDUSTRIAL CONTROL SYSTEMS

Authors: B. Copy, M. Mettälä, CERN, Geneva, Switzerland

Abstract

The production and exploitation of industrial control systems differ substantially from traditional information systems; this is in part due to constraints on the availability and change life-cycle of production systems, as well as their reliance on proprietary protocols and software packages with little support for open development standards [1]. The application of agile software development methods therefore represents a challenge which requires the adoption of existing change and build management tools and approaches that can help bridging the gap and reap the benefits of managed development when dealing with industrial control systems. This paper will consider how agile development tools such as Apache Maven for build management, Hudson for continuous integration or Sonatype Nexus for the operation of “definite media libraries” were leveraged to manage the development life-cycle of the CERN UAB framework [2], as well as other crucial building blocks of the CERN accelerator infrastructure, such as the CERN Common Middleware or the FESA project.

Objective

Have a global overview of your software engineering processes while relying on best-of-breed off-the-shelf products

Issue Management

This process focuses on collecting, prioritizing and refining customer demands and internal product quality feedback. Whether use cases, requests for new features or defects identified in existing products, such inputs must be classified and scheduled so as to reduce the risk of unaverted side effects and delays in delivery.

Change Management

This process ensures that changes are auditable, and grouped into coherent sets of modifications. Sets of modifications will eventually compose a release. Change management is often seen as a burden by software developers, but becomes a key activity when it is coupled with release management – for instance, when a release is scheduled to be deployed weeks or months away. The Exhibit framework provides great flexibility exactly what differentiates this version from the one that worked better a few weeks or months ago.

Dependency Management

As agile teams happily release ever improving deliveries, being able to orchestrate dependencies among these deliveries becomes an increasingly difficult task. Establishing clear policies and allowing the definition of dependency ranges becomes a very important aspect of software project management. The Exhibit framework allows to navigate between RDF triples once the SVN commit operations performed in the context of one or more particular software projects.

Conclusions

Integrating software engineering process data sources is useful in order to obtain a global vision of development and release activities. While most engineering process deliver access to structured data, integrating and correlating this information is currently a non-trivial task. While usage of RDF as a common data format has certainly proven a workable approach, certain limitations persist in the current support of RDF. The Exhibit framework for instance was designed highly interactive representation of low volumes of information.

References


JACOW ID WEPKS001

Authors: B. Copy, M. Mettälä, CERN, Geneva, Switzerland

10 - 14 October 2011