Up the Efficiency of your PLM Deployment

by Sameer Deodhar

Like any enterprise software system, product lifecycle management (PLM) deployments are never straightforward. Without thoughtful planning and a systematic approach, deployments end up with numerous problems that erode the potential business value of the system. Even problems that seem small – build failures, missing configurations, omitted software code, or incomplete functionality – can create chaos for a team that is under intense pressure to collaborate and deliver the right solution to the business.

An application lifecycle management (ALM) approach can address many PLM deployment challenges. ALM is the lifecycle management of enterprise software that includes definition, design, development, testing and deployment through a set of defined processes and tools. To be most successful with ALM, there are several leading practices companies should follow.

Agile Release Planning and Execution with ALM

As a basic rule of thumb, companies that are successful with PLM launch their system changes in a formal release process. They use an agile, iterative and collaborative approach very similar to the way that the top software companies themselves develop and launch software. ALM methodologies and tools support an agile way of planning and executing each release.

Regardless of whether you choose a waterfall or agile development method, companies should start with a carefully planned release that includes elements like new features, bug features and data corrections. Each release should include and plan out the following with as much detail as possible:

- Start and end date for the release
- Alignment of development, validation and deployment teams for the execution
- Correct numbering and sequencing of the release per the adopted scheme
- Detailed elements to include (new features, bug-fixes and data corrections)
- Exact timeline for deployment of test, pre-production and production environments
- Qualifying gate criteria for approval of a release as mandated by the validation team

One of the most valuable features of an ALM system is the dashboard-style visibility to project status and activities. ALM dashboards and features facilitate release planning and execution through quick visibility to the following:

- Placeholders for new features, bug fixes and data corrections in the form of tickets or issues which can be tied to specific milestones and releases
- Source code management for configurations and customizations
- Workflow driven ticket and issue lifecycle management
- Software build management for work package development
- Milestone and release management
- Continuous integration and build server

There’s a great example of a dashboard from an ALM tool here.

Release Management Basics

The typical output of the release is the build. A build consists of configurations and customizations that are the solutions for the tickets and issues in a particular release. Ideally, the development team creates these solutions iteratively and the validation team qualifies every build in a test or pre-production environment. Once the build is qualified it is moved to the production environment for general business use.

Companies that use a release management process like this always know what functionality to expect in their development, test, integration and production environments for each release. They can quickly and easily create new test and development environments, since it’s clear which builds should be deployed in which sequences. This means that development teams can start building new functionality faster, and that everything they are working on relates specifically to the requirements in the current release. This eliminates the introduction of unmanaged changes that typically cause chaos and distress.
PLM implementations typically involve a high degree of process redesign and evolving business requirements, so it makes sense to adopt a strategy that focuses on rapid results and embrace a more iterative and collaborative solution design approach. Following an ALM methodology and understanding release management basics during a PLM implementation can lead to significant benefits and more optimal PLM solutions.

The next part of this series will provide insights for managing the numerous elements of a PLM infrastructure. It will provide more details on how an ALM approach can be effectively applied to manage multiple application environments like development, testing and production.

 *Originally published on December 2nd, 2014*

What's your view? Add your question or comment
About the Author

Sameer Deodhar
sameer.deodhar@kalypso.com
Sameer is a senior consultant at Kalypso.

SURVEY
IF I HAD $100 to Maintain My PLM Solution

How would you spend $100 to solve issues around maintaining your PLM solution? Take our survey and let us know.

TAKE THE SURVEY