CDC Unified Process

Project Management

Newsletter

Supporting a Common Project Delivery Framework

Volume 6 | Issue 3 | March 2012

Release Strategy

System development decisions often have deep strategic, business, and cost implications. If not planned correctly ramifications of incorrect decisions may be felt long after the decisions have been made. On the periphery of system development one very important aspect addressing this issue may often be under emphasized or overlooked all together. This is the implementation, execution, and management of an effective software release strategy. This becomes exponentially important when building a custom application because of the many challenges facing development teams:

- Managing development activities
- Managing integration of new development into the existing application
- Managing application dependencies
- Managing new feature/function requests
- Managing compliance with departmental/federal regulations, mandates, and processes
- Managing security requirements (vulnerability assessment testing [VA])

To illustrate how complex a development environment can be, and how important an effective release strategy is, consider the following example. A development team is building a custom application and might be required to release a major new feature or function multiple times a year to meet the demand of its clients. This environment is further complicated by the management of defects, issues, risks, change requests, etc. resulting in the distribution of multiple minor patch or emergency releases. The team uses an iterative approach to development and performs a test build at least once a day. Additionally, the application may have dependencies and interdependencies on other applications. Managing software in this type of environment is often done using a source control management software application (Microsoft's Visual Source Safe [Visual Studio], IBM's Rational Clear Case, Borland's StarTeam, CVS, etc).

Deciding to release an application is often a tradeoff between early release and deferred release. Each alternative has its own sets of pros and cons that must be weighed against each other to determine maximum value for stakeholders. For example, rushing delivery benefits stakeholders with earlier release. However, it may reduce functionality and may decrease overall quality. As a result, future costs may rise in order to fix bugs and distribute patches. Deferring a release allows time for enhanced functionality and improved quality. However, this may incur additional development cost and may result in missed opportunities.

Release Strategy

It is sometimes difficult to accurately determine the most appropriate product release date, feature functionality, associated development costs, quality concerns, etc are all challenges needing to be considered. Proper development and implementation of a release strategy makes building and distributing software easier and more consistent for the performing organization and also outlines how and when product will be made available to the client. A release strategy includes information such as:

- Producing the product
  - ○ Development approach (Iterative, waterfall, etc)
  - ○ Functionality defined for each planned release
  - ○ Operating systems and Program languages used
- Product testing
  - ○ Quality requirements, measures, acceptable tolerances, and testing approach
- Product documentation
  - ○ Guides, release notes, and training material
- Packaging and distribution
  - ○ Physical or electronic distribution
- Migrating data
- Hosting the software
  - ○ Requirements for internally facing systems regarding hardware, software, security, etc.
  - ○ Requirements for externally facing systems regarding hardware, software, security, etc.
- Providing training to end-users

Release Management

Release management approaches may vary from organization to organization. However, regardless of which approach is used a best practice approach to release management is comprised of activities that effectively manage the planning, organization, development, testing, and implementation of new features and functions, defects, change requests, market demands, client expectations, etc into the application being developed. Some of the stages that a software release may go through as it works through the release process may include:

- Pre-Alpha - A pre-alpha release does not necessarily contain completed feature/function. This is often an interim product build, prior to testing, often to validate a particular piece of work or that development to this point hasn't broken the build process.
- Alpha - An alpha release does not necessarily contain all completed features/functions but does satisfy release requirements. An alpha release is often the first internal product build delivered to the testing group. It is often a preliminary build that is only partially complete and typically contains temporary code, comments, product breaks, etc.
- Beta - A beta release is the first product version released outside the performing organization for the purposes of real-world testing. A beta release includes all features/functions but often still contains known issues and bugs.
- Release Candidate - A release candidate contains all completed features/functions and is a product version that has the potential to be a final product. A release is called code complete when it is agreed that no additional "new" source code will be added to the release. However, there may still be changes to fix defects.
- General Availability (Gold) - A general availability release is the final version of a particular product. A gold release is stable and relatively bug-free with a quality exceeding the client's expectations.

For more information about release strategy, planning, or management, the Project Management Community of Practice or the CDC UP please visit the CDC Unified Process website at http://www.cdc.gov/cdcup/.

Project Management Community of Practice

- ***March 01, 2012***
  2012 Project Management Summit
- ***March 23, 2012***
  Leadership
- ***April 27, 2012***
  Contracting
- ***May 18, 2012***
  Cloud Computing at CDC
- ***June 22, 2012***
  Project Change Management
- ***July 27, 2012***
  PM Best Practices – A Panel Discussion
- ***August 24, 2012***
  Enterprise Performance Life Cycle (EPLC)
- ***September 28, 2012***
  A Conversation with CDC Policy Leadership
- ***October 26, 2012***
  The Value of Alternative Analysis
- ***December 07, 2012***
  Managing Risk

For more information on the Project Management Community of Practice visit the PMCoP website at http://www2.cdc.gov/cdcup/library/pmcop/

CDC Unified Process Presentations

The CDC UP offers a short overview presentation to any CDC employee and/or contractor group. Presentations are often performed at your facility, on a day of the week convenient for your group, and typically take place over lunch structured as one hour lunch-and-learn style meeting.

Contact the CDC Unified Process at cdcup@cdc.gov or visit http://www.cdc.gov/cdcup to arrange a short overview presentation for your group.

Contact the CDC Unified Process

The CDC Unified Process Project Management Newsletter is authored by Daniel Vitek, MBA, PMP and published by the Office of Surveillance, Epidemiology, and Laboratory Services.

For questions about the CDC UP, comments regarding this newsletter, suggestions for future newsletter topics, or to subscribe to the CDC UP Project Management Newsletter please contact the CDC UP at cdcup@cdc.gov

http://www.cdc.gov/cdcup/