Process Management in SCM

Author: Nina RajKumar

TABLE OF CONTENTS

Abstract

This paper presents the effectiveness of task-based Process Management in a typical software configuration management scenario. It goes on to highlight the flexibility of an integrated development process across the team environment. This paper also presents the features of some of currently available CM tools that support Process Management.

What is SCM?

Configuration Management is a very integral part of the software development lifecycle. It involves identifying the configuration of the project's product(s) at given points in time, systematically controlling changes to the configuration, and maintaining the integrity and traceability of the configuration throughout the life cycle.

Software Configuration Management (SCM) is defined as the ability to maintain separate versions of software source-code as changes are made to implement new features. While version control is an important component to any SCM strategy, it is by no means the sole criteria for effective SCM. Effective SCM consists of the following elements:

- Version control
- Configuration (Build) management
- Process management
- Workflow management
- Defect tracking
- Audit controls

What is Process Management?

Process management is defined in terms of milestones, rules and responsibilities of the software development process. It ensures that changes introduced to the software are carefully guided through a software development lifecycle with the highest possible level of quality. Today, development scenarios are highly complex involving large projects, multiple team members located across different locations.

In process-intensive areas such as configuration management, which provides a means for traceability and formally controlling the evolution of software, tools provide the key to unifying diverse project teams.

Many of the change request management tools available on the market integrate with the change management tool, and ensure that all bug fixes in the code are controlled and managed. These tools also help in implementing the Change management processes defined by the organization, which enables process management to be done easily and effectively.

A Typical Scenario

Let's examine a typical scenario in which Process Management plays a role. An organization plans to set up a process in order to ensure the quality of developed software is good. There are three groups involved in this process namely development, QA, Build and Production. Their roles and responsibilities are as follows:

Development

1. Implement changes to the source code based on received software change requests (SCRs).
2. Complete unit testing of the changes made.
3. Promote code to QA.

QA

1. Complete testing of changes made by developers.
2. Promote code to Build.

Build

1. Perform build of software.
2. Run verification tests.
3. Promote code to Production.

There are certain ground rules followed in this promotional model. They are:

- Developer cannot make any changes without a valid SCR.
- Developer cannot promote code without management signoff on test completion.
- Tester cannot promote code without management signoff on test completion.
- Build cannot promote code without management signoff on build completion.

These rules are validated using certain checkpoints:

1. All changes must be associated with a valid change request.
2. Developers must completely unit test their changes before sending the code to QA for testing.
3. Signoffs of completed test runs must be obtained before sending changes to BUILD.
    4. Verification Tests on final build mus*t* be completed before releasing to Production environment

Solution for Process Automation using a CM tool:

Advanced CM tools attempt to deliver a complete SCM solution by implementing and providing an open and flexible architecture. These tools provide support for version control and, through the use of triggers, promotion models and build scripts, partial support for process management. They provide an excellent task-based workflow management through the use of module associations. This overcomes the lengthy and error-prone manual procedures followed by organizations to implement a complete solution for process management.

A step by step solution is as follows: -

- Create a project containing the archives for the source code modules.
- Define a promotion hierarchy of
  - ✓ DEV
  - ✓ QA
  - ✓ BUILD
  - ✓ PROD.
- Use permissions to restrict promotions as follows:
  - ✓ Developers promote from DEV.
  - ✓ Testers promote from QA.
  - ✓ Builders promote from BUILD.
- Set up a workflow for the project
- Define fields to identify milestones in the workflow, i.e. Testing completed by QA, management signoff completed, etc.

Once the above setup has been completed the process would be as follows:

1. SCR is created.
2. Manager assigns SCR to developer.
3. Developer checks out modules based on SCR and makes changes.
4. Upon unit test completion, developer updates the appropriate fields and promotes the code to QA.
5. Tester receives notification that unit testing has been completed .Tester performs QA testing and Updates the appropriate fields indicating that the test has been successful.
6. Code is promoted to BUILD by the QA.
7. Build and Verification tests are completed
8. Code promoted to PROD.

CM Tools

1.Continuus CM Synergy - www.continuus.com

Continuus CM Synergy provides:

Complete team development capability: Continuus CM Synergy is a task-based, workflow-oriented environment where logical changes flow easily between groups regardless of their location. It includes all the tools necessary for comparing and merging code created by different developers, and for the immediate sharing of the newly merged code among teams.

Automatic notification of status changes: This provides development team members complete visibility to all changes in status relevant to their elements of a project regardless of their geographic location.

Automation of predictable change management processes: This capability enables timesaving concurrent development through distributed teams.

Development team effectiveness: Continuus CM Synergy promotes an independence that liberates developers from the mundane, repetitive, error-prone tasks associated with traditional software development processes.

Flexibility for supporting any type of project: Continuus CM Synergy can manage any level of "granularity" or configuration in code being passed among development teams, ranging from objects and components to entire applications.

Process flexibility: Teams can adapt to changes in the composition of the software organization, new processes and/or policies as well as temporary or emergency situations that may arise.

Reduced administrative overhead: For administrators, the system simplifies and consolidates administrative tasks. Through an easy-to-use graphical interface for configuration and set-up, the entire change management send and receive process can be automated if desired, resulting in minimal administrative overhead.

Reduced developer overhead: Distribution of software to multiple locations is a simple straight-forward process performed by the change management administrator and can be fully automated. This means that the use and management of DCM is seamless to the developer.

2.PVCS SCM Tools Suite - www.merant.com

LifeLine for PVCS provides such a perfect SCM solution. It addresses the gaps inherent in PVCS by linking the various tools together to effect a tighter level of integration and inter-operability. Using the strength and power of PVCS Tracker workflow manager and coupling it with the proven abilities of PVCS Version Manager, LifeLine for PVCS provides the level of integration and automation necessary to implement complete development task-based process management.

LifeLine for PVCS starts with an organization's defined SDLC and uses it to define the development process. There are no built-in model or rigid implementation standards and guidelines. A software development group tells the tool what its development process is and the tool provides the facilities to administer and manage this process.

It also supports the ability to delay the definition of a process and/or to change that process in mid-stream without causing any disruptions to the development schedule. It allows organizations to achieve this level of task-based process management by extending the reach and breadth of the PVCS Suite in the areas of process management. It provides a range of features that support process management including the ability to:

- Ensure that developers can only checkout modules with valid reasons.
  Developers can only work on SCRs that have been assigned to them.

  Enforce lifecycle rules by linking the PVCS Version Manager promotion groups with particular Tracker SCR *states*. This *state* of development reflects the values of Tracker fields that are designated by the administrator and are completely customizable.

  Ensure that auditors/managers have one centralized location to refer to for checking on development status. This is accomplished by synchronizing the PVCS Version Manager archives and the Tracker database.

- Send email notification as code is promoted from one lifecycle state to the next. This notification also serves to warn developers and concerned parties of changes made to modules other than the ones they are working on. In this way, developers are made aware of potential merge conditions in plenty of time for the merged code to be fully integration- and regression-tested for the release they are working on.

Conclusion:

A CM tool which provides high level of integration and automation necessary to implement a complete development task-based process management is sought by many software companies of today. These tools eliminate manual administrative error-prone tasks and permit procedural flexibility and adapt to changes in the composition of the software organization and its specific process, policy and methodology.

References:

1. *www.merant.com*

2 www.scmint.com

3.www.continuus.com