

Effective Process Configuration Management on AIX



By Tani Haque

Organizations that produce mission-critical and business-critical software on AIX cannot afford to ignore the consequences of releasing a product that incorporates flawed software and runs the risk of recall. Such consequences can be devastating to the customer and lead to major crises when mission-critical and business-critical applications are involved. Process configuration management ensures that the final software release is complete and incorporates all the changes implemented during the life cycle.

Tertiary care hospitals—hospitals of the last resort as they are also called—know that they cannot afford to make a mistake because the patient's life is at stake. A similar scenario, albeit not quite as dramatic, also takes place in the typical software engineering organization producing embedded systems on an AIX® platform. If the embedded software is imperfect, it can potentially kill the life of the product in which it is incorporated. It also can leave the organization with no choice but to recall this product—a very costly embarrassment.

Recalling products becomes necessary in cases where poor software can cause serious damages.

For example, flawed software that is integrated into the inflight navigation system of a commercial aircraft can potentially cause a plane crash. Among many other crises that can occur, software bugs can potentially create havoc in banking transactions, leaving customers frustrated and banking operations halted. For business-critical applications, software defects can threaten the very competitiveness of the organization, causing development delays, productivity dips, and significant cost overruns—not to mention unleashing scores of unsatisfied end users and customers clamoring for their money back.

Why Use Process CM?

Organizations using AIX and other platforms are always looking for ways to achieve greater product quality, reliability, serviceability, and performance in their production of mission-critical applications. Process Configuration Management (CM) can help these organizations produce quality, embedded software applications.

Traditional configuration management primarily provides narrow version control



Tani Haque

This article originally appeared in *CrossTalk, The Journal of Defense Software Engineering*, Volume 10 Number 4, April 1997. It has been updated and is now published with permission of *CrossTalk*.

functionality. This approach to CM is limited in scope because it does not allow process management and provides only unconnected islands of information to developers. Consequently, software engineers cannot bridge their efforts effectively or conduct cohesive development work in parallel and concurrently. Rather, they require automated process workflow and instant access to a centralized database of information about the progress of the development work, including the status of all software changes in the system.

Process configuration management provides the answers in distributed, heterogeneous environments. The advantage of process CM is that it combines process workflow with automation of all software changes across the life cycle—from development through maintenance.

Specifically, process CM automates the integration of version and change management on AIX and other platforms. It also considers the organizational, development, and CM processes of the organization. These processes typically cover all critical issues that impact the development organization, such as ways to achieve contractual requirements, defense standards, and other standards such as ISO 9000 or the Software Engineering Institute (SEI) certification. Other process examples include software engineering procedures, working practices for implementing process CM, and roles and responsibilities assigned to team members.

A process-based CM system can capture such processes. They can be combined to create a control plan, which will provide an important blueprint to effectively build embedded software in a cohesive development environment. The significance is that these processes can be duplicated and improved in the next project life cycle.

Tools of Choice

Currently, there is an array of CM tools from which to choose. The most appropriate ones for controlling the development of embedded applications on AIX are those that integrate change and version management while threading the processes underneath these functions. In a nutshell, these

Configuration Management Tools

Change tracking: Keeps a full audit trail of software change reports including design defects, software bugs, and change requests in the system. So, at any stage in the development cycle, software engineers know which change reports have been fixed and which ones are still pending.

Version management: Allows files to be checked in and out. It tracks the status of all revisions for each individual file and also maintains configuration threads (baselines) of files and their specific revisions. These files can be code or documents.

Automatic building: Allows software engineers to build different versions of the same product. It also enables the automatic control of derived files such as executables.

Software distribution: Enables developers to distribute software changes in parallel and concurrently across the development life cycle.

Process workflow: Allows software changes to be channeled automatically to the appropriate software development team member in the sequence of events of the development life cycle.

Incidents management: Allows software engineers to keep track of all the incidents raised in the system. This facilitates communication with users and customers via E-mail notifications about the status of these incidents.

Figure 1. Tools for configuration management

products automate all the CM functions, including defect tracking, version control, automatic building, software distribution, process workflow, and incident management. See Figure 1.

These process-driven CM solutions provide software engineers with the support they need to successfully implement parallel development, concurrent working, and distributed engineering. The result is an environment where software development is under control and ad hoc development is prevented.

Since these process CM products are based on client/server architecture, they can

be used easily in both homogeneous and heterogeneous environments. These systems enable organizations to absorb workstations into their projects and to have a strategy in place for managing multiple flavors of cooperating UNIX®, OpenVMS, Windows, and Windows NT™ operating systems. This is a major advantage.

The relational databases of such process CM products include information about the application. They also provide control mechanisms to regulate the use of this information, together with the procedures governing product development. For example, this information provides details about the product design, development of software modules and associated documentation, building of configurations, issuing of product items, user-defined rules and procedures, delegation of tasks, and the objects created.

Process CM systems also make it possible to establish special databases to accommodate separate activities implemented by many development groups working on AIX. These databases can feature and control several applications developed in parallel and concurrently.

How to Implement Change Management

The following examples show how software developers and their managers can effectively implement many change management activities using a process CM system in their environments.

Creating Change Documents

Users can easily create change documents, such as defects or change requests. They can, for example, enter various attribute fields that will define the specifics of the change documents, such as severity or importance. Once these attributes have been entered, users can change the application part, which is affected by a change document. When all necessary information is entered, users can then save the change document in the system. This document is now registered and actioned to be processed or authorized by the next person in line in the life cycle of that type of change document.

Opening Change Documents List

At any point in time, software engineers can open and check the change document list. The process CM tool displays various categories that can be further filtered by querying on their attributes. This allows users to restrict the selection of change documents displayed on the screen. This capability is particularly important to managers because it provides high visibility on all change documents that are currently in the system. These managers can instantly assess and report on the status of a project.

This functionality is also critical to software engineers. They can then easily check the list of change documents that fall under their responsibility in a pending list associated with that user. This pending list also provides users with a clear picture of completed tasks and assignments that remain to be done. Ultimately, this increases the accountability of everyone involved in the project, because all team members know their roles and responsibilities during each phase of the life cycle.

A process configuration management solution provides managers with many critical status reports showing the progress of the development effort.

Reviewing Change Documents

A process CM system enables users to review, update, or delegate change documents. The system ensures that only authorized users can implement the changes, since an edit option is only available to users with responsibility to revise these documents. Other users on the team do not have access to this option.

Classifying Change Documents

A process CM system allows users to classify and query newly created or existing change documents according to their relationships to other change document types. These users also can relate application parts to change documents. In this situation, the

hierarchical structure of the application parts represents its functional breakdown. For example, a product might contain hardware and software components represented as two distinct application parts in the system. Each part would then have subordinate parts representing sibling functions.

With this logical structure, managers responsible for modifying the hardware application parts can easily identify revisions they need to implement. And the same holds true for those team members responsible for modifying the software components. Such capability is critical when building embedded software so that all application parts related to the software, hardware, firmware, and documentation can be clearly delineated. So, again, there is no confusion about the managers' responsibilities and which components need to be modified.

Implementing Changes

Software engineers can implement changes to specific files. They do this by requiring that a valid change document be related to an item before that item can be extracted. The result is efficient tracking and the ability to audit change requests while facilitating the implementation of configuration builds and baselines.

Actioning Change Requests

A process CM system allows software engineers to action or move objects, such as change requests from one state in the life cycle to the next state in the life cycle. Using the actioning process, software engineers can advance the change document to a designated state in the life cycle. The system's process workflow capability automatically delivers the change request to the individual(s) responsible for the next action in the life cycle. For example, a change request may be actioned for consideration by a Change Control Board (CCB) authorized for scheduling work. The CCB could action this type of change request so it would appear in the pending list for the designated developers. The developers would, in turn, action the change request to perhaps a "ready for test" state.

This process allows each change document to pass through its normal path of

progression. The system will verify whether the user has the necessary privileges to action the change request form to its intended final life cycle. The system also promotes quality by requesting that the individual or CCB responsible for approving change requests determines that the work is performed correctly.

How to Implement Version Control

An appropriate process CM system for producing embedded software allows developers to implement version and revision control of their design and coding, workset management, build management, release control, maintenance and traceability, and process integration with change management. The version management function automates the versioning, building, release, and distribution of software across the life cycle. The following practical examples illustrate how such a function might be implemented with a process-based CM system.

Users can access a list showing the actual physical directory structure that contains the version objects. Project team members can refer to this physical directory structure to easily perform typical file operations in version management such as checking files in and out. They can also perform version management operations, such as extract (checkout) and return (checkin).

Software engineers using a process CM system can perform version management in parallel and concurrently. They do this through worksets that contain groups of objects. It is worth noting that a typical product may have a workset designated for functional changes, another for addressing maintenance, and yet another for customization.

By creating and maintaining worksets, users can concentrate on their activities and avoid interfering with other developers who are concurrently working on different aspects of the same application. This approach to version management significantly increases the productivity of the team and speeds the entire development effort. It also enables software engineers to deal with the complexities of producing large applications.

Before consolidating their work, users can compare their changes and make decisions about conflicting changes. This ensures that only wanted changes are merged and absorbed into the ongoing development effort. Under the process CM system, changes are merged by merging different worksets together and including them in the mainstream development workset.

Benefits of an Integrated Process CM Approach

A process configuration management solution, which integrates process workflow with the version and change management functions, provides managers with many critical status reports showing the progress of the development effort. Such deliverables may include traceability and management reports, change management reports, configuration management reports, among many other important metrics. This gives managers the ability to choose—at any given point in time—to redeploy staff and resources according to the changing needs of the project.

Software engineers can orchestrate their efforts harmoniously throughout the development life cycle. The result is an integrated approach that allows operations to flow smoothly. The flow of communication also greatly improves laterally and vertically, so everyone can be in sync to produce quality embedded software on AIX.

Finally, a proper CM system provides facilities for telecommuting users to implement configuration management from the Internet and other remote sites. These concurrent activities can be conducted under the umbrella of a set of common processes in a completely controlled and secure environment. This not only fosters cohesiveness in the development environment, but it also optimizes software engineering efforts across distributed, heterogeneous networks.

Conclusion

Today, organizations that produce mission-critical and business-critical software on AIX cannot afford to ignore the consequences of bringing to market a product that incorporates flawed software and risk the chance of recall. Such consequences can be

10 Benefits of Process CM at a Glance

The unique benefits of process CM to the organization include:

- ◆ Better management of software assets
- ◆ Effective parallel, concurrent, cross-platform, and Web developments
- ◆ Defined, repeatable, and improvable processes
- ◆ Higher software quality
- ◆ Proof to the customers that their requirements are being met
- ◆ Smooth process workflow
- ◆ Compliance with ISO 9000 and SEI certification
- ◆ Complete traceability of software changes
- ◆ Critical metrics such as CM reports and impact analysis
- ◆ High visibility over the entire project

devastating to the customer and lead to major crises when mission-critical and business-critical applications are concerned.

When software must be absolutely and positively right, process configuration management is the only way to go. It ensures that the final software release is complete and incorporates all the changes implemented during the life cycle.

Today's competitive market dictates the need to place the development of embedded software systems under active control of process CM. This puts the organization on a path to avoid serious software glitches that can lead to expensive product recalls and result in profit losses and unhappy customers. Call it an insurance policy, if you will. In today's complex development environment, process configuration management is mandatory. It's not just another piece of luxuryware.



Tani Haque, SQL Software, Vienna, VA. E-mail: info@sql.com. Phone: 703-760-0448. Web: www.sql.com. Mr. Haque is CEO and founder of SQL Software, a leader in process configuration management. He has over 20 years of experience in software engineering. He created the first process engine even before the Software Engineering Institute established the Software Maturity Model, and he integrated this leading-edge technology into PCMS .Dimensions, a comprehensive process CM solution by SQL Software.