

Using CM to Recapture Baselines for Y2K Compliance Efforts

Ronald Starbuck
International Billing Services, Inc.

Time is closing in on the year 2000 (Y2K) and those software organizations without product configuration baselines risk failure to meet this hard, "drop-dead" conversion deadline. The purpose of this article is to provide the basic methods essential to achieve the recapture of a product's software baseline to begin the Y2K conversion.

The foremost challenge facing today's software organizations is to make their products Y2K compliant. Failure to achieve compliance will result in immediate product obsolescence at the new millennium's arrival. At the heart of any successful Y2K conversion is the configuration management (CM) system, which has to be flexible enough to accomplish ongoing maintenance of routine fixes while integrating Y2K conversion code into the product [1].

The keystone of CM is the baseline [2]. Organizations that hastily developed software without CM may now be facing Y2K conversion efforts without product baselines. The lack of reference points in this vital area means that an organization cannot effectively support its software products [3].

Watts Humphrey best expresses the importance of CM to Y2K compliance in "Year 2000 Readiness Checklists." [4] He emphasizes that the lack of CM system capability "will most likely [create] problems that could be severe and unrecoverable." Humphrey implores software developers to implement CM immediately rather than wait until their schedule clears.

This article provides the first step to make software Y2K compliant: a method to recapture baselines of systems built without a CM system in place. The recapture process, ironically, requires that a CM system first be implemented.

CM System Defined

The CM system described by Humphrey constitutes more than just the adoption of high-end automated CM tools. Although tools are an important part of a CM system, they do not

compose the entire system; rather, it is based on the integration of basic CM functions—configuration identification, control, status accounting, and audits—into a cohesive process that uses defined procedures, documentation, automated tools, and practitioners. Humphrey observes that "The CM system maintains physical and electronic control of the organization's programs and data." [4] It ensures the integrity and reliability of source code, test data, and documentation over time through use of baselines. These software work products must always be kept current and correct to ensure reliable reproducibility and change.

Effective CM systems always originate from a well-written configuration management plan (CMP) [1]. This plan should be written to provide a clear understanding of how a baseline is created and maintained from the initial identification of components and documentation to their release and subsequent configuration control through a change management process. Details in the CMP should include how audits are performed to maintain the integrity and stability of the software product for further development. When properly executed, the CMP mirrors the organization's software process.

Establish Business Rules

Three vital considerations must be determined by management to set the tone for what is to be done for baseline recapture.

- **Consider impact to business revenue.** Due regard must be given to financial limitations; otherwise, the sequence and scope of the recapture

effort cannot be properly determined.

- **Ensure that software products under contract support are thoroughly reviewed.** Ascertain what the customer purchased for software maintenance to determine which artifacts (software work products) have to be accounted for in the support effort [2]. The range of possibilities can sweep the entire product spectrum from what is needed to support a full 30-year Department of Defense software-intensive system to a short-term interim application.
- **Identify which minimum artifacts are needed to support the products.** Identify the product's name, software performance requirements, software component lists, and documentation created for support. As elementary as this seems, as Humphrey points out, if they are not recorded, they are soon forgotten and lost. Even with automated CM tool assistance in the software process, documentation is not necessarily archived with the source code.

These business rules establish the requirements to be used by the Configuration Control Board (CCB) to determine types of products, required support artifacts, and the order of events to recapture lost product baselines.

Configuration Control Board

The CCB is the forum where business rules are put into action. Its function is central to manage and control baseline recapture efforts. Requisite responsibilities are defined in a CCB charter, which would outline the CCB's functions; thus,

- It guides the entire recapture effort from start to finish.
- It must resolve all problems and situations that arise during the effort.
- To be effective, it must have clear authority and scope.
- Its primary decision-making resource is the CM system.
- When needed, it takes into account organizational management considerations for decision making.
- It orchestrates all activities from the beginning to the approval of the baselines for CM establishment.
- It determines which products should be recaptured, the method to be used, and the order in which they should be done.
- It resolves all issues such as the most suitable product name to be used from among numerous documented aliases. (Multiple aliases often occur when a CM system was not originally in place.)

CCB members should have experience in many areas of software engineering, including CM, so that they can adequately advise the CCB chairman on such matters as product history, costs, contractual aspects, market conditions, quality requirements, and other proceedings. The chairman should have the authority to implement decisions, including those that require funding authorization. CM is the CCB's conduit to accomplish its configuring and processing decisions. All relationships and interfaces should be identified as part of the process plan for recapture as defined in the CMP.

Tools

CCB activities require a planning and decision-making tool to identify, track, and record findings revealed for each product. A generic example of one tool is shown in Figure 1. This checklist functions as a CM process action and status tool used to get tasks accomplished and monitor progress. Software organizations that use automated CM tools could have much of this information already available. A few characteristics of the checklist are as follows:

System Name			Recapture	yes	no	
Approach Used	Contracted	Internal	Category		Assessment	
			Essential	Not Essential	Current	Not Current
Software Version						
Component Composition (Attached)						
Compiler						
Version						
Tools						
Version						
Directory Name						
Date						
Requirements Documentation						
Publication Date						
Revision						
Maintenance Documentation						
Publication Date						
Revision						
Associated Documentation						
Publication Date						
Associated Test Documentation						
Publication Date						
Approve Disapprove						
Signed				Date		

Figure 1. *Baseline recapture checklist.*

- It is initiated at the direction of the CCB for each product under consideration.
- It ensures consistent organization and formatting of the recaptured data for CCB consideration.
- It includes fields for identification of the software product's name, software composition by version, documentation by revision, and any tools necessary to build the product.
- The "Approach Used" field identifies whether the recapture effort is for a contracted or internally produced item.
- Each list is given a unique CM control number.

- Those lists for contracted support products must include all contracted items.

The CM group and CCB are responsible for completing two highly important portions of the checklist. The first categorizes artifacts to be recaptured as essential or not essential. If an item is designated not essential (not mandatory but nice to have), the CCB chairman annotates the checklist accordingly, and the CM group establishes and maintains the item in its historical files for traceability. Artifact items checked as essential must be locatable and traceable. If they cannot be found or were never generated, a statement to that fact should be noted on the checklist.

The second important portion on the checklist is termed "Assessment." It specifies whether an artifact is current. The CM group records this evaluation after completing an analysis of the adequacy of the artifact to be released. When all items on the checklist have been assessed, the completed checklist provides all the information needed by the CCB chairman to approve or disapprove the baseline recapture effort. Approval signifies the sanctioning of the checklist's configuration information so that the CM group can establish the product's baseline and maintain all the processed checklists.

After CCB deliberations, the final decision rests with the CCB chairman to give the CM group the task of proceeding with investigation and coordination or of closing the checklist out as disapproved. To obtain information may require exploration of both on-line and backup storage such as tapes, disks, or cassettes, depending on the system legacy. Research of soft- and hard-copy documentation, preferably from original departmental documentation libraries, and one-on-one interviews with original system development and maintenance employees can be extremely useful to recapture the baseline. An intangible benefit from recapture is to uncover documentation that may still be useful to support the product. This previously unknown information should be entered on the checklist and validated regarding its relevance to the current released prod-

uct. The CCB will rule on its inclusion in the recaptured baseline.

When all research and information gathering is completed for a checklist, the CM group members evaluate it for completeness and write their initials on each item. They must determine whether an item is essential or not essential and current or not current with the released configuration. The completed checklist should include a detailed list of all software components called for in the performance requirements and the design documentation. If the checklist is incomplete, it is returned to the originator for completion. Unknown documents uncovered during research are identified by the CM group and brought to the CCB for resolution. Finally, the CCB must determine the adequacy and currency of recovered artifacts with the released product version. Checklist items not complete or adequately documented may require an impact estimate on cost to bring them into agreement with the released product. If the item was a contract deliverable not maintained since the initial release, a business rule may mandate its recovery. The CCB chairman will make that decision.

Recapture Process

The following is a summary of what, as a minimum, the CM group should do to recapture a baseline foundation to ensure parallel software support.

- Initiate a checklist based on known released products by product name.
- Identify the categories of artifacts for potential identification for baseline recapture and forward them to the CCB chairman for concurrence and direction.
- Follow the chairman's direction and coordinate the investigation. To obtain the information may require exploration of both on-line and backup electronic storage of media, depending on the legacy of the particular system.
- Search through such media types as tapes, cassettes, and disks and review associated hard-copy documentation through various departmental documentation libraries.

- Hold one-on-one interviews (if possible, with original team members). These constituents can provide invaluable insight into what happened and what was produced.
- Review the categories on the form and qualify each item as essential or not essential and current or not current with the released configuration. The checklist should contain list of all software components called for in the performance requirements and the design documentation.

After reviewing the checklists, the CCB returns them to the CM group for their assessment and processing. All artifacts uncovered are then analyzed for their adequacy and currency to the current released product version and provided to the board for reconciliation. Those items that are not current require an impact analysis for the estimated effort and resources to bring them into harmony with the software's released version level. Each item is authorized by the CCB chairman based on business rules, contractual commitments, or other higher-level management or product needs. The following sections provide further insight into what and how the software requirements and version information can be recaptured.

Requirements

It is essential to determine which performance requirements were to be achieved by the software. These specifications are generally produced before the code is designed and written and is the source from which the software design is based and coded. With incremental or evolutionary engineering models, the requirements are successively achieved; therefore, its documentation may be produced in sets. This document or set of documents could be in or part of the Statement of Work or in other stand-alone documents.

If created, an operational user's manual can include the system's operational requirements. Those mature organizations that incorporate software quality assurance (SQA) elements could have additional independent test plans that identify the functions tested and provide verification of produced arti-

facts. It also is possible that SQA has other records that further amplify, describe, or test the product.

Software Version

With these requirements noted in the checklist, the most complex task is to ascertain the relationships between the software artifacts that compose the released product. This decomposition must go down to the version level of build files and source code. CM tools can enable this with minimal effort. The hierarchy between the various components that compose software products can also make discerning the relationships easier. If tools are not employed, there should be some form of software repository from which software components can be identified for each product. The product's software components and the tools used to build them must be defined according to the version level used for baselining [3]. This will require checking records of system administrators to find what was used for a released product.

CM Status Review

Upon completion of each product's recapture effort, all checklists are returned to the CM group for a status review. The CM group determines readiness of the configuration data to be forwarded to the CCB for final baseline sanctioning or to address the particular need at the time to render a decision on some issue or problem. The CM group sorts the checklists into two types of packages: The first includes those deemed as current and ready for final determination by the CCB for baselining. The second are those that need further resolution by either the originator or the CM group. The checklists are verified by the CM group to ensure that the approach used to obtain the information followed the CCB's direction and that the consistency of the information complies with standardization, accuracy, and format for the recorded items.

Of high concern to the CM group is the inclusion of those items it identifies as essential for baselining, as shown in Figure 1. If not provided, a reason must

be attached to the checklist so the CCB can determine the need to re-create missing artifacts. The CM group ensures that each artifact identified has either a version or a revision number for each essential artifact. Any missing revision numbers found by the group are returned to the checklist implementers to obtain them. Those artifacts that are not current have to be acknowledged to the board so that a decision can be made regarding the need for an impact analysis to bring them into conformance.

This task is considerably easier when organizations use a common revision scheme in its version application. There will be those unfortunate instances where no revision status was used. This is quite prevalent in chaotic types of development as identified in the Software Engineering Institute Capability Maturity Model. The CM group also verifies that the tools and the revisions used to create the product were included. When completed, the CM group forwards the package to the board with its recommendations on the adequacy of data recaptured for baselining.

Reconciliation

The CCB reviews all packages and provides recommendations to the CCB chairman for final decision. Products not approved for the baseline could require more information or be rated "disapproved" based on business need. If necessary, the checklist is annotated and returned to the CM group for further processing. When the CCB chairman's signature and date appear on the checklist, it signifies the final decision from which the CM group takes action. For baseline recapture to occur, all product components and documentation not current with released software must be recovered and made current. The decision to proceed may depend on the cost required for the effort. Documentation often is the category most likely to be incomplete, especially when referenced by multiple names or aliases. When checklists are finally sanctioned by the CCB chairman, they are archived into the chosen database or CM tool by the CM group. They may also be stored manually. This act constitutes the re-

establishment of the product's baseline for authentication by the SQA group. It completes the recapture and provides the foundation needed to begin Y2K conversion.

Summary

Humphrey best expresses the importance of a CM system to achieve Y2K compliance when he states that the lack of a CM system will most likely create problems, some of which could be severe and unrecoverable. CM tools alone are not the entire CM system but part of it. Without an effective CM system, software organizations are likely to lose programs, misapply fixes, or use the wrong test or test data. Software organizations that sacrificed CM for expediency or other reasons are extremely vulnerable to problems related to Y2K conversion. An incomplete or absent baseline directly affects a software organization's ability to support its software. The only option available when a baseline does not exist is recapture. This article explained what is necessary from both management and the CM process to accomplish the recapture of product baselines—the key to completing conversion to Y2K. ♦

About the Author



Ronald Starbuck is a configuration manager at International Billing Services and is involved in improving customer systems' CM processes and process infrastructure. He has spent more than 20 years in various government positions and agencies such as lead programmer for the Navy's P-3 Orion Weapons Systems Simulator, configuration manager at the Sacramento Army Depot for test program sets, and a software quality assurance representative for the Defense Logistics Agency.

International Billing Services, Inc.
Mail Stop 7050
5220 Robert J. Mathews Parkway
El Dorado Hills, CA 95762-5712
Voice: 916-857-6857
E-mail: Ronald_Starbuck@uscs.com

References

1. Software Engineering Institute, CMU/SEI-93-TR-25, 1993, pp. A-5, A-22, A-45.
2. Starbuck, Ronald, "The Misplaced Backbone of Software Change Management," *Software QA*, Vol. 4. No. 5, 1997, pp. 19-21.
3. Bersoff, Edward, Vilas Henderson, and Stanley Siegal, *Software Configuration Management*, Prentice-Hall, Englewood Cliffs, N.J., 1980, pp. 85, 274.
4. Humphrey, Watts S., "Year 2000 Readiness Checklists," *CROSSTALK*, Software Technology Support Center, Hill Air Force Base, Utah, July 1998, pp. 24-28.

Coming Events

Sixth Annual ISO 9000 Conference

Dates: March 22-23, 1999

Location: Atlanta, Ga.

Theme: "New Direction for Continuous Improvement"

Sponsor: American Society for Quality

Topics: Upcoming Revisions to the International Organization for Standardization (ISO) 9000 Quality Management Standards, Technological Shortcuts to Accurate Process Documentation, Case Studies of Corrective Action Resulting in New Levels of Effectiveness, Cost Benefits of ISO 9000 to Secure Management Buy-In, and How Even Small Companies Can Use Internal Auditors Effectively.

To receive full conference program, call: 800-248-1946, ask for item B0424. Mention priority code CEJAPR8 to speed order.

Internet: <http://www.asq.org/products/conf.html>

Call for Papers: The Sixth International Symposium on Software Metrics

Dates: Nov. 5-6, 1999

Location: Boca Raton, Fla.

Theme: "Taking the Measure of New Technology"

Topics: The theme of the conference will focus on the application of measurement (through empirical studies) to understand and manage new software technologies (including their related tools and processes), such as commercial-off-the-shelf-based development, software architectures, object-oriented development, and Web-based applications.

Abstracts due: April 1, 1999

Contact: David Card, General Chairman, Software Productivity Consortium, 115 Windward Way, Indian Harbour, FL 32937

Voice: 703-742-7199

Fax: 703-742-7200

Call for Papers: Thirteenth Annual Advanced Software Engineering Education and Training (ASEET) Symposium

Dates: July 26-29, 1999

Location: U.S. Air Force Academy, Colorado Springs, Colo.

Theme: "Ada in the 21st Century: Academic, Government, and Industry Perspectives"

Topics: For a complete list of topics, refer to the Web page listed below.

Abstracts due: April 30, 1999

Full paper due: June 25, 1999

Send all submissions (MS Word or Adobe PDF) via E-mail to Lt. Col. J.A. Hamilton Jr. or Dr. Martin C. Carlisle.

E-mail: hamiltoj@spawar.navy.mil

E-mail: mcc@cs.usafa.af.mil

Internet: <http://www.acm.org/sigada/aseet>

INCOSE '99

Dates: June 6-10, 1999

Location: The Stakis Brighton Metropole Hotel, Brighton, England

Sponsor: The United Kingdom Chapter of the International Council on Systems Engineering (INCOSE).

Theme: "Systems Engineering: Sharing the Future"

Topics: INCOSE '99 seeks to exploit the confluence and synergy that we are seeing between various key issues being addressed by INCOSE worldwide. Come and share in the challenges of the breadth of applications, the diversity of techniques, and the overlap that systems engineering has with other disciplines.

Contact: Cass Jones, Professional Conference Management, Inc., 7916 Convoy Court, San Diego, CA 92111

Voice: 619-565-9921

Fax: 619-565-9954

E-mail: pcminc@pcmisandiego.com

Internet: <http://www.incose.org.uk>

5th Annual Joint Aerospace Weapon Systems Support, Sensors, and Simulation Symposium and Exhibition

Dates: June 13-17, 1999

Location: San Diego, Calif.

Theme: "Making Information Work for the Warfighter"

Sponsors: Embedded Computer Resources Support Improvement Program, U.S. Air Force, U.S. Army, U.S. Navy, and U.S. Marine Corps.

Contact: Dana Dovenbarger

Voice: 801-777-7411 DSN 777-7741 or 801-698-0132

Fax: 801-775-4932

E-mail: dovenbad@software.hill.af.mil

Internet: <http://www.jawswg.hill.af.mil>