Effective Software Configuration Management

Bob Ventimiglia
Lockheed Martin Aeronautical Systems

This article discusses effective software configuration management (CM) as the next step in the evolution of the configuration management practice. Effective CM stresses the technical aspects of CM, extends the concepts of classical CM, and meets the needs of both managers and developers.

In the beginning, there were rocks, dinosaurs, Fred Flintstone, and of course, classical CM, which was good for its time. But its time is now up. Its replacement is what I call "effective" CM. Effective CM is not your grandfather's CM. Effective CM is a nonadministrivia, nonclerical, integrated, cost-effective approach to implementing CM. In summary, effective CM is

- Cost-effective project insurance, i.e., product-integrity focused.
- A manager's and developer's best friend.
- A developer's tool, i.e., developer executed.
- A systems engineering function.
- Difficult to do right.
- No cost when done right (as part of daily development activities).

Effective CM is not

- Just version management.
- Just change management.
- Just a build-and-release function.
- An administrivia function.
- A trivial task.

The Institute of Electrical and Electronics Engineers (IEEE) definition of configuration management includes "a discipline applying technical and administrative direction and surveillance to. …" [1] The key words here are *technical, administrative,* and *surveillance.* Classical CM stresses administrative and surveillance and minimizes technical, which results in processes that developers see as obstacles to getting their job done. Effective CM stresses the technical while using tools to minimize the obstacles that can be caused by the administrative and surveillance activities. The IEEE definition further defines four functions performed by configuration management:

- Identification of configuration items.
- Change control of configuration items.
- Audit of configuration items.
- Recording and reporting of information needed to manage configuration items.

The International Organization for Standardization defines a configuration item as "a collection of hardware, software, and/or firmware, which satisfies an end-use function and is designated for configuration management." [2] The key part here is "which satisfies an end-use function and is designated for configuration management." Classical CM has typically defined configuration items as hard-copy documents and collections of completed source code. As this article will demonstrate, effective CM treats all files produced by the development process as configuration items.[1]

Classical CM focuses on the administrative direction and surveillance aspects of CM and downplays—in some cases ignores—the technical aspects of CM. Because it is labor intensive, classical CM applies an after-the-fact, reactive, "throw it over the wall when yer done" approach to CM for a relatively small number of configuration items. To many developers, this is an obstacle to the development process that adds little value.

Because of the lack of tools with which to manage configuration items, classical CM practitioners were and are highly detail-oriented people with a "green-eye-shades" mindset. I always envision Bob Cratchet in *A Christmas Carol* sitting with quill pen in hand, eyes protected by a green eye shade, slouched over Scrooge's account books, keeping detailed records of Scrooge's business transactions. Cratchet would make an excellent classical CM practitioner. Furthermore, the configuration items managed by classical CM are completed products such as documents and collections of source code with little or no support for in-process management of evolving designs and configurations or the environment used to generate them.

Businesses pressure us to do more with less and to speed development and change cycles. Changes in tool technology and new standards that represent world-class best practices (US 12207 and ISO 12207) require and support a revised approach to CM implementation—effective CM. Effective CM

- Is a system engineering function that is a pro-active, in-yer-face discipline that stresses the technical aspects of CM.
- Satisfies the needs of management as well as developers.
- Is unobtrusive to the point of being 100 percent developer executed.
- Is integral to the software development process.
- Supports the definition and implementation of the software development environment and process.
- Manages changes to all project components as they move through their development and approval cycles.
- Focuses on the "smallest work product of significance" to the development team.
- Follows the files and work products produced by the developer's tools.

Every file produced by the software development process can now be considered a configuration item. The number of objects managed by effective CM is orders of magnitude greater than those managed by classical CM. As a result, it carries the implicit requirement that it must be automated—automated by using CM tools that completely integrate process management (workflow), version, and change management, and that can act

Worth a Look

Configuration Management Readings

For further information about configuration management, the following sources may be useful.

Ben-Menachem, Mordechai, *Software Configuration Management,* McGraw-Hill, New York, 1994.

Berlack, Ronald, *Software Configuration Management,* John Wiley & Sons, New York, 1991.

Buckley, Fletcher J., *Implementing Configuration Management: Hardware, Software, and Firmware,* IEEE Computer Society Press, Los Alamitos, Calif., 1996.

Whitgift, David, *Methods and Tools for Software or Software Configuration,* John Wiley & Sons, New York, 1991.

Coordination for Team Productivity, *Software Configuration Management,* Addison-Wesley, Reading, Mass., 1986.

as the development environment and software repository.

Effective CM incorporates a paradigm shift that extends the concepts of classical CM to in-process management of software development work products, objects, entities, and artifacts, not just product-level documentation of product-level configuration items. It manages requirement identifications, not just requirement specification documents; elements of models as opposed to documentation that contains a completed model; interface messages instead of interface documents. Production of hard-copy documentation is avoided whenever possible by using the managed work products to document the system design. The level of control applied to each managed work product is based on the maturity level of that work product. As work products mature, change authorization migrates from author to software lead engineer to subsystem or system lead to program manager to customer. The level of control is kept to the lowest level practical throughout the products' lifecycle.

The effective CM process

- Supports and is built into the software development process.
- Is a core activity of the software development process.
- Helps manage the evolution of the software development work products.
- Focuses on management of the development process and on the work of and interactions between multiple developers.
- Applies a systems engineering view of CM to the software development process. (It is interesting to note that most industry and military standards identify CM as part of systems engineering.)
- Returns CM to its systems engineering roots and allows effective CM practitioners to act as systems engineers instead of clerks.
- Stresses the technical aspects of CM through an understanding of the development process and most important, by being part of the development process, by building CM into the development process, and by implementing tools that allow the developers to execute CM tasks with minimal intrusion into the development process.

Effective CM reinforces the belief that CM is good and that it is part of the solution, not part of the problem.

The systems integration approach to implementing effective CM begins with the development of a "living" software development plan (SDP). Effective CM practitioners participate in the creation of the SDP and the software development process to ensure that CM is built into the development process—a significant paradigm shift from the classical CM approach. Development methods are defined and a tool suite or system/software engineering environment (S/SEE) is selected that supports the development methods. CM participants ensure that the S/SEE includes tools to implement effective CM, e.g., completely integrated process, workflow, version, and change management CM tools that will be used by the developers. For effective CM to be successful, the CM practitioner must design for success by making the "right way" an "easy way" for developers to accomplish the tasks outlined in the SDP. One must never fail to recognize human nature. Developers will always take what they perceive to be the path of least resistance. Designed correctly, that way will be the effective CM way.

Summary

Because it requires less time on the clerical aspects than classical CM, effective CM allows a pro-active approach to maintain product integrity. It ensures the completeness and correctness of configured work products. It provides a stable environment in which to maintain the integrity of software development throughout the evolution of the project from concept to delivery to customers to product retirement. Its core focus is on problem avoidance and does not have to be cumbersome or complex. Effective CM eliminates the build difference problem where the engineering test build is different from the CM build. It eliminates the "wrong file" in the build problem and other problems related to process breakdowns that manual (classical) CM systems frequently have. Effective CM is program management's eyes and ears into the project. It provides the who, what, where, when, and how and provides complete and accurate in-process software configuration status, state, and volatility of changes, and identifies areas that need management attention. Effective CM is the next step in the evolution of CM practice. ◆

About the Author

Bob Ventimiglia is an internationally recognized expert in state-of-the-science CM. He is currently the environment, tools, and CM lead for the software engineering process department at Lockheed Martin Aeronautical Systems in Marietta, Ga. He leads the effort to enhance and evolve the Hercules C-130 program into an effective CM process. He has been the F-22 environmental control systems and brake control systems software manager, lead F-22 CM engineer, and chairman of the F-22 CM process configuration management system working group. Prior to joining Lockheed Martin, he held key software management and engineering positions with GE Aerospace Information Technology, GE Corporate Engineering and Manufacturing, Sanders Associates, Inc., and Hamilton Standard Division of United Technologies. He has a master's degree in engineering mechanics and a bachelor's degree in aeronautics and astronautics from New York University School of Engineering.

Lockheed Martin Aeronautical Systems
86 South Cobb Drive
Marietta, GA 30063-0685
Voice: 770-494-9791
Fax: 770-494-1345
E-mail: bventimi@hercii.mar.lmco.com (office)
bobev@technologist.com (home)

References

1. IEEE STD 610.12-1990, "IEEE Standard Glossary of Software Engineering Terminology."
2. ISO 10007:1995 (E), "Quality Management – Guidelines for Configuration Management."

Note

1. A more extensive list of definitions of CM—including those by Susan Dart that are coming close to a modern redefinition of CM—can be found at http://www.enteract.com/~bradapp/acme/.

Configuration Management Web Sites

Yahoo configuration management (CM) links
http://www.yahoo.com/Computers_and_Internet/Software/Programming_Tools/Software_Engineering/Configuration_Management

CM Yellow Pages (provided by André van der Hoek)
http://www.cs.colorado.edu/users/andre/configuration_management.html

Brad Appleton's home page and CM links
http://www.enteract.com/~bradapp

Software Technology Support Center (STSC) home page
http://www.stsc.hill.af.mil

Software Engineering Institute CM home page
http://www.sei.cmu.edu/technology/case/scm/scmHomePage.html

CM frequently asked questions from the Usenet CM group
http://www.iac.honeywell.com/Pub/Tech/CM/CMFAQ.html

Other Web Sites of Interest

CrossTalk
http://www.stsc.hill.af.mil/CrossTalk/crostalk.html

Association for Configuration and Data Management
http://www.acdm.org

Electronic Industries Association
http://www.eia.org/gd

Managing Standards home page
http://www.airtime.co.uk/users/wysywig/wysywig.htm

Data Interchange Standards Association
http://www.disa.org

International Organization for Standardization
http://www.iso.ch/welcome.html

Software Productivity Research
http://www.spr.com

Official Department of Defense Stock Point
http://www.dodssp.daps.mil