

Chapter 9

Configuration Management

CONTENTS

| | | |
|------------|--|----------|
| 9.1 | <u>INTRODUCTION</u> | 3 |
| 9.1.1 | <u>ROLE OF CHANGE</u> | 3 |
| 9.1.2 | <u>CONFIGURATION MANAGEMENT (CM)</u> | 3 |
| 9.2 | <u>PROCESS DESCRIPTION</u> | 4 |
| 9.2.1 | <u>FUNCTIONS OF CONFIGURATION MANAGEMENT</u> | 4 |
| 9.2.1.1 | <u>Identification</u> | 5 |
| 9.2.1.2 | <u>Control</u> | 5 |
| 9.2.1.3 | <u>Status Accounting</u> | 5 |
| 9.2.1.4 | <u>Auditing</u> | 6 |
| 9.2.2 | <u>CONFIGURATION MANAGEMENT PROCESS</u> | 6 |
| 9.2.2.1 | <u>Planning</u> | 6 |
| 9.2.2.2 | <u>Establishing Baselines</u> | 7 |
| 9.2.2.3 | <u>Controlling, Documenting, and Auditing</u> | 7 |
| 9.2.3 | <u>UPDATING THE CM PROCESS</u> | 7 |
| 9.3 | <u>CONFIGURATION MANAGEMENT CHECKLIST</u> | 7 |
| 9.3.1 | <u>CM PLANNING</u> | 7 |
| 9.3.2 | <u>ESTABLISHING BASELINES</u> | 8 |
| 9.3.3 | <u>CONTROLLING, DOCUMENTING, AUDITING</u> | 8 |
| 9.3.4 | <u>UPDATING THE PROCESS</u> | 8 |
| 9.4 | <u>REFERENCES</u> | 8 |
| 9.5 | <u>RESOURCES</u> | 8 |

This page intentionally left blank.

Chapter 9

Configuration Management

"But life is change, that is how it differs from the rocks, change is its very nature." – John Wyndham

9.1 Introduction

Change is a constant feature of software development. To eliminate change is to remove the opportunities to take advantage of lessons learned, incorporate advancing technology, and better accommodate a changing environment. Refusal to incorporate change can mean system limitations and early obsolescence, which in the world of technology can sign your system's death certificate before it is born. However, change is not universally benign and must be controlled in its introduction to a project.

9.1.1 Role of Change

All projects have as their main objective to change something. A system is being upgraded or replaced, presumably to provide better or greater functionality, ease of use, reduction of operating expenses, etc. As a project is executed, changes to the initial project plan and products are a natural occurrence. The following are common sources of changes:

- Requirements – the longer the delivery cycle, the more likely it is to happen.
- Changes in funding
- Technology advancements
- Solutions to problems
- Scheduling constraints
- Customer expectations
- Serendipitous (unexpected) opportunities for an improved system

Some of these changes may appear as options while others may be mandated from above or by circumstance, as in the loss of funding. In addition to these sources, many development efforts involve a progressive evolution or elaboration of capabilities and requirements. In projects of this nature the final product is expected to be the result of regular, controlled changes.

While all progress is accompanied by change, not all change is indicative of progress. If not properly handled, change can slip the schedule, affect the quality, and even kill the project. As a project draws closer to its completion, the impacts of change are more severe. [1] Clearly, a mechanism is needed to control change.

9.1.2 Configuration Management (CM)

In software development and other projects, proposed changes must be evaluated to determine their overall contribution to the project goals. Do they lead to improvements or do they ultimately impede or lower the quality of the project? Even those changes that are ultimately beneficial must be controlled in their introduction and implementation. Putting a bigger engine in a plane may improve its capabilities but it cannot be implemented until the aircraft structure has been found capable or been upgraded to support the increased weight and thrust.

Configuration Management is the process of controlling and documenting change to a developing system. It is part of the overall change management approach. As the size of an effort increases, so does the necessity of implementing effective CM. It allows large teams to work together in a stable environment, while still providing the flexibility required for creative work. [2] CM in a software environment is an absolute necessity.

CM has three major purposes: [1]

1. Identify the configuration of the product at various points on time.
2. Systematically control changes to the configuration.
3. Maintain the integrity and traceability of the configuration throughout the product life cycle.

CM accomplishes these purposes by answering and recording the answers to the change questions, who, what, when, and why, shown in Figure 9-1. [1] Being able to answer these questions is a sign of effective CM.

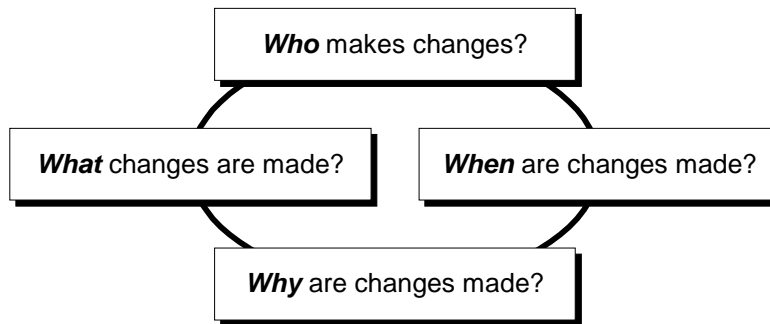


Figure 9-1 Configuration Management Questions

Effective CM provides the following essential benefits to a project:

1. Reduces confusion and establishes order.
2. Organizes the activities necessary to maintain product integrity.
3. Ensures correct product configurations.
4. Limits legal liability by providing a record of actions.
5. Reduces lifecycle costs.
6. Enables consistent conformance with requirements.
7. Provides a stable working environment.
8. Enhances compliance with standards.
9. Enhances status accounting.

In short, CM can provide cost effective project insurance when properly planned, organized, and implemented. [3] It must be integral to your overall project execution, and to your charter/customer agreement. [1] Proposed changes must be dealt with systematically, promptly, and honestly. [1] If the CM process is unreasonable or unresponsive, people will try to circumvent the process, leading to chaos and a loss of the benefits of true CM.

9.2 Process Description

While CM is a major element of a change control program, it is such a multifaceted discipline that it should be considered not simply as another activity, but as a program in and of itself. Establishing an effective CM program requires an understanding of CM functions and of the overall CM process.

The process presented herein is applicable to CM in general, including Software Configuration Management (SCM). If more details are needed for implementing SCM, please see documents specifically relating to SCM in Section 9.5, Resources.

9.2.1 Functions of Configuration Management

Configuration Management is comprised of four primary functions, Identification, Control, Status Accounting, and Auditing. These are shown in Figure 9-2, along with their sub-functions. All CM activity falls within the bounds of these functions.

9.2.1.1 Identification

This function identifies those items whose configuration needs to be controlled, usually consisting of hardware, software, and documentation. These items would probably include such things as specifications, designs, data, documents, drawings, software code and executables, components of the software engineering environment (compilers, linkers, loaders, hardware environment, etc.), and hardware components and assemblies. Project plans and guiding documents should also be included, especially the project requirements. A schema of names and numbers is developed for accurately identifying products and their configuration or version level. This must be done in accordance with project identification requirements. Finally, a baseline configuration is established for all configuration items and systems. Any changes to the baselines must be with the concurrence of the configuration control organization. [2]

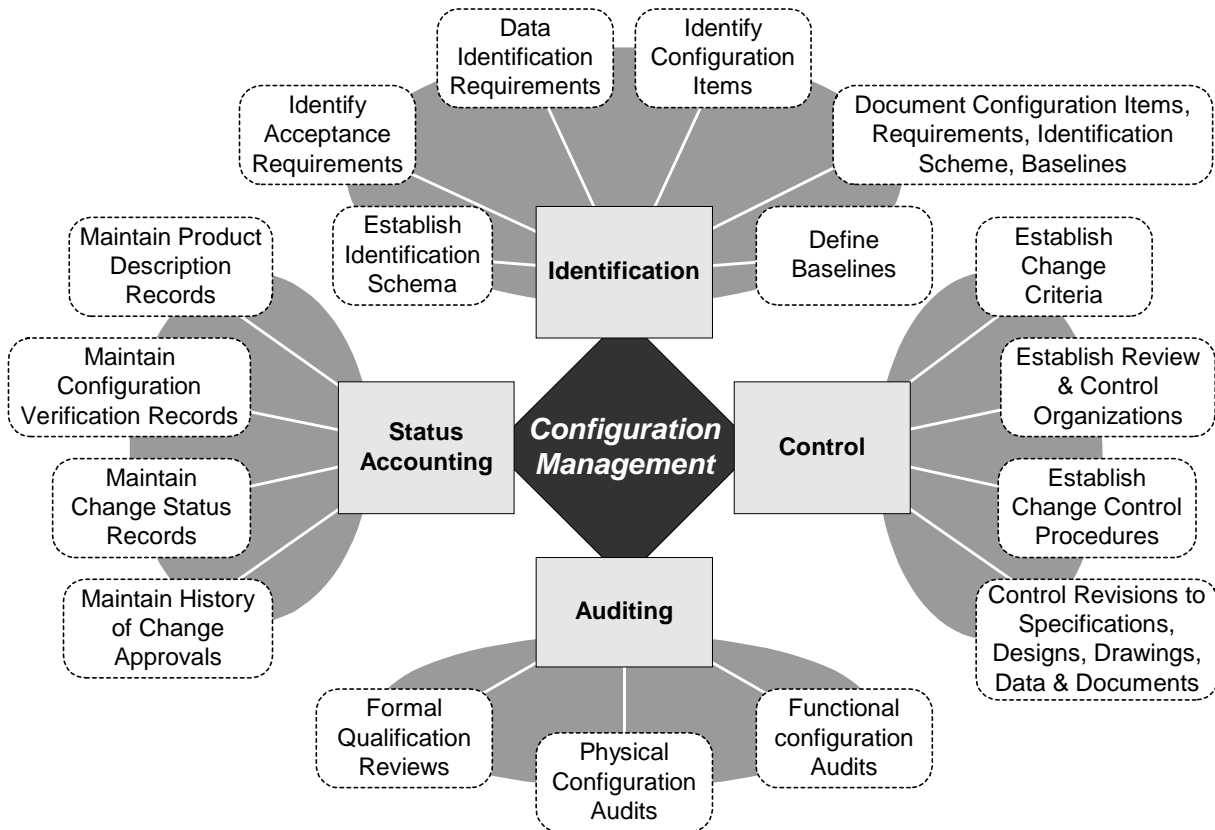


Figure 9-2 Major Functions of Configuration Management [2]

9.2.1.2 Control

Configuration control establishes procedures for proposing or requesting changes, evaluating those changes for desirability, obtaining authorization for changes, publishing and tracking changes, and implementing changes. This function also identifies those persons and organizations that have authority to make changes at various levels (configuration item, assembly, system, project, etc.) and those who make up the configuration control board(s). Additionally, various change criteria are defined as guidelines for the control organizations. Different types of configuration items or different systems will probably need different control procedures and involve different people. For example, software configuration control has different needs and involves different people than communications configuration control and would probably require different control rules and a different control board. [2]

9.2.1.3 Status Accounting

Status accounting is the documentation function of CM. Its primary purpose is to maintain formal records of established configurations and make regular reports of configuration status. These records should accurately describe the product, and are used to verify the configuration of the system for testing, delivery, and other activities. Status ac-

counting also maintains a history of change requests and authorizations, along with status of all approved changes. This includes the answers to the CM questions in Figure 9-1. [2]

9.2.1.4 Auditing

Effective CM requires regular evaluation of the configuration. This is done through the auditing function, where the physical and functional configurations are compared to the documented configuration. The purpose of auditing is to maintain the integrity of the baseline and release configurations for all controlled products. [2] Auditing is accomplished via both informal monitoring and formal reviews.

9.2.2 Configuration Management Process

Understanding of what CM is supposed to accomplish is one thing. Putting it into practice is another. As with most project activities CM begins with planning. With a plan, configuration baselines can be established. Following this initial configuration identification, the cyclical configuration control process is put into motion. These three major CM implementation activities are shown in Figure 9-3.

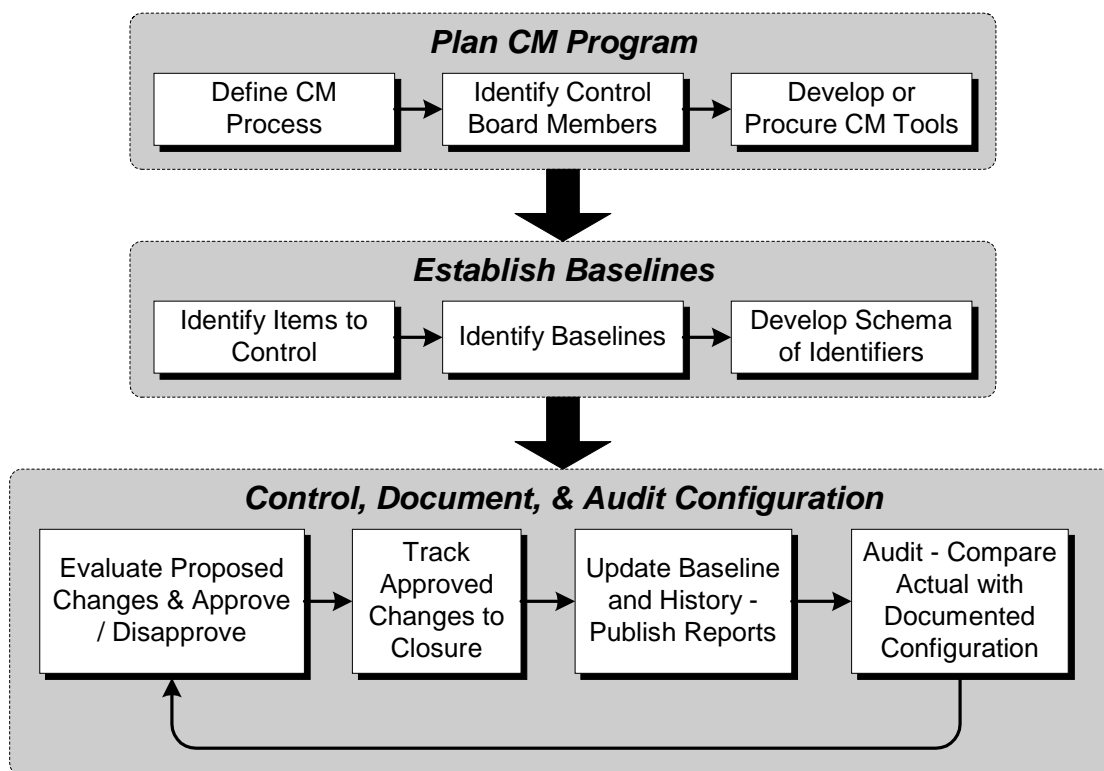


Figure 9-3 Configuration Management Implementation Process

9.2.2.1 Planning

Planning begins by defining the CM process and establishing procedures for controlling and documenting change. A crucial action is the designation of members of the Configuration Control Board (CCB). Members should be chosen who are directly or indirectly involved or affected by changes in configuration. For example, a software configuration change board would obviously be populated with representatives from different software teams, but software affects many more aspects of a project. There should also be representatives from the hardware, test, systems, security, and quality groups, as well as representatives from project management and possible other organizations. Not all changes would be reviewed by this august body. Changes occur at different system levels and affect different portions of the overall system. Many changes will probably only affect a small subset of the system and could therefore be reviewed and approved by a smaller group. Some sort of delineation of change levels should be made during planning to keep change decisions at the proper level. While software CM is essential, there may need to be other

CCBs to control change in other areas of the project. Again, this is something that should be worked out in the planning phase.

Various software tools exist which can facilitate the CM process flow and maintain configuration history. Use of a CM software tool is highly recommended. The temptation will be to choose a tool because it looks good in a demonstration and then build the CM process around it. The process should be defined first, and then a tool chosen to facilitate the process.

9.2.2.2 Establishing Baselines

Once the CM program exists on paper, it must be determined just what configurations it will control. The second major step of implementing effective CM is identifying what items, assemblies, code, data, documents, systems, etc. will fall under configuration control. With the configuration items identified, the baseline configuration must be identified for each item. For items that already exist it may prove to be nothing more than examining or reviewing, and then documenting. For those items that have not been developed yet, their configuration exists in the requirements database or in the project plans. Until they come into physical or software reality, changes to their configuration will consist only of changes to the requirements or plans.

Another essential activity in this step is developing a schema of numbers, letters, words, etc. to accurately describe the configuration revision, or version, for each general type of configuration item. There may be project requirements which dictate some type of nomenclature, or there may be an organizational or industry standard that can be used as the basis for configuration identification.

9.2.2.3 Controlling, Documenting, and Auditing

When the baselines have been established, the challenge becomes one of keeping the actual and documented configurations identical. Additionally, these baselines must conform to the configuration specified in the project requirements. This is an iterative process consisting of the four steps shown in Figure 9-3. All changes to the configuration are reviewed and evaluated by the appropriate configuration control representatives specified in the CM plan. The change is either approved or disapproved. Both approvals and disapprovals are documented in the CM history. Approved changes are published and tracked or monitored until they are implemented. The appropriate configuration baseline is then updated, along with all other applicable documents, and reports are published and sent to affected organizations indicating the changes that have occurred. At selected time intervals and whenever there appears to be a need, products and records are audited to ensure that:

- The actual configuration matches the documented configuration.
- The configuration is in conformance with project requirements.
- Records of all change activity are complete and up-to-date.

The controlling, documenting, auditing cycle is repeated throughout the project until its completion.

9.2.3 Updating the CM Process

It is unlikely a perfect CM program will be assembled during the initial planning stage. There will be learning and changes in the program that indicate a need for adjustments in the CM process. These may be any mixture of modifications to make it more efficient, responsive, or accurate. When changes in the CM process are needed, consider them as you would any other changes, get the approval of all participating organizations, and implement them as appropriate. It would be ironic indeed to have an unchanging change process.

9.3 Configuration Management Checklist

This checklist is provided to assist you in establishing an effective CM program. If you cannot answer a question affirmatively, you should carefully examine the situation and take appropriate action.

9.3.1 CM Planning

- 1. Have you planned and documented a configuration management process?
- 2. Have you identified Configuration Control Board members for each needed control board?

- 3. Has CM software been chosen to facilitate your CM process?

9.3.2 Establishing Baselines

- 4. Have all configuration items been identified?
- 5. Have baselines been established for all configuration items?
- 6. Has a descriptive schema been developed to accurately identify configuration items and changes to their configuration?

9.3.3 Controlling, Documenting, Auditing

- 7. Is there a formal process for documenting and submitting proposed changes?
- 8. Is the Configuration Control Board active and responsible in evaluating and approving changes?
- 9. Is there a “higher authority” to appeal to when the CCB gets “hung,” and can’t come to a consensus?
- 10. Are all changes tracked until they are fully implemented?
- 11. Are all changes fully documented in the baseline documents and change histories?
- 12. Are regular reports and configuration updates published and distributed to interested organizations?
- 13. Are regular audits and reviews performed to evaluate configuration integrity?
- 14. Are configuration errors dealt with in an efficient and timely manner?

9.3.4 Updating the Process

- 15. Is the CM program itself – its efficiency, responsiveness, and accuracy – evaluated regularly?
- 16. Is the CM program modified to include recommended improvements when needed?

9.4 References

- [1] Software Technology Support Center Course: *Life Cycle Software Project Management*, Configuration Management, 9 October 2001.
- [2] *Little Book of Configuration Management*, Software Program Managers Network, November 1998. Download at: www.spmn.com/products_guidebooks.html
- [3] *Program Manager’s Guide for Managing Software*, 0.6, 29 June 2001, Chapter 14: www.geia.org/sssc/G47/SWMgmtGuide%20Rev%200.4.doc

9.5 Resources

CM Today, Configuration management papers: www.cmtoday.com/yp/papers.html

Configuration Management Resource Guide: www.quality.org/config/CMResourceGuideMaster.doc

Configuration Management II Users Group: www.cmiug.com

Crosstalk Magazine: www.stsc.hill.af.mil/crosstalk/

- “Software Configuration Management Terminology”: www.stsc.hill.af.mil/crosstalk/1995/jan/terms.asp
- “Software Configuration Management: Function or Discipline?": www.stsc.hill.af.mil/crosstalk/1995/oct/cmfunct.asp
- “Stop-Gap Configuration Management”: www.stsc.hill.af.mil/crosstalk/1998/feb/stopgapcm.asp
- “Effective Software Configuration Management”: www.stsc.hill.af.mil/crosstalk/1998/feb/effectivecm.asp
- “Configuration Management Web Sites”: www.stsc.hill.af.mil/crosstalk/1999/mar/cmsites.asp
- “Demystifying Software Configuration Management”: www.stsc.hill.af.mil/crosstalk/1995/may/demystif.asp

- “Evaluation and Selection of Automated Configuration Management Tools”:
www.stsc.hill.af.mil/crosstalk/1995/nov/evaluati.asp
- “Software Configuration Management: A Discipline with Added Value”:
www.stsc.hill.af.mil/crosstalk/2001/jul/butler.asp
- “Introducing Process into Configuration Management”:
www.stsc.hill.af.mil/crosstalk/1996/jun/introduc.asp
- “Process-Based Configuration Management”: www.stsc.hill.af.mil/crosstalk/1997/apr/configuration.asp
- “Achieving the Best Possible Configuration Management Solution”:
www.stsc.hill.af.mil/crosstalk/1996/sep/achievin.asp
- “A Tutorial on Control Boards”: www.stsc.hill.af.mil/crosstalk/1999/mar/sorensen.asp
- “Adopting SCM Technology”: www.stsc.hill.af.mil/crosstalk/1996/mar/adopting.asp

Department of Energy, Configuration Management resources: http://cio.doe.gov/sqse/pm_conf.htm

Institute of Configuration Management: www.icmhq.com

MIL-HDBK-61A, Configuration Management Guidance: www.assist2.daps.dla.mil/quicksearch/

Program Manager’s Guide for Managing Software, 0.6, 29 June 2001, Chapter 14:

www.geia.org/sstc/G47/SWMgmtGuide%20Rev%200.4.doc

Software Engineering Institute, Software Configuration Management support:

www.sei.cmu.edu/legacy/scm/scmHomePage.html

This page intentionally left blank.