



Systems
Design &
Implementation
Series

Software Configuration Management

IDENTIFICATION, ACCOUNTING, CONTROL, & MANAGEMENT

STEVE J. AYER
FRANK S. PATRINOSTRO

TP31
A976

9460884

Software Configuration Management

Identification, Accounting,
Control, and Management

Steve J. Ayer
Frank S. Patrinostro



E9460884

McGraw-Hill, Inc.

New York St. Louis San Francisco Auckland Bogotá
Caracas Lisbon London Madrid Mexico Milan
Montreal New Delhi Paris San Juan São Paulo
Singapore Sydney Tokyo Toronto

Library of Congress Cataloging-in-Publication Data

Ayer, Steve J.

Software configuration management : identification, accounting, control, and management / Steve J. Ayer, Frank S. Patrinostro.

p. cm. — (McGraw-Hill software engineering series)

Includes index.

ISBN 0-07-002603-3

1. Software configuration management. I. Patrinostro, Frank S.

II. Title. III. Series.

QA76.76.C69A94 1992

005.1'068'4—dc20

92-4519

CIP

Copyright © 1992 by McGraw-Hill, Inc. All rights reserved. Printed in the United States of America. Except as permitted under the United States Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

1 2 3 4 5 6 7 8 9 0 DOC/DOC 9 7 6 5 4 3 2

ISBN 0-07-002603-3

The sponsoring editor for this book was Jerry Papke, the editing supervisor was Joseph Bertuna, and the production supervisor was Suzanne W. Babeuf. It was set in Century Schoolbook.

Printed and bound by R.R. Donnelley & Sons Company.

Information contained in this work has been obtained by McGraw-Hill, Inc. from sources believed to be reliable. However, neither McGraw-Hill nor its authors guarantee the accuracy or completeness of any information published herein and neither McGraw-Hill nor its authors shall be responsible for any errors, omissions, or damages arising out of use of this information. This work is published with the understanding that McGraw-Hill and its authors are supplying information but are not attempting to render engineering or other professional services. If such services are required, the assistance of an appropriate professional should be sought.

Software Configuration Management

McGraw-Hill Books of Interest

| | | |
|---------------|------------------|---|
| 0-07-061716-3 | Stone | <i>Inside ADW and IEF: The Promise and Reality of CASE</i> |
| 0-07-002604-1 | Ayer/Patrinostro | <i>Documenting the Software Development Process: A Handbook of Structured Techniques</i> |
| 0-07-030601-X | Howell | <i>Building Hypermedia Applications: A Software Development Guide</i> |
| 0-07-157879-X | Sodhi | <i>Software Requirements Analysis and Specification</i> |
| 0-07-062166-7 | Strauss/Ebenau | <i>Software Inspection Process</i> |
| 0-07-002603-3 | Ayer/Patrinostro | <i>Software Configuration Management: Identification, Accounting, Control, and Management</i> |

Preface

This book is intended to integrate the basic theories of configuration management and change control with the design, development, and implementation of computer-based management information systems. Such an integration emphasizes the importance of a systems development methodology (SDM) that breaks the systems life cycle into phases and tasks and identifies the deliverable items of each task. More importantly, it provides insights into how the MIS organizational structure responsible for the development of a computer system can apply technical and administrative direction to ensure that changes to a software configuration are processed in an orderly manner.

This book represents a significant departure from the numerous attempts that have been made to relate traditional engineering concepts of configuration management to the development and control of computer systems. In our judgment, much of the traditional material is essential; however, the unique thrust of software development makes it imperative to correlate the end-item orientation of the system life cycle with any discussion of configuration management and change control. This book is unique because it focuses on defining configuration identification baselines that correlate specifically with the phased development of a computer system.

The primary features of this book are as follows:

- At the outset the characteristics of the MIS environment in which software configuration management practices will be applied are emphasized.
- Project management is related to configuration management in terms of planning, project control, evaluation, change control, and project file management.
- Baselines are defined for maintaining control of changes to a software configuration identification.
- A labeling scheme for identifying the items that comprise a baseline identification is defined.
- Mechanisms for evaluating and approving or disapproving changes to a baseline identification are described and illus-

trated.

- The elements of the change management process from change initiation to installation monitoring are logically related.
- The functions of the change control board (CCB) that oversees maintenance of the information data base are delineated.
- Detailed procedures are provided for recording and reporting information on the status of proposed changes and the implementation of approved changes.
- The auditing function aimed at validating the satisfactory completion of a configuration item is explained and illustrated.

The subject matter of the six chapters has been arranged in what we believe to be a pedagogically sound sequence. Each chapter has a summary, and almost all have examples and illustrations.

The list of colleagues to whom the authors feel a sense of gratitude for ideas and suggestions is too long to enumerate here. With respect to this book; we are particularly grateful to Jerry Papke, senior editor at TAB/McGraw-Hill for making this book a reality.

Suggestions and comments on the text and the related materials are solicited.

Steve J. Ayer
Frank S. Patrinostro

Contents

| | |
|--------------------|-------------|
| Figures and Tables | <i>ix</i> |
| Preface | <i>xiii</i> |

| | | |
|-------------------|--|-----------|
| Chapter 1. | Overview of the Software Configuration Management Process | 1 |
| 1.0 | Introduction | 1 |
| 1.1 | Systems Development Methodology | 3 |
| 1.1.1 | Initial Investigation | 4 |
| 1.1.2 | Feasibility Study | 4 |
| 1.1.3 | Requirements Definition | 5 |
| 1.1.4 | External Design | 5 |
| 1.1.5 | Internal Design | 5 |
| 1.1.6 | Programming | 6 |
| 1.1.7 | Testing | 6 |
| 1.1.8 | Conversion | 7 |
| 1.1.9 | Implementation | 7 |
| 1.2 | Project Management Methodology | 7 |
| 1.2.1 | Planning | 8 |
| 1.2.2 | Project Control | 9 |
| 1.2.3 | Evaluating | 12 |
| 1.2.4 | Change Control | 13 |
| 1.2.5 | Project File | 16 |
| 1.3 | Configuration Management Methodology | 17 |
| 1.3.1 | Configuration Identification | 18 |
| 1.3.2 | Configuration Control | 22 |
| 1.3.3 | Configuration Accounting | 25 |
| 1.3.4 | Configuration Audit | 27 |
| 1.4 | Summary | 27 |
| Chapter 2. | Software Configuration Management Plan | 29 |
| 2.1 | Content Outline | 29 |
| 2.2 | Content Development Guidelines | 32 |

| | | |
|-------------------|--|------------|
| 2.3 | Summary | 63 |
| Chapter 3. | Software Configuration Identification | 65 |
| 3.1 | Task 1: Correlate the SDM Phases with the Configuration Baseline Identification | 67 |
| 3.1.1 | Baseline Management | 68 |
| 3.1.2 | Configuration Items | 69 |
| 3.2 | Task 2: Establish Identification Labeling Scheme | 70 |
| 3.3 | Task 3: Identify the Functional Baseline Configuration Items | 73 |
| 3.4 | Task 4: Identify the Allocated Baseline Configuration Items | 78 |
| 3.5 | Task 5: Identify the Design Baseline Configuration Items | 81 |
| 3.6 | Task 6: Identify the Product Baseline Configuration Items | 84 |
| 3.7 | Task 7: Identify the Operational Baseline Configuration Items | 90 |
| 3.8 | Summary | 93 |
| Chapter 4. | Software Configuration Control | 95 |
| 4.1 | Task 1: Define the Elements of the Change Management Process | 97 |
| 4.2 | Task 2: Establish Policies for Configuration Control | 98 |
| 4.3 | Task 3: Establish Configuration Management Standards | 102 |
| 4.4 | Task 4: Develop the Change Management Methodology | 117 |
| 4.5 | Task 5: Prepare Configuration Control Procedures | 123 |
| 4.6 | Task 6: Define Methods for Controlling Changes to Baseline Documents | 134 |
| 4.7 | Summary | 145 |
| Chapter 5. | Software Configuration Status Accounting | 147 |
| 5.1 | Task 1: Determine the Type of Logs Needed . . . | 150 |
| 5.2 | Task 2: Determine the Type, Frequency, and Distribution of Reports | 162 |
| 5.3 | Task 3: Determine Version Identification Documentation Requirements | 169 |
| 5.4 | Summary | 176 |

| | |
|--|------------|
| Chapter 6. Software Configuration Auditing | 177 |
| 6.1 Functional Configuration Audit | 178 |
| 6.1.1 Configuration Auditing Checklists | 179 |
| 6.1.2 Questionnaires | 184 |
| 6.1.3 Verification Forms | 188 |
| 6.2 Physical Configuration Audit | 192 |
| 6.2.1 Design Documentation Checklist | 192 |
| 6.2.2 Test Documentation Checklist | 194 |
| 6.2.3 Version Description Documentation Checklist | 195 |
| 6.2.4 Change Control Documentation Checklist | 196 |
| 6.3 Formal Qualification Review (FQR) | 197 |
| 6.3.1 Configuration Audit Report | 198 |
| 6.4 Summary | 199 |
| Glossary | 201 |
| Index | 207 |

List of Figures and Tables

| | | |
|-----------|--|-----|
| Fig. 1-1 | Project planning, control, and evaluation | 10 |
| Fig. 1-2 | The project review process | 12 |
| Fig. 1-3 | Development baselines | 14 |
| Fig. 1-4 | Change management policies | 15 |
| Fig. 1-5 | Baseline documents of the software development life cycle | 18 |
| Fig. 3-1 | Software configuration identification tasks | 66 |
| Fig. 3-2 | Baseline correlation of life-cycle stages | 67 |
| Fig. 3-3 | Hierarchy of an SCI labeling scheme | 71 |
| Fig. 3-4 | SCI version identification | 72 |
| Fig. 3-5 | Project file/phase end-document orientation of functional baseline | 76 |
| Fig. 3-6 | End-document orientation of the analysis process | 77 |
| Fig. 3-7 | Project file/phase end-document orientation of allocated baseline | 79 |
| Fig. 3-8 | Allocated baseline repository | 80 |
| Fig. 3-9 | Project file/phase end-document orientation of design baseline | 83 |
| Fig. 3-10 | Design baseline repository | 83 |
| Fig. 3-11 | Project file/phase end-document orientation of product baseline | 89 |
| Fig. 3-12 | Product baseline repository | 89 |
| Fig. 3-13 | Implementation documentation sources | 93 |
| Fig. 4-1 | Software configuration control tasks | 96 |
| Fig. 4-2 | Revision control documents | 134 |
| Fig. 4-3 | Review and approval points | 135 |
| Fig. 4-4 | Review copy status notation | 136 |
| Fig. 4-5 | Document review control form | 137 |
| Fig. 4-6 | Example of marking review copies | 138 |
| Fig. 4-7 | Formal baseline establishment | 139 |
| Fig. 4-8 | Example of a document sign-off sheet | 140 |
| Fig. 4-9 | Document change processing flow | 141 |
| Fig. 4-10 | Example of using the table of contents as a change control checklist | 142 |
| Fig. 4-11 | Example of a revision control form | 143 |
| Fig. 4-12 | Change control grid | 144 |

x List of Figures and Tables

| | | |
|-----------|---|-----|
| Fig. 5-1 | Software configuration accounting tasks | 148 |
| Fig. 5-2 | Data base change request log | 151 |
| Fig. 5-3 | Design problem report log | 152 |
| Fig. 5-4 | Document update transmittal log | 153 |
| Fig. 5-5 | Product log | 154 |
| Fig. 5-6 | Routine log | 155 |
| Fig. 5-7 | Site installation log | 156 |
| Fig. 5-8 | Software change proposal | 157 |
| Fig. 5-9 | Software modification report log | 158 |
| Fig. 5-10 | Software problem report log | 159 |
| Fig. 5-11 | Specification change notice log | 160 |
| Fig. 5-12 | Test status log | 161 |
| Fig. 5-13 | Change request status summary | 163 |
| Fig. 5-14 | Computer program inventory report | 164 |
| Fig. 5-15 | Computer program library transmittal summary | 165 |
| Fig. 5-16 | Software modification report summary | 166 |
| Fig. 5-17 | Software problem report summary | 167 |
| Fig. 5-18 | Test status summary report | 168 |
| Fig. 5-19 | Content arrangement of a version description document | 169 |
| Fig. 5-20 | Version identification form | 170 |
| Fig. 5-21 | Version application form | 171 |
| Fig. 5-22 | Differences-between-versions form | 172 |
| Fig. 5-23 | Software capabilities and limitations form | 173 |
| Fig. 5-24 | Adaptation information form | 174 |
| Fig. 5-25 | Installation instructions form | 175 |
| Fig. 6-1 | Configuration auditing process | 178 |
| Fig. 6-2 | FCA checklists | 180 |
| Fig. 6-3 | Functional requirements checklist | 181 |
| Fig. 6-4 | Allocated requirements checklist | 182 |
| Fig. 6-5 | Configuration item test checklist | 183 |
| Fig. 6-6 | Interview process | 184 |
| Fig. 6-7 | Auditing questionnaire form | 185 |
| Fig. 6-8 | Interview worksheet | 186 |
| Fig. 6-9 | Interview summary form | 187 |
| Fig. 6-10 | Verification process | 188 |
| Fig. 6-11 | Accuracy and validity verification form | 189 |
| Fig. 6-12 | Data integrity verification form | 190 |
| Fig. 6-13 | Performance verification form | 191 |
| Fig. 6-14 | Design documentation checklist | 193 |
| Fig. 6-15 | Test documentation checklist | 194 |
| Fig. 6-16 | Version description documentation checklist | 195 |
| Fig. 6-17 | Change control documentation checklist | 196 |

| | | |
|-----------|---|-----|
| Fig. 6-18 | Formal qualification review process | 197 |
| Fig. 6-19 | Configuration audit record | 198 |
| Table 3-1 | Functional baseline SCI labeling scheme | 73 |
| Table 3-2 | Allocated baseline SCI labeling scheme | 78 |
| Table 3-3 | Design baseline SCI labeling scheme | 81 |
| Table 3-4 | Product baseline SCI labeling scheme | 84 |
| Table 3-5 | Operational baseline SCI labeling scheme | 90 |
| Table 4-1 | Elements of a change control methodology | 97 |
| Table 4-2 | Software configuration control procedures | 124 |
| Table 5-1 | CSA logs | 150 |
| Table 5-2 | Configuration accounting reports | 162 |

Overview of the Software Configuration Management Process

1.0 Introduction

The objective of this book is to provide a methodological framework for software development and to define standards for managing the configuration items that comprise the baselines for management and control. In order to understand the role of configuration management in a software development environment, however, one must begin with dissecting the term *software configuration management*.

- *Software*, as defined in this book, includes the programs, procedures, routines, and all documents associated with the analysis, design, programming, conversion, and implementation of a computer system.
- *Configuration*, in a software context, is defined as the aggregate of deliverable items that result from the tasks performed during the phased development of the system. The configuration includes operating system programs used by computer equipment in responding to an application program; data base data that is stored in a form capable of being processed and operated on by a computer; system/subsystem specifications that define performance, organizational, and environmental and interfacing requirements; computer software documentation that explains the capabilities of the software, or provides operating instructions for using or maintaining the software; computer programs that

2 Overview of the Software Configuration Management Process

are subject to change; executable programs that are resident in read-only or erasable ROM memory and cannot be readily modified; program code used to translate or convert each operation into a computer-understandable language; support software designed to support the development, maintenance, and modification of other software; utility programs required for the generation of the operational and support programs; test software utilized in the testing of the program code; and computer software library materials that provide proper identification, storage, and protection.

- *Management* of the software configuration environment is defined as the administrative discipline that is exercised to identify and document the functional and physical characteristics of each software configuration item, control changes to those characteristics, record and report change processing and implementation status, and verify and validate software performance and the adequacy and completeness of documentation.

The following might be considered an appropriate definition of software configuration management:

Software configuration management is the process of exercising administrative surveillance and control of the sum total of computer equipment, computer software, and all associated documentation.

The definition can, of course, be amplified by further describing software configuration management as follows:

Software configuration management is the process of identifying, tracking, and controlling changes to software configuration items that define needs, evaluate the technical and cost feasibility of the system, and establish the functional requirements; identify specific hardware and software functions to be performed; set forth the design specifications, define the user procedures and controls, and establish plans for implementation and maintenance; transform the design specifications into executable code; establish specifications for testing and reporting on test results; and provide operational documentation, including program packages, user guides, operator manuals, and training materials.

From these definitions, perhaps we can begin to establish a framework for developing a software configuration management methodology. The implementation of a configuration management methodology must be integrated with a systems development methodology and project management methodology to provide a consistent, workable environment for software development, implementation, and control. The following pages outline the concepts of each of these components to management in a software environment.

1.1 Systems Development Methodology

The methodology used for systems development establishes the framework for software configuration identification and baseline management. The *systems development methodology* (SDM) breaks the systems development process into a series of manageable phases. A series of tasks and task steps define each phase. Each task or group of tasks results in one or more definable items that are deliverable to the project file. The aggregate of deliverable items that are cumulated at specified stages of the development process form the baselines for configuration management and change control.

The SDM model will differ from organization to organization, but in most methodologies the process revolves around five major functions of systems development concern:

- Define the functional specifications
- Allocate functional specifications to subsystems
- Prepare design specifications
- Program the design
- Maintain the system.

The approach to systems development that is associated with the configuration management process presented in this book is characterized by breaking down the systems development cycle into nine development phases:

- Initial investigation
- Feasibility study
- Requirements definition
- External design
- Internal design
- Programming
- Testing
- Conversion
- Implementation

4 Overview of the Software Configuration Management Process

1.1.1 Initial Investigation

The methodology begins with an initial investigation phase. The starting point of the initial investigation process is the issuance of a project request submitted by the manager of a user department (e.g., accounting department, purchasing department, sales department, etc.). The information on the project request generally includes a statement of the present system's problems and an explanation as to why a new system is being proposed. When the request is approved, the data processing organization will embark on the development effort. If the request is for a major enhancement to an existing system, an analyst, or a team of analysts, may be assigned the responsibility of conducting an initial investigation to further define the problems and needs identified by the project request. The project request document itself becomes the first item subject to configuration control.

The first task performed by the analysts assigned to conduct the initial investigation is to interview the probable users of the new system. The objective is to acquire a detailed understanding of the existing problems and the goals for the proposed system. The second task is to review the strengths and weaknesses of the proposed system. The third task focuses on developing a feasibility study plan. The final task is the preparation of an initial investigation phase end-document or initial investigation report to document the findings.

1.1.2 Feasibility Study

The tasks to be performed to determine the feasibility of the proposed project are geared to answer three questions: What needs to be done? How can it be done? What is the value of doing it? Expressed somewhat differently, the feasibility study phase of software development provides answers to questions that address the operational, technical, and economic feasibility of developing the proposed system.

The first group of tasks performed focuses on reviewing the present system to gain a thorough understanding of what it does and how it does it. The second group of tasks involves a preliminary analysis of the requirements that the proposed system must meet. The third group of tasks involves the identification of possible development alternatives. The fourth group of tasks focuses on comparing the estimated costs of development and operation to the anticipated benefits. The final task is to prepare a feasibility study phase-end document.

Using the information presented in the feasibility study document, management decides whether or not to proceed with the project.