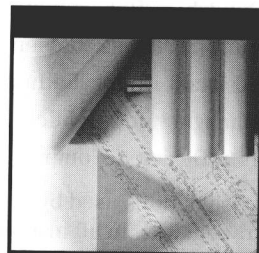


AntiPatterns and Patterns in Software Configuration Management

William J. Brown
Hays W. "Skip" McCormick III
Scott W. Thomas

TP31
B881



AntiPatterns and Patterns in Software Configuration Management

William J. Brown
Hays W. "Skip" McCormick III
Scott W. Thomas



E200000033

Wiley Computer Publishing



John Wiley & Sons, Inc.

NEW YORK • CHICHESTER • WEINHEIM • BRISBANE • SINGAPORE • TORONTO

Publisher: Robert Ipsen

Editor: Theresa Hudson

Assistant Editor: Kathryn A. Malm

Managing Editor: Micheline Frederick

Text Design & Composition: North Market Street Graphics

Designations used by companies to distinguish their products are often claimed as trademarks. In all instances where John Wiley & Sons, Inc., is aware of a claim, the product names appear in initial capital or ALL CAPITAL LETTERS. Readers, however, should contact the appropriate companies for more complete information regarding trademarks and registration.

This book is printed on acid-free paper. ∞

Copyright © 1999 by William J. Brown, Hays W. "Skip" McCormick III, Scott W. Thomas.
All rights reserved.

Published by John Wiley & Sons, Inc.

Published simultaneously in Canada.

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4744. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 605 Third Avenue, New York, NY 10158-0012, (212) 850-6011, fax (212) 850-6008, E-Mail: PERMREQ @ WILEY.COM.

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold with the understanding that the publisher is not engaged in professional services. If professional advice or other expert assistance is required, the services of a competent professional person should be sought.

Library of Congress Cataloging-in-Publication Data:

Brown, William J., 1954–

AntiPatterns and Patterns in software configuration management /
William J. Brown, Hays W. "Skip" McCormick III, Scott W. Thomas.
p. cm.

"Wiley computer publishing."

Includes index.

ISBN 0-471-32929-0 (cloth : alk. paper)

1. Software configuration management. I. McCormick, Hays W.,
1964– . II. Thomas, Scott W., 1957– . III. Title.

QA76.76.C69B76 1999

005.1'5—DC21

99-10822

CIP

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

***This book is dedicated to our families:
Luke and Matthew Brown,
Kim McCormick,
and
Susan Thomas***

***"For we ourselves also were sometimes foolish . . ."
Titus 3:3***



Preface

We wrote this book to help you manage your software development. It is our experience that a very large percentage of the people who manage software development efforts today possess only a limited understanding of and little experience with real software configuration management techniques. Furthermore, we believe that this is true not because managers don't want to know the principles, but rather because software configuration management seems so very hard to learn, even harder to implement, and virtually impossible to enforce. Many of us can remember dreading software configuration management as if it were some kind of arcane, mysterious black science fraught with scary "big brother" machinations—strictly enforced by a diabolical special police system that would be very likely to get us into a lot of trouble after having sucked the spark of creativity from our brains.

If you have any of these concerns, in even a mild way (we admit that we have rather overdramatized about the subject in the preceding paragraph), then we hope that this book will dispel them once and for all.

Unfortunately, much of the material many of us have read so far on software configuration management has tended to be dry and somnolent. Rather than repeat a very effective cure for insomnia, we have attempted in this book to shine the light of simplicity and understanding on this subject. Feedback from our last book on *AntiPatterns* indicated that *AntiPatterns* are both fun to read and an effective way to codify and encapsulate the kinds of wisdom and knowledge that many software people seek. Most people familiar

with the book said, “It’s funny. I liked the bit about . . .,” and then recounted how their personal experience matched some element of one of the many AntiPatterns they read. A lot of people even read the entire book!

That is exactly what we hope you will do with this book. We hope you will read it and use it, and we would like to think that you will find it both a proactive and reactive resource. And also that you’ll find it entertaining—because then you will discover two grand truths about software configuration management:

1. SCM is not scary, impossible, nightmarish, or evil; rather it’s understandable and based on common sense.
2. You can successfully implement and maintain an SCM program, and thereby give yourself a terrific protection against the possibility of your project failing.

We suggest that it is better for you to read through the entire book now (it’s not that thick), to gain some overall comprehension before you actually need it for a specific problem. However, if your project is in trouble right now, and you need to reference a Pattern or AntiPattern quickly, we suggest that you jump straight ahead to the Executive Summary and then read Appendix A to quickly identify and reference the appropriate and applicable Pattern or AntiPattern. You can then complete the book after resolving your crisis (right!)—or pull it out again when you encounter the next crisis.

The concept of Patterns can be somewhat complex and academically oriented as it relates to object-oriented software development. However, while we leverage those concepts in the development of this book, we are addressing predominately project management issues that are reality-based and practical in implementation. Therefore, this book is intended for the practitioner. It is intended for those project leaders, configuration managers, and test managers who are in the throes of a project and dealing with a crisis that has resulted from a misapplied Pattern (discipline or Key Process Area). The book will enable them to arrive at a refactored solution without first having to achieve a comprehensive understanding of object-oriented development or Pattern philosophy.

The book is divided into three parts. These include Part One: *Introduction to Patterns and AntiPatterns*; Part Two: *Software Engineering AntiPatterns and Patterns*; and Part Three: *Conclusions and Resources*.

Part One consists of two chapters. The first chapter provides an explanation of why we believe the material covered in the book is important, and includes an orientation to the topics covered. Additionally, it presents an overview of the concept of Patterns and AntiPatterns. This will equip the reader with a basic understanding of the philosophical concepts of Patterns. (We anticipate that many practitioners may not be familiar with the fundamental concepts of object-oriented programming and Patterns; while such an understanding is not absolutely necessary, it is nevertheless suggested.) Chapter Two, “Nil Desperandum,” is intended as an exemplar for all other AntiPatterns and Patterns contained within the book. We hope that it provides the reader, early on, with a foundation for the remainder of the book. It is the only AntiPattern in Part One and is broken out from Part Two for that reason only.

Part Two is the meat of the book. It is composed of three chapters that are broken down by system engineering disciplines. Specifically, these include Chapter Three: *Software Configuration Management Pattern and AntiPatterns*; Chapter Four: *Management and Process Patterns and AntiPatterns*; and Chapter Five: *Requirements and Testing Patterns and AntiPatterns*. The purpose of breaking out the Patterns and AntiPatterns in this manner is to assist the reader in quickly identifying where the Patterns and AntiPatterns are located by discipline. This book may be consumed in a linear fashion, but it can also be utilized as a reference that is applicable to the situation and environment of each reader. The structure of the AntiPatterns and Patterns breakout was accomplished so that when the reader refers back to the book, he or she will be able to quickly discern where to find the topic of interest. At the beginning of each chapter, a summary list of the Patterns and AntiPatterns is provided as well, for additional quick reference.

Part Three presents the book’s conclusions, as well as additional resources. The “conclusions” are essentially tabular listings of the AntiPatterns and Patterns, providing a summary for each, identifying the appropriate discipline, and the page where it is found. This part of the book provides an additional reference for the reader to quickly identify an applicable Pattern or AntiPattern. Finally, we have provided references that we believe will be invaluable for the practitioner, including listings of books, magazine articles, and Web sites that relate directly to the subject matter.



Acknowledgments

In our first book, *AntiPatterns: Refactoring Software Architectures and Projects in Crisis* (John Wiley & Sons, Inc., 1998), between the four coauthors, we listed some 98 people. We were trying to express that a lot of people helped us get to the point so that we could write *AntiPatterns*, and we wanted to recognize as many of them as we could.

That list has grown impossibly large since, as we have made loads of new friends at PLoP, OOPSLA and other conferences, several patterns study groups, and AntiPatterns tutorials all over the globe, in addition to literally hundreds of new Internet associates via the many Patterns discussion lists and Web spaces such as WikiWeb and AntiPatterns.com. We thank all of the many great people who have encouraged us, engaged us intellectually, and stimulated us passionately; they have all played a part in this new work.

Still, there are a few key people who have directly and substantially contributed to this work, so we will recognize them individually. First thanks to John Vlissides for his friendship and encouraging words and focused leadership in extending the Managerial AntiPatterns concept. To James Coplien for taking the time to personally and passionately exhort us to recognize, accept, and strive to meet our great responsibility for intellectual integrity as authors in the Patterns community. And to the Hillside Gang, especially Joe Yoder, Brian Foote, and Ralph Johnson; thank you guys for supporting us as we “took our lumps” at PLoP 98 and joined the team.

In terms of providing a greatly improved content, we thank John Marsh and Walt Stoneburner for their outstanding criticisms and

extensive appraisals of the manuscript, which drove greater care and attention to detail. Few people could motivate as well and effectively as John and Walt.

Finally, to the person without whom we would not have accomplished this work, who has mothered, scolded, cajoled, encouraged, and forced us to do the best work we are capable of, while caringly and very effectively shepherding the diverse personalities that make up this author team: our editor, Terri Hudson. She has been and continues to be the best visionary, partner, cheerleader, and editor we could ever have hoped for . . . and she is a wonderful friend. Thanks, Terri; let's do it again.



Author Biographies

William J. Brown is Product Engineer at Concept Five Technologies in McLean, Virginia. He is coauthor of *AntiPatterns: Refactoring Software, Architectures and Projects in Crisis* (John Wiley & Sons, Inc.), and is a columnist for *Distributed Computing*. Bill's background is in leading developments of distributed object systems, and he has experience with telecommunications, health, and military real-time systems, but his expertise is most formidable in the financial domain. He has worked both as architect and project manager for insurance services, treasury and derivative instruments, and banking product developments. Bill established a strong reputation at the OMG while he was architect for the Financial Domain Task Force and known as "Vlad the Impaler." His previous career as a social worker in behavior modification provides him with a unique insight into the sociopolitical aspects of software development.

Hays W. "Skip" McCormick is a lead engineer in the Distributed Components, Architectures and Frameworks Department at the MITRE corporation in McLean, Virginia. He is a coauthor of *AntiPatterns: Refactoring Software, Architectures and Projects in Crisis*, and has a diverse background in many information systems areas, including computer security, artificial intelligence, and distributed object computing. His most recent efforts have been focused in the areas of Knowledge Management and the pursuit of a truly useful reification of the elusive integrated, distributed collaboration system. Skip holds a B.S. degree in computer science from the United States Naval Academy at Annapolis, Maryland. He resides in Manas-

sas, Virginia, where he is the proud father of Teddy and Janine, and the happy husband of Kim.

Scott W. Thomas is a lead engineer and project leader in the Information Technology Division of the MITRE Corporation in McLean, Virginia. He has 20 years experience with government, commercial, and nonprofit organizations, in all facets of the system engineering lifecycle. He is currently involved with the development of secure collaborative environments. Scott also serves as a Lieutenant, Engineering Duty Officer, in the U.S. Naval Reserve, Commander Naval Surface Forces, U.S. Atlantic Fleet, Detachment 606. Scott holds a B.S. degree in industrial engineering and operations research, and an M.S. in systems engineering, both from the Virginia Tech at Blacksburg, Virginia. He resides in Vienna, Virginia, with his two children Shara and Ryan, and wife Susan.



Executive Summary

AntiPatterns and Patterns in Software Configuration Management will help you identify and overcome prevalent, recurring software configuration management and related management roadblocks to successful software delivery. The AntiPatterns identify common mistakes with refactored solutions, while the Patterns identify good working practices. This book will help you to solve existing problems at both organizational and project levels and to develop pragmatic working practices to stop some from ever occurring.

Many of us occasionally forget that we need some rigid controls, which at first blush appear to cause extra work and little tangible reward. This is commonly the case with software configuration management. In fact, the lack or incorrect implementation of SCM will cause more work, greater and more frequent failures, and wasted investment. In short, all software needs a level of software configuration management—unless it is intended to go straight into the bin. That said, the solution must be well balanced with the other software development processes to ensure that efficiency, rather than unproductive bureaucracy, is achieved.

The same is true for requirements traceability and testing, both of which also tend to be reduced under pressure of delivery. Again, if there is no discipline, the result will be undisciplined. The trick is to implement a minimalist process that is easy to follow and yields significant results. This book will cover the very common problems and propose practical solutions.

Culture and management start from the top, and if the executives who drive the direction of a company ignore attention to detail,

that practice will become the corporate norm. Executives should neither try to define nor implement critical product development management processes, but rather see the results in the form of status reports, approved software releases, and satisfied customers. Thus, executives of software development companies must understand both the problems incurred by nonexistent and badly implemented processes and the way to avoid and resolve them.

If you know of a project that has failed, most likely you can use the information in this book to identify at least one of the contributory causes; and, our previous book in the AntiPattern series, *Anti-Patterns: Refactoring Software, Architectures and Projects in Crisis*, can be used to identify the others. The intent is to preempt the problems where possible, and to implement a refactored solution where the problems already exist. In some cases, it will be too late, but that opens the opportunity to learn why the development failed and to prevent it next time. The industry seems to have become somewhat complacent about software development delivery failure, because it has been such a widely publicized fact for so long. But human nature acts against us in the form of social inertia in the workplace. This need not be the case. *Carpe Diem*: Seize the day.



Contents

Preface	vii
Acknowledgments	xi
Author Biographies	xiii
Executive Summary	xv
Part One	Introduction to Patterns and AntiPatterns
1	
Introduction	The Nature of a Pattern: A Brief Tutorial
	3
	Pattern and AntiPattern Views
	4
	Process and Management Pattern Template
	6
	AntiPattern Template
	8
	Quality without a Name
	11
	References
	14
Chapter 1	The Lost Disciplines: A System Engineering Perspective
	17
	Software Configuration Management
	20
	Requirements Development
	23
	Testing
	25
	Half a League Onward
	26
	References
	28
Chapter 2	The Father of All Management AntiPatterns
	29
	Nil Desperandum AntiPattern
	31

Part Two	Software Engineering AntiPatterns and Patterns	49
Chapter 3	Software Configuration Management Pattern and AntiPatterns	51
	Software Configuration Management	55
	Organization Pattern	
	Silver Bullet AntiPattern	63
	CM Takeover AntiPattern	73
	Developer-Software Configuration Management AntiPattern	83
	Decentralized Configuration Management AntiPattern	95
	Failure to Audit AntiPattern	111
	Object-Oriented Configuration Management AntiPattern	123
	Software Configuration Management Expert AntiPattern	145
	Postmortem Planning	163
Chapter 4	Management and Process Patterns and AntiPatterns	177
	P ² Pattern	179
	Détente AntiPattern	191
	Nirvana AntiPattern	199
	Enterprise System Fable AntiPattern	213
Chapter 5	Requirements and Testing Patterns and AntiPatterns	227
	Requirements Jeopardy AntiPattern	229
	Information Integrity Management Pattern	241
	What's on the Media AntiPattern	253
	Sure Thing AntiPattern	263
	Great Expectations AntiPattern	275
Part Three	Conclusions and Resources	289
Appendix A	AntiPatterns and Patterns Synopsis	291
Appendix B	Patterns and AntiPatterns Cross-References	299
Appendix C	Additional References	301
Index		305

PART

One

Introduction to Patterns and AntiPatterns
