
*Configuration Management
Principles and Practice*

The Agile Software Development Series

Alistair Cockburn and Jim Highsmith, Series Editors

For more information check out <http://www.awprofessional.com/>

Agile software development centers on four values identified in the Agile Alliance's Manifesto:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

The development of Agile software requires innovation and responsiveness, based on generating and sharing knowledge within a development team and with the customer. Agile software developers draw on the strengths of customers, users, and developers, finding just enough process to balance quality and agility.

The books in **The Agile Software Development Series** focus on sharing the experiences of such Agile developers. Individual books address individual techniques (such as Use Cases), group techniques (such as collaborative decision making), and proven solutions to different problems from a variety of organizational cultures. The result is a core of Agile best practices that will enrich your experience and improve your work.

Titles in the Series:

Alistair Cockburn, *Surviving Object-Oriented Projects*, ISBN 0-201-49834-0

Alistair Cockburn, *Writing Effective Use Cases*, ISBN 0-201-70225-8

Lars Mathiassen, Jan Pries-Heje, and Ojelanki Ngwenyama, *Improving Software Organizations: From Principles to Practice*, ISBN 0-201-75820-2

Alistair Cockburn, *Agile Software Development*, ISBN 0-201-69969-9

Jim Highsmith, *Agile Software Development Ecosystems*, ISBN 0-201-76043-6

Steve Adolph, Paul Bramble, Alistair Cockburn, and Andy Pols, *Patterns for Effective Use Cases*, ISBN 0-201-72184-8

Configuration Management Principles and Practice

Anne Mette Jonassen Hass

◆◆ Addison-Wesley

Boston • San Francisco • New York • Toronto • Montreal

London • Munich • Paris • Madrid

Capetown • Sydney • Tokyo • Singapore • Mexico City

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and Addison-Wesley was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The author and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

The publisher offers discounts on this book when ordered in quantity for bulk purchases and special sales. For more information, please contact:

U.S. Corporate and Government Sales
(800) 382-3419
corpsales@pearsontechgroup.com

For sales outside of the U.S., please contact:

International Sales
(317) 581-3793
international@pearsontechgroup.com

Visit Addison-Wesley on the Web: www.awprofessional.com

Library of Congress Cataloging-in-Publication Data

Hass, Anne Mette Jonassen.

Configuration management principles and practice / Anne Mette Jonassen Hass.

p. cm.

Includes bibliographical references and index.

ISBN 0-321-11766-2 (alk. paper)

1. Software configuration management. I. Title.

QA76.76.C69 H37 2003
005.1—dc21

2002034458

Copyright © 2003 by Pearson Education, Inc.

All rights reserved. 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, or otherwise, without the prior consent of the publisher. Printed in the United States of America. Published simultaneously in Canada.

For information on obtaining permission for use of material from this work, please submit a written request to:

Pearson Education, Inc.
Rights and Contracts Department
75 Arlington Street, Suite 300
Boston, MA 02116
Fax: (617) 848-7047

ISBN 0-321-11766-2
Text printed on recycled paper
1 2 3 4 5 6 7 8 9 10—MA—0706050403
First printing, January 2003

Contents

List of Figures	xxv
List of Tables	xxix
Foreword by Kim Caputo	xxxi
Foreword by Alistair Cockburn	xxxiii
Preface	xxxv
Introduction	xli

Part I *What Is Configuration Management?*..... 1

Chapter 1 *Definition of Configuration Management Used in This Book*..... 3

1.1	Configuration Management Activities.....	4
	Metadata.....	5
	Configuration Management Is Cyclic—or Is It?.....	5
	Quality Assurance Process.....	6
	Audit.....	7
1.2	Identification.....	7
	Inputs.....	8
	Outputs.....	8
	Process Descriptions.....	9
	Unique Identification.....	9

	Examples	9
	Authorization	11
	Roles	11
	Connection with Other Activities	11
1.3	Storage	12
	Library	13
	Main Processes	15
	Process Descriptions	16
	Roles	16
	Connection with Other Activities	17
	Example	17
1.4	Change Control	19
	Inputs	20
	Outputs	20
	Change Control Activities	20
	Usage of Metadata	21
	Consequence Analysis	22
	Roles	22
	Process Descriptions	23
	Connection with Other Activities	23
	Example	23
1.5	Status Reporting	23
	Inputs	25
	Outputs	25
	Process Descriptions	25
	Roles	26
	Connection with Other Activities	26
1.6	False Friends: Version Control and Baselines	26
	Version Control	27
	Baseline	27
Chapter 2 Configuration Management in Maturity Models		29
2.1	CMM Version 1.1	29
	CMM Maturity Levels	30
	Definition	31
	Activities	31

2.2	CMMI	32
	CMMI Process Areas	34
	Definition	35
	Goals	35
	Practice-to-Goal Relationships	35
	Capability and Maturity Levels	36
	Achieving Capability Levels	37
	Level 2 for All Process Areas	37
	Raising the Capability of the Configuration Management Process	38
2.3	ISO 15504 (SPICE) and BOOTSTRAP 3.2	40
	SPICE Process Model	40
	Definition	40
	Goals	40
	Best Practices	42
	Maturity Levels	42
	Maturity of Configuration Management	43

Chapter 3 *Configuration Management in International Standards* **45**

3.1	Overview of Related Standards	45
3.2	BS6488, DoD, IEEE	47
	BS6488	47
	DoD Mil-Std-973	48
	IEEE-Std-610.12-1990	48
3.3	ESA PSS-05-09	48
	Introduction from the Guide	49
3.4	GAMP	49
	Description from the Guide	50
3.5	ISO 9001:1994, ISO 9000-3, and ISO 9001:2000	50
	ISO 9001:1994	50
	ISO 9000-3	51
	ISO 9001:2000	52

Chapter 4 *Organizations Working with Configuration Management* . . . **55**

4.1	Institutions and Companies	55
	CM Today Yellow Pages	55

	Institute of Configuration Management	56
	Conferences.....	56
	Ovum.....	56
	Software Engineering Institute	56
4.2	Projects.....	57
	ACME.....	57
	AdCoMs	57
	DaSC	57
Chapter 5 Scoping the Configuration Management Task		59
5.1	Level of Ambition—Cost/Benefit Analysis.....	59
	Level of Ambition = Scope + Formalism	59
	Formalism for a Configuration Item	60
	Degrees of Formalism.....	61
	Earliest and Latest Extremes for Starting Configuration Management.....	61
	Formalism and Tools	62
	Expansion of Scope—from Candidate to Item.....	62
	No Rough Drafts—Please!	62
	Expansion from the Middle	63
5.2	Examples	64
5.3	Calculation of Profitability	67
	Expenses	68
	Savings	68
5.4	Pitfalls in Connection with Scoping	71
	Too Demanding.....	71
	Wrong	72
	Too Coarse or Too Fine.....	72
	Too Embracing or Too Exclusive.....	72
	Too Late or Too Early.....	73
5.5	How to Treat What Is Kept Outside	73
	Objects to Keep Outside	74
	Identification.....	74
	Storage	74

Part II Configuration Management Data 75

Chapter 6 What Can Be Placed under Configuration Management 77

6.1	Physical or Electronic Objects.....	77
	Configuration Item Class Hierarchy	77
	Physical Objects	78
	Electronic Objects	79
6.2	Types of Objects in Product Perspective	79
	Software	79
	Hardware.....	80
	Network.....	80
	Data.....	80
	Services	80
	Tools	81
6.3	Types of Objects in Project Perspective	81
	Life Cycle Activities.....	81
	Support Functions	82
	Tools	82
6.4	Types of Objects in Cross-Organizational Perspective	82
	Cross-Organizational Perspective	82
	Administrative Documents.....	82
	Company Product Assets	82
	Infrastructure.....	83
	Quality System	83
6.5	Deliveries under Configuration Management.....	83
	Examples.....	84
	Project Relationships	84
6.6	Deliveries for Planned Events Like Milestones.....	85
	Development Model	85
	Milestones.....	85

Chapter 7 What One Needs to Know about a Configuration Item..... 89

7.1	Overview of Metadata for a Configuration Item	89
	Data Elements.....	89
	Metadatabase Medium	90

	Other Data Elements	91
7.2	Metadata for Unique Identification	91
	Belongs To	91
	Name	91
	Version	93
	Status	93
	Date	93
	Storage Location	94
	Storage Medium	94
	Example of States for a Document	94
	Example of States for a Source Code Unit, Including in Build	94
7.3	Metadata for Authorization	96
	Producer	96
	Person Holding Overall Responsibility	97
	Person Responsible for Approval	97
	Ownership	97
7.4	Metadata for Relations to Other Configuration Items	97
	Traces To (and From!)	98
	Tracing Registration	99
	Importance of Tracing	99
	Produced With	100
	Derived From	100
	Consists Of	100
7.5	Metadata for Distribution	101
	May Be Distributed To	101
	Has Been Distributed To	101
 Chapter 8 What One Must Register for a Configuration Item		103
8.1	Item Approval	103
	Quality Approval	103
	Medium	104
	Content	104
	Examples	105
8.2	Release Request	106
	Medium	106
	Content	108
	Stock Control	109

	Examples	109
8.3	Event Registration.	111
	Life Cycle and Responsibility	111
	Content	112
	Created	113
	For Evaluation	113
	Under Decision	113
	Under Change	115
	Closed	115
	Classification.	115
	Examples	115
8.4	Change Request.	116
	Life Cycle and Responsibility	119
	Content	119
	Created	120
	Implemented.	120
	Approved	121
	New Events	121
	Examples	122

Chapter 9 *What Information Is Available for Configuration Items* **125**

9.1	Examples	125
	Release Note	125
	Item Status List	127
	Item History List.	127
	Item Composition List	127
	Trace Report	130
9.2	Configuration Management as Supplier of Measurements	132
	Ideas for Process Improvement	133

Part III *Roles in Configuration Management* **135**

Chapter 10 *People and Configuration Management* **137**

10.1	Configuration Management as a Career.	137
	Qualifications	138
10.2	Managing Configurations Is Everyone’s Job.	138

10.3 Understanding Team Roles.....139
 Putting Teams Together.....140

Chapter 11 Configuration Management Roles 141

11.1 Configuration Control Board141
 Skills and Knowledge142
 Multiple Boards.....143
 Managing Configurations of CCB Work Products.....143
 References.....143
11.2 Librarian144
 Tools146
 Managing Configuration of Library Work Products146
 References.....146
11.3 Person Responsible for Configuration Management.....146
 Planning Configuration Management148
 Managing Configuration Management Work Products.....148
 References.....148

Chapter 12 Organizational Roles..... 149

12.1 Management149
 Defining and Tracking Goals150
 Benefits150
 References.....150
12.2 Person Responsible for Assets151
 Different Process Descriptions.....151
 References.....152
12.3 Person Responsible for Operation.....152
 Configuration Management Responsibility.....152
 References.....153
12.4 Person Responsible for Process Management153
 Managing Configurations of Process Management Work Products154
 References.....155
12.5 Person Responsible for Environments and Tools.....155
 Managing Configurations of Environments and Tools156
 References.....156

12.6 Support/Helpdesk 156
 References 157

Chapter 13 *Project-Related Roles* **159**

13.1 Analyst 159
 Benefits 160
 References 160

13.2 Designer 160
 Benefits 161
 References 161

13.3 Programmer 161
 Benefits 162
 References 162

13.4 Integrator 162
 Benefits 163
 References 163

13.5 Tester 163
 Benefits 164
 References 164

13.6 Project Manager 164
 Benefits 165
 Managing Configurations of Project Management Work Products 165
 References 165

13.7 Person Responsible for Quality 165
 Managing Configurations of Quality Assurance Work Products 166
 References 166

13.8 Person Responsible for Customer Contact 167
 References 167

13.9 Person Responsible for Subcontractor Contact 167
 References 168

Chapter 14 *External Roles* **169**

14.1 Customer 169
 References 170

14.2 Subcontractor 171
 References 171

Part IV Configuration Management in Practice 173

Chapter 15 General Principles 175

15.1	Milestones	175
	Identification	175
	Generic Content Lists	177
	Storage	177
	Change Control	178
	Status Reporting	179
15.2	Document Handling	179
	Configuration Items or Deliveries	179
	Identification	180
	Authorization	182
	Tracing	182
	Storage	183
	Change Control	183
	Status Reporting	183
15.3	Emergency Changes	183
	Examples	184
	Principles for “Cheating”	184
	Avoid Cheating	185
	Examples Again	185

Chapter 16 Configuration Management in Development Activities . . . 187

16.1	Documentation Activities (Specifications and Design)	187
	Identification	188
16.2	Coding	189
	Unique Identification	189
	Authorization	190
	Tracing	190
	Storage	190
	Change Control	190
16.3	Integration	191
	Production Time	191
	Unique Identification	191
	Tracing	192

	Storage	192
	Change Control.....	192
16.4	Test	193
	Deliveries.....	193
	Identification.....	193
	Tracing	193
	Storage	194
	Change Control.....	194
16.5	Operational Use	194
	Configuration Management Considerations	195
	Release.....	195
	Event Registration	195
	Status Reporting.....	196
	Organizational Considerations.....	196
	Backup.....	197
16.6	Maintenance	197
	New Versions.....	197
	Configuration Management Considerations	197
	Example.....	198

Chapter 17 *Managing Configurations for Project Support Functions.* . 199

17.1	Project Management	199
	Example.....	200
	Deliveries.....	200
	Connection with Other Processes	200
	Identification.....	201
	Tracing	201
	Change Control.....	201
	Status Reporting.....	201
17.2	Configuration Management	202
	Milestone Deliveries	202
	Storage	202
	Change Control.....	203
	Status Reporting.....	203
17.3	Quality Assurance	203
	Connection with Other Processes	204
17.4	Subcontractor Management	204

Identification.....	205
Storage	205
Change Control.....	205
Status Reporting.....	206
Delivery	206

Chapter 18 *Managing Configurations in Different Development Models* **207**

18.1	Agile Development.....	207
	Configuration Management in Agile Development.....	208
	Empowered Teams	208
	Process Handling	209
	Environment and Support	209
	Requirements Management	210
	Working Together.....	210
	Frequent Delivery of Working Software.....	210
	Communication and Documentation	211
	Status Reporting.....	211
18.2	Frequent-Build Technique.....	211
	Planning Considerations	212
	Configuration Management Considerations	212
	Frequent Builds Are Not Frequent Storage.....	212
	Identification.....	213
	Building.....	213
	Storage	213
	Backtracking.....	213
	Change Control.....	214
	Example.....	214
18.3	Integrated Product Development.....	216
	Organizational Considerations.....	217
	Configuration Management Considerations	217
	Approach	217
18.4	Iterative Development	218
	Configuration Management Considerations	219
	Requirements Management	219
	Identification.....	220
	Storage	220

	Change Control.....	220
	Status Reporting.....	220
18.5	Sequential Development.....	221
	W-Model	222
	Configuration Management Considerations	222
	Identification.....	224
	Change Control.....	224
	Status Reporting.....	224
Chapter 19 Managing Configurations for Different Product Types		225
19.1	Composite Systems	225
	Design Considerations	226
	Configuration Management Considerations	226
	Identification.....	227
	Storage	227
	Change Control.....	227
	Status Reporting.....	227
19.2	Multiplatform	227
	Configuration Management Considerations	228
	Identification.....	228
19.3	Multivariants	228
	Examples.....	229
	Requirements Considerations	229
	Design Considerations	229
	Configuration Management Considerations	230
	Identification.....	230
	Storage	231
	Change Control.....	231
	Status Reporting.....	231
19.4	Safety-Critical Products	232
	Examples.....	232
	Configuration Management Considerations	233
19.5	Size of Product (Large and Small)	233
	Small Systems.....	233
	Large Systems.....	233
	Identification.....	234
	Storage	234

	Change Control	234
	Status Reporting	234
	Tool Considerations	234
19.6	Web Applications	235
	Examples	235
	Content Management	236
	Configuration Management Considerations	236
	Identification	237
	Storage	237
	Change Control	238
	Status Reporting	238
Chapter 20 <i>Managing Configurations under Special Conditions</i>		239
20.1	Multisite Development (Geographic Distribution)	239
	Example	239
	Organizational Considerations	240
	Configuration Management Considerations	241
	Identification	241
	Storage	241
	Change Control	242
	Status Reporting	242
	Example	242
20.2	Multiple Stakeholders	245
	Get an Overview of the Requirements	245
	Analyze the Requirements	245
	Describe the Fulfillment	246
	Conflict of Authority	246
20.3	Parallel Development	246
	Example	246
	Planning Considerations	247
	Configuration Management Considerations	247
	Identification	248
	Storage	248
	Change Control	248
	Status Reporting	249
	Tool Considerations	249

20.4 Tool Support..... 249
 Configuration Management Considerations 249

Chapter 21 *Managing Configurations for Cross-Organizational Functions* **251**

21.1 Company Infrastructure..... 251
 Organizational Considerations..... 251
 Identification..... 252
 Storage 252
 Change Control..... 252
 Status Reporting..... 253

21.2 Cross-Organizational Objects..... 253
 Configuration Management Considerations 253
 Identification..... 253
 Storage 254
 Change Control..... 254
 Status Reporting..... 254

21.3 External Reuse Component Development 254
 Examples..... 255
 Configuration Management Considerations 255
 Identification..... 255
 Storage 255
 Change Control..... 256
 Status Reporting..... 256

21.4 Internal Asset Development (Product-Line Approach) 257
 Examples..... 258
 Central Ownership of Components 258
 Configuration Management Considerations 258
 Identification..... 259
 Storage 259
 Change Control..... 260
 Status Reporting..... 261

21.5 Quality System, Including Process Management..... 261
 Configuration Management Considerations 262
 Responsibility 262
 Identification..... 262
 Storage 262

Change Control	263
Status Reporting	263

Part V Improving Configuration Management 265

Capability Levels	265
-----------------------------	-----

**Chapter 22 Getting Started on Configuration Management—
up to Capability Level 1 267**

22.1 How to Get Started from Nothing	267
Getting the Right People	267
Collecting Best Practices Internally	268
Looking at the Outside World	268
Focus	268
Look Ahead	269
22.2 First Steps Toward Configuration Management	269
Establish Baselines	269
Track and Control Changes	270
Minimum Documentation	271
Establish Integrity	271
22.3 Experiences in Implementing Configuration Management	272
Overall Conclusion	272
Datamat Ingegneria dei Sistemi	272
S.I.A. S.p.A.	273
Istiservice, S.p.a.	274
Event A/S	275
Sysdeco A/S	276

**Chapter 23 Planning Configuration Management—
up to Capability Level 2 277**

23.1 General Planning Advice	277
The Plan Itself	278
Connection to the Project	278
Template	279
23.2 Table of Contents for a Configuration Management Plan	279

23.3	Configuration Management Plan: Introduction	279
	Purpose	279
	Scope	280
	Vocabulary and Reference Lists	280
23.4	Configuration Management Plan: Management and Relations to the Environment	280
	Organization	281
	Responsibilities	281
	Interface Control	282
	Subcontractor Management	282
	Relevant Standards	282
23.5	Configuration Management Plan: Activities	282
	Identification	283
	Storage	283
	Change Control	284
	Status Reporting	284
23.6	Configuration Management Plan: Schedule	285
	Tasks	285
	Phases and Milestones	285
	Diagrams and Charts	286
23.7	Configuration Management Plan: Tools, Techniques, and Methods	286
	Tools	286
	Techniques and Methods	287
Chapter 24 <i>Processes for Configuration Management— up to Capability Level 3</i>		289
24.1	Processes in General	289
	Connection with Maturity Models	290
	Definitions	290
	A Process Is Like a Recipe	290
	Process Model	291
24.2	Configuration Management Processes—Overview	292
	Special Requirements for Configuration Management Processes	293
24.3	Configuration Management Process—Model Examples	293

Chapter 25 *Continuous Improvement of Configuration Management—
up to Capability Level 4 and 5* 295

25.1	General Software Process Improvement Advice	295
	Processes in Use	296
	Dissemination and Adaptation	296
	Companies at Capability Levels 4 and 5	297
25.2	Metrics for Controlling Configuration Management Performance	297
	Metrics in General	298
	Measuring Methods	300
	Measurement Plan	300
	Examples	300
25.3	Analyzing Metrics for Control and Improvement	301
	Statistics	302
	Balance Point	302
	Variation—What Is Normal	303
	Control Charts	303

Chapter 26 *Tool Support for Configuration Management* 305

26.1	Classes of Tools for Configuration Management	306
	Individual Support	306
	Project-Related Support	307
	Full, Company-Wide Process Support	307
	Who Should Use Which Tool?	308
26.2	Organizational Considerations	308
	Business Goals	308
	Buy It or Do It Yourself	309
	Environmental Constraints	309
	Legacy from the Past	309
	Financing	310
	Organizational Scope	310
	Ownership	310
	Planning for the Future	310
	Willingness to Change	311
26.3	Selecting a Configuration Management Tool	311
	Evaluation Group	311

	Evaluation Method	311
	Requirements	312
	Detailed Evaluation	312
	Nomination of the Winner	312
26.4	Requirements for Configuration Management Tools	312
	Integration with Other Tools	313
	Performance	313
	Scalability	313
	Usability	314
	Web Access	314
26.5	Requirements for the Tool Supplier	314
	Acquaintances	314
	Employees	314
	Financial Status	315
	Focus	315
	Tool Use	315
	Reputation	315
	Support Facilities	316
26.6	Customizing Configuration Management Tools	316
	One Tool or More	316
	Changing Tools or Processes	317
	From Class to Class	317

Appendix A *Configuration Management Process Model:
A Software Code Example 319*

Appendix B *Configuration Management Process Model:
A Tracing Example 333*

Appendix C *Agile SCM 343*

Glossary	349
Bibliography	351
Index	357

List of Figures

Figure I-1	Generic Development Model	<i>xliii</i>
Figure 1-1	Overview of Configuration Management Activities	4
Figure 1-2	Configuration Item Class and Instantiations	6
Figure 1-3	The Life of a Configuration Item Class.	6
Figure 1-4	Identification in Context.	8
Figure 1-5	Document Front Page	10
Figure 1-6	Test Cases	11
Figure 1-7	Storage in Context.	12
Figure 1-8	Example Library Structure.	18
Figure 1-9	Change Control in Context.	19
Figure 1-10	Change Control Process Diagram	24
Figure 1-11	Status Reporting in Context	25
Figure 2-1	CMM Version 1.1 Maturity Levels.	30
Figure 2-2	SPICE Process Area Model	41
Figure 2-3	SPICE Maturity Levels	43
Figure 5-1	Configuration Management Cost for One Item	60
Figure 5-2	Configuration Management Total Cost.	63
Figure 6-1	Configuration Item Class Hierarchy	78
Figure 6-2	Requirement Specification Delivery	84
Figure 6-3	Hardware-Related Delivery.	84

Figure 7-1	Overview of Metadata	90
Figure 7-2	Metadata for Unique Identification	92
Figure 7-3	Metadata for Authorization	96
Figure 7-4	Metadata for Relations to Other Configuration Items	98
Figure 7-5	Example of Tracing	99
Figure 7-6	Metadata for Distribution	101
Figure 8-1	Item Approval in Context.	104
Figure 8-2	Mail Message Used as Approval Registration.	106
Figure 8-3	Unit Approval Form—High Degree of Formalism	107
Figure 8-4	Release Request in Context	108
Figure 8-5	Mail Message Used as Release Registration	109
Figure 8-6	Release Request Form—High Degree of Formalism.	110
Figure 8-7	Excel Workbook Used for Event Registrations— Low Degree of Formalism	116
Figure 8-8	Full Event Life Cycle Registration Form— High Degree of Formalism.	117
Figure 8-9	Change Requests Derived from an Event Registration	119
Figure 8-10	Excel Workbook Used for Change Request Registrations	122
Figure 8-11	Change Request Life Cycle Registration Form— High Degree of Formalism.	123
Figure 9-1	Release Note Example 1	126
Figure 9-2	Release Note Example 2	126
Figure 9-3	Item Status List	127
Figure 9-4	Item History List	128
Figure 9-5	Item Composition List.	129
Figure 9-6	Trace Report, Part a	130
Figure 9-7	Trace Report, Part b	131
Figure 11-1	Multiple Configuration Control Boards	144
Figure 14-1	Customer, Contractor, and Subcontractor	170
Figure IV-1	Configuration Management Complexity for One Item	174
Figure 15-1	Document Delivery.	180
Figure 16-1	Single Requirement	188
Figure 16-2	Test-Related Tracings	194

Figure 17–1	Release Note for Full Delivery (Excerpt)	203
Figure 18–1	Initial Build Success, All Subsystems	215
Figure 18–2	Promotion of Code	215
Figure 18–3	Configurator Subsystem Beta Release August 24, 2001: 75% Successful	216
Figure 18–4	Salespoint Subsystem Beta Release August 17, 2001: 95% Successful	216
Figure 18–5	Stages in Iterative Development.	218
Figure 18–6	Pure Waterfall Development Model	221
Figure 18–7	W-Model.	223
Figure 19–1	Composite System—Example	226
Figure 19–2	Web Usage	236
Figure 20–1	Multisite Development—Sharing Items	240
Figure 20–2	Synchronization of Multisites	242
Figure 20–3	Synchronization	243
Figure 20–4	Multisite Build	244
Figure 20–5	Parallel Development and Merge.	247
Figure 20–6	Naming Conventions in Parallel Development	248
Figure 21.1	Usage Matrix	257
Figure 21–2	Use of Reuse Components in Projects	260
Figure 21–3	Event Registration Involving a Component.	261
Figure V–1	CMMI Capability Levels	266
Figure 23–1	RASIC Chart Example.	281
Figure 24–1	Simple Identification Process	291
Figure 24–2	Flowchart Conventions	294
Figure 25–1	Process Deployment and Improvement	296
Figure 25–2	Average and Range Control Chart	304
Figure 26–1	Configuration Management Tool Classes	306

List of Tables

Table I-1	Improvement Recommendations	xliv
Table 1-1	Overview of Change Control Phases	21
Table 2-1	Mapping from CMM v.1.1 Activities.	31
Table 2-2	Discipline Description in CMMI.	33
Table 2-3	Definition of Capability and Maturity Levels in CMMI	36
Table 2-4	Mapping from CMMI Activities	39
Table 2-5	Mapping from SPICE Best Practices	44
Table 3-1	Standards Overview.	46
Table 3-2	Mapping from ISO 9001:2000 Sections	53
Table 5-1	Activities with a Low Degree of Formalism	64
Table 5-2	Activities with a High Degree of Formalism	65
Table 5-3	Examples of Possible Savings Using a Configuration Management System	69
Table 6-1	Contents of Milestone Deliveries	87
Table 7-1	Configuration Item Name Parts and Their Functions	92
Table 7-2	Source Code Unit State Examples.	95

Table 8–1	Data Elements for Event Registration in the Creation Phase	113
Table 8–2	Data Elements for Event Registration in the Evaluation Phase	114
Table 8–3	Data Elements for Event Registration in the Decision Phase	114
Table 8–4	Data Elements for a Change Request in the Creation Phase	121
Table 8–5	Data Elements for a Change Request in the Implementation Phase	121
Table 10–1	Belbin Team Roles Descriptions	139
Table 15–1	Development in Milestone Deliveries	176
Table 15–2	Unique Identification of Milestone Deliveries	177
Table 15–3	Unique Identification for Documents	181
Table 15–4	Unique Identification of Subitems	182
Table 16–1	Document Types	188
Table 16–2	Unique Identification for Coding Items	189
Table 16–3	Unique Identification for Integration Items	192
Table 17–1	Project Management Document Types	201
Table 19–1	Safety Criticality Classification Example	232
Table 25–1	Metric and Measurement Terms	298
Table 25–2	Data Types	298
Table 25–3	Aspects of Metrics	299
Table 25–4	Scale Definitions	299
Table 25–5	Requirements for Metrics	300
Table 26.1	Buy It or Do It Yourself Considerations	309

Foreword by Kim Caputo

Solving the problems in configuration management can dramatically reduce the cost of rework, not to mention reduce the number of programmer headaches. I was fortunate to work in a company that did very well with configuration management practices on their proprietary systems. However, when they began software development on open systems, it was not so easy. Things that were second nature, that were so internalized that we didn't have to think about them anymore, suddenly became the things we didn't have the foresight to think about on the new systems. We began to have problems again. We had to relearn things that we thought we had learned before, and it was difficult to go back and learn them all over again. The explanations of the concepts, definitions, roles, and responsibilities in this book would have helped us then.

This book will also help those who have never had the appropriate level of discipline in their workplace for configuration management, especially those who have experienced horror stories like these:

- ◆ The Lost Software: "I know I wrote it, but I don't know where I put it."
- ◆ The Missing Links: "This used to work, but now it points to code that isn't there anymore."
- ◆ Stepping on each other's code: Developers doing different fixes in the same code area, overwriting each other.
- ◆ You Can't Go Back Again: New fixes are worse, and there's no "undo" button.
- ◆ You Can't Put It Together Again: Dropped a document with no page numbers, or dropped two documents, no titles on pages, which was which?

- ◆ Who's on First? What's on Second? Bug reported by customer, but don't know what version they have, don't know what fix to give them.
- ◆ "But I Know I Fixed It!"
 - Customer calls and says, "It's broken."
 - Programmer makes the fix, but forgets to check-in the change.
 - Software build is done without the fix (No one audited the baseline).
 - Exact same software shipped to the customer.
 - Customer calls and says, "It's still broken."
 - Programmer says, "But I know I fixed it!"

Configuration management is a cornerstone of software process improvement. (After all, if you can't manage your stuff, how can you tell whether your stuff has improved?) In *CMM Implementation Guide*, I wrote: "In the software industry, many of us have taken steps in software process improvement and made the steps our own, but perhaps many of us have not yet taken the more difficult steps of allowing ourselves to learn from each other and change under cross-cultural influence. It won't happen unless we share our experiences and our techniques. I am sharing my experiences and techniques, not to tell people to do it my way but rather to open the door for us to learn from each other throughout the industry and throughout the world. Perhaps I am not the first to open this door, and I hope that I am not the last. This is an invitation to the dance."

Across the world, nine time zones away, Anne Mette Jonassen Hass has answered the invitation and come through with a wonderful contribution. Here she shares her experiences and techniques for successful configuration management, with several possibilities for solutions that readers can take and make the steps their own. She also includes a wealth of references to reach more information for further learning. I am delighted with this contribution that takes up the call to influence our industry and our world.

—Kim Caputo
Mission Viejo, California

Foreword by Alistair Cockburn

Software configuration management and automated regression testing tools are the two development tools most critical to the success of the agile project. Over the last ten years, the version control and configuration management system was consistently cited to me as the top priority tool to install, both for agility-focused and plan-driven projects. No other tool even came close. (The editor and compiler are so integral that they don't get named.) Teams used to working with a version control and configuration management system refuse to operate without one.

Many teams find that once they have a satisfactory configuration management system in place they can do something more important to their project than merely coordinate their check-ins: They start experimenting with shorter and shorter periods between builds. (This is when the automated regression testing tool becomes important.)

Some teams run fully automated builds every half hour; these also run the suite of unit and system regression tests, post the results on a Web page, and email the owners of any failed code their failing test results! People on these teams report an increase in speed, agility, quality, and personal comfort, knowing they'll learn of unexpected errors within a half-hour of checking in their code.

One company is even experimenting with using such a continuous-build system to synchronize the work between India and the United States. They report that it is helping the two teams stay synchronized with each other across nine time zones.

It is therefore astonishing to see how many teams try to work without a configuration management system. Moreover, it can be frustratingly difficult to find information on the topic.

Anne Mette Hass manages in this book to capture both the heart of the subject and the variations needed in widely varying circumstances—a rare accomplishment. She knows, as you do, that some organizations run with heavy bureaucracy, some with little bureaucracy, some with little formality, some with great formality—and all need configuration management to smooth their collective work. She presents the topic from several angles: the work products, the job roles involved, the organizational issues, the tools, and various levels of formality and bureaucracy. In addition to her insights, Steve Berczuk and Brad Appleton describe, in their appendix, how the terms and practices can be used on the lightest of agile projects.

I have always found this subject daunting, and was pleased to find this text well presented and easy to digest. I could never have written this book; I'm glad that Anne Mette Hass has done it for us.

—Alistair Cockburn
Salt Lake City, Utah

Preface

My Life as a Software Professional

I have two—well, three really—passions in my professional life: test, configuration management, and process improvement.

I started my career as an all-around developer—a little requirements elicitation, a little analysis, a lot of coding and recoding, and some test—more than 20 years ago. During these first professional years, I always loved testing most—making my work run on the computer and enjoying the satisfaction of being told, in a factual and precise way, that something was wrong. This enabled me to carry out the correction and then finally enjoy the privilege of knowing that at least this error was a secret between me and the computer.

My experience grew, and my working teams grew. The problems grew. I wasn't always certain I had produced what I was supposed to and that I had tested everything. And sometimes an error would recur!

I got a job in which I was responsible for system and acceptance test in a company making software for the European Space Agency. For the first time in my then 12-year career, I heard the words *configuration management*. I had no clue as to what it was, but as I spent hours and hours trying to figure it out, discussing it with the person responsible for quality assurance and actually using parts of it in my daily work, I came to understand what a wonderful tool I had.

For the first time, I was able to trace my test cases to the requirements. I was able to tell, at any point, how many requirements I had covered in my test specification and how many were outstanding. I didn't have to encounter the frustration of having made test cases for requirements that weren't going to be implemented. Where I had

forgotten the reason for a turn in the work, I was able to find a previous version of my test specification and see why I had changed it. I loved it!

The last seven years, I've worked as a consultant, spending a good deal of my time on testing assignments of many types in many companies. One of the things I've learned from these assignments is that there is often a difference between what a customer asks for and what he really wants, what he needs (what you want to give him), and what you're able to give him.

Test consultants are often presented with a system to test without the right conditions for performing a professional test. The requirements may be in any state from nonexistent to brilliantly documented, with a pronounced bias toward the former. If requirements are present, they are most often not up to date. This is partly a requirement specification problem and partly a configuration management problem.

Testing requires resources in terms of time and people to perform the test. These resources are often all too scarce. This is a project management problem.

When test consultants plan and perform a test, they need to establish an overview not only of what has to be tested but also how the test is progressing, what errors have been found, and what the state of error correction is. These are configuration management issues.

It's tempting for a consultant to try to deliver what the customer really needs. However, this approach has some limitations and drawbacks. The art is to strike the right balance between what's needed and what's feasible. One of the things to keep in mind as a consultant is to keep up the standards but keep it light. So I try to keep up the configuration management standards as I solve the test assignment—hoping my customer will get an idea of what configuration management is and maybe ask for some assistance in that direction too.

Another part of my time is spent assessing software-producing companies using the BOOTSTRAP maturity model and method. Like the related Capability Maturity Model (CMM), this model includes configuration management. As an assessor in more than 40 assessments, I have time and again seen the blank look in people's eyes when I ask how they perform configuration management. The eyes are rarely less blank if I elaborate and ask about tracing between work products, production of error reports, or other detailed configuration management disciplines.

On the other hand, people are more than willing to talk about problems they've experienced due to lack of control over what is being implemented and tested—and when—and lack of control over what errors have occurred and which ones are being corrected and which are not.

Although configuration management is one of the basic disciplines for sound development (in CMM it is a key process area at level 2), many people go through a

considerable part of their careers without any idea of what it is and how it can ease their everyday tasks, just as I did. So I keep emphasizing its importance and very often recommend it as one of the first disciplines a company should work on when embarking on structured process improvement.

Creation of This Book

In 1999, the Danish organization *Datateknisk Forum*, an association of about 70 software-producing companies, asked me to write a book on configuration management. This was the result of a survey among the members as to what topic they needed a book on. Some of the comments and requirements that came back from the survey were

- ◆ How do you incorporate configuration management in the development process?
- ◆ How do you handle the fact that different kinds of work products, like documents and code, are treated differently?
- ◆ How do you obtain integration between different configuration management tools?
- ◆ How do you handle multisite development?
- ◆ How do you handle configuration management in relation to object-oriented development—component-based development?

I took on the assignment because in my own experience, configuration management has been of great value, not because I felt I knew much about it theoretically. I know much more now, and I hope I've conveyed some of the understanding, knowledge, and appreciation I've gained during my work on this book. If readers try at least some of the detailed disciplines, I hope they will experience the same enthusiasm about its usefulness that I did.

The book is based on literature as well as experience—and also on attitudes and opinions. It contains a lot of examples, advice, and recommendations that are not to be regarded as The Truth but primarily as the sum of a lot of experience—negative as well as positive.

When I learned that the book was to be published in the Agile Series, I knew little about agile development. But as I studied the values and principles, I found that I had practiced it in parts for years. Agile development is a wonderful idea, and one of the cornerstones of its success is configuration management, so it was a pleasure to be able to contribute to the series with one of my favorite disciplines.

The book may seem a bit heavy to some agilists, but I think it's better to discard some formality and detailed activities deliberately, knowing what one hasn't performed, than to just not perform it out of ignorance. So, agilists and others, read and choose!

Purpose of the Book

This book is not supposed to be a primer in configuration management. It does, however, start with an introduction to fundamental principles, to establish a basic understanding of the concepts used. The main part of the book discusses more advanced issues encountered when configuration management has to be implemented. The overall purpose of the book is twofold:

- ◆ To scare those who are engaging in configuration management! The book will give the reader an understanding of the complexity and comprehensiveness of the discipline. *Configuration management is not easy!* If you think it is, you'll be unable to solve its tasks in a professional way.
- ◆ To assuage the fear of those who are engaging in configuration management! The book will provide a fundamental understanding of the principles of the discipline, their interrelations and usage. *Configuration management is not difficult!* All you have to do is do it. If you understand it, it's much easier to specify and plan so it fulfills its purpose and becomes manageable.

It's assumed that the reader has some knowledge of other disciplines within software development, such as planning, design, test, and quality assurance.

Thanks

A lot of people have supported the creation of this book. I have no way of mentioning them all. First, I would like to thank the members and the board of *Datateknisk Forum* and my managers, Mr. Jørn Johansen and Mr. Ole Andersen, for believing in the idea and contributing to the contents.

I would also like to thank my colleagues (especially Ms. Elisabeth Broe Christensen and Mr. Robert Olesen), Mr. Lars Bendix of the University of Lund, Sweden, and not least my husband, Finn, for providing many pieces of good advice and good ideas, and for the interest and patience they have shown during my work on the book. My husband's wry way of looking at things is sometimes annoying but always enlightening—thanks, Finn, for being who you are!

The publisher and my editor, Mr. Ross Venables, deserve lots of thanks for their enthusiasm and encouragement, all the way from my first approach through the development of the manuscript to the complete book.

Last but not least, a big thanks goes to my longtime friend Ms. Pernille Lemvig-Fog and my father, Mr. Birger Jonassen, for their great help with the translation of the text into understandable English.

Introduction

I.1 CONFIGURATION MANAGEMENT IN COMPANY PERSPECTIVE

Every company or organizational unit in a company that develops products should consider configuration management. Configuration management becomes part of the general culture. This means it should be adjusted to the company culture, whether loose, rigorous, or in between. Configuration management may be viewed from different perspectives: people, product, project, cross-organizational, process, and tools. Each is briefly introduced below and discussed at greater length in the book.

People Perspective

Many people affect and are affected by configuration management by fulfilling the roles it involves. These may be categorized as configuration management roles, organizational roles, project-related roles, and external roles.

Product Perspective

Configuration management to be performed for a product depends on the nature of the product. Today, we find more and more complex products composed of different types of subproducts, such as software (applications), hardware (boxes, PCs, peripherals), networks (LAN, Internet), data (system data, parameter values), services (intangible

deliveries such as training and maintenance). Any product may have more or less—even no—emphasis on subproducts. A product may, for example, be

- ◆ A pure software product, delivered on a CD-ROM with no hardware, no initial data, no support or any other service, and no network connection
- ◆ A large control system, including
 - Software embedded in some hardware and in the network
 - PC software with a graphical user interface
 - Network connections for remote surveillance and support
 - Initial data and parameters set
 - Training courses and maintenance services included in the delivery

Products may be simple, complex, or somewhere in between. They may be harmless, with no great impact on human lives or other companies, like games or household equipment, or they may be safety-critical, like flight control systems or medical equipment. They may be developed as shrink-wrapped products, like a test tool, or as bespoke software, like a control system for a factory. Any product has a combination of these attributes.

Project Perspective

The work of developing and maintaining a product may be organized in one project or in a number of projects under different management during the product's lifetime. The project perspective is concerned with performing configuration management for a product in the project or projects during its life cycle. A product goes through a number of life cycle activities, for which configuration management should be considered. These may be preparation, requirements specification, design, production (e.g., coding and/or manufacturing), integration, testing, and operation and maintenance, as illustrated in Figure I-1.

The activities mentioned above are just building blocks that are arranged according to the chosen development model. A number of development models exist, such as the waterfall model (similar to Figure I-1), agile development, incremental development, and iterative development. Each subproduct may follow its own development model—for example, the software subproduct may follow an iterative development model, while the hardware subproduct follows a waterfall model.

As Figure I-1 also shows, a number of support functions exist for preparing, developing, operating, and maintaining a product. These functions, which may include

Project x					
Preparation	Development				Operation and maintenance
	Requirements	Design	Coding	Test	
Project management					
Quality assurance					
Configuration management					

Figure I-1 *Generic Development Model*

project management, quality assurance, and configuration management, should be performed during a product's entire lifetime. Performing these support functions produces objects, which must also be considered for configuration management. The development activities and support functions included in this book are based on the activities and support functions defined in maturity models.

Cross-Organizational Perspective

All companies have cross-organizational objects or assets for which configuration management should be considered: infrastructure, company product assets (such as components for reuse developed using a product-line approach), and company documentation (sales material, plans, quality system, process descriptions, and so on).

Process Perspective

Configuration management may well be the subject for process improvement. In fact, as soon as a company starts to consider configuration management, the process perspective needs to be taken into account. To sustain the work, processes must be understood and implemented and must continuously undergo improvement.

Process improvement and the concept of maturity models to support it, especially in software development, are becoming more and more common in the industry. In the Capability Maturity Model (CMM), configuration management plays a prominent part as a key process area at level 2. Another maturity model, used mostly

(and most) in Europe, is the BOOTSTRAP model. As part of a BOOTSTRAP assessment, a company is given a list of its five processes that most require improvement. As of early 2001, more than 50 BOOTSTRAP assessments had been performed in Denmark. Table I-1 shows the three most frequently appearing processes.

Table I-1 *Improvement Recommendations*

Number	Process	Appearances (%)
1.	Project management	75
2.	Configuration management	55
3.	Test	51

More than half the projects had problems in the way they implemented configuration management and needed to improve their practices. This made configuration management the second most frequent process.

Tools Perspective

It is virtually impossible to manage configuration management without one or more tools. Many tools are available, but many companies prefer to develop their own.

I.2 CONFIGURATION MANAGEMENT BETWEEN COMPANIES

Customers

Some companies have other companies as direct customers. In such cases, the customer’s demand for configuration management may influence how the discipline is carried out in the delivering company. Typically, delivered products form part of components in other products the customers take care of themselves, or the customer takes over responsibility for the finished product.

A company is a purchaser of products, but it may also act as a supplier—for instance, of a requirement specification or components. The customer’s attitude toward configuration management must be clarified where relevant.

Subcontractors

In some cases, subcontractors work for product-producing companies. The way the subcontractors perform configuration management may influence the way it's done in the producing company. Control of the subcontractors is a support function within software development and a discipline or process that ought to be present during a product's whole lifetime. It may be defined for the entire company, if the company has standard procedures for handling subcontractors. (This is not often the case.)

I.3 CONFIGURATION MANAGEMENT IN A BROADER PERSPECTIVE

The World at Large

Universities, research institutes, and companies work with configuration management at several levels. Standards within software development include configuration management as a discipline or a process. During recent years, work in connection with process improvement, including maturity models, has been augmented considerably. Configuration management is included as a process in the best-known maturity models. Furthermore, various institutions and large international projects work with configuration management. These aspects of research into configuration management have been included in the book to provide a larger perspective for what may sometimes seem like an isolated struggle.

A Little Philosophy

Configuration management is the existentialism of software development, because it answers the following questions for individual components or entire products:

- ◆ Who am I?
- ◆ Why am I here?
- ◆ Why am I who I am?
- ◆ Where do I belong?

Just as in “real life,” a certain amount of leisure is necessary for that kind of consideration, but if you have the leisure and use it in a reasonable way, it's possible to increase your quality of life—or, in this case, the quality of your products.

