

Lecture Notes in Computer Science

1500

Edited by G. Goos, J. Hartmanis and J. van Leeuwen

Springer

Berlin

Heidelberg

New York

Barcelona

Hong Kong

London

Milan

Paris

Singapore

Tokyo

Jean-Claude Derniame

Badara Ali Kaba David Wastell (Eds.)

Software Process: Principles, Methodology, and Technology



Springer

Series Editors

Gerhard Goos, Karlsruhe University, Germany
Juris Hartmanis, Cornell University, NY, USA
Jan van Leeuwen, Utrecht University, The Netherlands

Volume Editors

Jean-Claude Derniame
LORIA
Blv. des Aiguillettes, P.O. Box 239
F-54506 Vandoeuvre Cedex, France
E-mail: Jean-Claude.Derniame@loria.fr

Badara Ali Kaba
Polytechnical University of Bobo-Dioulasso
01, P.O. Box 1091, Bobo-Dioulasso 01, Burkina Faso
E-mail: kaba@esi.univ-ouaga.bf

David Wastell
University of Manchester, Department of Computer Science
Oxford Road, Manchester M13 9PL, UK
E-mail: was@cs.man.ac.uk

Cataloging-in-Publication data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Software process : principles, methodology, and technology / Jean-Claude Derniame ... (ed.). - Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ; Singapore ; Tokyo : Springer, 1999
(Lecture notes in computer science ; 1500)
ISBN 3-540-65516-6

CR Subject Classification (1998): K.6.3, D.2, K.6

ISSN 0302-9743

ISBN 3-540-65516-6 Springer-Verlag Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable for prosecution under the German Copyright Law.

© Springer-Verlag Berlin Heidelberg 1999
Printed in Germany

Typesetting: Camera-ready by author
SPIN 10638944 06/3142 - 5 4 3 2 1 0 Printed on acid-free paper

Foreword

*Jean Claude Derniame*¹

Software process technology is an emerging and strategic area that has already reached a reasonable degree of maturity, delivering products and significant industrial experiences. This technology aims at supporting the software production process by providing the means to model, analyse, improve, measure, and whenever it is reasonable and convenient, to automate software production activities. In recent years, this technology has proved to be effective in the support of many business activities not directly related to software production, but relying heavily on the concept of process (i.e. all the applications traditionally associated with workflow management). This book concentrates on the core technology of software processes, its principles and concepts as well as the technical aspect of software process support.

The contributions to this book are the collective work of the Promoter 2 European Working Group. This grouping of 13 academic and 3 industrial partners is the successor of Promoter, a working group responsible for creating a European software process community. Promoter 2 aims at exploiting this emerging community to collectively develop remaining open issues, to coordinate activities and to assist in the dissemination of results. The title “Software Process Modelling and Technology” [Fink94] was produced during Promoter 1. Being “project based”, it presented the main findings and proposals of the different projects then being undertaken by the partners. The present book is more ambitious for two reasons: it is “principles oriented” and it is intended to reflect our common understanding of the key concepts.

In order to produce it, we have adopted, from the beginning, an explicit “book writing” process and we have also described it with one of the available formalisms. This is used as an example in Appendix C to illustrate the discourse and to defend the thesis that software process technology can be exploited in other related domains. Each chapter has specific editors and contributors, and contributions have been discussed and amended before being integrated. The global editing has been decomposed into two facets, with the syntactic and semantic editing undertaken by Ali Kaba and myself, and a complete revision to transform our “Esprit English” into one more correct, with thanks to the IPG at Manchester for their enormous contribution.

Nancy, France

December 1998

1. Coordinator of the Promoter and Promoter 2 Working Groups. Promoter is a research working group funded by the ESPRIT programme under reference WG 21 185. The members can be reached at promoter@loria.fr.

Contents

1	The Software Process: Modelling and Technology	1
1.1	Introduction	1
1.2	The Perspective of this Book	2
1.3	Processes and Process Models	3
1.4	A Simple Example: Software Change	5
1.5	Process Modelling	7
1.5.1	Basic Elements	7
1.5.2	Process Model Levels	8
1.5.3	Process Model Views	9
1.6	Process-sensitive Software Engineering Environments	10
1.7	Meta-Process	11
1.8	Conclusion	12
2	Software Process — Standards, Assessments and Improvement	15
2.1	Introduction	15
2.2	Standard Processes	16
2.2.1	ISO 9000-3	16
2.2.2	PSS-05	17
2.2.3	ISO-12207	18
2.3	Assessment Methods	19
2.3.1	The Capability Maturity Model	19
2.3.2	Bootstrap	21
2.3.3	SPICE	21
2.3.4	Summary	22
2.4	Improvement Methods	22
2.4.1	Quality Improvement Paradigm	22
2.4.2	The Personal Software Process	23
2.4.3	Total Quality Management	24
2.5	Standards and Software Process Technology	25

3	Process Modelling Languages	27
3.1	Introduction	27
3.2	Requirements on Process Modelling Languages	28
3.2.1	Process Elements	29
3.2.2	PML Requirements and Meta-process Phases	31
3.3	Possible PML Technologies from Other Languages/Domains	34
3.3.1	Project Management	34
3.3.2	Formal Specification Languages	34
3.3.3	Informal Design Notations	35
3.3.4	Programming Languages	35
3.3.5	Database Languages	35
3.3.6	CASE Tools and Tool Integration Mechanisms	35
3.3.7	WorkFlow and Groupware	35
3.3.8	The PML Design Dilemma: One or Many PMLs?	36
3.4	Process Modelling Languages in the Promoter Context	38
3.4.1	The Survey Method	38
3.4.2	EPOS SPELL	39
3.4.3	SOCCA	39
3.4.4	Merlin	40
3.4.5	OIKOS	41
3.4.6	ALF	41
3.4.7	ADELE-TEMPO	42
3.4.8	SPADE	42
3.4.9	PEACE+	43
3.4.10	E3	44
3.4.11	PADM	44
3.4.12	Discussion	45
3.5	Other PMLs	47
3.5.1	APPL/A	47
3.5.2	MARVEL	48
3.5.3	Process Weaver	49
3.6	Possible Groups of PMLs and PSEEs	50
3.7	Conclusion	51
4	Meta-Process	53
4.1	Introduction	53
4.1.1	Overview	53
4.1.2	Meta-Process and Quality Improvement	55
4.1.3	Existing Meta-Processes	56
4.2	Requirements for a Meta-Process	59
4.3	A Model of the Meta-Process	61
4.3.1	Introduction	61
4.3.2	Control and Problem Solving	61
4.3.3	Consistency Management	63

4.3.4	Task Decomposition	65
4.3.5	Method Specialisation	66
4.3.6	Remarks on the Model	66
4.4	PROMOTER Reference Model (PRM)	67
4.4.1	Model Structure	67
4.4.2	Method Specialisation	70
4.4.3	Task Decomposition	71
4.4.4	Consistency Management	72
4.5	Validation of the PRM with Respect to Requirements	73
4.6	Empirical Justification of PRM	74
4.6.1	Introduction	74
4.6.2	The Customisation of PRM as QIP	74
4.6.3	The Customisation of PRM as PRISM	76
4.6.4	The Customisation of PRM as “Process Life-cycle”	76
4.6.5	Experience from Empirical Justification	78
4.7	Validation with respect to CMM	80
4.7.1	Introduction	80
4.7.2	Consistency Management	81
4.7.3	Task Decomposition View	81
4.7.4	Method Specialisation View	85
4.8	Validation of PRM with respect to Implementation	86
4.8.1	Introduction	86
4.8.2	ProcessWise Integrator	86
4.8.3	The Model	87
4.8.4	The Scenario	87
4.9	Conclusion	90
4.9.1	Requirements	91
4.9.2	Managing the Process Improvement Process	91
4.9.3	Looking at other Meta-Processes	91
4.9.4	Why Use a PRM?	92
4.9.5	The Way Forward	92
5	Architectural Views and Alternatives	95
5.1	Basic Components	95
5.1.1	A Reference Model for Architectures in PSEEs	95
5.1.2	Dialog Management	96
5.1.3	Process Management	98
5.1.4	Workspace Management	99
5.1.5	Repository Management	100
5.1.6	Communication Management	105
5.1.7	Tools	106
5.2	Architectures for Distributed PSEEs	107
5.2.1	Determinant Requirements on Architectures for Distributed PSEEs	107
5.2.2	Architectural Alternatives for Distributed PSEEs	108

5.3	Example Architecture: The Distributed PSEE Merlin	111
5.3.1	Instance View on the Merlin Architecture	111
5.3.2	Type View on the Merlin Architecture	111
6	Cooperation Control in PSEE	117
6.1	Introduction	117
6.1.1	Objective	117
6.1.2	An Illustrative Example	118
6.1.3	Organisation of the Chapter	123
6.2	Moving from Traditional to Advanced Applications	123
6.2.1	ACID Properties	123
6.2.2	From ACID to Non-ACID	124
6.2.3	From Flat to Nested	124
6.2.4	From Closed to Open	124
6.2.5	Hierarchical versus Layered	125
6.2.6	Homogeneous versus Heterogeneous	125
6.2.7	From Transient to Persistent	125
6.2.8	Available Advanced Transaction Models	125
6.2.9	Summary and Analysis	131
6.3	Impact of Cooperation Control on the Architecture of PSEE	133
6.3.1	Impact of the Repository on Consistency Maintenance	135
6.3.2	Workspaces: an Abstract Level to Support Flexibility	136
6.3.3	Predefined Synchronisation Strategies Layer	138
6.3.4	The Knowledge Management Layer	139
6.3.5	The Interface Layer	140
6.4	Current Work	141
6.4.1	The COO System	141
6.4.2	The MERLIN System	148
6.4.3	The ADELE System	153
6.4.4	The SPADE System	158
6.4.5	Other Facets of Cooperation	164
6.5	Conclusion	164
7	The Human Dimension of the Software Process	165
7.1	Introduction	165
7.2	Three Organisational Contexts of Software Development	166
7.2.1	In-house Development in "ACME Stores": the Fetish of Methodology	166
7.2.2	Case B: Implementing Quality Management in a Software House (Columbine)	168
7.2.3	Case C: User Involvement in the Development of a Medical Workstation	169
7.2.4	General Remarks on the Cases	171
7.3	The Social Dynamics of the Software Process	172
7.3.1	MIS Research on the Software Process	172

7.3.2	The Contribution of Software Psychology	176
7.3.3	Process Modelling and Enactment: Some Practical Experiences.	179
7.4	The Human Role in the Software Process:	
	Dowson's framework	182
7.4.1	Dowson's Framework	182
7.4.2	User Interaction	184
7.4.3	User Interaction, Learning and the Meta-Process	186
7.4.4	Interpersonal Interaction	187
7.5	A Human-Centred Approach to Software Process Support	189
7.5.1	The Need for an "Ecological Approach" in Software Process Research	190
7.5.2	Synergy with Computer Supported Cooperative Work	191
7.5.3	The Limits of the Process Enactment Paradigm	192
7.5.4	The Software Process is a Learning Process	194
7.6	Conclusion	196

8	Software Process: Key Issues and Future Directions	201
8.1	Introduction	201
8.2	Summary of Key Issues	201
8.2.1	Process Modelling Languages	202
8.2.2	The Meta-Process	202
8.2.3	PSEE Architecture	203
8.2.4	Cooperation Control	204
8.2.5	Social Aspects	205
8.3	Wider Applications	207
8.4	Future Trends	210
8.4.1	Evolution of Software Development Practice	210
8.4.2	Technology Evolution	211
8.4.3	Application Domain Evolution	212

Appendix

A	Lifecycle (Sub) Process Demonstration Scenario (ISPW 9)	217
A.1	Background	217
A.2	Introduction	217
A.3	Problem Reporting and Change Process	218
A.4	Sub-scenarios	219
B	Annotated Bibliography on PSEE/PML	223
B.1	PMLs	223
B.1.1	Japanese and American PSEEs	223
B.1.2	European PSEEs	223

**C Case Study Demonstrating the Wider Applicability
of the PSEE Paradigm 227**

C.1 Introduction 227

C.2 Informal Formulation of the Example 227

C.3 A Preliminary Discussion of the Example 228

C.4 A First Level of Process Modelling 230

C.5 A Top-Down LCPS Model for the Example Process 235

C.6 Discussion of the Example Process Models 239

C.7 Conclusion 243

D Assessment Framework for PSEEs 245

D.1 Product 246

D.2 Activity 248

D.3 Workspace 254

D.4 Cooperation 257

D.5 Process and Meta-process Support 263

D.6 Process Tracking and Time Constraints 268

D.7 Human and Social Aspects: Costs and Benefits 273

Glossary 277

References 281

Index 305