Software Product Lines in Action

Frank J. van der Linden · Klaus Schmid
Eelco Rommes

Software Product Lines in Action

The Best Industrial Practice in Product Line Engineering

With 90 Figures and 9 Tables

Springer

Frank J. van der Linden
Philips Medical Systems
Veenpluis 4-6
5684 PC Best, The Netherlands
frank.van.der.linden@philips.com

Eelco Rommes
Philips Research
Prof. Holstlaan 4
5656 AA Eindhoven, The Netherlands
eelco.rommes@xs4all.nl

Klaus Schmid
Universität Hildesheim
Institut für Informatik
Samelsonplatz 1
31141 Hildesheim, Germany
schmid@sse.uni-hildesheim.de

Library of Congress Control Number: 2007923180

ACM Computing Classification (2007): D.2, K.6.3, H.4

ISBN 978-3-540-71436-1 Springer 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, reuse of illustrations, recitation, broadcasting, reproduction on microfilm 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. Violations are liable for prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media
springer.com
© Springer-Verlag Berlin Heidelberg 2007

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: by the authors
Production: Integra Software Services Pvt. Ltd., Puducherry, India
Cover design: KünkelLopka, Heidelberg

Printed on acid-free paper 45/3100/Integra 5 4 3 2 1 0

Foreword

Software product lines represent perhaps the most exciting paradigm shift in software development since the advent of high-level programming languages. Nowhere else in software engineering have we seen such breathtaking improvements in cost, quality, time to market, and developer productivity, often registering in the order-of-magnitude range. Let me say that again: *often registering in the order-of-magnitude range*. Just to be clear, we are talking about software systems built for around one-tenth the cost. With around one-tenth the faults. Delivered in around one-tenth the time. If you know of another way to achieve such a staggering combination of better-faster-and-cheaper, please let me know.

Equally exciting to me are the strategic benefits that accrue to a product line organization, in the form of market agility. At the Software Engineering Institute, we have recorded case study after case study of companies succeeding in one market area with a product line approach, and then taking their production capability to a nearby, under-exploited area of the market, and quickly rising to market dominance in that area as well. And why not? If you can outperform your competitors by order-of-magnitude levels, it's hard to imagine what could keep you from becoming a market leader.

It seems very clear that software product line practice, as a viable and attractive option for software development, is without doubt here to stay. While the underlying concepts are straightforward enough – building a family of related products or systems by planned and careful reuse of a base of generalized software development assets – the devil can be in the details. Successful product line practice can involve organizational change, business process change, and technology change. Bringing comprehensive change to a software development organization isn't easy, and false starts are expensive – or fatal. Better to learn from the experts than strike out through the wilderness on your own.

Which brings us to this book: I fully expect that *Software Product Lines in Action* will become one of the foundational references of this quickly evolving field. It's the most comprehensive treatment of product line practice in

existence today. It's all here, the concepts, a full approach, a holistic treatment of product line practice from the standpoint of business, process, and technology, an analysis method, and a rich collection of case studies. In fact, a big reason to be a fan of this book is its wonderful collection of case studies. Nothing teaches like experience, and the unprecedented ten case studies represent (to my knowledge) the largest collection of experiential software product line reports ever gathered in one reference. More than just third-hand reporting, however, the authors themselves have been integral leaders on many of the case studies on which they report. They have been important contributors to this field almost since it was a field, and can rightly take credit for helping to make software product line practice a known, repeatable, software development approach. In fact, they helped make this field a field. I'm proud to call them colleagues. After you read this book and launch a successful software product line of your own, you'll be proud to call them colleagues too.

Austin, Texas,
January 2007 *Paul Clements*

Preface

The software industry is challenged with a continuous drive to improve its engineering practice. Software has to be produced ever faster and more reliable. Increasingly, complex systems are produced with constant, or even diminishing, numbers of people.

Software product line engineering is a strategic approach to developing software. It impacts business, organisation and technology alike and is a proven way to develop a large range of software products and software-intensive systems fast and at low costs, while at the same time delivering high-quality software.

This book captures the wealth of knowledge that eight companies have gathered during the introduction of the software product line engineering approach in their daily practice.

Who This Book Is For

This book is meant for anyone who is interested in the practical side of product line engineering. Those who consider to use a product line approach in their organisations, those who are about to start one and those who want to improve their current practices will find useful information. This book presents a broad view on product line engineering so that both managers and technical specialists will benefit from reading it. Specific emphasis is given to providing real-world data to support managers in deciding on the potential adoption of product line engineering in their organisations. We believe that best practices are best communicated along with what goes wrong if one fails to adhere to them. This book is also a tool on how to do it right (or wrong), and to learn from the experiences of others.

Background knowledge of product line engineering is not required, but the reader is expected to be familiar with current software engineering practices, or to have some experience in software development.

Readers who want a detailed introduction to the subject are referred to the textbook *Software Product Line Engineering* [106], which describes the foundations, principles and techniques of software product line engineering.

What You Will Learn from Reading This Book

This book gives a practical overview to software product line engineering, driven from industrial experiences that were collected from organisations of varying sizes and domains. Practitioners themselves report on practical implementation: *from practitioners to practitioners.*

This book is complemented with business-related information regarding the benefits and drawbacks of the approach. It not only shows how software product lines can improve the software creation process, but also describes problems that may occur and how companies have solved them in their respective contexts.

The core of this book contains ten case studies, covering small and large organisations, acting in all kinds of domains, with different degrees of domain and process maturity. These companies work on a large variety of software intensive systems including medical imaging, mobile phones, software for televisions, utility control, supervision and management, financial services and car electronics.

The reader will

- understand the relevant aspects, regarding business, architecture, process and organisational issues, of applying software product line engineering.
- learn about the current practice of product line engineering in leading companies of different sizes, operating in several countries and working in various domains.
- have the information for performing an informed analysis on the applicability, or improvement, of the product line approach to his or her own organisation.
- have information about the first steps in transitioning, or improving, the product line approach in his or her organisation.

The Case Studies

Starting in the 1990s, massive investments were made in Europe in the area of product line engineering. This was done both inside companies and as a part of large projects in which companies, research institutes and academia from many different countries co-operated, for example the ESAPS, CAFÉ and FAMILIES projects. One of the results was a flourishing community of product line engineering research and practice in Europe.

The case studies in this book reflect the experiences of companies that were involved in these projects. Each case study was written with experts from the case in question.

The majority of these studies deals with software intensive systems, mainly because the software intensive systems industry in general is more advanced when it comes to software product line engineering. There are several reasons for that.

- Experience with platforms and customisation in other engineering disciplines is often already present in these companies.
- Their customers are used to choosing from a range of systems, each with different properties.
- Pure software customers are often acquainted to and mostly accept the one-size-fits-all system, where they can adapt the system to their specific needs.

However, as more and more information systems — especially business-oriented systems — must be adapted to business-oriented workflows, product line engineering becomes increasingly important. Also for these systems, the adaptation and customisation costs for the client may become prohibitive if systems are delivered with too much diverse and undocumented variability.

The Structure of This Book

This book consist of three parts: a high-level introduction to software product line engineering, ten industrial case studies and their analysis.

Part I — Aspects of Software Product Line Engineering

This part sets a common framework for the description of our industrial case studies. It covers the four major concerns of software product line engineering: Business, Architecture, Process and Organisation. These *BAPO* concerns are a main organising principle of this part. Each of them is explained in detail in a separate chapter. In addition, the Family Evaluation Framework is based on these concerns.

Chapter 1 Product Line Engineering Approach provides the basics you need to understand the book. It explains what product line engineering is, provides an overview of the major aspects (BAPO) and introduces the main topics of software product line engineering: variability and the use of a platform.

Chapter 2 Business explains the business aspects of software product line engineering. It deals with the motivation to initiate or continue with this approach and it explains the economical aspects of software product line engineering.

Chapter 3 Architecture deals with the technical aspects of product lines, most importantly how to deal with variability.

Chapter 4 Process describes the processes for software product line engineering. It describes the separation between domain and application engineering, and the relation between these two life-cycles.

Chapter 5 Organisation deals with roles and responsibilities, structures and distribution of the work.

Chapter 6 The Family Evaluation Framework introduces a BAPO-based framework that can be used to evaluate software product line engineering in larger or smaller parts of companies.

Part II — Experience Reports

This part is the body of the book. It consists of eight experience reports from ten different companies of various sizes and working on various domains.

Chapter 7 Experiences in Product Line Engineering describes the origins of the experiences, the part of software product line engineering that is covered and the formats used within each experiment. The following chapters each describe the company's (or division's) experiences.

Chapter 8 AKVAsmart shows a small company introducing a product-line approach for its range of fish-farming products.

Chapter 9 Bosch Gasoline Systems describes how a product line organisation was set up and executed for a large supplier of automotive products.

Chapter 10 DNV Software deals with the introduction of a product line in ship classification software.

Chapter 11 market maker Software AG shows the business impact of the introduction of a product line on a small company producing financial software.

Chapter 12 Nokia Mobile Phones gives information on the way a product line improves the way to deal with quality requirements, in a large telecom product company.

Chapter 13 Nokia Networks shows another part of this big company, describing the impact of the complex organisation to the product line development.

Chapter 14 Philips Consumer Electronics Software for Televisions shows how all BAPO concerns are affected by the introduction of a product line within this large company

Chapter 15 Philips Medical Systems describes another part of this large company, and how they took a different approach in the introduction of product lines.

Chapter 16 Siemens Medical Solutions shows the difficulties a big company may have to introduce only partially a product line.

Chapter 17 Telvent gives details of the application of an architecture pattern for product lines in the network management domain.

Part III — Conclusions

In this part, conclusions are drawn from an analysis of the set of experiences described. It summarises the lessons learned and provides general guidelines on how to get started with software product line engineering.

Chapter 18 Analysis reflects on the experiences and looks at them from a BAPO and FEF perspective.

Chapter 19 Starting with Software Product Line Engineering presents the steps that need to be taken to successfully make the transition towards software product line engineering, using examples from Part II.

Chapter 20 Outlook looks at trends and expectations for the future. It also describes the challenges that still need to be solved.

Acknowledgements

We thank Eureka/ITEA, BMBF (Germany), SenterNovem (Netherlands) and all other public authorities for funding the projects ESAPS (1999–2001), CAFÉ (2001–2003) and FAMILIES (2003–2005). This book is based on the experience obtained in these projects. However, a lot of additional work was done by a lot of people from the various organisations in order to achieve a quality of the case studies that we can now present here. This goes well beyond project-based work. Each of them is named for the respective case studies, as they were strongly involved and contributed to writing the final case studies.

The BAPO model of product line engineering concerns was a result of the Composable Architectures project conducted at Philips Research between 1998 and 2002. The authors thank their organisations for giving them time to finish the book. These are Philips Medical Systems, Philips Research, Fraunhofer IESE and the University of Hildesheim.

Timo Käkölä provided valuable comments on an earlier version of the book. Monika Lamping did a great job at proof-reading and performing corrections on it and Dennis Stender provided assistance with some pictures. Several figures, viz. 1.1, 1.2, 1.4, 1.5, 2.3, 4.1, 4.2 and 5.1 are copies from the book Software Product Line Engineering [106].

And finally, we would like to thank Ralf Gerstner for accompanying this project for such a long time.

Contents

Part I Aspects of Software Product Line Engineering

1 The Product Line Engineering Approach 3
1.1 Motivation 3
1.2 A Brief History of Software Product Line Engineering 5
1.3 Fundamentals of the Software Product Line Engineering Approach 6
1.4 Variability Management 8
1.4.1 Types of Variability 8
1.4.2 Variability Representation 9
1.4.3 Application Engineering and Variability 11
1.5 Business-Centric 12
1.6 Architecture-Centric 14
1.7 Two-Life-Cycle Approach 14
1.8 The BAPO Model 16
1.9 Summary 19

2 Business 21
2.1 Motivation 21
2.2 Product Line Markets 22
2.2.1 Product Definition Strategy 22
2.2.2 Market Strategies 23
2.2.3 The Product Line Life-Cycle 24
2.2.4 The Relation of Strategy and Product Line Engineering 26
2.3 Product Line Economics 27
2.3.1 Economic Results of Product Line Engineering 27
2.3.2 A Simple Model of Product Line Economics 28
2.3.3 Advanced Aspects of Product Line Economics 29
2.4 Product Management and Scoping 31
2.4.1 Product Portfolio Management 31
2.4.2 Domain Potential Analysis 33
2.4.3 Asset Scoping 34
2.5 Summary 35

3 Architecture ... 37
3.1 Motivation ... 37
3.2 Architecture Concerns ... 38
3.2.1 Architecturally Significant Requirements ... 38
3.2.2 Conceptual Architecture ... 39
3.2.3 Structure ... 39
3.2.4 Texture ... 39
3.3 Product Line Architecting ... 40
3.3.1 Basic Variability Techniques ... 40
3.3.2 Concrete Variation Mechanisms ... 41
3.4 Evaluation ... 42
3.5 Evolution ... 43
3.5.1 End of Life ... 44
3.6 Summary ... 44

4 Process ... 47
4.1 Motivation ... 47
4.2 The Software Product Line Engineering Framework ... 48
4.3 Domain Engineering ... 49
4.3.1 Product Management ... 49
4.3.2 Domain Requirements Engineering ... 49
4.3.3 Domain Design ... 51
4.3.4 Domain Realisation ... 51
4.3.5 Domain Testing ... 52
4.4 Application Engineering ... 53
4.4.1 Application Requirements Engineering ... 53
4.4.2 Application Design ... 54
4.4.3 Application Realisation ... 54
4.4.4 Application Testing ... 54
4.5 Process Maturity: CMMI ... 55
4.5.1 Maturity Levels ... 55
4.5.2 Structure of CMMI Models ... 56
4.6 Summary ... 57

5 Organisation ... 59
5.1 Motivation ... 59
5.2 Roles and Responsibilities ... 61
5.2.1 Product Manager ... 61
5.2.2 Domain Requirements Engineer ... 62
5.2.3 Domain Architect ... 63
5.2.4 Domain Developer ... 63
5.2.5 Domain Tester ... 64
5.2.6 Domain Asset Manager ... 64
5.2.7 Application Requirements Engineer ... 64
5.2.8 Application Architect ... 65

5.2.9 Application Developer 65
5.2.10 Application Tester 65
5.3 Organisational Structures 66
5.3.1 Product-Oriented Organisation 67
5.3.2 Process-Oriented Organisation 69
5.3.3 Matrix Organisation 70
5.3.4 Testing 70
5.3.5 Asset Management 72
5.3.6 Product Management 74
5.4 Geographical Distribution 76
5.5 Collaboration Schemes 77
5.6 Summary 78

6 The Family Evaluation Framework 79
6.1 Motivation 79
6.2 Structure 80
6.3 Business Dimension 82
6.3.1 Level 1: Project-Based 82
6.3.2 Level 2: Aware 83
6.3.3 Level 3: Managed 84
6.3.4 Level 4: Measured 85
6.3.5 Level 5: Optimised 85
6.4 Architecture Dimension 85
6.4.1 Level 1: Independent Development 87
6.4.2 Level 2: Standardised Infrastructure 87
6.4.3 Level 3: Software Platform 87
6.4.4 Level 4: Variant Products 88
6.4.5 Level 5: Configuring 88
6.5 Process Dimension 88
6.5.1 Level 1: Initial 90
6.5.2 Level 2: Managed 90
6.5.3 Level 3: Defined 91
6.5.4 Level 4: Quantitatively Managed 93
6.5.5 Level 5: Optimising 93
6.6 Organisation Dimension 93
6.6.1 Level 1: Project 95
6.6.2 Level 2: Reuse 95
6.6.3 Level 3: Weakly Connected 95
6.6.4 Level 4: Synchronised 96
6.6.5 Level 5: Domain-Oriented 96
6.7 Applying the FEF 97
6.7.1 Complex Organisations 97
6.7.2 Example 100
6.8 Connection to Other Approaches 104
6.9 Summary 105

Part II Experience Reports

7 Experiences in Product Line Engineering . . . 111
7.1 Experimental Software Engineering . . . 112
7.2 Experience Reports on Product Line Development . . . 114
7.3 Case Study Basics . . . 115
7.3.1 Setting Up Case Studies . . . 115
7.3.2 The Case Study Format . . . 116
7.4 Overview of the Case Studies . . . 118

8 AKVAsmart . . . 121
8.1 Introduction . . . 122
8.2 Motivation . . . 122
8.2.1 Case Description . . . 122
8.2.2 Market Drivers . . . 125
8.3 Approach . . . 125
8.4 Architecture . . . 126
8.4.1 The Framework . . . 127
8.4.2 Examples of Plug-ins . . . 128
8.5 Results and Impact Evaluation . . . 129
8.6 Lessons Learned . . . 131
8.7 Outlook . . . 131

9 Bosch Gasoline Systems . . . 133
9.1 Introduction . . . 134
9.2 Motivation . . . 134
9.3 Approach . . . 136
9.3.1 Business Strategy . . . 136
9.3.2 Work Products: Software Architecture . . . 137
9.3.3 Software Components . . . 140
9.3.4 Processes and Methods . . . 141
9.3.5 Tool Environment . . . 143
9.3.6 Organisation . . . 144
9.4 Lessons Learned . . . 144
9.4.1 Management Role . . . 144
9.4.2 Product and Process Excellence – Product Line Engineering and CMMI . . . 146
9.5 Summary . . . 147

10 DNV Software . . . 149
10.1 Introduction . . . 150
10.2 Motivation . . . 151
10.3 Approach . . . 152
10.3.1 First Generation Product Line Engineering . . . 152
10.3.2 Second Generation Product Line Engineering . . . 155

10.4 Results and Impact Evaluation . . . 162
10.5 Lessons Learned . . . 164
10.6 Outlook . . . 165

11 market maker Software AG . . . 167
11.1 Introduction . . . 168
11.2 Motivation . . . 168
11.3 Adoption Process . . . 172
11.3.1 Fast Time to Market . . . 172
11.3.2 New Team . . . 172
11.3.3 Early Focus on Applications . . . 172
11.3.4 No Separation of Domain and Application Engineering Teams . . . 173
11.3.5 Encapsulation of Legacy Systems . . . 173
11.3.6 Simple Architectural Style . . . 173
11.3.7 Effective Communication . . . 174
11.3.8 Immediate and Reliable Decisions . . . 174
11.3.9 Coaching . . . 174
11.3.10 Small Investments . . . 174
11.4 Current Process . . . 175
11.4.1 Business . . . 175
11.4.2 Architecture . . . 175
11.4.3 Process . . . 180
11.4.4 Organisation . . . 184
11.5 Results and Impact Evaluation . . . 186
11.6 Lessons Learned . . . 187
11.7 Summary . . . 189

12 Nokia Mobile Phones . . . 191
12.1 Introduction . . . 192
12.2 Motivation . . . 192
12.3 Approach . . . 193
12.3.1 Typing and Quality Characteristics . . . 195
12.3.2 Traceability . . . 195
12.3.3 The ART Environment . . . 197
12.4 Example: Security . . . 199
12.5 Lessons Learned . . . 204
12.6 Outlook . . . 205

13 Nokia Networks . . . 207
13.1 Introduction . . . 208
13.2 Motivation . . . 208
13.3 Approach . . . 211

13.4 Lessons Learned . . . 214
13.5 Outlook . . . 216

14 Philips Consumer Electronics Software for Televisions . . . 219
14.1 Introduction . . . 220
14.2 Motivation . . . 220
14.3 Approach . . . 223
14.4 Business Aspects . . . 224
14.5 Architecture . . . 224
14.6 Process . . . 227
14.7 Organisation . . . 229
14.8 Results . . . 229
14.9 Lessons Learned . . . 230

15 Philips Medical Systems . . . 233
15.1 Introduction . . . 234
15.2 Motivation . . . 234
15.3 Approach . . . 235
15.3.1 Adoption Approach . . . 235
15.3.2 Current Development Approach . . . 239
15.4 Results and Impact Evaluation . . . 245
15.5 Lessons Learned . . . 246
15.6 Outlook . . . 247

16 Siemens Medical Solutions . . . 249
16.1 Introduction . . . 250
16.2 Motivation . . . 251
16.3 Approach . . . 251
16.3.1 Adoption Process . . . 251
16.3.2 Current Process . . . 252
16.4 Results and Impact Evaluation . . . 261
16.5 Lessons Learned . . . 262
16.6 Summary . . . 263

17 Telvent . . . 265
17.1 Introduction . . . 266
17.2 Motivation . . . 266
17.3 Approach . . . 268
17.3.1 Organisation and Business . . . 269
17.3.2 Using the Abstract Factory Pattern . . . 269
17.3.3 Introducing the Dynamic Abstract Factory Pattern . . . 270
17.3.4 Reusing the Dynamic Abstract Factory Pattern . . . 272
17.4 Lessons Learned . . . 274

Part III Conclusions

18 Analysis 277
18.1 Motivation 277
18.1.1 Complexity 277
18.1.2 Variability and Commonality 278
18.1.3 Efficiency and Costs 279
18.1.4 Reuse and Architecture 279
18.1.5 Quality 279
18.2 Business 280
18.2.1 FEF Evaluations 281
18.3 Architecture 281
18.3.1 FEF Evaluations 282
18.4 Process 283
18.4.1 Evaluations 283
18.5 Organisation 284
18.5.1 FEF Evaluations 284
18.6 Summary 285
18.6.1 How to Do It 285
18.6.2 Guidelines 286
18.6.3 Benefits 287
18.6.4 Concerns 287
18.6.5 Evaluations 288

19 Starting with Software Product Line Engineering 289
19.1 Decide 290
19.1.1 Define Business Strategy and Vision 290
19.1.2 Learn About Software Product Line Engineering 291
19.1.3 Perform a Risk Analysis 291
19.2 Prepare 294
19.2.1 Gain Support 294
19.2.2 Set Concrete Goals 295
19.2.3 Scope the Product Line 296
19.2.4 Evaluate the Organisation 298
19.2.5 Plan the Transition 299
19.3 Transition 300
19.3.1 Roll Out and Institutionalise 300
19.3.2 Evolving the Product Line 302
19.4 Conclusion 303

20 Outlook 305
20.1 Where We Are 305
20.2 Current Shortcomings of Product Line Engineering 306
20.2.1 Methodological Shortcomings 307
20.2.2 Technology and Tools 309
20.3 Going Beyond Product Lines 310
20.4 Product Line Engineering for Practitioners 311

Glossary 313

References 317

About the Authors 327

Index 329