Contents

Part 1: Product Line Management

1. A Scenario-Based Method for Software Product Line Architecting 3
1.1. Introduction 3
1.1.1. Research Questions.....4
1.1.2. Existing Architecting Methods 4
1.1.3. The Use of Scenarios in Architecting 5
1.1.4. Applicability of Scenario-Based Architecting 6
1.1.5. Structure of This Chapter.....6
1.2. Research Method 6
1.3. Method Overview 8
1.3.1. The Views.....8
1.3.2 The Process.....11
1.4. Scenario-Based Architecting Applied.....14
1.4.1. Running Example: The 3D Cathlab.....14
1.4.2. Strategic Scenarios.....15
1.4.3. Explore Architecture Choices.....17
1.4.4. Create Architecture Scenarios 24
1.4.5. Evaluate Candidate Architectures.....29
1.4.6. Select Architecture.....39
1.4.7. Artifacts in the CAFCR Views 40
1.5. Conclusions and Future Research.....49
References 50

2. Strategic Scenario-Based Valuation of Product Line Roadmaps.....53
2.1. Introduction 53
2.2. Research Question 54
2.3. Research Method 55
2.4. Overview of Our Value Evaluation Approach.....56
2.4.1. Net Present Value Calculations 57
2.4.2. Scenario-Based Value Evaluation 57
2.5. Existing (Product Line) Cost and Value Models 59
2.5.1. COCOMO II and Function Points 60
2.5.2. Breakdown of Product Line Cost.....61
2.5.3. Product Line Engineering Cost Reduction Model 62
2.5.4. NPV-Based Product Line Adoption Modeling.....65
2.5.5. CBAM 66
2.5.6. Combining the Models 67

2.6. Product Line Pitfalls and Benefits 69
2.6.1. Pitfall: Platform Over-Design and Perfectionism 69
2.6.2. Pitfall: Short-Term Focus 71
2.6.3. Pitfall: Lack of Vision and Clear Decision Making (No Constancy) 72
2.6.4. Benefit: Time-to-Market Reduction 72
2.6.5. Benefit: Cross-Product Compatibility....75
2.7. A Case "Inspired By Reality"....76
2.7.1. Description of the Case....76
2.7.2. Strategic Scenario 1: Level of Alignment of Business Goals 77
2.7.3. Strategic Scenario 2: Similarity of Functionality....81
2.7.4. Strategic Scenario 3: Evolving System Functionality....82
2.7.5. Summary....85
2.8. Conclusions and Future Research 86
References 88

3. Experiences and Expectations Regarding the Introduction of Systematic Reuse in Small- and Medium-Sized Companies 91
3.1. Introduction 91
3.2. Method and Sample of the Study....93
3.2.1. Method of the Study 93
3.2.2. Sample of the Study....95
3.2.3. Overview of Participating Organizations 96
3.3. State of Practice of Systematic Reuse in the Case Study....98
3.4. Reuse–Invest Specific Results....100
3.4.1. Risk Analysis....101
3.4.2. The Organization's Attitude to Risk....106
3.4.3. Economic Analysis of the Investment on Systematic Reuse 107
3.4.4. Reuse Potential Analysis 109
3.5. Reuse–Check Analysis Results....111
3.5.1. Identified Reuse Situations Description....112
3.5.2. Current State of Reuse Practice Analysis 116
3.5.3. Identified Strengths and Major Problems 118
3.5.4. Improvement Actions Reported to the Organizations....119
3.6. Conclusions and Future Research 121
3.6.1. Current State of Practice of Reuse in the Organizations....122
3.6.2. Reuse Analysis as an Investment....122
3.6.3. Current Situation Characterization....122
3.6.4. Future Research....123
References 124

Part 2: Product Line Requirements Engineering

4. Capturing Product Line Information from Legacy User Documentation 127
4.1. Introduction 127
4.1.1. Outline 129

4.1.2. Research Approach 129
4.2. Problem 130
4.2.1. Product Line Engineering 130
4.2.2. Product Line Modeling 131
4.2.3. User Documentation as Information Source 132
4.3. Related Work 133
4.3.1. Classification 134
4.3.2. Classified Approaches 137
4.4. Metamodel 138
4.4.1. Overview 138
4.4.2. User Documentation Model 139
4.4.3. Requirements Concept Model 141
4.4.4. Variability Model 141
4.4.5. Product Line Artifact Model 141
4.4.6. Extraction Patterns 143
4.4.7. Using the Metamodel 147
4.5. Method 147
4.5.1. Method Overview 148
4.5.2. Preparation 149
4.5.3. Search 150
4.5.4. Selection 151
4.6. Validation of the Approach 151
4.6.1. Industrial Case Study 151
4.6.2. Controlled Experiment 154
4.7. Conclusions and Future Research 156
References 157

5. Scenario-Based Application Requirements Engineering 161
5.1. Introduction 161
5.1.1. Requirements Engineering within Product Line Engineering 161
5.1.2. Application Requirements Engineering 163
5.1.3. Challenges During Application Requirements Engineering 164
5.1.4. Structure of the Chapter 165
5.2. Related Work 166
5.2.1. Requirements Derivation in Product Line Engineering 166
5.2.2. Requirements Reuse in Product Line Engineering 168
5.2.3. Summary of the Related Work 169
5.3. The Orthogonal Variability Modeling Approach 169
5.3.1. Overview of the OVM-A 170
5.3.2. Variability Model for the E-Shop Example 171
5.3.3. Relations Between the Variability Model and Product Line Scenarios 173
5.3.4. Summary of the OVM-A 174
5.4. Use of the Orthogonal Variability Modeling Approach During Application Requirements Engineering 174
5.4.1. Requirements Elicitation 176
5.4.2. Requirements Negotiation 178

5.4.3. Requirements Documentation....181
5.4.4. Requirements Validation....185
5.5. Discussion of the Proposed Approach....188
5.5.1. Industrial Experiences with the OVM-A....188
5.5.2. Experiences in a Laboratory Case Study....190
5.5.3. Validation of the Approach....191
5.6. Conclusions and Future Research....192
References....193

6. Consolidated Product Line Variability Modeling....195
6.1. Introduction....195
6.2. Variability in Standard Languages Exemplified by UML 2.0....197
6.2.1. Introducing the Watch Product Line and its Description in UML 2.0....197
6.2.2. Variability by Means of Templates....198
6.2.3. Variability by Plug-Ins (Component-Based Approach)....199
6.2.4. Variability by Specialization and Redefinition....201
6.3. Variability by Enhancing Languages....202
6.3.1. Earlier Efforts....203
6.3.2. Consolidated Variability Metamodel....206
6.3.3. Variability Mechanisms Expressed by Annotations to UML....211
6.3.4. Management of Variability in UML State Machines....218
6.3.5. Prototype Model Tool Integration....221
6.4. Domain-Specific Languages....227
6.4.1. Similar Efforts: Software Factories....227
6.4.2. Supporting Variability Directly in the Language....228
6.4.3. Supporting Product Derivation Using Generators....231
6.4.4. Defining DSM Support....233
6.5. Evaluation....233
6.5.1. Evaluation Criteria Relative to an Evaluation Reference Model....233
6.5.2. Approaches....234
6.5.3. Evaluation Results....236
6.5.4. Evaluation with Respect to Conventional Systems Engineering....237
6.6. Conclusions and Future Research....239
References....240

Part 3: Product Line Architecture

7. Dealing with Architectural Variation in Product Populations....245
7.1. Introduction....245
7.1.1. The Problem....245
7.1.2. Overview....247
7.2. Architectural Variation....248
7.2.1. The Nature of Architectural Variation....248
7.2.2. Avoiding Architectural Variation....249
7.3. Textural Variation Points....252
7.3.1. Patterns as Architecture Building Blocks....253

7.3.2. Encoding Textural Variation 254
7.3.3. Support for Product Architecture Design 256
7.3.4. Support for Reusable Component Design....257
7.4. Preliminary Validation....257
7.4.1. Philips Equipment Control Platform....258
7.4.2. Composable Image Processor....262
7.4.3. The BRIX Platform....265
7.5. Related Work....269
7.6. Conclusions and Future Research....270
References 272

8. A Software Product Line Reference Architecture for Security 275
8.1. Introduction 275
8.2. Security Architecture Design....277
8.2.1. Encoding Architectural Knowledge....277
8.2.2. Security Design....278
8.2.3. Security Architecture 279
8.2.4. Security Architecture for Software Product Lines 279
8.3. Conceptual Model of the Reference Architecture....280
8.3.1. Security Submodel....281
8.3.2. Architecture Submodel 282
8.3.3. Decision Support Submodel 285
8.4. Quality Model....287
8.5. Decision Model....289
8.5.1. Integrity 289
8.5.2. Confidentiality 294
8.5.3. Availability 295
8.5.4. Accountability 298
8.6. Security Architecture Language 300
8.6.1. Tactics....300
8.6.2. Patterns 302
8.7. Using the Reference Architecture....318
8.7.1. Architecture Derivation 318
8.7.2. Architecture Evaluation 319
8.7.3. Evolution of the Reference Architecture 319
8.8. Validation 320
8.8.1. The Quality Model....320
8.8.2. The Decision Model 321
8.8.3. The Security Architecture Language 322
8.8.4. Summary....322
8.9. Related Work....323
8.10. Conclusions and Future Research....324
References 324

9. Architecture Reasoning for Supporting Product Line Evolution: An Example on Security....327
9.1. Introduction 327

9.2. Software Product Line Architecture 329
9.3. Architecture Recovery 332
9.3.1. Architecture Recovery Methods 334
9.3.2. Architecture Recovery Tools 336
9.3.3. The Process for Architecture Recovery 336
9.4. Architectural Conformance 338
9.5. Conformance and Recovery with Respect to Security 341
9.5.1. Countermeasures 344
9.5.2. Specification of the Security Agent 345
9.6. The Case Study on Security for Distributed Systems 347
9.6.1. Conformance Between Oscar and the OSGi Standard 349
9.6.2. Conformance Between the OSGi and the CIM Standard 352
9.7. Security Model Validation 358
9.7.1. Generic Scenario 360
9.7.2. Criteria 362
9.7.3. Implementation Technologies 364
9.7.4. System Validation 365
9.8. Conclusions and Future Research 366
References 368

10. A Method for Predicting Reliability and Availability at the Architecture Level 373
10.1. Introduction 373
10.2. A Literature Survey of Applicable Methods and Techniques for R&A Prediction 375
10.2.1. Requirement Engineering 375
10.2.2. Architecture Design 376
10.2.3. R&A Analysis 378
10.3. Overview of the RAP Method 379
10.4. Introduction of a Case Example 381
10.5. The First Phase: Defining Reliability and Availability Goals 384
10.5.1. Description of the Steps of the First Phase 384
10.5.2. Applying the Steps to the Case Example 389
10.6. The Second Phase: Representing Reliability and Availability in Architectural Models 395
10.6.1. Description of the Steps of the Second Phase 395
10.6.2. Applying the Steps to the Case Example 398
10.7. The Third Phase: Evaluating Reliability and Availability 404
10.7.1. Description of the Steps of the Third Phase 405
10.7.2. Applying the Steps to the Case Example 411
10.8. Discussion 417
10.9. Conclusions and Future Research 419
References 420

Part 4: Product Line Testing

11. Product Line Use Cases: Scenario-Based Specification and Testing of Requirements 425
11.1. Introduction 425
11.2. PLUC Notation 427
11.2.1. Specification of a PLUC 430
11.3. PUC Derivation from PLUC 433
11.4. Using PLUCs for Derivation of Test Scenarios 434
11.4.1. PLUTO: A Methodology to Derive Test Scenarios 435
11.4.2. An Example 437
11.4.3. Extending the Methodology 439
11.5. Related Work 442
11.6. Conclusions and Future Research 443
References 444

12. System Testing of Product Lines: From Requirements to Test Cases 447
12.1. Introduction 447
12.2. Overview of the Approach 449
12.2.1. From the Product Line Requirements to Product-Specific Requirements 449
12.2.2. Simulating Product-Specific Requirements 450
12.2.3. Generation of the Test Objectives 451
12.2.4. Generation of the Test Scenarios 451
12.2.5. Behavioral Test Patterns and Synthesis of Test Cases 451
12.2.6. An Illustrative Example of Product Line 451
12.3. An Enhanced Use Case Model for Product 453
12.3.1. Enhancing Use Cases with Parameters and Contracts 453
12.3.2. Expressing Variability at the Use Case Level 454
12.4. Simulating the Use Cases 456
12.4.1. The Simulation Model 456
12.4.2. Exhaustive Simulation and Building of a Behavioral Graph 457
12.4.3. Simulating Each Product 458
12.5. Test Objectives 458
12.6. Test Case Generation 462
12.6.1. Generating Test Scenarios 462
12.6.2. Test Scenarios and Test Cases 467
12.6.3. Test Synthesis Tools 468
12.6.4. Using Behavioral Test Patterns 469
12.7. Results and Discussion 470
12.7.1. Test Generated for the 3 Products 471
12.7.2. Study of the Generated Test Efficiency for Demonstration Edition 471
12.7.3. Discussion on the Benefits and Limitations of the Approach 473
12.7.4. Related Work 474
12.8. Conclusions and Future Research 475
References 476

13. The ScenTED Method for Testing Software Product Lines 479
13.1. Introduction 479
13.1.1. Strategies for Testing Product Lines 479
13.1.2. The ScenTED Method 481
13.1.3. Overview 482
13.2. Basics of the ScenTED Method 482
13.2.1. Use Case Based Testing 482
13.2.2. Information Model of ScenTED 483
13.3. ScenTED in Domain Engineering 484
13.3.1. Activities for System Testing 484
13.3.2. Activities for Integration Testing 491
13.4. ScenTED in Application Engineering 493
13.4.1. Creating Application Test Artifacts for System Testing 494
13.4.2. Creating Application Test Artifacts for Integration Testing 502
13.4.3. Ensure the Correct Binding 507
13.4.4. Reuse of Application Artifacts 510
13.5. ScenTED at Siemens Medical Solutions – A Case Study 513
13.5.1. Product Line Development at Siemens Medical Solutions HS 513
13.5.2. Objectives of the ScenTED Introduction 514
13.5.3. Lessons Learned 514
13.5.4. Summary of Results 516
13.6. Conclusions and Future Research 517
References 518

Part 5: Specific Product Line Engineering Issues

14. Incremental Systems Integration within Multidisciplinary Product Line Engineering Using Configuration Item Evolution Diagrams 523
14.1. Introduction 523
14.2. Configuration Management and Problems with Integration 526
14.2.1. Extensions Needed for SCM 527
14.3. Solving the Problems by Using the Configuration Item Evolution Diagram (CIED) 528
14.3.1. Requirements of the Proposed Solution 528
14.3.2. Symbols used in the CIED 529
14.3.3. How the CIED should be Used in Practice 535
14.3.4. Simple Examples of a CIED 536
14.3.5. Linking Test Documentation to Design Documentation 538
14.3.6. A Practical Example of Using a CIED 540
14.3.7. Configuration Item Cycle Times 544
14.4. A Preliminary Validation of the Proposed Solution 545
14.4.1. Comparing Two Case Studies to Illustrate the Usefulness of the CIED 545
14.4.2. Comparing the Two Projects 548
14.4.3. Objective Evidence 549
14.4.4. Qualitative Evidence from Interviews 550

14.5. Conclusions and Future Research 552
References 554

15. Software Product Line Engineering with the UML: Deriving Products 557
15.1. Introduction 557
15.2. Deriving Static Aspects 558
15.2.1. The Mercure Product Line 558
15.2.2. PL Static Architecture as UML Class Diagrams 559
15.2.3. Product Line Constraints 561
15.2.4. From Product Line Models to Product Models 563
15.3. Deriving Behavioral Aspects 567
15.3.1. The Banking Product Line 567
15.3.2. Product Line Behaviors as UML 2.0 Sequence Diagrams 568
15.3.3. Deriving Product Behaviors 574
15.3.4. Implementation and Validation 582
15.4. Related Work 583
15.5. Conclusions and Future Research 585
References 586

16. Evaluation Framework for Model-Driven Product Line Engineering Tools 589
16.1. Introduction 590
16.2. Combining Model-Driven Development and Product Line Engineering 591
16.3. Tool Evaluation Framework 594
16.3.1. Characteristics Elicitation 594
16.3.2. Evaluation Characteristics 597
16.4. Examples of Tool Evaluations 600
16.4.1. The Evaluated Tools 600
16.4.2. A Common Example 601
16.4.3. Atlas Transformation Language (ATL) 602
16.4.4. UML Model Transformation Tool (UMT) 604
16.4.5. ArcStyler 607
16.4.6. XMF-Mosaic 609
16.5. Evaluation of the Framework 613
16.5.1. The Tool Evaluation Framework 613
16.5.2. The Tools Evaluated 614
16.5.3. Applicability of Results 614
16.5.4. Related Work 615
16.6. Conclusions and Future Research 616
References 617

Glossary 619
Index 625

http://www.springer.com/978-3-540-33252-7

Software Product Lines
Research Issues in Engineering and Management
Käkölä, T.; Dueñas, J.C. (Eds.)
2006, XXXIV, 635 p. 251 illus., Hardcover
ISBN: 978-3-540-33252-7