

Contents

Part 1 Introduction	1
1 What is Rationale and Why Does It Matter?	3
1.1 Introduction	3
1.1.1 The Scope and Value of Rationale in Software Engineering	3
1.1.2 Objectives of This Chapter	5
1.2 A Rough Sketch of Research on Rationale.....	5
1.2.1 Argumentative Approaches to Rationale.....	5
1.2.2 Rationale Methods That Go Beyond Argumentation	12
1.3 Why Rationale Matters	13
1.3.1 The Usefulness of Rationale for Artifact Creation	13
1.3.2 The Usefulness of Rationale for Software Engineering	16
1.4 Summary and Conclusions	22
2 What Makes Software Different.....	25
2.1 Introduction	25
2.1.1 Rationale for Software Artifacts versus Rationale for Physical Artifacts.....	25
2.1.2 Objectives of This Chapter	26
2.2 The Roles of the Computer.....	26
2.2.1 Comparison of the Roles of the Computer in the Lifecycles of Physical and Software Artifacts	27
2.2.2 The Significance for Rationale Management in Software Engineering.....	28
2.3 Iteration in Development	32
2.3.1 The Role of Iteration in Different Types of Development	32
2.3.2 Implications of Iteration for Rationale Management in Software Engineering	33
2.4 Summary and Conclusion.....	35
3 Rationale and Software Engineering	37
3.1 Introduction	37
3.1.1 Software Engineering	37
3.1.2 Software Engineering Rationale.....	38

3.1.3 Objectives of This Chapter	38
3.2 Rationale and the Software Process	38
3.2.1 Software Process Definition and Implementation	38
3.2.2 Rationale and SE Process Decision-Making	39
3.3 Rationale and Project Management	41
3.4 Rationale and Software Development	43
3.4.1 Why Capture Software Engineering Rationale?	43
3.4.2 What are the Uses of Software Engineering Rationale?	44
3.4.3 When can Software Engineering Rationale be Used in Software Development?	45
3.4.4 How Can We Support Software Engineering Rationale Use in Software Development?	47
3.5 Summary and Conclusions	47
4 Learning from Rationale Research in Other Domains.....	49
4.1 Introduction	49
4.1.1 Research on Rationale in other Domains.....	49
4.1.2 Objectives of This Chapter	50
4.2 Domain-Oriented Design Environments Using PHI	50
4.2.1 PHIDIAS and JANUS	50
4.2.2 Discussion.....	53
4.3 Automating the Capture of Design Rationale with CAD	55
4.3.1 The Rationale Capture Problem	55
4.3.2 Solution Approach: Automating the Capture of Rationale.....	56
4.3.3 Implementation: The Rationale Construction Framework	57
4.3.4 Discussion.....	58
4.4 Parameter Dependency Networks as Design Rationale.....	59
4.4.1 The DRIVE System and Parameter Dependency Networks....	59
4.4.2 Discussion.....	60
4.5 Case-Based Reasoning as Design Rationale.....	61
4.5.1. From Automated Case-Based Reasoning to Case-Based Design Aids	61
4.5.2 Discussion.....	64
4.6 Summary and Conclusions	66
5 Decision-Making in Software Engineering.....	67
5.1 Introduction	67
5.1.1 General	67
5.1.1 Objectives of this Chapter	67
5.2 Decision-Making Problems	68
5.2.1 Where Decisions Go Wrong.....	68
5.2.2 Poor Decisions in Software	69

5.3 Naturalistic Decision-Making.....	71
5.3.1 Background.....	71
5.3.2 The Recognition-Primed Decision Model.....	72
5.4 Rationale as a Resource for Decision-Making	73
5.4.1 Classical Decision-Making.....	74
5.4.2 Naturalistic Decision-making	75
5.5 Summary and Conclusions	76
Part 2 Uses for Rationale.....	77
6 Presentation of Rationale	79
6.1 Introduction	79
6.1.1 General	79
6.1.2 Objectives of This Chapter	79
6.2 Codifying Rationale Semiformally	80
6.2.1 The rationale for rationale notations.....	80
6.2.2 Hypermedia Presentations of Rationale.....	81
6.2.3 Using Semiformal Rationales	82
6.3 Codifying Rationale Informally.....	83
6.4 Directions.....	85
6.4.1 Reusable Rationale Databases	86
6.4.2 Multi-Scale Presentations of Rationale	87
6.4.3 Integrated Presentation	87
6.5 Summary and Conclusions	89
7 Evaluation.....	91
7.1 Introduction	91
7.1.1 Argumentation-Based Rationale.....	91
7.1.2 Scenario-Based Rationale.....	93
7.1.3 Objectives of This Chapter	93
7.2 Evaluating the Rationale.....	94
7.2.1 Completeness.....	94
7.2.2 Correctness	95
7.3 Evaluating the Decisions	96
7.3.1 Comparing Alternatives.....	96
7.3.2 Combining Inputs from Multiple Developers.....	97
7.3.3 Handling Uncertainty	98
7.4 Scenario-Based Evaluation.....	100
7.5 Summary and Conclusions	101
8 Support for Collaboration.....	103
8.1 Introduction	103

8.1.1 General	103
8.1.2 Objectives of This Chapter	103
8.2 Software Development as Collaborative Work	104
8.2.1 Collaboration Is Inescapable	104
8.2.2 Collaboration Entrains Challenges	105
8.3 Collaboration Supports Rationale.....	106
8.3.1 Collaboration Externalizes Rationales.....	107
8.3.2 Software Development Communities of Practice.....	108
8.4 Rationale Supports Collaboration.....	110
8.4.1 Awareness.....	110
8.4.2 Coordination.....	111
8.5 Summary and Conclusions	112
9 Change Analysis.....	113
9.1 Introduction	113
9.1.1 Issues with Change in Software Development	113
9.1.2 Objectives of This Chapter	115
9.2 Types of Software Changes	115
9.2.1 Functional Requirement Change	116
9.2.2 Nonfunctional Requirement Change	117
9.2.3 Changing Assumptions.....	118
9.2.4 Structural Changes.....	119
9.2.5 Defect Correction	119
9.3 Change Impact Assessment.....	120
9.4 Consistency Management.....	121
9.5 Summary and Conclusions	122
Part 3 Rationale and Software Engineering.....	123
10 Rationale and the Software Lifecycle.....	125
10.1 Introduction	125
10.1.1 Software Engineering Process	125
10.1.2 Objectives of This Chapter	126
10.2 Development Activities and Rationale	126
10.2.1 Project Planning and Management	126
10.2.2 Requirements.....	127
10.2.3 Design.....	127
10.2.4 Implementation.....	128
10.2.5 Verification and Validation	128
10.2.6 Maintenance	129
10.2.7 Retirement	129
10.3 Software Lifecycle Models.....	129

10.3.1 Sequential Models	129
10.3.2 Iterative Models	131
10.3.3 Other Models	133
10.4. Software Process Improvement	136
10.4.1 CMM	136
10.4.2 Personal Software Process	137
10.5. Summary and Conclusions	138
11 Rationale and Requirements Engineering.....	139
11.1 Introduction	139
11.1.1 Requirements Engineering	139
11.1.2 Objectives of This Chapter	140
11.2 Obtaining Requirements	140
11.2.1 Requirements Elicitation	140
11.2.2 Achieving Consensus	142
11.2.3 Requirements Inconsistency	143
11.2.4 Requirements Prioritization	144
11.3 Requirements Traceability	144
11.4 Rationale and Nonfunctional Requirements	146
11.4.1 Nonfunctional Requirement Categorization	146
11.4.2 The NFR Framework	147
11.4.3 SEURAT Argument Ontology and NFR Prioritization	148
11.4.4 NFRs and Conflict Representation and Detection	148
11.5 Goal-Based Requirements Engineering	149
11.5.1 Goal-Based Requirements Analysis	149
11.5.2 Goal-Oriented Requirements Engineering	150
11.5.3 Relationship to Rationale	151
11.6 Adapting to Changing Requirements	152
11.7 Summary and Conclusions	153
12 Rationale and Software Design.....	155
12.1 Introduction	155
12.1.1 The Nature and Importance of Software Design Rationale	156
12.1.2 Objectives of This Chapter	156
12.2 Relating Rationale Approaches to Software Design Processes	157
12.2.1 Decision-centric and Usage-centric Rationale Approaches	157
12.2.2 Prescriptive and Descriptive Roles of Rationale Approaches	159
12.2.3 Rationale for Design Space Analysis and Deeper Reflection	164
12.3 Specific Approaches that Integrate Rationale into Software Design	167

12.3.1 Rationale and Software Architecture.....	167
12.3.2 Strategies for Fitting Rationale into Architectural Design Processes.....	172
12.4 Summary and Conclusions	173
13 Rationale and Software Verification, Validation, and Testing.....	175
13.1 Introduction	175
13.1.1 Verification, Validation, and Testing	175
13.1.2 Software Testing Issues	176
13.1.3 Objectives of This Chapter	177
13.2 Types of Software VV&T	177
13.2.1 Inspection	177
13.2.2 Unit Testing	178
13.2.3 Integration Testing.....	179
13.2.4 System Testing	179
13.3 Rationale Support for Software VV&T	180
13.3.1 Rationale and Testability	180
13.3.2 Rationale and Test Case Prioritization	181
13.3.3 Rationale, Testing, and Component Selection.....	182
13.4 Software Testing Rationale.....	183
13.4.1 Testing Rationale	183
13.4.2 Uses for Testing Rationale	184
13.5 Summary and Conclusions	184
14 Rationale and Software Maintenance	187
14.1 Introduction	187
14.1.1 Software Maintenance and Evolution.....	187
14.1.2 Objectives of This Chapter	188
14.2. Types of Software Maintenance	188
14.3 Improving Maintainability	190
14.3.1 Designing for Maintenance	190
14.3.2 System Reengineering	191
14.4 Software Maintenance Support.....	193
14.4.1 Maintenance Prediction	193
14.4.2 Impact Assessment	193
14.4.3 Program Comprehension	194
14.4.4 Maintenance Recovery	196
14.4.5 Maintenance Rationale	197
14.5 Summary and Conclusions	198
15 Rationale and Software Reuse	199
15.1 Introduction	199

15.1.1 Software Reuse	199
15.1.2 Objectives of This Chapter	200
15.2 Reuse: Concepts and Categories	200
15.2.1 Types of Reuse	200
15.2.2 Types of Rationale for Reuse	202
15.2.3 Reusable Rationale	203
15.3 Applying Rationale	203
15.3.1 Rationale and Patterns	203
15.3.2 Rationale and Component-Based Software Engineering	205
15.3.3 Rationale and Software Product Lines	206
15.3.4 Rationale and COTS-Based Software Engineering	208
15.4. Summary and Conclusions	209
Part 4 Frameworks for Rationale-Based Software Engineering	211
16 A Conceptual Framework	213
16.1 Introduction	213
16.1.1 What a Conceptual Framework Should Do	213
16.1.2 Objectives of This Chapter	214
16.2 General Goals of Rationale Usage in Software Engineering	214
16.3 Rationale: Types of Approaches, Specific Approaches, and Methods	215
16.4 Decision-centric Rationale in Software Engineering	216
16.4.1 Decision-Making in Rationale Approaches	216
16.4.2 Question Answering in Software Engineering	218
16.4.3 Using Decision-centric Rationale in the Full Spectrum of SER	223
16.5 Usage-centric Rationale in Software Engineering	227
16.6 Rationale and Iterative Software Development	228
16.6.1 A Rationale-Based Account of Iterative Development	229
16.6.2 Principles for Rationale Approaches to Support Iterative Development	230
16.6.3 Supporting Iterative Development by Combining Decision- centric and Usage-centric Rationale	234
16.7 Challenges to Rationale Usage	235
16.7.1 Solving the Capture Problem	235
16.7.2 Solving the Delivery Problem	238
16.8 Summary and Conclusions	239
17 An Architectural Framework	241
17.1 Introduction	241

17.1.1 An Integrative Architecture for Rationale-Based Software Engineering.....	241
17.1.2 Objectives of This Chapter	242
17.2 The Need for an Integrative Approach to Rationale Management	243
17.2.1 Representing and Integrating All Types of Software Engineering Rationale	243
17.2.2 Alleviating the Capture and Delivery Problems.....	243
17.3 Framework of an Integrative Architecture for Rationale Management in Software Engineering	248
17.3.1 An Overview of the Framework.....	248
17.3.2 Workings of the Rational Management System	249
17.3.3 Integration with External Systems.....	251
17.4 Summary and Conclusions	254
18 Rationale-Based Software Engineering: Summary and Prospect	255
18.1 Introduction	255
18.1.1. Rationale as an Aid to Software Engineering.....	255
18.1.2 Objectives of This Chapter	256
18.2 Summary of the Book.....	256
18.3 The Challenges of Future Software Development.....	258
18.3.1 Managing Change.....	258
18.3.2 Managing the Increasing Scale, Complexity, and Longevity of Software Projects	258
18.4 The Promise of Rationale-Based Software Engineering	259
18.4.1 Rationale and the Management of Change.....	260
18.4.2 Using Rationale to Manage the Increasing Scale, Complexity, and Longevity of Software Projects.....	261
18.5 Challenges for Rationale-Based Software Engineering.....	261
18.5.1 Addressing the Capture Problem	262
18.5.2 Addressing the Delivery Problem.....	265
18.6. Summary and Conclusions	266
Bibliography	269
Glossary	295
Index.....	311



<http://www.springer.com/978-3-540-77582-9>

Rationale-Based Software Engineering

Burge, J.E.; Carroll, J.M.; McCall, R.; Mistrík, I.

2008, XXXV, 316 p., Hardcover

ISBN: 978-3-540-77582-9