

---

# Contents

---

<b>Introduction</b> . . . . .	xiii
<b>Chapter 1. Realization of a Software Application</b> . . . . .	1
1.1. Introduction. . . . .	1
1.2. Implementation process . . . . .	2
1.3. Quality control . . . . .	3
1.3.1. Introduction . . . . .	3
1.3.2. Software quality assurance plan . . . . .	4
1.4. Conclusion . . . . .	6
<b>Chapter 2. Requirements Management</b> . . . . .	7
2.1. Introduction. . . . .	7
2.2. Three key concepts . . . . .	8
2.2.1. Introduction . . . . .	8
2.2.2. Process requirement . . . . .	9
2.2.3. Requirement . . . . .	9
2.2.4. Attributes. . . . .	13
2.2.5. Allocation versus refinement . . . . .	15
2.2.6. Traceability . . . . .	18
2.3. Process . . . . .	21
2.3.1. Introduction . . . . .	21
2.3.2. Two approaches. . . . .	23
2.3.3. Implementation of tools . . . . .	24
2.4. Conclusion . . . . .	26
<b>Chapter 3. Documentation Structure</b> . . . . .	29
3.1. Introduction. . . . .	29
3.2. Documentation . . . . .	29
3.2.1. Introduction . . . . .	29

3.2.2. Formalization of documents . . . . .	30
3.2.3. Signature and acceptance . . . . .	30
3.3. Quality control . . . . .	31
3.3.1. The company's quality process. . . . .	31
3.3.2. Project quality control . . . . .	32
3.4. Implementation process . . . . .	35
3.5. Configuration management . . . . .	36
3.6. Defect and non-conformity management . . . . .	37
3.6.1. Presentation . . . . .	37
3.6.2. Defect formalization . . . . .	37
3.6.3. Impact of changes . . . . .	38
3.7. Conclusion . . . . .	41
<b>Chapter 4. Requirements Specification of a Software Application . . . . .</b>	<b>43</b>
4.1. Introduction. . . . .	43
4.2. Requirements management . . . . .	44
4.2.1. Presentation . . . . .	44
4.2.2. Requirements manager. . . . .	45
4.3. Specification know-how . . . . .	46
4.4. Quality characteristics of a software product . . . . .	48
4.5. Objectives of the software requirements specification phase. . . . .	50
4.6. Risk assessment . . . . .	55
4.7. System requirements specification document: SwRS . . . . .	56
4.7.1. Presentation . . . . .	56
4.7.2. Characterization of the content of the SwRS. . . . .	57
4.7.3. Identification of software boundaries . . . . .	58
4.7.4. Identification of the operation modes . . . . .	60
4.7.5. Identification of parametrization data . . . . .	63
4.7.6. Identification of services/functions . . . . .	64
4.7.7. Behavior description . . . . .	64
4.8. Specifications formalization . . . . .	67
4.8.1. Modeling for verification . . . . .	67
4.8.2. Text specification versus graphical specification . . . . .	69
4.8.3. Formal method . . . . .	70
4.9. Considering COTS and reused components. . . . .	70
4.10. Specification verification . . . . .	71
4.11. Conclusion . . . . .	71
4.12. Appendix: SwRS template . . . . .	72
<b>Chapter 5. Modeling . . . . .</b>	<b>75</b>
5.1. Introduction. . . . .	75
5.2. Modeling . . . . .	76
5.2.1. Presentation . . . . .	76

---

5.2.2. Model . . . . .	76
5.2.3. Different types of modeling . . . . .	81
5.3. Model transformation . . . . .	82
5.3.1. Presentation . . . . .	82
5.3.2. Different types of transformation . . . . .	82
5.3.3. Abstraction . . . . .	84
5.3.4. Implementation . . . . .	85
5.3.5. Element transformation . . . . .	87
5.3.6. Code generation . . . . .	88
5.3.7. Summary . . . . .	89
5.4. Model refinement . . . . .	91
5.5. Methods . . . . .	93
5.5.1. Semiformal method . . . . .	93
5.5.2. Structured method . . . . .	94
5.5.3. Formal method . . . . .	95
5.5.4. Computer-aided specification tools . . . . .	96
5.5.5. Summary . . . . .	96
5.6. Conclusion . . . . .	96
<b>Chapter 6. Formalization . . . . .</b>	<b>97</b>
6.1. Introduction . . . . .	97
6.2. Approach example . . . . .	97
6.2.1. Context diagram . . . . .	98
6.2.2. Structured method . . . . .	102
6.2.3. Modeling . . . . .	106
6.2.4. Formal method and/or technique . . . . .	121
6.2.5. Synthesis . . . . .	142
6.3. Model verification . . . . .	143
6.4. Setting up of formal methods . . . . .	144
6.5. Implementation of formal methods . . . . .	146
6.5.1. Conventional processes . . . . .	146
6.5.2. Process taking formal methods into account . . . . .	147
6.5.3. Issues . . . . .	149
6.6. Software application maintenance . . . . .	151
6.7. Conclusion . . . . .	151
<b>Chapter 7. Software Specification Verification Stage . . . . .</b>	<b>155</b>
7.1. Introduction . . . . .	155
7.2. Verification . . . . .	156
7.2.1. Presentation . . . . .	156
7.2.2. Verifications of the software requirement specification . . . . .	156
7.2.3. Overview . . . . .	170
7.3. Conclusion . . . . .	171

---

<b>Chapter 8. Component Versus Module</b> . . . . .	173
8.1. Introduction. . . . .	173
8.2. Reusability, maintainability and continuity of service. . . . .	174
8.3. Module and component. . . . .	176
8.3.1. Presentation . . . . .	176
8.3.2. Module . . . . .	176
8.3.3. Component. . . . .	179
8.3.4. Data encapsulation versus information hiding . . . . .	181
8.4. Conclusion . . . . .	181
<b>Chapter 9. Software Application Architecture</b> . . . . .	183
9.1. Introduction. . . . .	183
9.2. Objective of the architecture phase of a software application . . . . .	184
9.3. Software architect . . . . .	186
9.4. Software architecture description document . . . . .	187
9.4.1. Introduction . . . . .	187
9.4.2. Constraint on the methodology . . . . .	187
9.4.3. Description of the interfaces with the environment . . . . .	188
9.4.4. Typical architecture . . . . .	189
9.4.5. Component, module and interface . . . . .	191
9.4.6. Architecture principle . . . . .	193
9.4.7. Description of the software application architecture . . . . .	204
9.5. Verification of the software application architecture . . . . .	208
9.6. Consideration of COTS and reused components . . . . .	209
9.6.1. Introduction . . . . .	209
9.6.2. Preexisting component. . . . .	210
9.6.3. COTS . . . . .	210
9.6.4. Reusability . . . . .	212
9.7. Model approach . . . . .	212
9.8. Conclusion . . . . .	214
<b>Chapter 10. Software Application Architecture Verification</b> . . . . .	215
10.1. Introduction . . . . .	215
10.2. Verification . . . . .	216
10.2.1. Reminder . . . . .	216
10.2.2. Software application architecture verifications. . . . .	216
10.2.3. Synthesis . . . . .	230
10.3. Conclusion . . . . .	230
<b>Chapter 11. Software Application Design.</b> . . . . .	231
11.1. Introduction . . . . .	231
11.2. Component . . . . .	231
11.3. Purpose of the design phase . . . . .	232

---

11.4. Designer . . . . .	234
11.5. Software component specification document . . . . .	234
11.5.1. Objective . . . . .	234
11.5.2. Constraint on the methodology . . . . .	235
11.5.3. Methodology characteristic . . . . .	236
11.6. Software component specification document . . . . .	237
11.6.1. Principles . . . . .	237
11.6.2. Functional decomposition . . . . .	240
11.6.3. Modular or programming unit decomposition . . . . .	241
11.6.4. Data description . . . . .	241
11.6.5. Description of the algorithms . . . . .	242
11.6.6. Requirement traceability . . . . .	243
11.7. Design verification . . . . .	245
11.8. Consideration of COTS and reused components . . . . .	246
11.8.1. Introduction . . . . .	246
11.8.2. Preexisting code . . . . .	247
11.8.3. COTS . . . . .	247
11.9. Conclusion . . . . .	248
<b>Chapter 12. Software Application Component Design Phase Verification . . . . .</b>	<b>249</b>
12.1. Introduction . . . . .	249
12.2. Verification . . . . .	249
12.2.1. Reminder . . . . .	249
12.2.2. Software application component design phase verifications . . . . .	250
12.2.3. Synthesis . . . . .	259
12.3. Conclusion . . . . .	260
<b>Chapter 13. Software Application Coding . . . . .</b>	<b>261</b>
13.1. Introduction . . . . .	261
13.2. Coding phase objective . . . . .	262
13.3. IMPlementer . . . . .	264
13.4. Code production . . . . .	264
13.4.1. Manual coding or automatic generation . . . . .	264
13.4.2. Executable generation sequence . . . . .	266
13.5. Principles to be implemented in the context of the coding phase . . . . .	267
13.5.1. Compliance with the SwCD . . . . .	267
13.5.2. Programming rules . . . . .	268
13.6. The choice of design language . . . . .	271
13.6.1. Introduction . . . . .	271
13.6.2. Essential properties . . . . .	271
13.6.3. Analysis of languages . . . . .	273

13.7. Consideration of COTS and reused components . . . . .	284
13.8. Coding phase verification. . . . .	284
13.9. Conclusion . . . . .	284
<b>Chapter 14. Programming Rules . . . . .</b>	<b>287</b>
14.1. Introduction . . . . .	287
14.2. Characteristics of programming rules. . . . .	287
14.2.1. Introduction . . . . .	287
14.2.2. General rules . . . . .	290
14.2.3. Compilation rules . . . . .	290
14.2.4. Formatting rules . . . . .	291
14.2.5. Documentation conventions . . . . .	292
14.2.6. Software engineering conventions . . . . .	292
14.2.7. Object names. . . . .	294
14.2.8. Variables management . . . . .	296
14.2.9. Rules related to safety principles . . . . .	299
14.3. Programming rules verification . . . . .	300
14.4. Conclusion . . . . .	301
<b>Chapter 15. Coding Verification . . . . .</b>	<b>303</b>
15.1. Introduction . . . . .	303
15.2. Verification . . . . .	303
15.2.1. Reminder . . . . .	303
15.2.2. Methodology. . . . .	304
15.2.3. Compliance with coding rules. . . . .	306
15.3. Dead code and unreachable code . . . . .	307
15.3.1. Introduction . . . . .	307
15.3.2. Unreachable code . . . . .	309
15.3.3. Dead code . . . . .	309
15.3.4. Inhibited code . . . . .	310
15.4. Conclusion . . . . .	310
<b>Chapter 16. Version Sheet of the Software Application . . . . .</b>	<b>311</b>
16.1. Introduction . . . . .	311
16.2. Generating the executable . . . . .	311
16.2.1. Principles . . . . .	311
16.3. Identification of the executable . . . . .	314
16.3.1. Introduction . . . . .	314
16.3.2. Identification of the executable . . . . .	314
16.3.3. Executable protection. . . . .	314
16.4. Version datasheet of the application software. . . . .	315
16.4.1. Presentation of the need . . . . .	315
16.4.2. Implementation . . . . .	315

16.4.3. Compatibility . . . . .	317
16.4.4. Summary . . . . .	317
16.5. Conclusion . . . . .	317
16.6. Appendix: SwVS. . . . .	318
<b>Conclusion</b> . . . . .	319
<b>Glossary</b> . . . . .	321
<b>Bibliography</b> . . . . .	329
<b>Index</b> . . . . .	339