

# **Software Verification and Validation for Practitioners and Managers**

**Second Edition**

Steven R. Rakitin



Artech House  
Boston • London  
[www.artechhouse.com](http://www.artechhouse.com)

# Contents

	<b>Preface to the Second Edition</b>	<b><i>xxi</i></b>
	<b>Acknowledgments</b>	<b><i>xxv</i></b>
	<b>Part I Introduction</b>	<b>1</b>
<b>1</b>	<b>Software in Perspective</b>	<b>3</b>
1.1	The Software Crisis	3
1.2	The Elusive Silver Bullet	5
1.3	Other Attempts to Resolve the Crisis	5
1.3.1	Formal Proof of Correctness	5
1.3.2	Independent Verification and Validation	6
1.3.3	Software Quality Assurance	6
1.3.4	Cleanroom Process	6
1.4	Understanding the Nature of Software	7
1.5	Software Process Improvement Initiatives	8
1.5.1	SEI Capability Maturity Model (CMM <sup>SM</sup> )	8

1.5.2	ISO SPICE	9
1.5.3	Bootstrap	10
1.5.4	ISO 12207	11
1.5.5	Trillium	11
1.6	Summary	12
	References	13
	Web Resources	14
<b>2</b>	<b>Software Development Life-Cycle Models</b>	<b>17</b>
2.1	The Waterfall Model	18
2.2	Concurrent Development Model	21
2.3	The Rapid Prototyping Model	22
2.4	The Spiral Model	24
2.5	Hybrid Models	27
2.6	Model-Based Development	27
2.7	Object-Oriented Models	29
2.8	Summary	32
	References	32
<b>3</b>	<b>Software Development Process</b>	<b>35</b>
3.1	Software Development Process FAQs	36
3.2	Summary	42
	References	42
<b>4</b>	<b>Economic Justification</b>	<b>45</b>
4.1	Economic Justification	47
4.2	Software Defect Cost Models	49
4.3	Measuring the Cost of Quality	53
4.4	Summary	54
	References	55

---

	Selected Bibliography	55
	<b>Part II Overview of Software Verification Activities</b>	<b>57</b>
	Reference	58
<b>5</b>	<b>The Inspection Process</b>	<b>59</b>
5.1	Inspection Process FAQs	61
5.2	Summary	71
	References	71
	Selected Bibliography	71
	Web Resources	72
<b>6</b>	<b>Applying the Inspection Process</b>	<b>73</b>
6.1	Attributes of a Good Process	73
6.1.1	Institutionalizing Inspections	74
6.1.2	Real-Life Experiences	75
6.2	Requirements Inspections	76
6.2.1	Objectives and Prerequisites	77
6.2.2	Requirements Inspection Process	77
6.3	Design Inspection	79
6.3.1	Objectives and Prerequisites	80
6.3.2	Design Inspection Process	80
6.4	Code Inspection	83
6.4.1	Objectives and Prerequisites	83
6.4.2	Code Inspection Process	83
6.5	Test Script Inspection	86
6.5.1	Objectives and Prerequisites	86
6.5.2	Test Procedure Inspection Process	86
6.6	Summary	88
	References	88
<b>7</b>	<b>Software Quality Metrics</b>	<b>91</b>

7.1	Strategy for Implementing a Software Metrics Program	92
7.2	Software Quality Metrics Framework	93
7.2.1	Definitions	94
7.2.2	The Framework	94
7.2.3	Applying the Software Quality Metrics Methodology	95
7.3	Metrics That Support Software Verification Activities	103
7.3.1	Complexity	103
7.3.2	Defect Metrics	105
7.3.3	Product Metrics	106
7.3.4	Process Metrics	106
7.4	Summary	107
	References	108
	Web References	108
<b>8</b>	<b>Configuration Management</b>	<b>109</b>
8.1	Software Configuration Management Basics	111
8.1.1	Definitions	112
8.1.2	Example of a Manufacturing Process	112
8.2	Identification	114
8.2.1	Naming and Labeling	115
8.2.2	Version Control	116
8.2.3	Software Configuration Items	118
8.3	Baseline Management	118
8.3.1	Workspace Management	123
8.3.2	Baseline Change Assessment	124
8.3.3	Version Management	124
8.4	Auditing and Reporting	124
8.4.1	Auditing	125
8.4.2	Configuration Status Accounting	125
8.4.3	Reports, Record Collection, and Retention	126

---

8.5	Summary	126
	References	128
	<b>Part III Overview of Software Validation Activities</b>	<b>129</b>
	Reference	130
<b>9</b>	<b>Testing</b>	<b>131</b>
9.1	Levels, Methods, and Types of Tests	133
9.1.1	Test Levels	133
9.1.2	Test Methods	140
9.1.3	Test Types	141
9.2	Concurrent Development/Validation Testing Model	144
9.2.1	Informal Validation	145
9.2.2	Validation Readiness Review	146
9.2.3	Formal Validation	147
9.3	Test Planning	149
9.3.1	Test Plan	149
9.3.2	Test Procedure	154
9.3.3	Test Report	155
9.4	Summary	155
	References	156
	Web Resources	157
<b>10</b>	<b>Validation Metrics</b>	<b>159</b>
10.1	Time Measures	160
10.1.1	Find-Fix Cycle Time	160
10.1.2	Cumulative Test Time	161
10.2	Test Coverage Metrics	161
10.2.1	Code Coverage	161
10.2.2	Requirements Coverage	163
10.3	Quality Metrics	163
10.3.1	Defect Removal Percentage	164
10.3.2	Defects Reported in Each Baseline	164

---

10.3.3	Defect Detection Efficiency	164
10.4	Summary	165
	References	165
<b>11</b>	<b>Software Reliability Growth</b>	<b>167</b>
11.1	Definitions	168
11.2	The Test-Analyze-Fix Process	168
11.3	Reliability Growth Modeling	169
11.3.1	Objectives of Modeling	170
11.3.2	Types of Models	170
11.3.3	Model Assumptions	172
11.3.4	Model Selection Process	173
11.3.5	Applying the Selected Model	174
11.3.6	Reliability Modeling Tools	174
11.4	Summary	176
	References	177
	Web Resources	177
	<b>Part IV Predictable Software Development</b>	<b>179</b>
	Reference	180
<b>12</b>	<b>Motivation for Becoming Predictable</b>	<b>181</b>
12.1	Introduction to Predictable Software Development	182
12.2	Characteristics of Unpredictable Organizations	186
12.3	Characteristics of Predictable Organizations	188
12.4	Management Can Change the Organization	188
12.5	Summary	192
	References	193
<b>13</b>	<b>Balancing Quality, Features, and Schedule</b>	<b>195</b>
13.1	Quality	197
13.1.1	The Impact of Poor Quality	198

---

13.1.2	Quality and Risk	200
13.2	Features	202
13.3	Schedules	203
13.3.1	Most Projects Are Scheduled Backwards	204
13.3.2	We Don't Teach Estimating and Scheduling Skills	204
13.3.3	We Don't Cultivate Software Project Management Skills	205
13.3.4	We Don't Manage Risk and Commitments	206
13.3.5	We Don't Manage Change	206
13.4	Balancing Quality, Features, and Schedule	207
13.5	Summary	207
	References	211
	Selected Bibliography	212
<b>14</b>	<b><u>Accurate Estimating and Scheduling</u></b>	<b>213</b>
14.1	Why Estimates and Schedules Are Wrong Most of the Time	214
14.2	A Typical Scheduled Backwards Project	217
14.3	Software Estimating Techniques	218
14.3.1	Function Points and Feature Points	219
14.3.2	COCOMO II	219
14.3.3	Wideband Delphi Method	220
14.4	Scheduling Techniques	222
14.4.1	PERT and CPM	222
14.4.2	The Yellow Sticky Method	224
14.5	Summary	225
	References	226
	Selected Bibliography	226
	Web Resources	227
<b>15</b>	<b><u>Balancing People, Process, and Product</u></b>	<b>229</b>
15.1	Process	229



15.1.1	Economic Motivation	230
15.1.2	The Process-Oriented Organization	231
15.1.3	Finding the Right Process	233
15.2	People	235
15.2.1	Provide Motivation	237
15.2.2	Reduce Turnover	239
15.2.3	Build Effective Teams	239
15.2.4	Best Practices for Managing People	240
15.3	Product	242
15.3.1	Project Postmortems	244
15.3.2	Triage Process	245
15.3.3	Root-Cause Analysis	245
15.4	Summary	246
	References	247
<b>16</b>	<b>Managing Commitment and Risk</b>	<b>249</b>
16.1	Managing Commitments	249
16.2	Risk	252
16.3	Risk-Management Techniques	252
16.4	Summary	256
	References	256
	<b>Appendix A:</b>	
	<b>Inspection Roles and Responsibilities</b>	<b>257</b>
A.1	Roles	257
A.2	Responsibilities	258
A.2.1	Moderator	258
A.2.2	Producer	259
A.2.3	Reader	260
A.2.4	Inspectors	260
A.2.5	Recorder (Optional Role)	261
A.2.6	Manager	261

---

	<b>Appendix B:</b>	
	<b>A Sample Inspection Process</b>	<b>263</b>
B.1	Planning	264
B.1.1	Objectives	264
B.1.2	Entry Criteria	264
B.1.3	Activities	264
B.1.4	Exit Criteria	265
B.1.5	Metrics	266
B.2	Overview Meeting (Optional)	266
B.2.1	Objective	266
B.2.2	Entry Criteria	266
B.2.3	Activities	266
B.2.4	Exit Criteria	266
B.2.5	Metrics	267
B.3	Preparation	267
B.3.1	Objective	267
B.3.2	Entry Criteria	267
B.3.3	Activities	267
B.3.4	Exit Criteria	268
B.3.5	Metrics	268
B.4	Inspection Meeting	268
B.4.1	Objective	268
B.4.2	Entry Criteria	268
B.4.3	Activities	268
B.4.4	Exit Criteria	269
B.4.5	Metrics	270
B.5	Follow-Up	270
B.5.1	Objective	270
B.5.2	Entry Criteria	270
B.5.3	Activities	270
B.5.4	Exit Criteria	270
B.5.5	Metrics	271

**Appendix C:**  
**Inspection Process Forms** **273****Appendix D:**  
**Inspection Checklists** **277**

D.1	Requirements Inspection Checklist	277
D.2	Design Inspection Checklist: High-Level Design	278
D.3	Design Inspection Checklist: Detailed Design	279
D.4	Code Inspection Checklist for C Code	282
D.5	A C++ Code Inspection Checklist	284
D.5.1	Variable Declarations	284
D.5.2	Data Usage	287
D.5.3	Initialization	288
D.5.4	Macros	288
D.5.5	Sizing of Data	289
D.5.6	Dynamic Allocation	290
D.5.7	Pointers	291
D.5.8	Casting	291
D.5.9	Computation	291
D.5.10	Conditionals	292
D.5.11	Flow Control	293
D.5.12	Assignment	294
D.5.13	Argument Passing	295
D.5.14	Return Values	295
D.5.15	Function Calls	296
D.5.16	Files	296
D.5.17	Errors Due to Implicit Type Conversions	297
D.5.18	Errors Due to Loss of “Precision” in Return Values	298
D.5.19	Loop Checklist	299
D.5.20	Copyright Notices	299
D.6	Test Procedure Inspection Checklist	300

---

**Appendix E:**  
**Attributes of Good Requirements Specifications**      **301**

---

**Appendix F:**  
**Sample Criteria for Selecting Modules for**  
**Code Inspection**      **303**

---

**Appendix G:**  
**Sample Software Development Process Based**  
**on the Waterfall Model**      **305**

---

G.1	Requirements Analysis Phase	305
G.2	Requirements Definition Phase	307
G.3	Design Phase	308
G.4	Coding Phase	310
G.5	Testing Phase	312
G.6	Maintenance Phase	313

**Appendix H:**  
**Document Outlines**      **315**

---

H.1	Product Concept Document	316
H.2	Software Requirements Specification (SRS)	316
H.3	Software Design Description (SDD)	318
H.4	Software Development Plan	319
H.5	Software Quality Assurance Plan	323
H.6	Software Validation Test Plan	324
H.7	Software Validation Test Procedure	326
H.8	Software Validation Test Report	326
H.9	Software Validation Test Script	327
H.10	Software Configuration Management Plan	328

H.11	Software Release Procedure	330
------	----------------------------	-----

---

**Appendix I: Test Cases for the Triangle Program** **331**

Reference	332
-----------	-----

---

**Appendix J :  
Software Reliability Models** **333**

J.1	Jelinski-Moranda Model	333
J.2	Geometric Model	334
J.3	Schick-Wolverton Model	335
J.4	Goel-Okumoto Nonhomogeneous Poisson Process	335
J.5	Generalized Poisson Model	336
J.6	Brooks-Motley Model	336

---

**Appendix K:  
The Yellow Sticky Method** **339**

K.1	Start with a Complete Software Requirements Specification	340
K.2	Group Requirements into “Must Haves” and “Wants”	340
K.3	Commit to Deliver Only the Must Haves, Not the Wants	341
K.4	Yellow Sticky Estimating Rules	341
K.5	Identifying Tasks and Creating Initial Estimates	342
K.6	Building the Schedule Going Forward	343
K.7	Negotiate Based on Factual Information	345
K.8	Manage the Project to the Schedule	346
K.9	Benefits	346

---

**Appendix L:  
Software Development Best Practices** **347**

---

L.1	Define Requirements First	349
L.2	Binary Quality Gates at the Inch-Pebble Level	354
L.3	Risk Management	357
L.4	Peer Reviews	357
L.5	Project-wide Visibility of Project Plan	357
L.6	Defect Tracking Against Quality Targets	357
L.7	People-Aware Management	360
	References	361
<b>Appendix M:</b>		
<b>Software Quality Best Practices</b>		<b>363</b>
	Selected Bibliography	366
<b>Appendix N:</b>		
<b>Project Postmortems</b>		<b>367</b>
<b>Appendix O:</b>		
<b>Root-Cause Analysis</b>		<b>371</b>
	Reference	374
<b>About the Author</b>		<b>375</b>
<b>Index</b>		<b>377</b>