
Contents

1	Introduction	1
1.1	Introduction	1
1.1.1	The Software Engineering Challenge	2
1.2	History of Software Failures	4
1.3	Background to Software Quality	5
1.3.1	What Is Software Quality?	5
1.3.2	Early Quality Management	6
1.3.3	Total Quality Management	6
1.3.4	Software Quality Control	7
1.4	History of Quality	8
1.4.1	Shewhart	8
1.4.2	Deming	10
1.4.3	Juran	12
1.4.4	Crosby	15
1.4.5	Watts Humphrey	17
1.4.6	Miscellaneous Quality Gurus	19
1.5	Modern Software Quality Management	20
1.5.1	Software Inspections	20
1.5.2	Software Testing	21
1.5.3	Software Quality Assurance	21
1.5.4	Problem Solving Techniques	22
1.5.5	Cost of Quality	24
1.5.6	Software Process Improvement	25
1.5.7	Software Metrics	26
1.5.8	Customer Satisfaction	26
1.5.9	Assessments (Appraisals)	29
1.5.10	Total Quality Management	29
1.6	Miscellaneous	30
1.6.1	Organization Culture and Change	30
1.6.2	Law of Negligence	31
1.6.3	Quality and the WEB	31
1.7	Review Questions	32
1.8	Summary	32

2	Software Engineering	35
2.1	Introduction	35
2.2	What Is Software Engineering?	38
2.3	Challenges in Software Engineering	40
2.4	Software Processes and Lifecycles	42
2.4.1	Waterfall Lifecycle	43
2.4.2	Spiral Lifecycles	44
2.4.3	Rational Unified Process	45
2.4.4	Agile Development	46
2.5	Activities in Waterfall Lifecycle	47
2.5.1	Business Requirements Definition	48
2.5.2	Specification of System Requirements	48
2.5.3	Design	49
2.5.4	Implementation	50
2.5.5	Software Testing	50
2.5.6	Maintenance	52
2.6	Software Inspections	53
2.7	Software Project Management	53
2.8	CMMI Maturity Model	54
2.9	Formal Methods	55
2.10	Review Questions	56
2.11	Summary	56
3	Project Management	59
3.1	Introduction	59
3.2	Project Start Up and Initiation	61
3.3	Estimation	62
3.3.1	Estimation Techniques	63
3.3.2	Work Breakdown Structure	64
3.4	Project Planning and Scheduling	65
3.5	Risk Management	67
3.6	Quality Management in Projects	69
3.7	Project Monitoring and Control	70
3.8	Managing Issues and Change Requests	71
3.9	Project Board and Governance	72
3.10	Project Reporting	73
3.11	Project Closure	73
3.12	Prince 2 Methodology	74
3.13	Review Questions	76
3.14	Summary	76
4	Requirements, Design and Development	77
4.1	Introduction	77
4.2	Requirements Engineering	78
4.2.1	Requirements Elicitation and Specification	79
4.2.2	Requirements Analysis	81

4.2.3	Requirements Verification and Validation	81
4.2.4	Managing Changes to Requirements	82
4.2.5	Requirements Traceability	82
4.3	Architecture Design	84
4.4	Design and Development	86
4.5	Review Questions	87
4.6	Summary	87
5	Configuration Management	89
5.1	Introduction	89
5.2	Configuration Management System	93
5.2.1	Identify Configuration Items	93
5.2.2	Document Control Management	93
5.2.3	Source Code Control Management	94
5.2.4	Configuration Management Plan	95
5.3	Change Control	95
5.4	Configuration Management Audits	98
5.5	Review Questions	98
5.6	Summary	99
6	Software Inspections	101
6.1	Introduction	101
6.2	Economic Benefits of Software Inspections	103
6.3	Informal Reviews	104
6.4	Structured Walkthrough	104
6.5	Semi-formal Review Meeting	105
6.6	Fagan Inspections	108
6.6.1	Fagan Inspection Guidelines	109
6.6.2	Inspectors and Roles	110
6.6.3	Inspection Entry Criteria	110
6.6.4	Preparation	110
6.6.5	The Inspection Meeting	112
6.6.6	Inspection Exit Criteria	114
6.6.7	Issue Severity	114
6.6.8	Defect Type	114
6.7	Automated Software Inspections	116
6.8	Review Questions	117
6.9	Summary	117
7	Software Testing	119
7.1	Introduction	119
7.2	Test Process	121
7.3	Test Planning	125
7.4	Test Case Design and Definition	126
7.5	Test Reporting and Project Sign-off	127
7.6	Testing and Quality Improvement	128

7.7	Traceability of Requirements	129
7.8	Test Tools	130
7.8.1	Test Management Tools	130
7.8.2	Miscellaneous Testing Tools	131
7.9	E-commerce Testing	131
7.10	Review Questions	133
7.11	Summary	133
8	Supplier Selection and Management	135
8.1	Introduction	135
8.2	Planning and Requirements	136
8.3	Identifying Suppliers	137
8.4	Prepare and Issue RFP	137
8.5	Evaluate Proposals and Select Supplier	138
8.6	Formal Agreement	138
8.7	Managing the Supplier	139
8.8	Acceptance of Software	139
8.9	Rollout	140
8.10	Review Questions	140
8.11	Summary	140
9	Software Quality Assurance	143
9.1	Introduction	143
9.2	Audit Planning	146
9.3	Audit Meeting	147
9.4	Audit Reporting	148
9.5	Follow Up Activity	149
9.6	Audit Escalation	149
9.7	Review of Audit Activities	149
9.8	Review Questions	149
9.9	Summary	150
10	Software Metrics	151
10.1	Introduction	151
10.2	The Goal Question Metric Paradigm	152
10.2.1	Goal	153
10.2.2	Question	154
10.2.3	Metrics	154
10.3	The Balanced Scorecard	154
10.4	Metrics for an Organization	156
10.4.1	Customer Satisfaction Metrics	157
10.4.2	Process Improvement Metrics	158
10.4.3	Human Resources and Training Metrics	160
10.4.4	Project Management Metrics	162
10.4.5	Development Quality Metrics	164
10.4.6	Quality Audit Metrics	167

10.4.7	Customer Care Metrics	168
10.4.8	Miscellaneous Metrics	170
10.5	Implementing a Metrics Program	172
10.5.1	Data Gathering for Metrics	173
10.6	Problem-Solving Techniques	174
10.6.1	Fishbone Diagram	175
10.6.2	Histograms	177
10.6.3	Pareto Chart	178
10.6.4	Trend Graphs	179
10.6.5	Scatter Graphs	180
10.6.6	Metrics and Statistical Process Control	181
10.7	Review Questions	182
10.8	Summary	183
11	ISO 9000	185
11.1	Introduction	185
11.2	Motivation for ISO 9000	186
11.3	ISO 9000	187
11.3.1	Quality Management System	187
11.3.2	Management Responsibility	189
11.3.3	Resource Management	190
11.3.4	Product or Service Realization	191
11.3.5	Measuring, Analysis, and Improvement	193
11.4	Implementing ISO 9001	195
11.5	ISO 9000 and Improvement	195
11.5.1	Self-Assessment Process	195
11.5.2	ISO 9001 Certification Process	195
11.6	Review Questions	197
11.7	Summary	198
12	Software Process Improvement	199
12.1	Introduction	199
12.2	What Is a Software Process?	200
12.3	What Is Software Process Improvement?	202
12.4	What Are the Benefits of Software Process Improvement?	203
12.5	What Models Are Used in Software Process Improvement?	204
12.6	Process Mapping	206
12.7	Process Improvement Initiatives	206
12.8	Barriers to Success	207
12.9	Review Questions	208
12.10	Summary	208
13	Capability Maturity Model Integration	211
13.1	Introduction	211
13.2	The CMMI	214

13.3	CMMI Maturity Levels	217
13.3.1	CMMI Representations	220
13.4	Categories of CMMI Processes	222
13.5	CMMI Process Areas	222
13.6	Components of CMMI Process Areas	223
13.6.1	SG 1 – Manage Requirements	227
13.7	SCAMPI Appraisals	229
13.8	Review Questions	231
13.9	Summary	231
14	Setting Up a CMMI Initiative	233
14.1	Introduction	233
14.2	Approach to Continuous Improvement	234
14.3	CMMI Improvement Structure and Teams	236
14.3.1	Setting Up the SEPG Team	237
14.3.2	Setting Up the Steering Group	240
14.3.3	Setting Up Dedicated Improvement Sub-teams	241
14.3.4	Role of the CMMI Project Manager	242
14.3.5	Risks to Success	242
14.4	Planning the Improvement Cycle	243
14.4.1	Appraisals	243
14.4.2	CMMI Project Plan	244
14.4.3	CMMI Project Schedule	245
14.4.4	CMMI Kick-off Session	245
14.5	Implementation of Improvements	246
14.5.1	Process Mapping	246
14.5.2	Layout of Templates	248
14.5.3	Layout of Procedures and Guidelines	248
14.6	Piloting the Process	249
14.7	Rolling Out Process	249
14.8	Review Questions	250
14.9	Summary	250
15	SCAMPI Appraisals	253
15.1	Introduction	253
15.2	Planning and Requirements for the Appraisal	256
15.2.1	Analyze Requirements	256
15.2.2	Develop Appraisal Plan	257
15.2.3	Select and Prepare Team	258
15.2.4	Obtain and Analyze Initial Evidence	259
15.2.5	Prepare for Conducting Appraisal	260
15.3	Conducting the Appraisal	261
15.3.1	Prepare Participants	261
15.3.2	Examine Objective Evidence	262
15.3.3	Document Objective Evidence	264

15.3.4	Verify Objective Evidence	265
15.3.5	Validate Preliminary Findings	266
15.3.6	Generate Appraisal Results	266
15.4	Reporting the Results	267
15.4.1	Deliver Appraisal Results	267
15.4.2	Archive Appraisal Results	267
15.5	Review Questions	268
15.6	Summary	268
16	Software Engineering Tools	271
16.1	Introduction	271
16.2	Tools for Project Management	272
16.3	Tools for Requirements	274
16.4	Tools for Design and Development	278
16.5	Tools for Configuration Management and Change Control	280
16.6	Tools for Code Analysis and Code Inspections	283
16.7	Tools for Testing	284
16.8	Review Questions	286
16.9	Summary	286
17	Formal Methods	289
17.1	Introduction	289
17.2	Why Should We Use Formal Methods?	291
17.3	Applications of Formal Methods	292
17.4	Tools for Formal Methods	293
17.5	Approaches to Formal Methods	294
17.5.1	Model-Oriented Approach	294
17.5.2	Modelling	295
17.5.3	Axiomatic Approach	296
17.6	Proof and Formal Methods	296
17.7	The Future of Formal Methods	298
17.8	The Vienna Development Method	298
17.9	VDM [*] , the Irish School of VDM	299
17.10	The Z Specification Language	301
17.11	The B Method	302
17.12	Predicate Transformers and Weakest Preconditions	302
17.13	The Process Calculii	303
17.14	Finite State Machines	304
17.15	The Parnas Way	305
17.16	Usability of Formal Methods	306
17.16.1	Why Are Formal Methods difficult?	307
17.16.2	Characteristics of a Usable Formal Method	308
17.17	Review Questions	308
17.18	Summary	309

18	Z Formal Specification Language	311
18.1	Introduction	311
18.2	Sets	314
18.3	Relations	315
18.4	Functions	316
18.5	Sequences	318
18.6	Bags	319
18.7	Schemas and Schema Composition	320
18.8	Reification and Decomposition	322
18.9	Proof in Z	323
18.10	Review Questions	324
18.11	Summary	325
19	Unified Modelling Language	327
19.1	Introduction	327
19.2	Overview of UML	328
19.3	UML Diagrams	330
19.3.1	Advantages of UML	335
19.4	Rational Unified Process	335
19.5	Review Questions	337
19.6	Summary	337
20	Epilogue	339
20.1	The Future of Software Quality	341
	Glossary	343
	References	347
	Index	351



<http://www.springer.com/978-3-319-06105-4>

Introduction to Software Quality

O'Regan, G.

2014, XXVII, 354 p. 124 illus., Softcover

ISBN: 978-3-319-06105-4