Contents

Dedication and Acknowledgments xiii

Preface xv

Chapter 1. Introduction 1

1.1. Definition 1
1.2. Why and for who are these books? 3
1.2.1. Why? 3
1.2.2. Who is this book for? 4
1.2.3. Organization of this book 5
1.3. Examples 5
1.4. Limitations 7
1.5. Why test? 7
1.6. MOA and MOE 8
1.7. Major challenges 9
1.7.1. Increased complexity 9
1.7.2. Significant failure rate 12
1.7.3. Limited visibility 12
1.7.4. Multi-sources and complexity 13
1.7.5. Multi-enterprise politics 14
1.7.6. Multiple test levels 16
1.7.7. Contract follow-up, measures, reporting and penalties 18
1.7.8. Integration and test environments 19
1.7.9. Availability of components 20
1.7.10. Combination and coverage 21
1.7.11. Data quality 21
1.7.12. Flows, pivots and data conversions 22

1.7.13. Evolution and transition. 23
1.7.14. History and historization 23
1.7.15. Impostors. 24

Chapter 2. Software Development Life Cycle 27

2.1. Sequential development cycles 28
2.1.1. Waterfall. 29
2.1.2. V-cycle. 30
2.1.3. Spiral and prototyping 32
2.1.4. Challenges of sequential developments. 34
2.2. Incremental development cycles 34
2.2.1. Challenges of incremental development 35
2.3. Agile development cycles. 35
2.3.1. Agile Manifesto. 36
2.3.2. eXtreme Programming 39
2.3.3. Challenges of iterative cycles 40
2.3.4. Lean 47
2.3.5. DevOps and continuous delivery 49
2.3.6. Agile development challenges. 52
2.4. Acquisition. 53
2.5. Maintenance 54
2.6. OK, what about reality?. 55

Chapter 3. Test Policy and Test Strategy. 59

3.1. Test policy 59
3.1.1. Writing test policy 60
3.1.2. Scope of the test policy. 60
3.1.3. Applicability of the test policy. 61
3.2. Test strategy 61
3.2.1. Content of a test strategy. 63
3.2.2. Test strategies and Taylorism 65
3.2.3. Types of test strategies. 66
3.2.4. Test strategy and environments 70
3.3. Selecting a test strategy 71
3.3.1. “Completeness” of the strategy 71
3.3.2. Important points in the strategy 72
3.3.3. Strategy monitoring. 73
3.3.4. Shift left, costs and time 74
3.3.5. “Optimal” strategy 76
3.3.6. Ensuring success 77

3.3.7. Why multiple test iterations? . . . 78
3.3.8. Progress forecast . . . 80
3.3.9. Continuous improvements . . . 81

Chapter 4. Testing Methodologies . . . 83

4.1. Risk-based tests (RBT) . . . 83
4.1.1. RBT hypothesis . . . 84
4.1.2. RBT methodology . . . 84
4.1.3. RBT versus RRBT . . . 90
4.1.4. Reactions to risks . . . 90
4.1.5. Risk computation . . . 91
4.1.6. RBT synthesis . . . 97
4.1.7. Additional references . . . 97
4.2. Requirement-based tests (TBX) . . . 97
4.2.1. TBX hypothesis . . . 98
4.2.2. TBX methodology . . . 99
4.2.3. TBX calculation . . . 99
4.2.4. TBX synthesis . . . 100
4.3. Standard-based (TBS) and systematic tests . . . 101
4.3.1. TBS hypothesis . . . 101
4.3.2. TBS calculation . . . 101
4.3.3. TBS synthesis . . . 102
4.4. Model-based testing (MBT) . . . 102
4.4.1. MBT hypothesis . . . 103
4.4.2. MBT calculation . . . 104
4.4.3. MBT synthesis . . . 104
4.5. Testing in Agile methodologies . . . 105
4.5.1. Agile “test” methodologies? . . . 105
4.5.2. Test coverage . . . 105
4.5.3. Hypothesis . . . 112
4.5.4. Calculation methods . . . 114
4.5.5. Synthesis . . . 116
4.6. Selecting a multi-level methodology . . . 116
4.6.1. Hypothesis . . . 117
4.6.2. Calculation . . . 118
4.7. From design to delivery . . . 119

Chapter 5. Quality Characteristics . . . 121

5.1. Product quality characteristics . . . 122
5.2. Quality in use . . . 125
5.3. Quality for acquirers . . . 125

5.4. Quality for suppliers . . . 126
5.5. Quality for users . . . 126
5.6. Impact of quality on criticality and priority . . . 127
5.7. Quality characteristics demonstration . . . 128
5.7.1. Two schools . . . 128
5.7.2. IADT proofs . . . 129
5.7.3. Other thoughts. . . . 130

Chapter 6. Test Levels . . . 131

6.1. Generic elements of a test level. . . . 132
6.1.1. Impacts on development cycles . . . 133
6.1.2. Methods and techniques . . . 134
6.1.3. Fundamental principles . . . 134
6.2. Unit testing . . . 137
6.3. Component integration testing . . . 139
6.3.1. Types of interfaces to integrate . . . 140
6.3.2. Integration challenges . . . 140
6.3.3. Integration models . . . 141
6.3.4. Hardware–software integration tests . . . 142
6.4. Component tests . . . 143
6.5. Component integration tests . . . 144
6.6. System tests . . . 145
6.7. Acceptance tests or functional acceptance . . . 147
6.8. Particularities of specific systems . . . 148
6.8.1. Safety critical systems . . . 148
6.8.2. Airborne systems . . . 148
6.8.3. Confidentiality and data security . . . 149

Chapter 7. Test Documentation . . . 151

7.1. Objectives for documentation. . . . 152
7.2. Conformity construction plan (CCP). . . . 153
7.3. Articulation of the test documentation . . . 153
7.4. Test policy . . . 154
7.5. Test strategy . . . 155
7.6. Master test plan (MTP) . . . 156
7.7. Level test plan . . . 158
7.8. Test design documents . . . 159
7.9. Test case specification. . . . 160
7.10. Test procedure specification. . . . 160
7.11. Test data specifications . . . 161
7.12. Test environment specification . . . 161

7.13. Reporting and progress reports 161
7.14. Project documentation 162
7.15. Other deliverables 163

Chapter 8. Reporting 165

8.1. Introduction 165
8.2. Stakeholders 167
8.3. Product quality 168
8.4. Cost of defects 168
8.5. Frequency of reporting 170
8.6. Test progress and interpretation 170
8.6.1. Requirements coverage. 171
8.6.2. Risk coverage 172
8.6.3. Component or functional coverage 174
8.7. Progress and defects 175
8.7.1. Defect identification 176
8.7.2. Defects fixing 177
8.7.3. Defect backlog 177
8.7.4. Number of reopened defects 179
8.8. Efficiency and effectiveness of test activities 180
8.9. Continuous improvement 181
8.9.1. Implementing continuous improvements 181
8.10. Reporting attention points 184
8.10.1. Audience 184
8.10.2. Usage 185
8.10.3. Impartiality. 185
8.10.4. Evolution of reporting. 186
8.10.5. Scrum reporting 187
8.10.6. KANBAN reporting. 188
8.10.7. Test design reporting 188
8.10.8. Test execution reporting 189
8.10.9. Reporting software defects 190
8.10.10. UAT progress reporting 194
8.10.11. Reporting for stakeholders 194

Chapter 9. Testing Techniques 197

9.1. Test typologies. 197
9.1.1. Static tests and reviews. 198
9.1.2. Technical tests. 198
9.2. Test techniques. 199
9.3. CRUD. 200

9.4. Paths (PATH) 200
9.4.1. Operation 200
9.4.2. Coverage. . . . 204
9.4.3. Limitations and risks 204
9.5. Equivalence partitions (EP). . . . 204
9.5.1. Objective. . . . 207
9.5.2. Operation 207
9.5.3. Coverage. . . . 207
9.5.4. Limitations and risks 207
9.6. Boundary value analysis (BVA) 207
9.6.1. Objective. . . . 208
9.6.2. Operation 208
9.6.3. Coverage. . . . 208
9.6.4. Limitations and risks 208
9.7. Decision table testing (DTT) 208
9.7.1. Objective. . . . 209
9.7.2. Operation 209
9.7.3. Coverage. . . . 212
9.7.4. Limitations and risks 212
9.8. Use case testing (UCT) 212
9.8.1. Objective. . . . 212
9.8.2. Operation 213
9.8.3. Coverage. . . . 213
9.8.4. Limitations and risks 213
9.9. Data combination testing (DCOT) 214
9.9.1. Objective. . . . 214
9.9.2. Operation 214
9.9.3. Coverage. . . . 214
9.9.4. Challenge 214
9.10. Data life cycle testing (DCYT) 215
9.10.1. Objective 215
9.10.2. Operation. . . . 215
9.10.3. Coverage 215
9.10.4. Challenge. . . . 215
9.11. Exploratory testing (ET). . . . 216
9.11.1. Objective 216
9.11.2. Operation. . . . 216
9.11.3. Coverage 217
9.11.4. Limitations and risks 217
9.12. State transition testing (STT) 218
9.12.1. Objective 218
9.12.2. Operation. . . . 219
9.12.3. Coverage 219

9.13. Process cycle testing (PCT) . . . 219
9.13.1. Objective . . . 219
9.13.2. Operation . . . 219
9.13.3. Coverage . . . 220
9.13.4. Limitations and risks . . . 220
9.14. Real life testing (RLT) . . . 221
9.14.1. Objective . . . 221
9.14.2. Operation . . . 221
9.14.3. Coverage . . . 222
9.14.4. Limitations and risks . . . 222
9.15. Other types of tests . . . 223
9.15.1. Regression tests or non-regression tests (NRTs) . . . 223
9.15.2. Automated tests . . . 224
9.15.3. Performance tests . . . 225
9.15.4. Security tests . . . 226
9.16. Combinatorial explosion . . . 227
9.16.1. Orthogonal array testing (OAT) . . . 228
9.16.2. Classification tree testing (CTT) . . . 229
9.16.3. Domain testing (DOM) . . . 230
9.16.4. Built-in tests (BIT, IBIT, CBIT and PBIT) . . . 231

Chapter 10. Static Tests, Reviews and Inspections . . . 233

10.1. What is static testing? . . . 235
10.2. Reviews or tests? . . . 236
10.2.1. What is a review? . . . 236
10.2.2. What can be subjected to reviews? . . . 236
10.3. Types and formalism of reviews . . . 237
10.3.1. Informal or ad hoc reviews . . . 239
10.3.2. Technical reviews . . . 239
10.3.3. Checklist-based reviews . . . 240
10.3.4. Scenario-based reviews . . . 240
10.3.5. Perspective-based reviews (PBRs) . . . 241
10.3.6. Role-based reviews . . . 241
10.3.7. Walkthrough . . . 241
10.3.8. Inspections . . . 241
10.3.9. Milestone review . . . 242
10.3.10. Peer review . . . 242
10.4. Implementing reviews . . . 242
10.5. Reviews checklists . . . 243
10.5.1. Reviews and viewpoint . . . 243
10.5.2. Checklist for specifications or requirements review . . . 244
10.5.3. Checklist for architecture review . . . 245

10.5.4. Checklist for high-level design review 247
10.5.5. Checklist for critical design review (CDR) 248
10.5.6. Checklist for code review 250
10.6. Defects taxonomies 251
10.7. Effectiveness of reviews 252
10.8. Safety analysis 253

Terminology 255

References 263

Index 269

Summary of Volume 2 271