The Ultimate Guide to ASQMS

Practical Implementation of the Automotive Software Quality Management System Standard

Reinald Melster

Melster Consulting GmbH

PRELIMINARY

Contents

I. Foundations 27

1. Motivation 29
1.1. The Need for an Organizational Standard 29
1.2. The Industry Shift 30
1.2.1. Software Complexity and Scale 31
1.2.2. The Quality Paradox 31
1.2.3. Quality Challenges 32
1.2.4. Why Software Changes Everything 33

2. Introduction 35
2.1. The Need for Action 36
2.2. Scope Changes 36
2.3. Origins of ASQMS 37
2.4. Key Features of ASQMS 38
2.5. Standard Structure and Conventions 39
2.5.1. Document Structure 39
2.5.2. Software Classification and Categories of Practices 40
2.5.3. Requirement Language 41
2.6. ASQMS vs ASPICE 41
2.7. Normative References 43
2.8. Target Audience 44
2.8.1. Prerequisites and Scope 44
2.9. Book Structure 45

PRELIMINARY

3. Process Overview 47
3.1. System Management 48
3.1.1. Context of the Organization 48
3.1.2. Quality Management System Fundamentals . . . 49
3.1.3. Personnel Management 49
3.1.4. Actions to Address Risks and Opportunities . . 50
3.1.5. Performance Evaluation 50
3.1.6. Improvement 51
3.2. Supporting Processes 52
3.2.1. Documentation Management 52
3.2.2. Software Tool Chain Management 53
3.2.3. Equipment and Facilities 53
3.2.4. Knowledge Management 54
3.2.5. Configuration Management 54
3.2.6. Problem Resolution Management 55
3.2.7. Change Request Management 55
3.2.8. Review Management 56
3.2.9. Quality Assurance 56
3.2.10. Externally Supplied Products and Services . . . 57
3.3. Operational Processes 58
3.3.1. Project Management 58
3.3.2. Requirements Development and Management . 59
3.3.3. Architectural Design 59
3.3.4. Software Detailed Design and Implementation . 60
3.3.5. Unit Testing 60
3.3.6. Integration and Integration Testing 61
3.3.7. Verification Testing 61
3.3.8. Machine Learning Design and Data Management 62
3.3.9. Supplier Software Development Management . 63
3.3.10. Software Release 63
3.3.11. Software Deployment 64
3.3.12. Software Maintenance 64
3.3.13. User Information Management 65

PRELIMINARY

3.3.14. Software Termination 65

II. System Management 67

4. Context of the Organization 71

4.1. Why This Process Matters 72
4.2. ASQMS Requirements Overview 73
4.3. Challenges and Best Practices 74
4.3.1. Challenge: Keeping Context Information Current 74
4.3.2. Best Practice: Triggering Context Reviews on Defined Events 74
4.3.3. Challenge: Identifying the Full Set of Relevant Parties 75
4.3.4. Challenge: Connecting Context Analysis to QMS Processes 75
4.4. Interfaces to Other Processes 76

5. Quality Management System Fundamentals 79

5.1. ASQMS Requirements Overview 80
5.2. Why This Process Matters 83
5.2.1. QMS Scope Extends Beyond In-Vehicle Software 83
5.2.2. Tailoring for Type I and Type II Software 85
5.2.3. Leadership Commitment Has Specific Obligations 86
5.3. Challenges and Best Practices 87
5.3.1. Challenge: Defining QMS Scope for the Automotive Software Ecosystem 87
5.3.2. Best Practice: Scope Definition Using Software Type and Deployment Model 88
5.3.3. Challenge: Tailoring Decisions Without Adequate Justification 89
5.3.4. Best Practice: Tailoring Justification Matrix . . . 89
5.3.5. Challenge: Making the Quality Policy Actionable 90
5.3.6. Best Practice: Policy-Objective-Metric Cascade . 91

PRELIMINARY

5.4. Interfaces to Other Processes 92

6. Personnel Management 95
6.1. Why This Process Matters 96
6.2. ASQMS Requirements Overview 97
6.3. Challenges and Best Practices 99
6.3.1. Challenge: Rapidly Evolving Technology Requires Continuous Upskilling 99
6.3.2. Best Practice: Competence Radar and Annual Review Cycle 99
6.3.3. Challenge: Balancing Internal and External Staffing 100
6.3.4. Best Practice: Sourcing Matrix with Competence Anchoring 101
6.3.5. Challenge: Demonstrating Competence Evidence 102
6.3.6. Best Practice: Role-Based Competence Profiles . 102
6.4. Interfaces to Other Processes 103

7. Actions to Address Risks and Opportunities 105
7.1. Why This Process Matters 106
7.2. ASQMS Requirements Overview 106
7.3. Challenges and Best Practices 107
7.4. Interfaces to Other Processes 107

8. Performance Evaluation 111
8.1. Why This Process Matters 112
8.2. ASQMS Requirements Overview 113
8.3. Challenges and Best Practices 114
8.3.1. Challenge: Metrics Without Decision Triggers . . 115
8.3.2. Challenge: End-User Satisfaction in Vehicle Software 116
8.3.3. Challenge: Management Review as a Formal Event 117
8.4. Interfaces to Other Processes 118

PRELIMINARY

9. Improvement **121**
9.1. Why This Process Matters 122
9.2. ASQMS Requirements Overview 123
9.3. Challenges and Best Practices 124
9.3.1. Challenge: Confusing Correction with Corrective Action 124
9.3.2. Best Practice: Classify Non-Conformities at Entry 125
9.3.3. Challenge: Root Cause Analysis Depth 125
9.3.4. Best Practice: Apply Five-Why Analysis to Software Defects 126
9.3.5. Challenge: Continuous Improvement Without Momentum 126
9.3.6. Best Practice: Time-Box Improvement Capacity . 127
9.3.7. Best Practice: Use Retrospectives as Input to Continuous Improvement 127
9.3.8. Best Practice: Measure Test Automation Maturity 128
9.4. Interfaces to Other Processes 128

III. Supporting Processes **131**

10. Documentation Management **133**
10.1. Why This Process Matters 134
10.2. ASQMS Requirements Overview 134
10.3. Challenges and Best Practices 135
10.3.1. Challenge: Distinguishing Documentation Management from Configuration Management . . . 136
10.3.2. Best Practice: Collaborating on Information Item Identification 137
10.3.3. Best Practice: Collaborating on System Selection and Setup 137
10.3.4. Best Practice: Separating Internal and External Document Concerns 138
10.3.5. Challenge: Deciding Where Version Numbers Live 138

PRELIMINARY

10.3.6. Best Practice: Maintaining Version Numbers in Both Locations . . . 139
10.3.7. Best Practice: Combining Automatic and Manual Versioning . . . 140
10.4. Interfaces to Other Processes . . . 141

11. Software Tool Chain Management 143
11.1. Why This Process Matters . . . 143
11.2. ASQMS Requirements Overview . . . 144
11.3. Challenges and Best Practices . . . 145
11.3.1. Challenge: Tool Qualification for Safety-Critical Development . . . 146
11.3.2. Best Practice: Embedding Tool Qualification in Tool Chain Management . . . 147
11.3.3. Challenge: Tool Integration and Data Consistency 149
11.3.4. Best Practice: Single Source of Truth with Read-Only Tool Integration . . . 150
11.3.5. Challenge: Access Rights Management Across Tool Chain . . . 153
11.3.6. Best Practice: Centralized Access Management Through Single Sign-On . . . 154
11.3.7. Challenge: Tool Lifecycle and End-of-Support Management . . . 158
11.3.8. Best Practice: Risk-Based Tool Lifecycle Strategy 159
11.4. Interfaces to Other Processes . . . 163

12. Equipment and Facilities 165
12.1. Why This Process Matters . . . 166
12.2. ASQMS Requirements Overview . . . 166
12.3. Challenges and Best Practices . . . 167
12.3.1. Challenge: Shared Test Resources Across Multiple Projects . . . 168
12.3.2. Best Practice: Centralized Test Resource Management . . . 170

PRELIMINARY

12.3.3. Best Practice: Virtual Test Environments for Early Validation 173
12.3.4. Challenge: Maintenance of Complex Test Equipment 176
12.3.5. Best Practice: Dedicated Equipment Maintenance Team 177
12.3.6. Best Practice: Equipment Configuration Baseline and Verification 179
12.3.7. Challenge: Ownership and Responsibility for Cloud Infrastructure 182
12.3.8. Best Practice: Shared Responsibility Model with Documented Matrix 184
12.3.9. Best Practice: SLA-Based Governance with Audit Rights 187
12.3.10. Challenge: Duplicate Equipment Management Structures 190
12.3.11. Best Practice: Leveraging Existing IATF Equipment Management Infrastructure 192
12.4. Interfaces to Other Processes 195

13. Knowledge Management 197
13.1. Why This Process Matters 198
13.2. ASQMS Requirements Overview 198
13.3. Challenges and Best Practices 199
13.4. Interfaces to Other Processes 199

14. Configuration Management 201
14.1. Why This Process Matters 202
14.2. ASQMS Requirements Overview 202
14.3. Challenges and Best Practices 203
14.3.1. Challenge: Unclear Branching Criteria 203
14.3.2. Challenge: Undefined Merging Criteria 204
14.3.3. Best Practice: Establishing Structured Criteria for Branching and Merging 205

PRELIMINARY

14.3.4. Challenge: Lack of Clear Responsibility for Backup Management 210
14.3.5. Best Practice: Formalizing Project-Specific Backup Management 211
14.3.6. Best Practice: Project-Specific Definition of Backup Requirements 212
14.3.7. Best Practice: Formal Agreement and Monitoring of Backup Implementation 213
14.3.8. Challenge: Archiving at Project and Product End-of-Life 214
14.3.9. Best Practice: Comprehensive and Systematic Archiving 216
14.3.10. Challenge: Status Control in Multi-Level Baselining 216
14.3.11. Best Practice: Baseline of Baselines in Hierarchical System Development 217
14.3.12. Best Practice: Integration of Configuration Management and Change Management 219
14.3.13. Best Practice: Interface to Release Management . 220
14.4. Interfaces to Other Processes 220

15. Problem Resolution Management 223
15.1. Why This Process Matters 224
15.2. ASQMS Requirements Overview 224
15.3. Challenges and Best Practices 225
15.3.1. Challenge: Finding an Appropriate Status Model 225
15.3.2. Best practice: Status model 226
15.3.3. Best practice: Status transitions for Problem . . . 228
15.3.4. Challenge: Confusion Between Urgent Problems and Alert Notifications 229
15.3.5. Best Practice: Urgent Resolution Strategy 231
15.3.6. Best Practice: Pre-Analysis 232
15.3.7. Challenge: Classification of Problems 234

PRELIMINARY

15.3.8. Best Practice: Classification of Problems Along the Architecture 235
15.3.9. Best Practice: Classification of Problems Along the Process Steps 236
15.3.10. Best Practice: Interface to Change Management . 238
15.3.11. Best Practice: Tool Support with Mandatory Fields 240
15.3.12. Challenge: Trend Analysis 241
15.4. Interfaces to Other Processes 243

16. Change Request Management 245
16.1. Why This Process Matters 246
16.2. ASQMS Requirements Overview 246
16.3. Challenges and Best Practices 247
16.3.1. Challenge: Defining a Clear and Practical Status Model 247
16.3.2. Best Practice: Status Model 248
16.3.3. Best Practice: Status Transitions for Change Requests 251
16.3.4. Challenge: Change vs. Heavy Development . . . 253
16.3.5. Best Practice: Apply Change Request Management Only After Baselining 254
16.3.6. Best Practice: Two Levels of Approval 255
16.3.7. Challenge: Implementation Before Approval . . 256
16.3.8. Best Practice: Two Levels of Acceptance – Internal and Stakeholder 257
16.3.9. Challenge: Late Changes 258
16.3.10. Best Practice: Structured Change Handling Under Time Pressure 259
16.3.11. Challenge: Informal Change Requests Over the Phone 260
16.3.12. Best Practice: Ensure Formalization of All Change Requests 261

PRELIMINARY

16.3.13. Challenge: Uncontrolled Changes to Project Scope (Scope Creep) 262
16.3.14. Best Practice: Apply Change Management to Scope Modifications 263
16.3.15. Challenge: Change Request Management Limited to Product Artifacts 265
16.3.16. Best Practice: Include PMT Changes 267
16.3.17. Best Practice: Ensure Traceability Between Problems and Change Requests 269
16.3.18. Challenge: Missing Traceability Between Change Requests and Changed Artifacts 270
16.3.19. Best Practice: Ensure Traceability Between Change Requests and Changed Artifacts 271
16.3.20. Best Practice: Document Changes in the Release Notes 272
16.3.21. Best Practice: Include Change Requests in the Project Scope 272
16.4. Interfaces to Other Processes 273

17. Review Management 275
17.1. Why This Process Matters 276
17.2. ASQMS Requirements Overview 276
17.3. Challenges and Best Practices 277
17.4. Interfaces to Other Processes 277

18. Quality Assurance 279
18.1. Why This Process Matters 280
18.2. ASQMS Requirements Overview 280
18.3. Challenges and Best Practices 281
18.3.1. Challenge: The QA Independence Paradox . . . 282
18.3.2. Best Practice: Two-Level Quality Assurance Structure 283
18.3.3. Best Practice: Disciplinary Separation of Quality Engineers 284

PRELIMINARY

18.3.4. Best Practice: Establishing QA Independence . . 286
18.3.5. Best Practice: Four-Eyes Principle for Quality Verification 287
18.3.6. Challenge: "End-of-Line" vs. "Built-In" Quality Mindset 288
18.3.7. Best Practice: Constructive Quality Assurance Approach 290
18.3.8. Best Practice: Regular Compliance Checks 291
18.3.9. Best Practice: Self-Organized Reviews in Mature Teams 292
18.3.10. Challenge: Efficient Reviews 294
18.3.11. Challenge: Reviews are Misunderstood as Inspections 295
18.3.12. Best Practice: Comprehensive Review Method Definitions 296
18.3.13. Best Practice: Tool-Supported Review Implementation 297
18.3.14. Best Practice: Full Reviews versus Delta Reviews 298
18.3.15. Best Practice: Review Termination for Immature Artifacts 300
18.3.16. Challenge: Balancing the Use of Checklists in Reviews 302
18.3.17. Best Practice: Appropriate Checklist Level and Frequency 303
18.4. Interfaces to Other Processes 305

19. Externally Supplied Products and Services 307
19.1. Why This Process Matters 308
19.2. ASQMS Requirements Overview 308
19.2.1. Selection and Qualification of Potential Suppliers 309
19.2.2. Existing Supplier Performance Management . . 310
19.2.3. Free and Open Source Software Management . . 311
19.3. Challenges and Best Practices 312

PRELIMINARY

19.4. Interfaces to Other Processes 312

IV. Operational Processes 315

20. Project Management 317
20.1. Why This Process Matters 318
20.2. ASQMS Requirements Overview 319
20.3. Challenges and Best Practices 320
20.3.1. Challenge: Unclear Project Boundaries 321
20.3.2. Best Practice: Define Scope Across Three Dimensions 321
20.3.3. Best Practice: Apply SMART Criteria 322
20.3.4. Challenge: Maintenance Phase Scope Confusion 323
20.3.5. Best Practice: Separate Development and Maintenance Activities 324
20.3.6. Challenge: Activity and Task Terminology Confusion 325
20.3.7. Best Practice: Apply Work Breakdown Hierarchy 326
20.3.8. Challenge: Conflicting Agile and ASPICE Terminologies 328
20.3.9. Best Practice: Establish Clear Terminology Mapping 329
20.3.10. Challenge: Multi-Level Lifecycle Complexity . . 330
20.3.11. Best Practice: Define Lifecycle Hierarchy 331
20.3.12. Challenge: Multi-Level Project Management Coordination 334
20.3.13. Best Practice: Define Three Levels of Project Management 335
20.3.14. Challenge: Competency Management Challenges 341
20.3.15. Best Practice: Implement Structured Competency Artifacts 342
20.3.16. Best Practice: Define Standardized Competency Levels 346

PRELIMINARY

20.3.17. Challenge: Critical Path Management Complexity 347
20.3.18. Best Practice: Implement Systematic Critical Path Analysis 349
20.3.19. Best Practice: Critical Path Management in Agile Environments 351
20.3.20. Challenge: Project Artifact Consistency Management 353
20.3.21. Best Practice: Implement Structured Consistency Management 355
20.4. Interfaces to Other Processes 357

21. Requirements Development and Management 359
21.1. Why This Process Matters 360
21.2. ASQMS Requirements Overview 362
21.3. Challenges and Best Practices 364
21.3.1. Challenge: Limited stakeholder integration . . . 364
21.3.2. Challenge: Stakeholder expectations 365
21.3.3. Challenge: Customer defines very detailed software requirements 367
21.3.4. Best practice: Derive and agree on a common understanding 368
21.3.5. Best practice: Accept "solutions" as design proposals 371
21.3.6. Challenge: Different levels of requirements . . . 372
21.3.7. Best practice: Define a common approach 374
21.3.8. Best practice: Lifecycle for requirements 376
21.3.9. Challenge: System requirements are copied to software requirements level 378
21.3.10. Best practice: Define allocation via HSI, not by duplication 379
21.3.11. Best practice: Create a Requirements Engineering Strategy Across Domains 380
21.4. Interfaces to Other Processes 381

PRELIMINARY

22. Architectural Design **383**
22.1. Why This Process Matters 383
22.2. ASQMS Requirements Overview 386
22.3. Challenges and Best Practices 387
22.3.1. Challenge: Uncertainty about how to start designing the architecture 388
22.3.2. Best practice: Create a Context Diagram First . . 388
22.3.3. Challenge: Architectural decisions 389
22.3.4. Best practice: Integrate rationale into architecture changes 393
22.3.5. Challenge: Understanding and representing dynamic behavior 394
22.3.6. Best practice: Represent system-wide dynamic behavior explicitly 394
22.3.7. Best practice: Document absence of dynamic behavior explicitly 395
22.3.8. Challenge: How to deal with reuse 396
22.3.9. Best practice: Clearly define the supplied software components 397
22.3.10. Best practice: Agree on the Responsibility for Testing 398
22.3.11. Challenge: Interfaces Are Described Only Informally 399
22.3.12. Best practice: Create a database for all interfaces 399
22.3.13. Challenge: Estimation of resource consumption 401
22.3.14. Best practice: Start with Measuring the Actual Consumption 402
22.3.15. Best practice: Agree on Runtime Consumption . 403
22.3.16. Best practice: Define Component Requirements . 404
22.3.17. Challenge: Implement First 405
22.3.18. Best practice: Detailed Design as Part of the Source Code 405

PRELIMINARY

22.3.19. Challenge: Traceability from Architecture to Requirements 406
22.3.20. Best practice: Define the Components in the Requirements Management Tool 407
22.4. Interfaces to Other Processes 408

23. Software Detailed Design and Implementation 411
23.1. Why This Process Matters 411
23.2. ASQMS Requirements Overview 414
23.3. Challenges and Best Practices 415
23.3.1. Best practice: Detailed Design as Part of the Source Code 415
23.4. Interfaces to Other Processes 416

24. Unit Testing 419
24.1. Why This Process Matters 420
24.2. ASQMS Requirements Overview 422
24.3. Challenges and Best Practices 423
24.4. Interfaces to Other Processes 423

25. Software Integration and Verification 425
25.1. Why This Process Matters 425
25.2. ASQMS Requirements Overview 428
25.3. Challenges and Best Practices 429
25.4. Interfaces to Other Processes 429

26. Verification Testing 433
26.1. ASQMS Requirements Overview 434
26.2. Interfaces to Other Processes 435

27. Software Supplier Management 437
27.1. Why This Process Matters 437
27.2. ASQMS Requirements Overview 440
27.3. Challenges and Best Practices 442
27.4. Interfaces to Other Processes 442

PRELIMINARY

28. Software Release **445**
28.1. Why This Process Matters 445
28.2. ASQMS Requirements Overview 448
28.3. Challenges and Best Practices 449
28.4. Interfaces to Other Processes 450

29. Software Deployment **453**
29.1. Why This Process Matters 453
29.2. ASQMS Requirements Overview 457
29.3. Challenges and Best Practices 458
29.4. Interfaces to Other Processes 458

30. Software Maintenance **461**
30.1. Why This Process Matters 461
30.2. ASQMS Requirements Overview 465
30.3. Challenges and Best Practices 467
30.4. Interfaces to Other Processes 467

31. User Information Management **469**
31.1. Why This Process Matters 469
31.2. ASQMS Requirements Overview 472
31.3. Challenges and Best Practices 474
31.4. Interfaces to Other Processes 474

32. Software Termination **477**
32.1. Why This Process Matters 477
32.2. ASQMS Requirements Overview 481
32.3. Challenges and Best Practices 482
32.4. Interfaces to Other Processes 482

33. Conclusion **485**
33.1. Personal Consultation 486

PRELIMINARY

V. Appendices **489**

34. Software Classification **491**

35. Implementation Guide **493**

36. ASPICE Mapping **495**

37. Certification **497**

38. Glossary **499**

Index **513**

39. About the Author **519**

PRELIMINARY