
Contents at a Glance

Chapter 1.	Introduction and Definitions of Software Best Practices	1
Chapter 2.	Overview of 50 Software Best Practices	39
Chapter 3.	A Preview of Software Development and Maintenance in 2049	177
Chapter 4.	How Software Personnel Learn New Skills	227
Chapter 5.	Software Team Organization and Specialization	275
Chapter 6.	Project Management and Software Engineering	351
Chapter 7.	Requirements, Business Analysis, Architecture, Enterprise Architecture, and Design	437
Chapter 8.	Programming and Code Development	489
Chapter 9.	Software Quality: The Key to Successful Software Engineering	555
Index		645

Contents

Foreword	xv
Acknowledgments	xvii
Introduction	xix

Chapter 1. Introduction and Definitions of Software Best Practices	1
What Are “Best Practices” and How Can They Be Evaluated?	7
Multiple Paths for Software Development, Deployment, and Maintenance	10
Paths for Software Deployment	12
Paths for Maintenance and Enhancements	14
Quantifying Software Development, Deployment, and Maintenance	16
Critical Topics in Software Engineering	19
Overall Ranking of Methods, Practices, and Sociological Factors	23
Summary and Conclusions	36
Readings and References	36
Chapter 2. Overview of 50 Software Best Practices	39
1. Best Practices for Minimizing Harm from Layoffs and Downsizing	41
2. Best Practices for Motivation and Morale of Technical Staff	45
4. Best Practices for Selection and Hiring of Software Personnel	50
5. Best Practices for Appraisals and Career Planning for Software Personnel	50
6. Best Practices for Early Sizing and Scope Control of Software Applications	51
7. Best Practices for Outsourcing Software Applications	53
8. Best Practices for Using Contractors and Management Consultants	58
9. Best Practices for Selecting Software Methods, Tools, and Practices	59
10. Best Practices for Certifying Methods, Tools, and Practices	64
11. Best Practices for Requirements of Software Applications	70
12. Best Practices for User Involvement in Software Projects	72
13. Best Practices for Executive Management Support of Software Applications	74
14. Best Practices for Software Architecture and Design	75
15. Best Practices for Software Project Planning	77

16. Best Practices for Software Project Cost Estimating	79
17. Best Practices for Software Project Risk Analysis	81
18. Best Practices for Software Project Value Analysis	83
19. Best Practices for Canceling or Turning Around Troubled Projects	84
20. Best Practices for Software Project Organization Structures	87
21. Best Practices for Training Managers of Software Projects	89
22. Best Practices for Training Software Technical Personnel	91
23. Best Practices for Use of Software Specialists	92
24. Best Practices for Certifying Software Engineers, Specialists, and Managers	94
25. Best Practices for Communication During Software Projects	97
26. Best Practices for Software Reusability	99
27. Best Practices for Certification of Reusable Materials	101
28. Best Practices for Programming or Coding	107
29. Best Practices for Software Project Governance	109
30. Best Practices for Software Project Measurements and Metrics	110
31. Best Practices for Software Benchmarks and Baselines	112
32. Best Practices for Software Project Milestone and Cost Tracking	115
33. Best Practices for Software Change Control Before Release	117
34. Best Practices for Configuration Control	119
35. Best Practices for Software Quality Assurance (SQA)	120
36. Best Practices for Inspections and Static Analysis	124
37. Best Practices for Testing and Test Library Control	128
38. Best Practices for Software Security Analysis and Control	132
39. Best Practices for Software Performance Analysis	134
40. Best Practices for International Software Standards	136
41. Best Practices for Protecting Intellectual Property in Software	136
42. Best Practices for Protecting Against Viruses, Spyware, and Hacking	139
43. Best Practices for Software Deployment and Customization	154
44. Best Practices for Training Clients or Users of Software Applications	156
45. Best Practices for Customer Support of Software Applications	157
46. Best Practices for Software Warranties and Recalls	158
47. Best Practices for Software Change Management After Release	160
48. Best Practices for Software Maintenance and Enhancement	161
49. Best Practices for Updates and Releases of Software Applications	164
50. Best Practices for Terminating or Withdrawing Legacy Applications	166
Summary and Conclusions	167
Readings and References	167
Chapter 3. A Preview of Software Development and Maintenance in 2049	177
Introduction	177
Requirements Analysis Circa 2049	179
Design Circa 2049	182
Software Development Circa 2049	184
User Documentation Circa 2049	186
Customer Support in 2049	188
Deployment and Customer Training in 2049	190

Maintenance and Enhancement in 2049	191
Software Outsourcing in 2049	195
Software Package Evaluation and Acquisition in 2049	204
Technology Selection and Technology Transfer in 2049	207
Enterprise Architecture and Portfolio Analysis in 2049	210
A Preview of Software Learning in 2049	213
Due Diligence in 2049	216
Certification and Licensing in 2049	218
Software Litigation in 2049	221
Summary and Conclusions	225
Readings and References	225
Chapter 4. How Software Personnel Learn New Skills	227
Introduction	227
The Evolution of Software Learning Channels	228
What Topics Do Software Engineers Need to Learn Circa 2009?	230
Software Engineering Specialists Circa 2009	233
Varieties of Software Specialization Circa 2009	236
Approximate Ratios of Specialists to General Software Personnel	241
Evaluating Software Learning Channels Used by Software Engineers	243
Software Areas Where Additional Education Is Needed	266
New Directions in Software Learning	267
Summary and Conclusions	268
Curricula of Software Management and Technical Topics	268
Readings and References	273
Chapter 5. Software Team Organization and Specialization	275
Introduction	275
Quantifying Organizational Results	276
The Separate Worlds of Information Technology and Systems Software	277
Colocation vs. Distributed Development	278
The Challenge of Organizing Software Specialists	281
Software Organization Structures from Small to Large	284
One-Person Software Projects	284
Pair programming for software development and maintenance	286
Self-Organizing Agile Teams	289
Team Software Process (TSP) Teams	293
Conventional Departments with Hierarchical Organization Structures	298
Conventional Departments with Matrix Organization Structures	304
Specialist Organizations in Large Companies	308
Software Maintenance Organizations	309
Customer Support Organizations	322
Software Test Organizations	328
Software Quality Assurance (SQA) Organizations	342
Summary and Conclusions	348
Readings and References	349

Chapter 6. Project Management and Software Engineering	351
Introduction	351
Software Sizing	359
Software Progress and Problem Tracking	403
Software Benchmarking	408
Summary and Conclusions	433
Readings and References	434
Chapter 7. Requirements, Business Analysis, Architecture, Enterprise Architecture, and Design	437
Introduction	437
Software Requirements	439
Statistical Analysis of Software Requirements	442
Business Analysis	468
Software Architecture	470
Enterprise Architecture	475
Software Design	479
Summary and Conclusions	484
Readings and References	485
Chapter 8. Programming and Code Development	489
Introduction	489
A Short History of Programming and Language Development	490
Why Do We Have More than 2500 Programming Languages?	492
Exploring the Popularity of Programming Languages	495
How Many Programming Languages Are Really Needed?	499
Creating a National Programming Language Translation Center	501
Why Do Most Applications Use Between 2 and 15 Programming Languages	504
How Many Programmers Use Various Programming Languages?	506
What Kinds of Bugs or Defects Occur in Source Code?	509
Logistics of Software Code Defects	512
Preventing and Removing Defects from Application Source Code	518
Forms of Programming Defect Prevention	520
Forms of Programming Defect Removal	529
Economic Problems of the “Lines of Code” Metric	537
Summary and Conclusions	552
Readings and References	552
Chapter 9. Software Quality: The Key to Successful Software Engineering	555
Introduction	555
Defining Software Quality	558
Measuring Software Quality	585
Defect Prevention	600

Software Defect Removal	613
Software Quality Specialists	619
Summary and Conclusions on Software Specialization	632
The Economic Value of Software Quality	633
Summary and Conclusions	642
Readings and References	643
Index	645