

Contents

Preface xiii

Part 1

1 What Makes a Good Software Manager?	3
1.1 People Perspective	3
1.2 Business Perspective	6
1.3 Process Perspective	7
1.3.1 Successful Process Techniques	8
1.3.2 Best Practices	9
1.3.3 Management “Secrets”	10
1.4 Key Thoughts in This Chapter	11
References	11
2 Four Basics That Work	13
2.1 People, Process, and Product	13
2.1.1 People	14
2.1.2 Process	14
2.1.3 Product	15
2.1.4 Balancing the 3Ps	17
2.2 Visibility	18
2.2.1 Basic Visibility Techniques	20
2.2.2 Using the Techniques	24
2.3 Configuration Management	25
2.3.1 The CM Plan	25
2.3.2 Basic Baselines	26
2.3.3 Activities	28
2.3.4 CM People	31
2.3.5 CM Sketch	33

2.4	Standards	35
2.5	Key Thoughts in This Chapter	36
	References	36
3	What Doesn't Work and Why	39
3.1	When the 3Ps Are Out of Balance	39
3.2	When There's Not Enough Visibility	40
3.3	When Configuration Management is Missing or Abused	41
3.4	When Standards are Dismissed	44
3.5	Key Thoughts in This Chapter	45
	Reference	46
4	Managing a Project Day by Day	47
4.1	Balancing the 3Ps to Create a Good Environment	48
4.1.1	Emotional Safety	48
4.1.2	Emphasis on Team Empowerment	49
4.1.3	High Degree of Personal Interaction	49
4.1.4	Good Balance of Work and Rest	49
4.1.5	Structure that Promotes Success	50
4.2	Visibility: Project Control in a Simple Equation	52
4.2.1	Collecting Status	52
4.2.2	Collection Guidelines	53
4.2.3	Making Status Visible and Undistorted	62
4.2.4	Analyzing the Situation	81
4.2.5	Taking Action	82
4.2.6	Making and Communicating Decisions	83
4.2.7	Making a Decision Visible	84
4.2.8	Keeping the Environment Good	86
4.2.9	Managing an External Supplier	88
4.3	CM: Managing Baselines with Milestones	89
4.4	Looking to Standards for Help	90
4.5	Key Thoughts in This Chapter	91
	References	92

Part 2

5	Requirements	97
5.1	Balancing the 3Ps: Requirements Analysis, Documentation, and Management	98
5.1.1	Selecting the Requirements Engineer	99
5.1.2	Interviewing Customers	100
5.1.3	Conducting Group Meetings	103
5.1.4	Diffusing Tense Situations	104
5.1.5	Evolving Requirements	106
5.1.6	Requirements Versus Design	107
5.1.7	Requirements Management	108

5.2	Visibility: Making Requirements Known	110
5.2.1	An Overview of Techniques	111
5.2.2	Facilitated Meetings—JAD	113
5.2.3	Design by Walking Around	115
5.2.4	System Storyboarding Technique	117
5.2.5	Concept of Operations	118
5.2.6	Mind Maps	119
5.2.7	Gilb Charts	120
5.2.8	Diagram of Effects	121
5.2.9	Rapid Prototyping	123
5.2.10	Software Diagrams	123
5.2.11	The Software Requirements Specification	133
5.2.12	Database Support	143
5.3	Using CM	143
5.4	Using Standards	145
5.5	Key Thoughts in this Chapter	148
	References	151
6	Planning	155
6.1	Elements of a Good Plan	156
6.2	Balancing the 3Ps: Selecting the Process	161
	Are Software Problems Unique?	162
6.2.1	Prototyping	164
6.2.2	Rapid Application Development	165
6.2.3	Microsoft Process	166
6.2.4	Spiral Process	168
6.2.5	Agile Development Methods	172
6.2.6	Process Improvement Mechanisms	179
6.3	Making the Project Visible: Planning Techniques	196
6.3.1	Project Context	196
6.3.2	Creating a Task Network	198
6.3.3	The Critical Path	203
6.3.4	The Critical Chain	204
6.3.5	Cards on the Wall Planning	207
6.4	Making the Project Visible: Estimating Techniques	208
6.4.1	Rayleigh Model	209
6.4.2	PSP's Probe	215
6.4.3	A Technique for Simple Estimation	220
6.4.4	Judging an Estimate	221
6.4.5	Tailoring Techniques to the Process Model	222
6.5	Configuration Management	226
6.6	Standards	227
6.6.1	All-in-One Military and Commercial Standards	227
6.6.2	Documenting the Plan	231
6.7	Key Thoughts in this Chapter	233
	References	234

7 Risk Management	237
7.1 A Task Overview	239
7.2 Balancing The 3Ps: Uncertainty and Choice	239
7.2.1 Risk Identification	240
7.2.2 Risk Planning	245
7.2.3 Risk Control	246
7.2.4 Risk Monitoring	247
7.2.5 Risk Directing and Staffing	248
7.3 Making Risk Visible	248
7.3.1 Risk Estimating	248
7.3.2 Risk Evaluation	249
7.3.3 Risk Analysis Products	250
7.4 Other Ways to Manage Risk	252
7.4.1 Testing	253
7.4.2 Scenario Planning	253
7.4.3 Whiteboard Risk Management	254
7.5 Configuration Management	254
7.6 Using Standards	255
7.7 Key Thoughts in this Chapter	255
References	256

Part 3

8 Design	259
8.1 The Challenge of the 3Ps	260
8.1.1 Managing Creativity	260
8.1.2 Reducing Design Frustration	262
8.1.3 Evaluating and Selecting from Design Alternatives	268
8.2 Visibility—Expressing the Design	273
8.2.1 Words	273
8.2.2 Pictures	274
8.3 Design in the Code	275
8.4 Design and Process	277
8.5 Designing with COTS	278
8.6 Configuration Management	280
8.6.1 Configuration Control Boards	281
8.6.2 Design Documents	281
8.6.3 The CM Plan	282
8.6.4 Tracing Requirements	283
8.7 Standards: Writing the SDD	284
8.7.1 Contents	284
8.7.2 Organization	286
8.8 Key Thoughts in this Chapter	287
References	289

9	Integration and Testing	291
9.1	Some I&T Myths	292
9.2	Managing the 3Ps: People	294
9.3	Managing the 3Ps: Process	296
9.3.1	Common Testing Problems	296
9.3.2	IDEA	297
9.3.3	Verification and Validation	299
9.4	Visibility: Testing Techniques and Details	299
9.4.1	Elements of Effective Testing	300
9.4.2	Black Box Testing	301
9.4.3	White Box Testing	304
9.4.4	Combining White Box and Black Box Testing	304
9.4.5	Integration Testing	305
9.4.6	Acceptance Testing	308
9.4.7	Regression Testing	309
9.4.8	Cleanroom Testing	309
9.4.9	Test-Driven Development	313
9.5	Configuration Management	314
9.5.1	How Testing Relates to Other Activities	314
9.5.2	Controlling Test Artifacts	316
9.5.3	Using the Requirements Verification Traceability Matrix	317
9.6	Standards: Documenting the Test Plan	318
9.7	Key Thoughts in this Chapter	320
	References	321
10	Software Maintenance	323
10.1	What is Maintenance	324
10.1.1	Maintenance or Development?	324
10.1.2	Maintenance Activities	324
10.1.3	Why Use Configuration Management?	325
10.1.4	Why Is It So Expensive and Difficult?	325
10.1.5	Why We Should Not Neglect Maintenance	326
10.2	Balancing the 3Ps: Managing the Maintainers	327
10.3	Balancing the 3Ps: Managing the Process	328
10.4	Balancing the 3Ps: Making the Most of the Product	329
10.5	Visibility: Understanding the Maintenance Stages	331
10.5.1	Identification and Classification	331
10.5.2	Analysis	333
10.5.3	Design	335
10.5.4	Implementation	335
10.5.5	System Test	336
10.5.6	Acceptance Test	336
10.5.7	Delivery	336

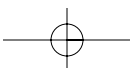
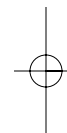
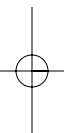
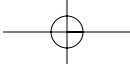
10.6 Configuration Management	337
10.6.1 Keeping Baselines Straight	338
10.6.2 Managing Releases	338
10.6.3 Pacing the Process	339
10.7 Using Standards	340
10.8 Key Thoughts in this Chapter	340
References	341

Part 4

11 Cookbook	345
11.1 Essentials	347
11.1.1 Use Journals and Decision Records	347
11.1.2 Perform All CM Activities	348
11.1.3 Manage Day by Day	348
11.1.4 Use Standards	348
11.1.5 Conduct a Retrospective	348
11.2 Opt: A Waterfall Project	349
11.2.1 Context	349
11.2.2 Project Details	349
11.3 System Upgrade: An Evolutionary Project	362
11.3.1 Context	362
11.3.2 Project Details	362
11.4 CTRAN: A Spiral Project	376
11.4.1 Context	376
11.4.2 Risks	377
11.4.3 Project Details	378
11.4.4 Quadrants of the Spiral	383
11.4.5 Cycles of the Spiral	385
11.5 Other Software Projects	389
11.5.1 My Little Spreadsheet	389
11.5.2 Let's Make a Web Page	391
11.5.3 An Agile Methods Project	393
11.6 Key Thoughts in this Chapter	395
References	397

Appendix A Documents for the OPT Project	397
A.1 OPT Executive Sponsor Memorandum	397
A.2 OPT Project Context Document	398
A.3 OPT Configuration Management Plan	400
A.4 OPT Concept of Operations	407
A.5 OPT Software Requirements Specification	410
A.6 OPT Software Project Management Plan	419
A.7 OPT Software Design Description	431

Appendix B Configuration Management	439
B.1 Will The Real CM Please Stand Up?	439
B.2 The Main Ingredients	440
B.3 Baselines	441
B.3.1 Basics	441
B.3.2 Applying Baselines in a Waterfall Project	442
B.3.3 Applying Baselines in a Nonwaterfall Project	444
B.3.4 Documenting Baselines	445
B.3.5 Baseline Contents	445
B.3.6 Interface Control Documents	446
B.4 CM Activities	448
B.4.1 Identification	448
B.4.2 Control	448
B.4.3 Auditing	450
B.4.4 Status Accounting	451
B.5 CM People	451
B.5.1 Configuration Control Boards	451
B.5.2 CM Staff	452
B.5.3 Project Manager	452
B.6 CM Plan	453
B.7 A CM Sketch	454
B.8 Summary	456
References	457
Appendix C Structured Analysis and Design	459
C.1 Structured Analysis	460
C.1.1 Environmental Model	460
C.1.2 Preliminary Behavioral Model	461
C.1.3 Final Behavioral Model	463
C.1.4 Finished Essential Model	465
C.2 Structured Design	466
C.2.1 User Implementation Model	466
C.2.2 Systems Implementation Model	467
C.2.3 Program Implementation Model	469
References	470
Appendix D Annotated Bibliography	471
D.1 Process	471
D.2 Visibility	472
D.3 People	473
D.4 Journals	474
Index	475
About the Author	485



Preface

I still like to write software. The material in this book and the relative success of the first edition have caused my own senior managers to pull me away from writing software and push me more into managing software and systems projects. Although I miss writing software, I do like to manage projects.

The first edition of this book sold well, at least that is what the fine people at the IEEE Computer Society have told me. It has been most gratifying to hear from people who have read the first edition. Some people have told me that they generally “like the book,” whereas others have pointed to specific pages and related how something that has worked for me also worked for them. Recently, a project manager told me how her team had never liked the concept of earned value, but something clicked when they saw the figures in the first edition. Since then, they bother her until she updates the earned value charts and posts them publicly so everyone can see how they are progressing.

Although gratified at hearing this, I also winced a bit because I felt that I had erred in some of my descriptions of earned value. There were several other things in the first edition that I wished I had described differently. That is the main reason for the second edition of *The Software Project Manager's Handbook*.

Another reason for the second edition is that I have learned a few things in the past five years that I have seen work at work. I wanted to update the book so that I could remember and use them better.

Several events have occurred in my life since the first edition that have had a significant impact on the second edition. First, I was able to meet Jerry Weinberg and attend several of his seminars. The ideas I had read in his books came to life. I am most grateful to Jerry and to some of the fine people I met at his seminars. Notable among these are Johanna Rothman, Dale Emery, Steve Smith, Elizabeth Hendrickson, Don Gray, Esther Derby, Naomi Karten, and Jerry's wife Dani. I had the great pleasure of attending one of Jerry's writers workshops. That wonderful week led me to write another book on project management with Roy O'Bryan—*It Sounded Good When We Started*—also published by the IEEE Computer Society and Wiley.

A second event that occurred since the first edition of this book was that I