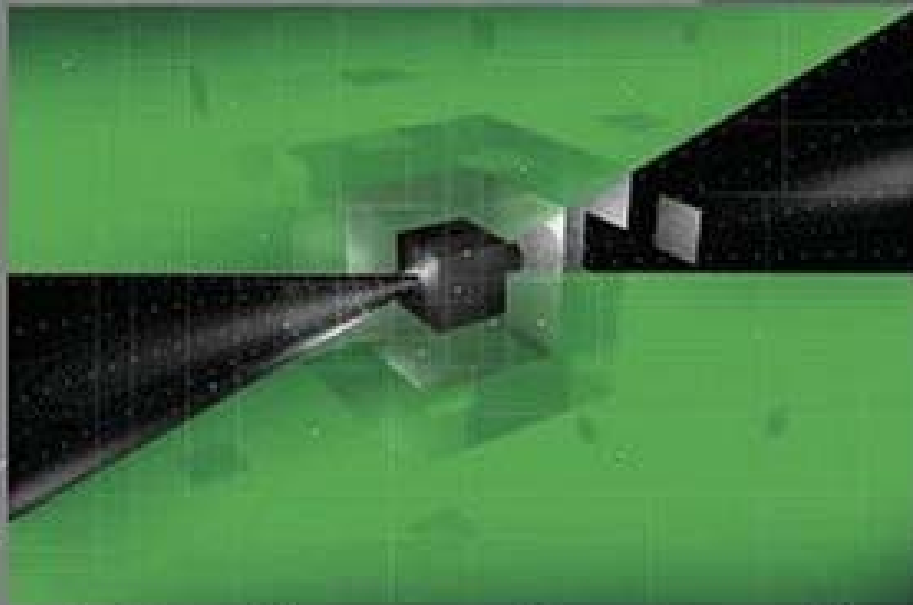


Microsoft

BEST PRACTICES

SOFTWARE ESTIMATION



Demystifying the Black Art

Steve McConnell

Two-time winner of *Software Development* magazine's Jolt Award

This material is excerpted from *Software Estimation: Demystifying the Black Art* by Steve McConnell (Redmond, Wa.: Microsoft Press).

© 2006 All Rights Reserved.

Table of Contents

<i>Welcome</i>	xv
<i>Acknowledgments</i>	xxi
<i>List of Equations</i>	xxiii
<i>List of Figures</i>	xxv

Part I Critical Estimation Concepts

1	What Is an “Estimate”?	3
	1.1 Estimates, Targets, and Commitments	3
	1.2 Relationship Between Estimates and Plans	4
	1.3 Communicating about Estimates, Targets, and Commitments	5
	1.4 Estimates as Probability Statements	6
	1.5 Common Definitions of a “Good” Estimate	9
	1.6 Estimates and Project Control	11
	1.7 Estimation’s Real Purpose	13
	1.8 A Working Definition of a “Good Estimate”	14
	Additional Resources	14
2	How Good an Estimator Are You?	15
	2.1 A Simple Estimation Quiz	15
	2.2 Discussion of Quiz Results	16
	How Confident Is “90% Confident”?	16
	How Wide Should You Make Your Ranges?	18
	Where Does Pressure to Use Narrow Ranges Come From?	18
	How Representative Is This Quiz of Real Software Estimates?	19
3	Value of Accurate Estimates	21
	3.1 Is It Better to Overestimate or Underestimate?	21
	Arguments Against Overestimation	21
	Arguments Against Underestimation	22
	Weighing the Arguments	23

What do you think of this book?
We want to hear from you!

Microsoft is interested in hearing your feedback about this publication so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit: www.microsoft.com/learning/booksurvey/

3.2	Details on the Software Industry's Estimation Track Record	24
	How Late Are the Late Projects?	26
	One Company's Experience	26
	The Software Industry's Systemic Problem	27
3.3	Benefits of Accurate Estimates	27
3.4	Value of Predictability Compared with Other Desirable Project Attributes	29
3.5	Problems with Common Estimation Techniques	30
	Additional Resources	31
4	Where Does Estimation Error Come From?	33
4.1	Sources of Estimation Uncertainty	34
4.2	The Cone of Uncertainty	35
	Can You Beat the Cone?	37
	The Cone Doesn't Narrow Itself.	38
	Accounting for the Cone of Uncertainty in Software Estimates.	39
	Relationship Between the Cone of Uncertainty and Commitment	40
	The Cone of Uncertainty and Iterative Development	40
4.3	Chaotic Development Processes	41
4.4	Unstable Requirements.	42
	Estimating Requirements Growth	43
4.5	Omitted Activities	44
4.6	Unfounded Optimism.	46
4.7	Subjectivity and Bias	47
4.8	Off-the-Cuff Estimates	49
4.9	Unwarranted Precision	51
4.10	Other Sources of Error	52
	Additional Resources.	53
5	Estimate Influences	55
5.1	Project Size	55
	Why Is This Book Discussing Size in Lines of Code?.	56
	Diseconomies of Scale	56
	When You Can Safely Ignore Diseconomies of Scale.	60
	Importance of Diseconomy of Scale in Software Estimation	61
5.2	Kind of Software Being Developed.	61
5.3	Personnel Factors.	63
5.4	Programming Language	64
5.5	Other Project Influences.	65
5.6	Diseconomies of Scale Revisited	70
	Additional Resources.	72

Part II Fundamental Estimation Techniques

6	Introduction to Estimation Techniques	77
	6.1 Considerations in Choosing Estimation Techniques	77
	What's Being Estimated	77
	Project Size	78
	Software Development Style	78
	Development Stage	80
	Accuracy Possible	80
	6.2 Technique Applicability Tables	81
7	Count, Compute, Judge	83
	7.1 Count First	84
	7.2 What to Count	85
	7.3 Use Computation to Convert Counts to Estimates	86
	7.4 Use Judgment Only as a Last Resort	88
	Additional Resources	89
8	Calibration and Historical Data	91
	8.1 Improved Accuracy and Other Benefits of Historical Data	91
	Accounts for Organizational Influences	92
	Avoids Subjectivity and Unfounded Optimism	93
	Reduces Estimation Politics	93
	8.2 Data to Collect	95
	Issues Related to Size Measures	95
	Issues Related to Effort Measures	96
	Issues Related to Calendar Time Measures	97
	Issues Related to Defect Measures	97
	Other Data Collection Issues	98
	8.3 How to Calibrate	98
	8.4 Using Project Data to Refine Your Estimates	99
	8.5 Calibration with Industry Average Data	100
	8.6 Summary	102
	Additional Resources	102

viii Table of Contents

9	Individual Expert Judgment	105
	9.1 Structured Expert Judgment	106
	Who Creates the Estimates?	106
	Granularity	106
	Use of Ranges	107
	Formulas	108
	Checklists	110
	9.2 Compare Estimates to Actuals	110
	Additional Resources	112
10	Decomposition and Recomposition	113
	10.1 Calculating an Accurate Overall Expected Case	113
	The Law of Large Numbers	115
	How Small Should the Estimated Pieces Be?	116
	10.2 Decomposition via an Activity-Based Work Breakdown Structure	117
	10.3 Hazards of Adding Up Best Case and Worst Case Estimates	118
	Warning: Math Ahead!	119
	What Went Wrong?	119
	10.4 Creating Meaningful Overall Best Case and Worst Case Estimates	120
	Computing Aggregate Best and Worst Cases for Small Numbers of Tasks (Simple Standard Deviation Formula)	121
	Computing Aggregate Best and Worst Cases for Large Numbers of Tasks (Complex Standard Deviation Formula)	122
	Creating the Aggregate Best and Worst Case Estimates	124
	Cautions About Percentage Confident Estimates	126
	Additional Resources	126
11	Estimation by Analogy	127
	11.1 Basic Approach to Estimating by Analogy	127
	Step 1: Get Detailed Size, Effort, and Cost Results for a Similar Previous Project	128
	Step 2: Compare the Size of the New Project to a Similar Past Project	129
	Step 3: Build Up the Estimate for the New Project's Size as a Percentage of the Old Project's Size	130
	Step 4: Create an Effort Estimate from the Size of the New Project Compared to the Previous Project	131
	Step 5: Check for Consistent Assumptions Across the Old and New Projects	131

11.2	Comments on Uncertainty in the Triad Estimate	132
	Estimation Uncertainty, Plans, and Commitments	133
12	Proxy-Based Estimates	135
12.1	Fuzzy Logic	136
	How to Get the Average Size Numbers	136
	How to Classify New Functionality	137
	How Not to Use Fuzzy Logic	137
	Extensions of Fuzzy Logic	138
12.2	Standard Components	138
	Using Standard Components with Percentiles	140
	Limitations of Standard Components	141
12.3	Story Points	142
	Cautions About Ratings Scales	143
12.4	T-Shirt Sizing	145
12.5	Other Uses of Proxy-Based Techniques	147
12.6	Additional Resources	147
13	Expert Judgment in Groups	149
13.1	Group Reviews	149
13.2	Wideband Delphi	150
	Effectiveness of Wideband Delphi	152
	“The Truth Is Out There”	154
	When to Use Wideband Delphi	154
	Additional Resources	155
14	Software Estimation Tools	157
14.1	Things You Can Do with Tools That You Can’t Do Manually	157
14.2	Data You’ll Need to Calibrate the Tools	162
14.3	One Thing You Shouldn’t Do with a Tool Any More than You Should Do Otherwise	162
14.4	Summary of Available Tools	163
	Additional Resources	164
15	Use of Multiple Approaches	165
	Additional Resources	169

x Table of Contents

16	Flow of Software Estimates on a Well-Estimated Project	171
	16.1 Flow of an Individual Estimate on a Poorly Estimated Project	171
	16.2 Flow of an Individual Estimate on a Well-Estimated Project	172
	16.3 Chronological Estimation Flow for an Entire Project	173
	Estimation Flow for Large Projects	174
	Estimation Flow for Small Projects	175
	16.4 Estimate Refinement	175
	16.5 How to Present Reestimation to Other Project Stakeholders	176
	When to Present the Reestimates	177
	What If Your Management Won't Let You Reestimate?	178
	16.6 A View of a Well-Estimated Project	179
17	Standardized Estimation Procedures	181
	17.1 Usual Elements of a Standardized Procedure	181
	17.2 Fitting Estimation into a Stage-Gate Process	182
	17.3 An Example of a Standardized Estimation Procedure for Sequential Projects	185
	17.4 An Example of a Standardized Estimation Procedure for Iterative Projects . .	188
	17.5 An Example of a Standardized Estimation Procedure from an Advanced Organization	190
	17.6 Improving Your Standardized Procedure	192
	Additional Resources	193

Part III Specific Estimation Challenges

18	Special Issues in Estimating Size	197
	18.1 Challenges with Estimating Size	197
	Role of Lines of Code in Size Estimation	198
	18.2 Function-Point Estimation	200
	Converting from Function Points to Lines of Code	202
	18.3 Simplified Function-Point Techniques	203
	The Dutch Method	203
	GUI Elements	204
	18.4 Summary of Techniques for Estimating Size	205
	Additional Resources	206

19	Special Issues in Estimating Effort.	207
	19.1 Influences on Effort	207
	19.2 Computing Effort from Size	209
	Computing Effort Estimates by Using Informal Comparison to Past Projects	209
	What Kinds of Effort Are Included in This Estimate?	210
	19.3 Computing Effort Estimates by Using the Science of Estimation	210
	19.4 Industry-Average Effort Graphs	210
	19.5 ISBSG Method	216
	19.6 Comparing Effort Estimates	218
	Additional Resources	219
20	Special Issues in Estimating Schedule.	221
	20.1 The Basic Schedule Equation	221
	20.2 Computing Schedule by Using Informal Comparisons to Past Projects	223
	20.3 Jones's First-Order Estimation Practice	224
	20.4 Computing a Schedule Estimate by Using the Science of Estimation	225
	20.5 Schedule Compression and the Shortest Possible Schedule	226
	20.6 Tradeoffs Between Schedule and Effort	228
	Schedule Compression and Team Size	229
	20.7 Schedule Estimation and Staffing Constraints	230
	20.8 Comparison of Results from Different Methods	231
	Additional Resources	232
21	Estimating Planning Parameters	233
	21.1 Estimating Activity Breakdown on a Project	233
	Estimating Allocation of Effort to Different Technical Activities	233
	Estimating Requirements Effort	234
	Estimating Management Effort	235
	Estimating Total Activity	235
	Adjustments Due to Project Type	236
	Example of Allocating Effort to Activities	237
	Developer-to-Tester Ratios	237
	21.2 Estimating Schedule for Different Activities	238
	21.3 Converting Estimated Effort (Ideal Effort) to Planned Effort	239
	21.4 Cost Estimates	241
	Overtime	241

xii Table of Contents

	Is the Project Cost Based on Direct Cost, Fully Burdened Cost, or Some Other Variation?	241
	Other Direct Costs	241
21.5	Estimating Defect Production and Removal	241
	Estimating Defect Removal	242
	An Example of Estimating Defect-Removal Efficiency	243
21.6	Estimating Risk and Contingency Buffers	245
21.7	Other Rules of Thumb	247
21.8	Additional Resources	247
22	Estimate Presentation Styles	249
22.1	Communicating Estimate Assumptions	249
22.2	Expressing Uncertainty	251
	Plus-or-Minus Qualifiers	251
	Risk Quantification	251
	Confidence Factors	252
	Case-Based Estimates	254
	Coarse Dates and Time Periods	255
22.3	Using Ranges (of Any Kind)	256
	Usefulness of Estimates Presented as Ranges	256
	Ranges and Commitments	257
	Additional Resources	257
23	Politics, Negotiation, and Problem Solving	259
23.1	Attributes of Executives	259
23.2	Political Influences on Estimates	260
	External Constraints	260
	Budgeting and Dates	261
	Negotiating an Estimate vs. Negotiating a Commitment	261
	What to Do if Your Estimate Doesn't Get Accepted	262
	Responsibility of Technical Staff to Educate Nontechnical Stakeholders	262
23.3	Problem Solving and Principled Negotiation	263
	A Problem-Solving Approach to Negotiation	264
	Separate the People from the Problem	264
	Focus on Interests, Not Positions	265
	Invent Options for Mutual Gain	266
	Insist on Using Objective Criteria	268
	Additional Resources	270

Table of Contents **xiii**

A	Estimation Sanity Check	271
B	Answers to Chapter 2 Quiz, “How Good an Estimator Are You?”	273
C	Software Estimation Tips	275
	Bibliography	287
	Index.	295

What do you think of this book?
We want to hear from you!

Microsoft is interested in hearing your feedback about this publication so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit: www.microsoft.com/learning/booksurvey/

