

Contents

Acknowledgments	xv
1. Introduction	1
1.1 Objective /	1
1.2 Approach /	2
1.3 Motivation /	3
1.4 Summary /	5
References /	6
2. What to Measure	7
2.1 Method 1: The Goal Question Metrics Approach /	9
2.2 Method 2: Decision Maker Model /	10
2.3 Method 3: Standards Driven Metrics /	10
2.4 Extension to GQM: Metrics Mechanism /	11
2.5 What to Measure Is a Function of Time /	12
2.6 Summary /	12
Problems /	13
Project /	13
References /	13

3. Measurement Fundamentals **15**

- 3.1 Initial Measurement Exercise / 15
- 3.2 The Challenge of Measurement / 16
- 3.3 Measurement Models / 16
 - 3.3.1 Text Models / 16
 - 3.3.2 Diagrammatic Models / 18
 - 3.3.3 Algorithmic Models / 18
 - 3.3.4 Model Examples: Response Time / 18
 - 3.3.5 The Pantometric Paradigm: How to Measure Anything / 19
- 3.4 Meta-Model for Metrics / 20
- 3.5 The Power of Measurement / 21
- 3.6 Measurement Theory / 22
 - 3.6.1 Introduction to Measurement Theory / 22
 - 3.6.2 Measurement Scales / 23
 - 3.6.3 Measures of Central Tendency and Variability / 24
 - 3.6.3.1 Measures of Central Tendency / 25
 - 3.6.3.2 Measures of Variability / 25
 - 3.6.4 Validity and Reliability of Measurement / 27
 - 3.6.5 Measurement Error / 28
- 3.7 Accuracy Versus Precision and the Limits of Software Metrics / 30
- 3.8 Summary / 31
- Problems / 31
- Projects / 33
- References / 33

4. Measuring Size **34**

- 4.1 Physical Measurements of Software / 34
 - 4.1.1 Measuring Lines of Code / 35
 - 4.1.2 Language Productivity Factor / 35
 - 4.1.3 Counting Reused and Refactored Code / 37
 - 4.1.4 Counting Nonprocedural Code Length / 39
 - 4.1.5 Measuring the Length of Specifications and Design / 39
- 4.2 Measuring Functionality / 40
 - 4.2.1 Function Points / 41
 - 4.2.1.1 Counting Function Points / 41
 - 4.2.1.2 Function Point Example / 45

4.2.1.3	Converting Function Points to Physical Size /	47
4.2.1.4	Converting Function Points to Effort /	47
4.2.1.5	Other Function Point Engineering Rules /	48
4.2.1.6	Function Point Pros and Cons /	49
4.2.2	Feature Points /	50
4.3	Summary /	51
	Problems /	51
	Project /	52
	References /	53
5.	Measuring Complexity	54
5.1	Structural Complexity /	55
5.1.1	Size as a Complexity Measure /	55
5.1.1.1	System Size and Complexity /	55
5.1.1.2	Module Size and Complexity /	56
5.1.2	Cyclomatic Complexity /	58
5.1.3	Halstead's Metrics /	63
5.1.4	Information Flow Metrics /	65
5.1.5	System Complexity /	67
5.1.5.1	Maintainability Index /	67
5.1.5.2	The Agresti-Card System Complexity Metric /	69
5.1.6	Object-Oriented Design Metrics /	71
5.1.7	Structural Complexity Summary /	73
5.2	Conceptual Complexity /	73
5.3	Computational Complexity /	74
5.4	Summary /	75
	Problems /	75
	Projects /	77
	References /	78
6.	Estimating Effort	79
6.1	Effort Estimation: Where Are We? /	80
6.2	Software Estimation Methodologies and Models /	81
6.2.1	Expert Estimation /	82
6.2.1.1	Work and Activity Decomposition /	82
6.2.1.2	System Decomposition /	83
6.2.1.3	The Delphi Methods /	84

6.2.2	Using Benchmark Size Data /	85
6.2.2.1	Lines of Code Benchmark Data /	85
6.2.2.2	Function Point Benchmark Data /	87
6.2.3	Estimation by Analogy /	88
6.2.3.1	Traditional Analogy Approach /	89
6.2.3.2	Analogy Summary /	91
6.2.4	Proxy Point Estimation Methods /	91
6.2.4.1	Meta-Model for Effort Estimation /	91
6.2.4.2	Function Points /	92
6.2.4.3	Object Points /	94
6.2.4.4	Use Case Sizing Methodologies /	95
6.2.5	Custom Models /	101
6.2.6	Algorithmic Models /	103
6.2.6.1	Manual Models /	103
6.2.6.2	Estimating Project Duration /	105
6.2.6.3	Tool-Based Models /	105
6.3	Combining Estimates /	107
6.4	Estimating Issues /	108
6.4.1	Targets Versus Estimates /	108
6.4.2	The Limitations of Estimation: Why? /	109
6.4.3	Estimate Uncertainties /	109
6.5	Estimating Early and Often /	112
6.6	Summary /	113
	Problems /	114
	Projects /	116
	References /	116

7. In Praise of Defects: Defects and Defect Metrics

118

7.1	Why Study and Measure Defects? /	118
7.2	Faults Versus Failures /	119
7.3	Defect Dynamics and Behaviors /	120
7.3.1	Defect Arrival Rates /	120
7.3.2	Defects Versus Effort /	120
7.3.3	Defects Versus Staffing /	120
7.3.4	Defect Arrival Rates Versus Code Production Rate /	121
7.3.5	Defect Density Versus Module Complexity /	122
7.3.6	Defect Density Versus System Size /	122

7.4	Defect Projection Techniques and Models /	123
7.4.1	Dynamic Defect Models /	123
7.4.1.1	Rayleigh Models /	124
7.4.1.2	Exponential and S-Curves Arrival Distribution Models /	127
7.4.1.3	Empirical Data and Recommendations for Dynamic Models /	128
7.4.2	Static Defect Models /	129
7.4.2.1	Defect Insertion and Removal Model /	129
7.4.2.2	Defect Removal Efficiency: A Key Metric /	130
7.4.2.3	Static Defect Model Tools /	132
7.5	Additional Defect Benchmark Data /	133
7.5.1	Defect Data by Application Domain /	133
7.5.2	Cumulative Defect Removal Efficiency (DRE) Benchmark /	134
7.5.3	SEI Levels and Defect Relationships /	134
7.5.4	Latent Defects /	135
7.5.5	A Few Recommendations /	135
7.6	Cost Effectiveness of Defect Removal by Phase /	136
7.7	Defining and Using Simple Defect Metrics: An Example /	136
7.8	Some Paradoxical Patterns for Customer Reported Defects /	139
7.9	Answers to the Initial Questions /	140
7.10	Summary /	140
	Problems /	141
	Projects /	142
	References /	142

8. Software Reliability Measurement and Prediction

144

8.1	Why Study and Measure Software Reliability? /	144
8.2	What Is Reliability? /	144
8.3	Faults and Failures /	145
8.4	Failure Severity Classes /	145
8.5	Failure Intensity /	146
8.6	The Cost of Reliability /	147
8.7	Software Reliability Theory /	148
8.7.1	Uniform and Random Distributions /	148

8.7.2	The Probability of Failure During a Time Interval /	150
8.7.3	$F(t)$: The Probability of Failure by Time T /	151
8.7.4	$R(t)$: The Reliability Function /	151
8.7.5	Reliability Theory Summarized /	152
8.8	Reliability Models /	152
8.8.1	Types of Models /	152
8.8.2	Predicting Number of Defects Remaining /	154
8.9	Failure Arrival Rates /	155
8.9.1	Predicting Failure Arrival Rates Using Historical Data /	155
8.9.2	Engineering Rules for MTTF /	156
8.9.3	Musa's Algorithm /	157
8.9.4	Operational Profile Testing /	158
8.9.5	Predicting Reliability Summary /	161
8.10	But When Do I Ship? /	161
8.11	System Configurations: Probability and Reliability /	161
8.12	Answers to Initial Question /	163
8.13	Summary /	164
	Problems /	164
	Project /	165
	References /	166

9. Response Time and Availability

167

9.1	Response Time Measurements /	168
9.2	Availability /	170
9.2.1	Availability Factors /	172
9.2.2	Outage Scope /	173
9.2.3	Complexities in Measuring Availability /	173
9.2.4	Software Rejuvenation /	174
9.2.4.1	Software Aging /	175
9.2.4.2	Classification of Faults /	175
9.2.4.3	Software Rejuvenation Techniques /	175
9.2.4.4	Impact of Rejuvenation on Availability /	176
9.3	Summary /	177
	Problems /	178
	Project /	179
	References /	180

10. Measuring Progress	181
10.1 Project Milestones /	182
10.2 Code Integration /	185
10.3 Testing Progress /	187
10.4 Defects Discovery and Closure /	188
10.4.1 Defect Discovery /	189
10.4.2 Defect Closure /	190
10.5 Process Effectiveness /	192
10.6 Summary /	194
Problems /	195
Project /	196
References /	196
11. Outsourcing	197
11.1 The “O” Word /	197
11.2 Defining Outsourcing /	198
11.3 Risk Management and Outsourcing /	201
11.4 Metrics and the Contract /	203
11.5 Summary /	206
Problems /	206
Projects /	207
References /	207
12. Financial Measures for the Software Engineer	208
12.1 It’s All About the Green /	208
12.2 Financial Concepts /	209
12.3 Building the Business Case /	209
12.3.1 Understanding Costs /	210
12.3.1.1 Salaries /	210
12.3.1.2 Overhead costs /	210
12.3.1.3 Risk Costs /	211
12.3.1.4 Capital Versus Expense /	213
12.3.2 Understanding Benefits /	216
12.3.3 Business Case Metrics /	218
12.3.3.1 Return on Investment /	218
12.3.3.2 Payback Period /	219
12.3.3.3 Cost/Benefit Ratio /	220
12.3.3.4 Profit and Loss Statement /	221

12.3.3.5	Cash Flow / 222	
12.3.3.6	Expected Value / 223	
12.4	Living the Business Case / 224	
12.5	Summary / 224	
	Problems / 227	
	Projects / 228	
	References / 230	
13.	Benchmarking	231
13.1	What Is Benchmarking? / 231	
13.2	Why Benchmark? / 232	
13.3	What to Benchmark / 232	
13.4	Identifying and Obtaining a Benchmark / 233	
13.5	Collecting Actual Data / 233	
13.6	Taking Action / 234	
13.7	Current Benchmarks / 234	
13.8	Summary / 236	
	Problems / 236	
	Projects / 236	
	References / 237	
14.	Presenting Metrics Effectively to Management	238
14.1	Decide on the Metrics / 239	
14.2	Draw the Picture / 240	
14.3	Create a Dashboard / 243	
14.4	Drilling for Information / 243	
14.5	Example for the Big Cheese / 247	
14.6	Evolving Metrics / 249	
14.7	Summary / 250	
	Problems / 250	
	Project / 251	
	Reference / 251	
Index		252