Contents

Preface, xvii
Acknowledgments, xix
Authors, xx1

PART 1  Fundamentals of Measurement and Experimentation

CHAPTER 1  Measurement: What Is It and Why Do It?  3

1.1 MEASUREMENT IN EVERYDAY LIFE  4
   1.1.1 What Is Measurement?  4
   1.1.2 "What Is Not Measurable Make Measurable"  7

1.2 MEASUREMENT IN SOFTWARE ENGINEERING  11
   1.2.1 Neglect of Measurement in Software Engineering  12
   1.2.2 Objectives for Software Measurement  14
      1.2.2.1 Managers  14
      1.2.2.2 Developers  15
   1.2.3 Measurement for Understanding, Control, and Improvement  16

1.3 SCOPE OF SOFTWARE METRICS  17
   1.3.1 Cost and Effort Estimation  18
   1.3.2 Data Collection  18
   1.3.3 Quality Models and Measures  19
   1.3.4 Reliability Models  19
   1.3.5 Security Metrics  20
   1.3.6 Structural and Complexity Metrics  20
   1.3.7 Capability Maturity Assessment  20
### 3.1 Processes 91

- 3.1.1 Processes
- 3.1.2 Products 92
  - 3.1.2.1 **External Product Attributes**
  - 3.1.2.2 **Internal Product Attributes**
  - 3.1.2.3 **The Importance of Internal Attributes**
  - 3.1.2.4 **Internal Attributes and Quality Control and Assurance**
  - 3.1.2.5 **Validating Composite Measures**
- 3.1.3 Resources
- 3.1.4 Change and Evolution

### 3.2 Determining What to Measure 100

- 3.2.1 **Goal-Question-Metric Paradigm**
- 3.2.2 Measurement for Process Improvement 105
- 3.2.3 Combining GQM with Process Maturity 108

### 3.3 Applying the Framework 110

- 3.3.1 Cost and Effort Estimation 110
- 3.3.2 Productivity Measures and Models 112
- 3.3.3 Data Collection
- 3.3.4 Quality Models and Measures 113
- 3.3.5 Reliability Models 113
- 3.3.6 Structural and Complexity Metrics 114
- 3.3.7 Management by Metrics 115
- 3.3.8 Evaluation of Methods and Tools 116

### 3.4 Software Measurement Validation 117

- 3.4.1 Validating Prediction Systems 117
- 3.4.2 Validating Measures 119
- 3.4.3 A Mathematical Perspective of Metric Validation 120

### 3.5 Performing Software Measurement Validation 121

- 3.5.1 A More Stringent Requirement for Validation 122
- 3.5.2 Validation and Imprecise Definition 124
- 3.5.3 How Not to Validate 125
- 3.5.4 Choosing Appropriate Prediction Systems 126
Chapter 4  •  Empirical Investigation

4.1  PRINCIPLES OF EMPIRICAL STUDIES

4.1.1  Control of Variables and Study Type 135
4.1.2  Study Goals and Hypotheses 139
4.1.3  Maintaining Control over Variables 141
4.1.4  Threats to Validity 143
4.1.5  Human Subjects 144

4.2  PLANNING EXPERIMENTS 145

4.2.1  A Process Model for Performing Experiments 145

4.2.1.1  Conception 146
4.2.1.2  Design 146
4.2.1.3  Preparation 150
4.2.1.4  Execution 150
4.2.1.5  Analysis 150
4.2.1.6  Dissemination and Decision-Making 150

4.2.2  Key Experimental Design Concepts 151

4.2.2.1  Replication 153
4.2.2.2  Randomization 154
4.2.2.3  Local Control 155

4.2.3  Types of Experimental Designs 157

4.2.3.1  Crossing 159
4.2.3.2  Nesting 159

4.2.4  Selecting an Experimental Design 161

4.2.4.1  Choosing the Number of Factors 162
4.2.4.2  Factors versus Blocks 164
4.2.4.3  Choosing between Nested and Crossed Designs 165

4.2.4.4  Fixed and Random Effects 169
CHAPTER 6  •  Analyzing Software Measurement Data 225

6.1 STATISTICAL DISTRIBUTIONS AND HYPOTHESIS TESTING 226
  6.1.1 Probability Distributions 226
  6.1.2 Hypothesis Testing Approaches 231

6.2 CLASSICAL DATA ANALYSIS TECHNIQUES 232
  6.2.1 Nature of the Data 233
    6.2.1.1 Sampling, Population, and Data Distribution 233
    6.2.1.2 Distribution of Software Measurements 236
    6.2.1.3 Statistical Inference and Classical Hypothesis Testing 239
  6.2.2 Purpose of the Experiment 241
    6.2.2.1 Confirming a Theory 241
    6.2.2.2 Exploring a Relationship 242
  6.2.3 Decision Tree 243

6.3 EXAMPLES OF SIMPLE ANALYSIS TECHNIQUES 243
  6.3.1 Box Plots 243
  6.3.2 Bar Charts 247
  6.3.3 Control Charts 248
  6.3.4 Scatter Plots 250
  6.3.5 Measures of Association 252
  6.3.6 Robust Correlation 253
  6.3.7 Linear Regression 255
  6.3.8 Robust Regression 257
  6.3.9 Multivariate Regression 259

6.4 MORE ADVANCED METHODS 259
  6.4.1 Classification Tree Analysis 259
  6.4.2 Transformations 261
  6.4.3 Multivariate Data Analysis 264
    6.4.3.1 Principal Component Analysis 264
    6.4.3.2 Cluster Analysis 267
    6.4.3.3 Discriminant Analysis 267
6.5 MULTICRITERIA DECISION AIDS

6.5.1 Basic Concepts of Multicriteria Decision-Making

6.5.2 Multiattribute Utility Theory

6.5.3 Outranking Methods

6.5.4 Bayesian Evaluation of Multiple Hypotheses

6.6 OVERVIEW OF STATISTICAL TESTS

6.6.1 One-Group Tests

   6.6.1.1 Binomial Test

   6.6.1.2 Chi-Squared Test for Goodness of Fit

   6.6.1.3 Kolmogorov-Smirnov One-Sample Test

   6.6.1.4 One-Sample Runs Test

   6.6.1.5 Change-Point Test

6.6.2 Two-Group Tests

   6.6.2.1 Tests to Compare Two Matched or Related Groups

   6.6.2.2 Tests to Compare Two Independent Groups

6.6.3 Comparisons Involving More than Two Groups

6.7 SUMMARY

EXERCISES

REFERENCE

FURTHER READING

CHAPTER 7 Metrics for Decision Support: The Need for Causal Models

7.1 FROM CORRELATION AND REGRESSION TO CAUSAL MODELS

7.2 BAYES THEOREM AND BAYESIAN NETWORKS

7.3 APPLYING BAYESIAN NETWORKS TO THE PROBLEM OF SOFTWARE DEFECTS PREDICTION

   7.3.1 A Very Simple BN for Understanding Defect Prediction

   7.3.2 A Full Model for Software Defects and Reliability Prediction

   7.3.3 Commercial Scale Versions of the Defect Prediction Models
PART II  Software Engineering Measurement

CHAPTER 8  Measuring Internal Product Attributes: Size 335

8.1 PROPERTIES OF SOFTWARE SIZE 336

8.2 CODE SIZE 339

8.2.1 Counting Lines of Code to Measure Code Size 339
8.2.2 Halstead’s Approach 344
8.2.3 Alternative Code Size Measures 346
8.2.4 Dealing with Nontextual or External Code 347

8.3 DESIGN SIZE 348

8.4 REQUIREMENTS ANALYSIS AND SPECIFICATION SIZE 350

8.5 FUNCTIONAL SIZE MEASURES AND ESTIMATORS 351

8.5.1 Function Points 352

8.5.1.1 Function Points for Object-Oriented Software 355

8.5.1.2 Function Point Limitations 356

8.5.2 COCOMO II Approach 358

8.6 APPLICATIONS OF SIZE MEASURES 360

8.6.1 Using Size to Normalize Other Measurements 360
8.6.2 Size-Based Reuse Measurement 361
8.6.3 Size-Based Software Testing Measurement 363

8.7 PROBLEM, SOLUTION SIZE, COMPUTATIONAL COMPLEXITY 364

8.8 SUMMARY 365

EXERCISES 366

FURTHER READING 368
9.4.4 Object-Oriented Reuse Measurement 422
9.4.5 Design Pattern Use 423

9.5 NO SINGLE OVERALL "SOFTWARE COMPLEXITY" MEASURE 425

9.6 SUMMARY 428

EXERCISES 429

APPENDICES TO CHAPTER 9 433

A.1 McCabe's Testing Strategy 433
A.1.1 Background 433
A.1.2 The Strategy 434
A.2 Computing Test Coverage Measures 436

FURTHER READING 437

CHAPTER 10 • Measuring External Product Attributes 441

10.1 MODELING SOFTWARE QUALITY 442
10.1.1 Early Models 443
10.1.2 Define-Your-Own Models 447
10.1.3 ISO/IEC 9126-1 and ISO/IEC 25010 Standard Quality Models 447

10.2 MEASURING ASPECTS OF QUALITY 449
10.2.1 Defects-Based Quality Measures 450
10.2.1.1 Defect Density Measures 450
10.2.1.2 Other Quality Measures Based on Defect Counts 455

10.3 USABILITY MEASURES 456
10.3.1 External View of Usability 457
10.3.2 Internal Attributes Affecting Usability 459

10.4 MAINTAINABILITY MEASURES 460
10.4.1 External View of Maintainability 462
10.4.2 Internal Attributes Affecting Maintainability 463

10.5 SECURITY MEASURES 466
10.5.1 External View of Security 467
10.5.2 Internal Attributes Affecting Security 470