1.

Record Nr.	UNISA990000580060203316
Autore	DI BELLO, Stelio
Titolo	La cucina del Cilento / Stelio Di Bello
Pubbl/distr/stampa	Napoli : T. Marotta, 1980
Edizione	[2. ed.]
Descrizione fisica	94 p ; 21 cm
Disciplina	641.594574
Collocazione	XV.1.A. 508(V G 479)
Lingua di pubblicazione	Italiano
Formato	Materiale a stampa
Livello bibliografico	Monografia

2.

Record Nr.	UNINA9910830234603321
Autore	Laird Linda M. <1952->
Titolo	Software measurement and estimation : a practical approach / / Linda M. Laird, M. Carol Brennan
Pubbl/distr/stampa	Hoboken, New Jersey : , : John Wiley & Sons, , 2006 [Piscataqay, New Jersey] : , : IEEE Xplore, , [2006]
ISBN	1-280-46844-0 9786610468447 0-470-24780-0 0-471-79253-5 0-471-79252-7
Descrizione fisica	1 online resource (276 p.)
Collana	Quantitative software engineering series ; ; 2
Altri autori (Persone)	BrennanM. Carol <1954->
Disciplina	005.1 005.14
Soggetti	Software measurement Software engineering
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia

Note generali	Description based upon print version of record.
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Acknowledgments -- 1. Introduction -- 1.1 Objective -- 1.2 Approach -- 1.3 Motivation -- 1.4 Summary -- References -- Chapter 1 Side Bar -- 2. What to Measure -- 2.1 Method 1: The Goal Question Metrics Approach -- 2.2 Extension to GQM: Metrics Mechanism is Important -- 2.3 Method 2: Decision Maker Model -- 2.4 Method 3: Standards Driven Metrics -- 2.5 What to Measure is a Function of Time -- 2.6 Summary -- References -- Exercises -- Project -- 3. Fundamentals of Measurement -- 3.1 Initial Measurement Exercise -- 3.2 The Challenge of Measurement -- 3.3 Measurement Models -- 3.3.1 Text Models -- 3.3.2 Diagrammatic Models -- 3.3.3 Algorithmic Models -- 3.3.4 Model Examples: Response Time -- 3.3.5 The Pantometric Paradigm - How to Measure Anything -- 3.4 Meta-Model for Metrics -- 3.5 The Power of Measurement -- 3.6 Measurement Theory -- 3.6.1 Introduction to Measurement Theory -- 3.6.2 Measurement Scales -- 3.6.3 Measures of Central Tendency and Variability -- 3.6.3.1 Measures of Central Tendency -- 3.6.3.2 Measures of Variability -- 3.6.4 Validity and Reliability of Measurement -- 3.6.5 Measurement Error -- 3.7 Accuracy versus Precision and the Limits of Software Measurement -- 3.7.1 Summary -- 3.7.2 Problems -- 3.7.3 Project -- References -- 4. Measuring the Size of Software -- 4.1 Physical Measurements of Software -- 4.1.1 Measuring Lines of Code -- 4.1.1.1 Code Counting Checklists -- 4.1.2 Language Productivity Factor -- 4.1.3 Counting Reused and Refactored Code -- 4.1.4 Counting Non-Procedural Code Length -- 4.1.5 Measuring the Length of Specifications and Design -- 4.2 Measuring Functionality -- 4.2.1 Function Points -- 4.2.1.1 Counting Function Points -- 4.2.2 Function Point Counting Exercise -- 4.2.3 Converting Function Points to Physical Size -- 4.2.4 Converting Function Points to Effort -- 4.2.5 Other Function Point Engineering Rules -- 4.2.6 Function Point Pros and Cons -- 4.3 Feature Points -- 4.4 Size Summary -- 4.5 Size Exercises -- 4.6 Theater Tickets Project. References -- 5. Measuring Complexity -- 5.1 Structural Complexity -- 5.1.1 Size as a Complexity Measure -- 5.1.1.1 System Size and Complexity -- 5.1.1.2 Module Size and Complexity -- 5.1.2 Cyclomatic Complexity -- 5.1.3 Halstead's Metrics -- 5.1.4 Information Flow Metrics -- 5.1.5 System Complexity -- 5.1.5.1 Maintainability Index -- 5.1.5.2 The Agresti-Card System Complexity Metric -- 5.1.6 Object-Oriented Design Metrics -- 5.1.7 Structural Complexity Summary -- 5.2 Conceptual Complexity -- 5.3 Computational Complexity -- 5.4 Complexity Metrics Summary -- 5.5 Complexity Exercises -- 5.6 Projects -- References -- 6. Estimating Effort -- 6.1 Effort Estimation - Where are we? -- 6.2 Software Estimation Methodologies and Models -- 6.2.1 Expert Estimation -- 6.2.1.1 Work and Activity Decomposition -- 6.2.1.2 System Decomposition -- 6.2.1.3 The Delphi Methods -- 6.2.2 Using Benchmark Size Data -- 6.2.2.1 Lines of Code Benchmark Data -- 6.2.2.2 Function Point Benchmark Data -- 6.2.3 Estimation by Analogy -- 6.2.3.1 Traditional Analogy Approach -- 6.2.3.2 Analogy Summary -- 6.2.4 Proxy Point Estimation Methods -- 6.2.4.1 Meta-Model for Effort Estimation -- 6.2.4.2 Function Points -- 6.2.4.2.1 COSMIC Function Points -- 6.2.4.3 Object Points -- 6.2.4.4 Use Case Sizing Methodologies -- 6.2.4.4.1 Use Case Points Methodology -- 6.2.4.4.2 Example: Use Case Point Methodology Example: Home Security System -- 6.2.4.4.3 Use Case Point Methodology Effectiveness -- 6.2.5 Custom Models -- 6.2.6 Algorithmic Models -- 6.2.6.1 Manual Models -- 6.2.6.2 Estimating Project Duration -- 6.2.6.3 Tool Based Models -- 6.3 Combining Estimates -- 6.4 Estimating Issues -- 6.4.1 Targets vs. Estimates -- 6.4.2 The Limitations of Estimation - Why? --

Sommario/riassunto	This book serves as a practical guide to metrics and quantitative software estimation, beginning with the foundations of measurement and metrics, and then focuses on techniques and tools for estimation of the required effort and the resulting quality of a software project.