Books

Software Metrics: A Rigorous & Practical Approach, Second Edition, Norman E. Fenton and Shari Lawrence Pfleeger, International Thomson Computer Press, London, 1997. 638 pp. (ISBN 0-534-95600-9).

Many of us have seen the "Dilbert" comic strip by Scott Adams, where The Boss complains to Wally that his laptop weighs too much. Wally suggests that he "delete files to lower the weight."

Wally's silly suggestion is funny, not because we know that software has no weight, but because many of us know of "bosses" who ask for software measurements that do not make sense.

Although software has no physical presence, managers, hopefully unlike The Boss in "Dilbert," need to be able to quantify software attributes. Indeed, software is often the most expensive component of a product. Yet, it is not visible. It is easy to see when a building is half completed, but the statement that coding is "90 percent complete" carries little useful information.

The field of software measurement, or software metrics, has grown up out of two needs. One is the need for software developers to be able to manage the software development process. For example, developers need to determine the resources (i.e., programmers) or time needed to deliver a product or whether the software is "good enough" to release. The other need comes from researchers, who want to be able to objectively define and measure software attributes in order to better understand software engineering.

Unfortunately, developers and managers have sometimes adopted any available measurement scheme, even if it had not been scientifically validated, because of the need for numbers. Meanwhile, the development of well-defined and validated software measures has been very slow. *Software Metrics: A Rigorous & Practical Approach, Second Edition*, by Norman Fenton and Shari Lawrence Pfleeger, shows practitioners how to evaluate and apply software measurements for specific purposes, and it provides guidance for researchers interested in defining and validating ways to quantify important software attributes.

I have used the first edition of this book, *Software Metrics: A Rigorous Approach* by Norman Fenton, as a text for graduate courses in software engineering measurement since it was published in 1991. I can't speak for my students, but when I've used the first edition in my classes I've received very good course evaluations. When it was first published, the first edition was the only software metrics book that took a rigorous, measurement theory approach, and covered a broad range of metrics topics. The first edition contains many good examples, and is very well written. Its comprehensive annotated bibliography is one of its strengths; the annotations are critiques and not just abstracts.

Software Metrics: A Rigorous & Practical Approach, Second Edition, is 638 pages long, nearly twice as long as the first edition. It is divided into three parts: Part I describes fundamentals of measurement theory, measurement frameworks, empirical studies, data collection and analysis; Part II provides details on how to measure attributes of products, processes, and resources; Part III focuses on the practical issues of setting up a measurement program and conducting empirical studies. Improvements over the first edition include expanded and updated references, improved explanations, and more exercises. Each chapter now has a further readings section with pointers to relevant references.

The most notable improvement is expanded coverage of practical issues. Chapter 12 describes process prediction techniques such as COCOMO, and their

0018-8670/97/$5.00 © 1997 IBM

limitations. Chapter 13, "Planning a Measurement Program," is new; it replaces Chapter 7 in the first edition. This chapter shows how to ensure that the measurements taken actually satisfy the organization's objectives. Measurement programs need to be continuously revised to adapt to changing environments. Chapter 13 also addresses alternative measurement frameworks such as the SEI Capability Maturity Model (CMM), Software Process Improvement and Capability dEtermination (SPICE), and ISO 9000. Chapter 14 describes the use of software measurement at several organizations including Motorola, AT&T, NASA, Hewlett-Packard, Contel, Siemens, and Hitachi. These improvements in the book are reflected in the new title, which adds the word "Practical" to the title of the first edition.

A new chapter on empirical research (Chapter 15) provides a lucid description of the decision process often used to adopt new software engineering techniques and tools. Rather than rely on empirical evidence, decisions are made based on the "credibility of the seller or advocate." As a result, many inappropriate techniques and tools are adopted. Chapter 15 provides an excellent description of how to effectively evaluate software engineering methods.

Other improvements include an emphasis on the Goal Question Metric (GQM) paradigm, process improvement and evaluation using the SEI CMM, and an expanded description of how to validate software measures and models. Chapter 2, the measurement theory chapter, includes a very clear discussion of the qualitative and intuitive basis that underlies all good software measures. There is an expanded explanation of function points and their limitations. The material on software quality measurement is greatly expanded.

Although this book includes a vast amount of information, it is written in a clear and entertaining style. There are many insightful observations. For example, Chapter 2 shows parallels between the data needed by professional gamblers and that needed to make software engineering predictions. Like gamblers, software managers need significant information about past events to limit the inherent riskiness of making predictions.

Software Metrics: A Rigorous & Practical Approach, Second Edition, really answers the needs of a wide range of readers including practitioners, students, and researchers. It can be used as a reference book, tutorial, and text and belongs on the bookshelf of all software engineers.

James M. Bieman
Department of Computer Science
Colorado State University
Fort Collins
Colorado

Note—The books reviewed are those the Editor thinks might be of interest to our readers. The reviews express the opinions of the reviewers.