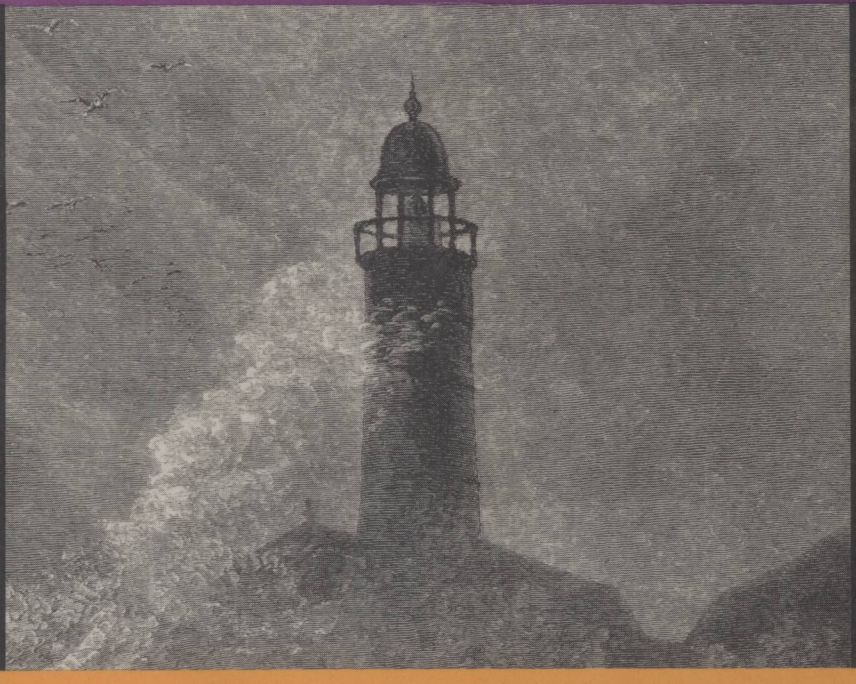
Professional Software

Software Engineering Concepts

HENRY LEDGARD *with* JOHN TAUER



Volume I

Professional Software

Volume I

Software Engineering Concepts

HENRY LEDGARD *with* JOHN TAUER

Addison-Wesley Publishing Company

Reading, Massachusetts • Menlo Park, California • Don Mills, Ontario
Wokingham, England • Amsterdam • Sydney • Singapore • Tokyo
Madrid • Bogotá • Santiago • San Juan

Library of Congress Cataloging-in-Publication Data

Ledgard, Henry F., 1943—
Professional software.

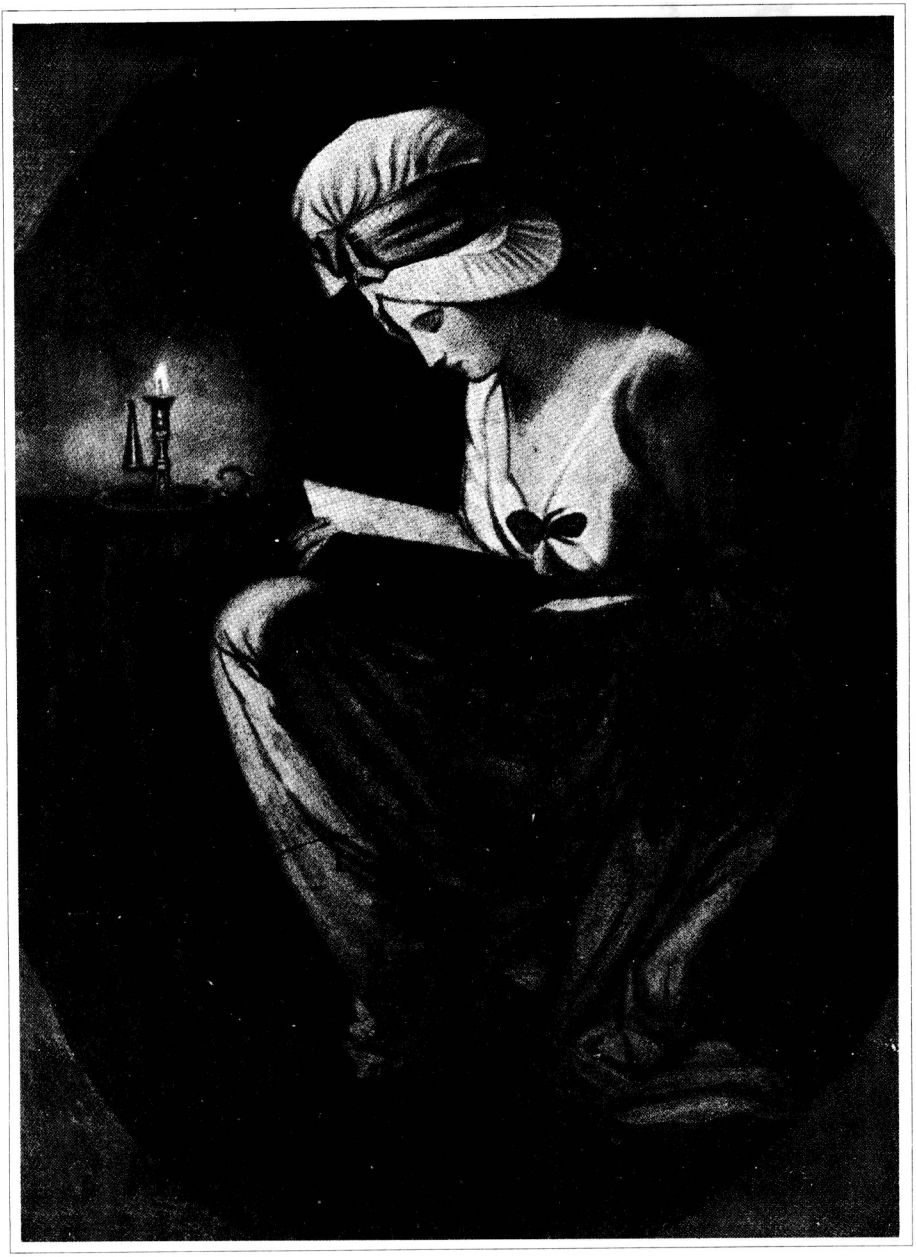
Contents: Software engineering — Programming practice.
Includes bibliographies and indexes.
1. Computer software—Development. 2. Electronic digital computers—Programming. I. Title.
QA76.76.D47L43 1987 005 87-1760
ISBN 0-201-12231-6 (v. 1)
ISBN 0-201-12232-4 (v. 2)

ABCDEFGHIJ-AL-8987

Professional Software

Volume I

Software Engineering Concepts



Preface

This work treats a number of *practical* issues in software development. The issues raised here are not new but fundamental to the day-to-day concerns of the practicing professional.

In the discipline of computer science, there is a distinction to make. This is between a fundamental approach, which is necessarily quantitative, and a qualitative aspect that, while less easy to define, is equally important.

New tools, algorithms, design notations, structured editors, powerful workstations, rapid prototyping—these are exciting developments. But we must also keep sight of the more qualitative roots of our profession. These roots include:

- discipline
- teamwork
- craftsmanship
- quality

My experience suggests that, in our desire to discover new technologies, we are not always sharpening our understanding of concepts like the software development process or program readability.

Organization of This Work

The work is organized around a series of topics. Each topic relates, directly or indirectly, to the quality of software. The topics are collected into two broad categories in separate volumes.

Volume I. *Software Engineering.* This volume treats several issues that are mainly relevant to the writing of large programs. This is the area generally called *software engineering.* This includes topics such as the need to reaffirm the importance of the software lifecycle and the organization of programming teams. This section also treats the difficulties users encounter in software, an aspect of software engineering called "software human factors" or "software human engineering." Here, for example, the idea of user testing is discussed. User testing is a remarkably simple and pragmatic idea—before we ship completed systems we ought to have typical users try the system so that we can learn from their difficulties.

Volume II. *Programming Practice.* The second volume deals exclusively with programs themselves. It treats topics like the role of program comments, the persistence of global variables, and the use of packages. The underlying theme of these essays is to convey the need for careful craftsmanship in programming.

Both Volumes I and II conclude with an extended example. In developing this example, a good number of ideas mentioned in previous essays are put to practice. In Volume I, the example is developed in stages, with particular attention to the software engineering process. The final program is given as an appendix. In Volume II, the final program itself is examined. The program is presented again; this time with annotations on programming practice.

This work certainly does not address all the issues in software development. These include critical topics like planning, management techniques, resource estimating, design notations, and testing.

This book has been typeset using a monospaced font (both bold and nonbold) for programs. Monospaced typefaces, with or without bold, are most appropriate for programs. They promote readability and, I believe, give the best appearance for printing programs.

Acknowledgments

A number of people have personally influenced this work, either through my thinking about the issues or by motivating my desire to take on this effort. These include William Cave, David Gries, Jean Ichbiah, Michael Marcotty, Harlan Mills, and Andrew Singer.

Jon Hueras, over many years, has in his quiet ways continued to demonstrate the finest in software engineering.

Richard Rasala (Northeastern University), Jerry Waxman (Queens College, New York), Richard Rinewalt (University of Texas at Arlington), Charles Engelke (University of Florida in Gainesville) provided helpful review comments on an earlier version of this work. Bernhard Weinberg (Michigan State University, East Lansing), Steven Wartik (University of Virginia, Charlottesville), Richard Vidale (Boston University), and Daniel McCracken (City University of New York) thoughtfully reviewed this particular work.

The Philips courses, where I gained a clearer understanding of software quality, were under the brilliant and inspired direction of Allen Macro, assisted by John Buxton. Their book, *The Craft of Software Engineering,* is a fundamental work in the field and is based on years of experience and thinking. Nat Macon, my colleague, as well as the participants in the Philips courses, strengthened my commitment to excellence and teaching.

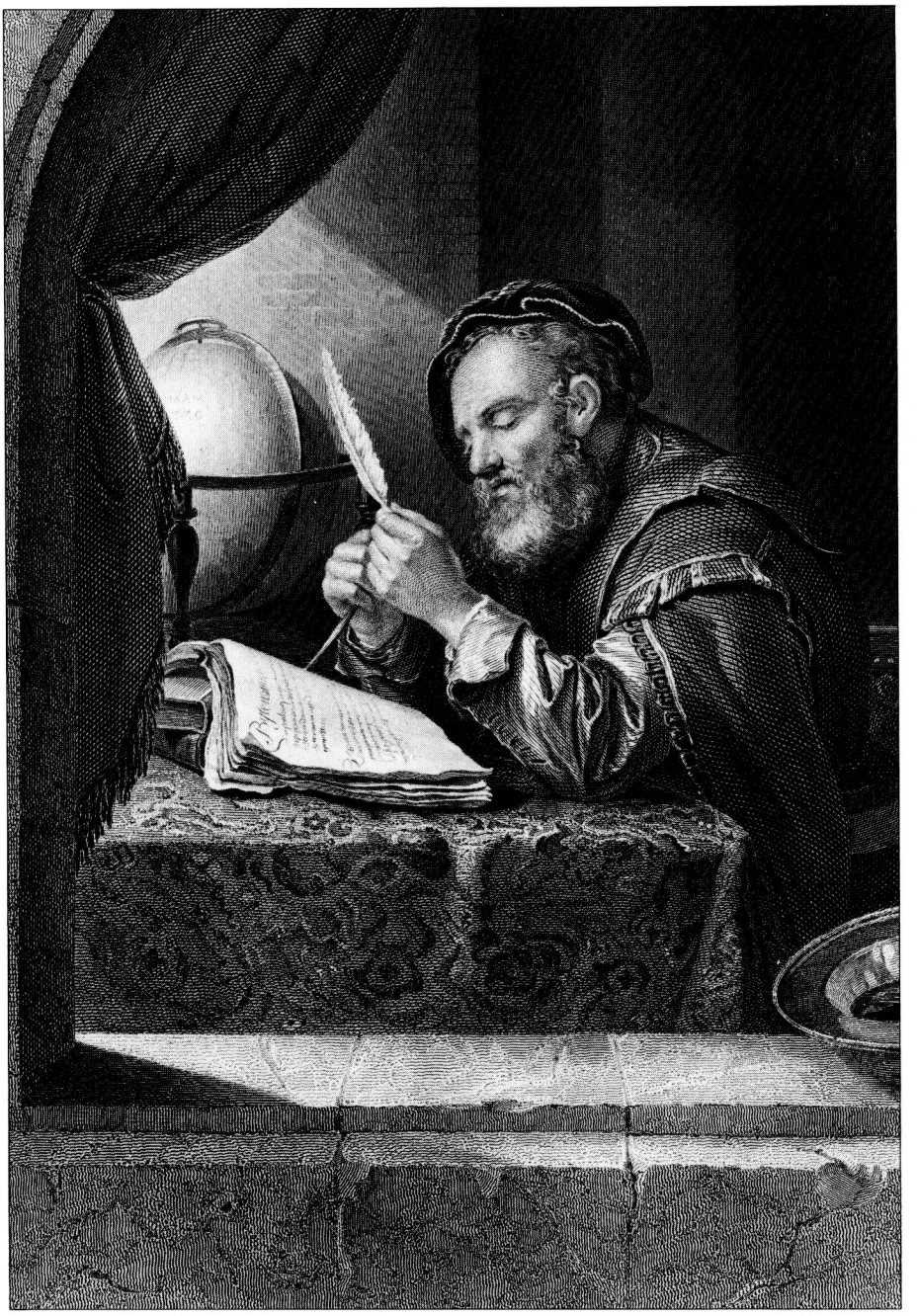
Howard Karger provided the photographs of the light blub and the spider web. Christine Lee provided the computer-generated piece of art depicting the cats and the graphic for program layout.

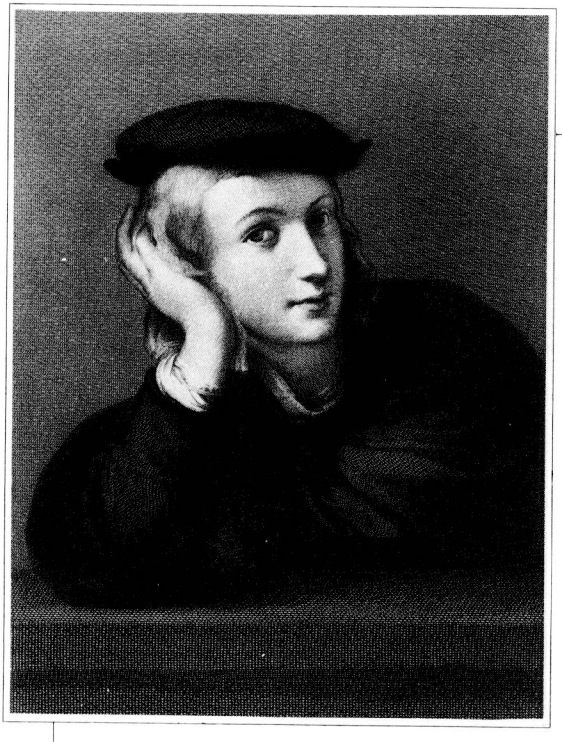
John Tauer, a professional from another discipline, assisted me on this work with his gifted pen and mind. He turned a table at Daisy's into a forum for discussing the very heart of professional practice.

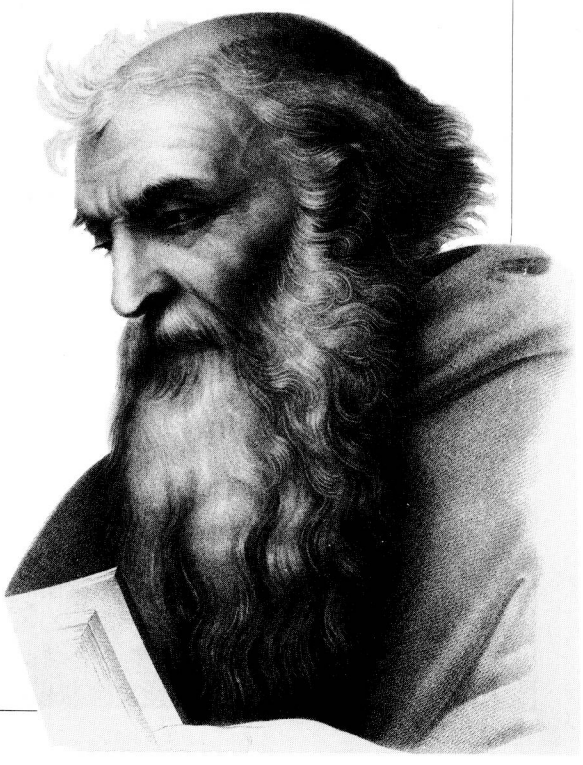
H. L.

To The Reader

This book is my considered opinion about professional practice. It is derived from teaching students and professionals and from participating in numerous software efforts. The thoughtful reader may, in places, have good reason to hold other views. This should not confuse our common goal, the pursuit for excellence.







Contents

Contents for Volume II
Programming Practice

1

Programmers: The Amateur versus The Professional

At the outset, it should be understood that an endeavor to compare an amateur and a professional does not describe the full hierarchy in programming or any other vocation or avocation. The complete spectrum might be: the ignorant, the novice, the aficionado, the amateur, the professional, and the master.

Music, expressly classical music, comes to mind if all six rankings were to be defined. The novice appreciates Tchaikovsky's 1812 Overture every Fourth of July when the cannons go off along the Charles River in Boston. The novice learns the notes of the scale in order to play an instrument (the guitar, usually). The aficionado can tell the difference between Mozart and Mahler, the amateur can play the first two movements of the "Moonlight" sonata, and the professional can play all 32 Beethoven sonatas. The master writes the music that eventually finds its way into the repertoire.

The distinction between amateur and professional that we are making, however, is not a simple matter. Consider an analogy with a golfer. Given some basic talent, capable instruction, and a good deal of practice, a young player can play close to par golf and decide to pursue a career as a professional. After a perfunctory declaration or surviving the qualification tournaments, he may end up as a club pro or be invited to the Master's