SOFTWARE ENGINEERING

A Practitioner's Approach

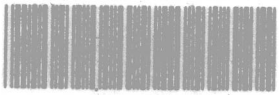
~~Second~~ Edition

Roger S. Pressman, Ph.D.

SOFTWARE ENGINEERING

A Practitioner's Approach

Second Edition



Roger S. Pressman, Ph.D.

President, R.S. Pressman & Associates, Inc.
and
Adjunct Professor of Computer Engineering
University of Bridgeport

McGraw-Hill Book Company

New York St. Louis San Francisco Auckland Bogotá Hamburg Johannesburg
London Madrid Mexico Milan Montreal New Delhi Panama
Paris São Paulo Singapore Sydney Tokyo Toronto

This book was set in Times Roman by Publication Services.
The editor was Gerald A. Gleason;
the production supervisor was Phil Galea;
the cover was designed by Infield + D'Astolfo.
Project supervision was done by Publication Services.
R.R. Donnelley & Sons Company was the printer and binder.

SOFTWARE ENGINEERING
A Practitioner's Approach

1 2 3 4 5 6 7 8 9 0 DOCDOC 8 9 4 3 2 1 0 9 8 7

ISBN 0-07-050783-X

Library of Congress Cataloging-in-Publication Data
Pressman, Roger S.
Software engineering.
Includes bibliographies and index.
1. Electronic digital computers—Programming.
I. Title.
QA76.6.P73 1987 005.1 86-18549
ISBN 0-07-050783-X

SOFTWARE ENGINEERING
A Practitioner's Approach

McGraw-Hill Series in Software Engineering and Technology

Consulting Editor

Peter Freeman, *University of California, Irvine*

Cohen: *Ada as a Second Language*
Fairley: *Software Engineering Concepts*
Howden: *Functional Program Testing and Analysis*
Jones: *Programming Productivity*
Kolence: *An Introduction to Software Physics: The Meaning of Computer Measurement*
Musa, Iannino and Okumoto: *Software Reliability: Measurement, Prediction and Application*
Pressman: *Software Engineering: A Practitioner's Approach*

ABOUT THE AUTHOR

Roger S. Pressman is a nationally recognized industry consultant in software engineering. He received a B.S.E. from the University of Connecticut, an M.S. from the University of Bridgeport, and a Ph.D. in Engineering from the University of Connecticut. He has had nearly two decades of industry experience, holding both technical and management positions with responsibility for the development of software for engineered products and systems.

In addition to his industry experience, Dr. Pressman was Bullard Associate Professor of Computer Engineering at the University of Bridgeport and Director of the University's Computer Aided Design and Manufacturing Center. He remains at the university as an adjunct Associate Professor of Computer Engineering.

Dr. Pressman is President of R.S. Pressman & Associates, Inc., a consulting firm specializing in management and technical problem solving in software engineering. Clients include many large corporations as well as smaller high technology companies. The firm specializes in software engineering training and has developed courses and other training products that have been used by over 10,000 industry professionals.

Dr. Pressman is author of many technical papers and another book, *Numerical Control and Computer Aided Manufacturing*. He is a member of the ACM, IEEE, and Tau Beta Pi, Phi Kappa Phi, Pi Tau Sigma, and Eta Kappa Nu.

TO MY PARENTS

PREFACE

In the five years since the first edition of *Software Engineering: A Practitioner's Approach*, software engineering has grown from infancy to early adolescence. Today, software engineering is recognized as a legitimate discipline and "software engineer" has replaced "programmer" as the job title of preference. Software engineering methods and procedures have been adopted successfully in a broad spectrum of industry applications. Managers and practitioners alike recognize the need for a more disciplined approach to software development.

But the problems that were described in the preface to the first edition remain with us. Many companies still develop software haphazardly. Many professionals and students are unaware of modern methods. In addition, debate and controversy about the true nature of the software engineering approach continue. The status of software engineering is a study in contrasts. Attitudes have changed, progress has been made, but much remains to be done before the discipline reaches maturity.

The second edition of *Software Engineering: A Practitioner's Approach* is intended to provide one element of a foundation from which the bridge from adolescence to maturity can be constructed. The second edition, like the first, is intended for both students and practitioners, maintaining the same format and style of the first edition. The book retains its appeal as a guide to the industry professional and a comprehensive introduction to the student at the upper level undergraduate or first year graduate level.

Like the first edition, software engineering methods are presented in the chronological sequence that they are applied during software development. However, the second edition is more than a simple update. The book has be restructured to emphasize new and important software engineering methods and techniques. Rather than maintaining a strict life cycle view, the second edition presents generic activities that are performed regardless of the software engineering paradigm that has been chosen.

Chapters that have been retained from the first edition have been expanded and revised to reflect current trends and techniques. Major new sections have been added to chapters on computer system engineering, software project planning, analysis methodologies, programming languages and coding, testing strategies and software maintenance.

New chapters on analysis and design fundamentals have been added to provide a foundation for the methods that are introduced in later chapters. In addition, new chapters present object-oriented design, real-time design, software test case design techniques, and software quality assurance. These chapters have been added to reflect new software engineering methods that are rapidly gaining acceptance in the industry. Many new examples, problems, and points to ponder have been added and the *Further Readings* sections (one of the more popular tidbits in the first edition) have been expanded and updated for every chapter.

The software engineering literature is expanding at an explosive rate. Once again, my thanks to the many authors who have provided additional insight, ideas, and commentary in the years since the first edition. Many have been referenced within the pages of each chapter. All deserve credit for their contribution to this rapidly evolving field. I also wish to thank the reviewers of the second edition: Robert Glass, Seattle University; Ernest H. Goldman, University of Bridgeport; Medi-Harandi, University of Illinois-Urbana-Champaign; Marvin Zelkowitz, University of Maryland; and John Musa, ATT Bell Labs. Their comment and criticism have been invaluable.

The content of the second edition of *Software Engineering: A Practitioner's Approach* has been shaped by the hundreds of industry professionals, university professors and students who have taken the time to communicate their suggestions, criticisms and ideas. In addition, my personal thanks go to our many industry clients, who certainly teach me as much or more than I can teach them.

Finally, to Barbara, Mathew, and Michael, my love and thanks for tolerating my travel schedule, understanding the evenings at the office, and encouraging the second edition of "the book."

Roger S. Pressman

PREFACE TO THE FIRST EDITION

In the brief history of the electronic digital computer, the 1950s and 1960s were decades of hardware. The 1970s were a period of transition and a time of recognition of software. The decade of software is now upon us. In fact, advances in computing may become limited by our inability to produce quality software that can tap the enormous capacity of 1980-era processors.

During the past decade we have grown to recognize circumstances that are collectively called the *software crisis*. Software costs escalated dramatically, becoming the largest dollar item in many computer-based systems. Schedules and completion dates were set but rarely kept. As software systems grew larger, quality became suspect. Individuals responsible for software development projects had limited historical data to use as guides and less control over the course of a project.

A set of techniques, collectively called *software engineering*, has evolved as a response to the software crisis. These techniques deal with software as an engineered product that requires planning, analysis, design, implementation, testing, and maintenance. The goal of this text is to provide a concise presentation of each step in the software engineering process.

The contents of this book closely parallel the software life cycle. Early chapters present the planning phase, emphasizing system definition (computer systems engineering), software planning, and software requirements analysis. Specific techniques for software costs and schedule estimation should be of particular interest to project managers as well as to technical practitioners and students.

In subsequent chapters emphasis shifts to the software development phase. The fundamental principles of software design are introduced. In addition, descriptions of two important classes of software design methodology are presented in detail. A variety of software tools are discussed. Comparisons among techniques and among tools are provided to assist the practitioner and student alike. Coding style is also stressed in the context of the software engineering process.

The concluding chapters deal with software testing techniques, reliability, and software maintenance. Software engineering steps associated with testing are de-

scribed and specific techniques for software testing are presented. The current status of software reliability prediction is discussed and an overview of reliability models and program correctness approaches is presented. The concluding chapter considers both management and technical aspects of software maintenance.

This book is an outgrowth of a senior-level/first-year-graduate course in software engineering offered at the University of Bridgeport. The course and this text cover both management and technical aspects of the software development process. The chapters of the text correspond roughly to major lecture topics. In fact, the text is derived in part from edited versions of transcribed notes of these lectures. Writing style is therefore purposely casual and figures are derived from viewgraphs used during the course.

Software Engineering: A Practitioner's Approach may be used in a number of ways for various audiences. The text can serve as a concise guide to software engineering for the practicing manager, analyst, or programmer. It can also serve as the basic text for an upper-level undergraduate or graduate course in software engineering. Lastly, the text can be used as a supplementary guide for software development early in computer science or computer engineering undergraduate curricula.

The software engineering literature has expanded rapidly during the past decade. I gratefully acknowledge the many authors who have helped this new discipline evolve. Their work has had an important influence on this book and my method of presentation. I also wish to acknowledge Pat Duran, Leo Lambert, Kyu Lee, John Musa, Claude Walston, Anthony Wasserman, Marvin Zelkowitz, and Nicholas Zvegintzov, the reviewers of this book, and Peter Freeman, the series editor. Their thoughtful insights and suggestions have been invaluable during the final stages of preparation. Special thanks go to Leo Lambert and his colleagues from the Computer Management Operation, General Electric Company, who have allowed me to tap their broad collective experience during my long association with them. In addition, to the students at the University of Bridgeport and the hundreds of software professionals and their managers who have attended short courses that I have taught, my thanks for the arguments, the ideas, and the challenges that are essential in a field such as ours.

Finally, to Barbara, Mathew, and Michael, my love and thanks for tolerating the genesis of book number two.

Roger S. Pressman

CONTENTS