An Introduction to Software Quality Control

Chin-Kuei Cho, PhD

The MITRE Corporation
and
The George Washington University

A [illegible]science Publication
JOHN WILEY & SONS
New York • Chichester • Brisbane • Toronto

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional service. If legal advice or other expert assistance is required, the services of a competent professional person should be sought. *From a Declaration of principles jointly adopted by a Committee of the American Bar Association and a Committee of Publishers.*

***Library of Congress Cataloging in Publication Data*:**

Cho, Chin-Kuei, 1937-
An introduction to software quality control

(Business data processing)
"A Wiley-Interscience publication."
Includes indexes.
1. Electronic digital computers—Programming—
Quality control. 2. Computer programs—Quality control.
3. Software engineering.
I. Title.

QA76.6.C45 001.64'25 80-15244
ISBN 0-471-04704-X

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

Business Data Processing
A WILEY SERIES

Personnel Implications for Business Data Processing
Robert A. Dickmann

Design of Sequential File Systems
Thomas R. Gildersleeve

Management Reporting Systems
James M. McKeever in collaboration with Benedict Kruse

A Guide to Packaged Systems
Robert V. Head

Control of the Information System Development Cycle
Robert I. Benjamin

Decision Tables: Theory and Practice
Solomon L. Pollack, Harry T. Hicks, Jr., and William J. Harrison

An Introduction to Data Base Design
John K. Lyon

The Organization of the Data Processing Function
Frederic G. Withington

COBOL Support Packages: Programming and Productivity Aids
Stanley M. Naftaly, Bruce G. Johnson, and Michael C. Cohen

Modular Programming in COBOL
Russell M. Armstrong

Functional Analysis of Information Networks
Hal B. Becker

Functional Analysis of Information Processing
Grayce M. Booth

Effective Use of ANS COBOL Computer Programming Language
Laurence S. Cohn

The Data Base Administrator
John K. Lyon

Software Reliability: Principles and Practices
Glenford J. Myers

The Psychology of Business Systems
William C. Ramsgard

The Art of Software Testing
Glenford J. Myers

Data Base: Structured Techniques for Design, Performance, and Management
S. Atre

An Introduction to Software Quality Control
Chin-Kuei Cho

Motivating and Managing Computer Personnel
J. Daniel Couger and Robert A. Zawacki

An Introduction to Software Quality Control

To

My Parents
and
Yin Cho
Shin-Gan Wu

Foreword

Although there exists abundant literature on techniques and methodologies of software development, most such methods are concerned with a very specific phase in the development cycle, and few consider the software development process as a manufacturing process. Furthermore, we still lack good techniques for achieving specific software quality attributes such as reliability, performance, and maintainability. Therefore, I am particularly pleased to read Dr. Cho's manuscript, *An Introduction to Software Quality Control*, in which statistical quality control techniques are systematically applied to the software development phases from requirements specification to system acceptance test.

The key concept in this book is the specification of input domain to the software system by means of the *symbolic input attribute decomposition* (SIAD) tree. The SIAD tree not only will force the development of well-defined user requirements before design and implementation, but its usage throughout the development cycle will also impose discipline in design and implementation. Furthermore, a test plan based on the SIAD tree can be designed and implemented concurrently with the development. The quality control comes from both the imposed discipline as well as the systematic application of statistical sampling techniques using the SIAD tree. It is interesting to observe that the author has implicitly developed a new "operational reliability" model. Of course, the quality attributes covered in this book are much more than just reliability.

This book is written with practitioners in mind. The style of writing is to first pinpoint problems to be encountered in different development phases, and then discuss techniques to systematically overcome these problems. The author has, in fact, introduced a uniform development methodology, since the designs of both the system and the test are based on the SIAD tree concept at different levels. Furthermore, each relevant chapter always includes a good survey and evaluation of current practices as well as ample examples for illustrations.

I find the idea of SIAD tree to be refreshingly original. The book is very readable and practical. I consider it to be a major contribution to the software engineering discipline.

Raymond T. Yeh

College Park, Maryland

Preface

Good quality software can save human lives!
Good quality software will save millions of dollars!
Good quality software can change the computer industry!

This book is a timely response to the urgent need for good quality software in the software industry. For the first time, the industry is considered as a manufacturing industry. The software is considered as a manufacturing plant working raw data into usable data. The well-established, powerful, and widely used quality control in modern manufacturing industries can be applied to controlling the quality of the data produced by the software, with proper definitions of input domain and designs of product units.

This book provides a systematic introduction to software quality control through prevention and promotion channels based on current software development methods and statistical quality control principles respectively. Top-down, bottom-up, structured programming, and model-driven techniques are widely discussed and used in the industry. Similarly, in manufacturing industries statistical quality control has been a powerful and widely used tool to ensure the quality of finished products such as light bulbs, tires, television sets, and automobiles. Principles of this tool (including sampling, testing, measurements, inspections, defective cause findings, improvements, statistical inferences, and acceptance sampling) are being applied to manufacturing processes from the acceptance of raw materials and the inspection of intermediate products to the acceptance of finished goods. A newly developed method called quality programming, which is a synthesis of the above-mentioned software development methods and which employs the quality control principles, is introduced throughout the book.

The contents of this book are divided into three major parts. A brief survey of the current state of software development, together with the nature of quality control and software quality control, is introduced first. Part 1 deals with the fundamental concepts in statistics required to understand the principles of quality control.

Part 2, consisting of Chapters 7 through 10, presents concepts in producing good quality software through practices such as good requirement specifications, structured design, structured programming, and structured module interfacing to prevent errors from entering the software. One important concept, called the symbolic input attribute decomposition (SIAD) tree, is introduced. The tree is used to represent the input domain of the software from which input data (or "raw material") are constructed. Its use will force the development of well-defined user requirements before software design and implementation begin. The reason for doing this is very simple. Defective raw materials always lead to defective products regardless of how correct and reliable a manufacturing plant is in any manufacturing industry. The raw data input to the software must not be defective or else defective output data is inevitable. If the input domain of the software is not well defined, it is doubtful that the input data constructed from the input domain will be of good quality with respect to the design of the software. The defective input data may be one of the most serious causes of current unreliable software. The SIAD tree is also used to facilitate concurrent development of software design and software test design and serves as a convenient vehicle for verifying the designs for consistency, completeness, and correctness.

Part 3, consisting of Chapters 11 through 14, explains the promotion of software quality through practices such as testing, validation, verification, and debugging. A brief survey of existing approaches and their advantages and disadvantages is given. A major idea in this book is the development of a quantified software output data defective rate estimate and software output data acceptance planning based on the input domain and product units by statistical methods. It is shown step-by-step how the estimate and acceptance of the output data is implemented. Examples are given to illustrate the application of the approach to numerical and nonnumerical programs. The application of the estimate and acceptance approach to a FORTRAN and a COBOL compiler is also given.

Materials presented in this book result from the author's more than 10 years' software engineering and data base development hands-on experience. The book is introductory in nature and is a self-contained treatment. No special prerequisite is necessary except some knowledge and/or experience in the FORTRAN and COBOL language. It is designed to be used as a reference and as a textbook for senior and first-year graduate-level computer science or software engineering courses. Practitioners in the computer industry such as programmers, analysts, software designers, test personnel, data processing personnel, programming project managers, company executives, and contract administrators would benefit the most from using this book.

Some materials in Chapters 2 to 6 are based on the book

I. Guttman and S. S. Wilks, *Introductory Engineering Statistics*, John Wiley & Sons, Inc., 1965.

The contribution of John Wiley & Sons, Inc., is acknowledged.

I would like to thank Arthur E. Hutt and Gerald T. Papke for their farsightedness in bringing this book into being. I am deeply grateful to many of my friends and colleagues for their help and suggestions. In particular, Richard A. Rucker, Robert Hart, Frank Rawlinson, Sam Wang, Robert M. Tarakan, Elizabeth Eckl, Ramona J. Briggs, Dr. Sou Yang Liu, Dr. Peter Gross, Dr. David S. Alberts, Dr. Jeffrey A. Krauss, Dr. Robert S. Conker, Dr. Michael C. Chen, Dr. Ned Chapin, Dr. Bruce H. Barnes, and Dr. Thomas P. Keenan. The IBM 370/148 computer made available for the book by the George Washington University through E. Michael Hamilton and Kay K. Beach, the contribution of the Department of Electrical Engineering and Computer Science at the George Washington University to the preparation of the manuscript, and the help from Glenn Morgan are appreciated. It should be pointed out that the opinions and views presented in the book are solely the author's and do not necessarily represent the opinions and views of the people mentioned above.

I would also like to thank my wife Yush-Chye and my lovely Mary and Jennifer for their patience in allowing me to devote my time to this project.

CHIN-KUEI CHO

Rockville, Maryland
July 1980

Acknowledgment

The following acknowledgment is reprinted from **American National Standard Programming Language COBOL, X3.23-1974,** published by the American National Standards Institute, Inc.

COBOL is an industry language and is not the property of any company or group of companies, or of any organization or group of organizations. No warranty, expressed or implied, is made by any contributor or by the CODASYL Programming Language Committee as to the accuracy and functioning of the programming system and language. Moreover, no responsibility is assumed by any contributor, or by the committee, in connection herewith. The authors and copyright holders of the copyrighted material used herein

FLOW-MATIC (trademark of Sperry Rand Corporation), Programming for the UNIVAC^{R} I and II, Data Automation Systems copyrighted 1958, 1959, by Sperry Rand Corporation; IBM Commercial Translator Form No. F 28-8013, copyrighted 1959 by IBM; FACT, DSI 27A5260-2760, copyrighted 1960 by Minneapolis-Honeywell

have specifically authorized the use of this material in whole or in part, in the COBOL specifications. Such authorization extends to the reproduction and use of COBOL specifications in programming manuals or similar publications.

How to Use This Book

As this book is a self-contained treatment intended for readers of different backgrounds, a guide to the readers is offered.

A. For scientific application readers:

a. If you have statistical and quality control background, then read:

Chapters 1, 4, 7, 8, 9, 10, 11, 12, 13

b. If you have statistical background only, then read:

Chapters, 1, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13

c. If you have no statistical and quality control background, then read:

Chapters 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13

B. For commercial and data processing application readers:

a. If you have statistical and quality control background, then read:

Chapters 1, 4, 7, 8, 9, 11, 12, 14

b. If you have statistical background only, then read:

Chapters 1, 4, 5, 6, 7, 8, 9, 11, 12, 14

c. If you have no statistical and quality control background, then read:

Chapters 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 14

An Introduction to Software Quality Control

Contents