Handbook of Software Reliability Engineering

Handbook of Software Reliability Engineering

Michael R. Lyu Editor in Chief

IEEE COMPUTER SOCIETY PRESS
Los Alamitos, California
Washington Brussels Tokyo

McGraw-Hill
New York San Francisco Washington, D.C. Auckland Bogota
Caracas Lisbon London Madrid Mexico City Milan Montreal New
Delhi San Juan Singapore Sydney Tokyo Toronto

Library of Congress Cataloging-in-Publication Data

Handbook of software reliability engineering / Michael R. Lyu, editor in chief.
p. cm. Includes index. ISBN
0-07-039400-8 (alk. paper)
1. Computer software—Reliability—Handbooks, manuals, etc. I. Lyu,
Michael R. QA76.76.R44H36 1995
005.1 —dc20 95-46468 CIP

McGraw-Hill

A Division of The McGraw-Hill Companies

1234567890 BKP/BKP 9009876

P/N 039401-6
PART OF
ISBN 0-07-039400-8

The sponsoring editor for this book was Marjorie Spencer, the editing supervisor was Christine H. Furry, and the production supervisor was Suzanne W. B. Rapcavage. This book was set in Century Schoolbook by North Market Street Graphics.
Printed and bound by Quebecor / Book Press. This

book is printed on acid-free paper.

To my wife C. Felicia Lyu,
for her love, understanding, and
support throughout this project

Contents

Part 2 Practices and Experiences

Part 3 Emerging Techniques

Contributors

Sarah Brocklehurst *City University, London* (CHAP. 4)
Ram Chillarege *IBM Watson Research* (CHAP. 9)
Mary Donnelly *AT&T Bell Laboratories* (CHAP. 6)
Joanne Bechta Dugan *University of Virginia* (CHAP. 15)
Bill Everett *AT&T Bell Laboratories* (CHAP. 6)
William Farr *Naval Surface Warfare Center* (CHAP. 3)
Gene Fuoco *AT&T Bell Laboratories* (CHAP. 5)
Joseph R. Horgan *Bellcore* (CHAP. 13)
Nancy Irving *AT&T Bell Laboratories* (CHAP. 5)
Ravi K. Iyer *University of Illinois* (CHAP. 8)
Wendell Jones *BNR Incorporated* (CHAP. 11)
Bruce Juhlin *U S West* (CHAP. 5)
KaramaKanoun *LAAS-CNRS, Toulouse, France* (CHAPS. 2,10)
Nachimuthu Karunanithi *Bellcore* (CHAP. 17)
Taghi Khoshgoftaar *Florida Atlantic University* (CHAP. 12)
Diane Kropfl *AT&T Bell Laboratories* (CHAP. 5)
Jean-Claude Laprie *LAAS-CNRS, Toulouse. France* (CHAPS. 2,10)
Inhwan Lee *Tandem Computers, Inc.* (CHAP. 8)
BevLittlewood *City University, London* (CHAP. 4)
Michael R. Lyu *AT&T Bell Laboratories, Editor* (CHAPS. 1, 7, 16, APP. B)
Yashwant Malaiya *Colorado State University* (CHAP. 17)
Aditya P. Mathur *Purdue University* (CHAP. 13)
David McAllister *North Carolina State University* (CHAP. 14)
John Munson *University of Idaho* (CHAP. 12)
John Musa *AT&T Bell Laboratories* (CHAPS. 5, 6)
Alien Nikora *Jet Propulsion Laboratory* (CHAP. 7)
GeorgeStark *Mitre Corporation* (APP. A)

Robert Tausworthe *Jet Propulsion Laboratory* (CHAP. 16) **MIaden Vouk** *North Carolina State University* (CHAPS. 11, 14) **Geoff Wilson** *AT&T Bell Laboratories* (CHAP. 6)

Foreword

Alfred V.Aho
Columbia University

In complex software systems, reliability is the most important aspect of software quality, but one that has often been the most elusive to achieve. Since more and more of the world's activities and systems are dependent on software, achieving the appropriate level of software reliability consistently and economically is crucial. Software failures make newspaper headlines because at best they inconvenience people and in extreme cases kill them.

It is refreshing to see a book that has the potential to make a significant improvement to software reliability. The *Handbook of Software Reliability Engineering* is an important milestone in the history of software reliability engineering. Michael R. Lyu has assembled a team of leading experts to document the best current practices in the field. The coverage is comprehensive, including material on fault prevention, fault removal, fault tolerance, and failure forecasting. Theory, models, metrics, measurements, processes, analysis, and estimation techniques are presented. The book is filled with proven methods, illustrative examples, and representative test results from working systems in the field. An important component of the book is a set of reliability tools that can be used to apply the techniques presented.

The subject is treated with the rigor that is characteristic of a mature engineering discipline. The book stresses mathematical models for evaluating reliability trade-offs, and shows how these models can be applied to the development of software systems.

With the publication of this Handbook, the field of software reliability engineering has come of age. This book is must reading for all software engineers concerned with software reliability.

Alfred V. Aho

Foreword

Richard A. DeMillo
Purdue University and Bellcore

Early in this exhaustive treatment of what may be the single most critical aspect of modern software development, the editor says "Mature engineering fields classify and organize proven solutions in handbooks so that most engineers can consistently handle complicated but routine designs." The reliability engineering of software has become mature with the appearance of this Handbook.

In my graduate software engineering course, I motivate the importance of early test planning with reliability requirement setting examples. It is, in my experience an issue about which success or failure of major systems projects revolve. In the early 1980s I led the DOD's software testing and reliability analysis team for the final operational tests of the now-famous Patriot Missile System. The questions? What was the required system reliability? Was the operational test data consistent with these requirements? Not many people know how close Patriot came to being rejected as a viable weapons system—not because the system itself was bad, but rather because the reliability engineering was so flawed that developers could not determine how reliable it really was. This crisis could have been avoided had software reliability engineering practice been systematized and applied in the manner advocated by this Handbook.

Reliability theory and engineering statistics textbooks ignore software, for the most part. Software engineering textbooks generally ignore reliability theory. Classroom teachers of the subject are forced to the kind of anecdotal material mentioned above, perhaps augmented by special-purpose supplementary readings Even worse, software reliability theory has a reputation for facileness that has been encouraged by the many contributors who try to apply hardware reliability models mutatis mutandis to the very different (and more difficult) problems of software reliability.

So, when I was asked by the editor to review this Handbook, I agreed eagerly. On the one hand, a "real" handbook would be of inestimable help to practitioners, decision makers, teachers, and students. On the other hand, a spotty or imbalanced treatment would only make matters worse. I said I would offer my comments only after reading the entire book.

The first thing I did when I received the manuscript was to check it against my classroom "staples." There for the first time in book form was a coherent approach to developing reliability requirements. There also was a discussion of the relationship between software test and reliability estimation, the impact of software architecture on reliability, error studies and software fault classification, tools and methods extracted from best-practice benchmarks of the best reliability labs in the world, actual data. It was all there—and in pretty much the same form in which I would have presented it myself. The editor even included exercises to make it suitable for classroom use.

Encouraged, I read the manuscript front to back. This is a book that will be the standard by which the field is measured for years to come. It is thorough, correct, readable, and so current that it actually anticipates results that have not appeared in archival journals yet. It contains the best work of many of the founders of the field. It contains innovations by some of the rising stars. It is, however, more than anything else a Handbook in the tradition of the classic handbooks of mathematics, physics, and engineering. It does not present software reliability as a silver bullet. It does not attempt to proscribe the complex system usages that would require skill and training on the part of software developers. Rather it seeks to ". . . classify and organize proven solutions ... so that most engineers can consistently handle complicated but routine designs." In this it succeeds, far beyond my expectations. It clearly establishes software reliability engineering as a mature engineering discipline.

Richard A. DeMillo

Preface

Ever since I entered the field of software reliability engineering some years ago, I have been looking for a book that exclusively and comprehensively deals with software reliability subjects that interest me, as both a researcher and a practitioner. I wasn't able to find one. So I started this project by inviting the leading experts in this field to contribute chapters for this book. I laid out the framework of the book, identified its essential components, and integrated them by maintaining completeness and avoiding redundancies. As an editor, my duty is to ensure breadth, while the chapter authors treat the subjects of their delegated chapters in depth.

This is a handbook on software reliability engineering. The theme underlying the book is the formulation, application, and evaluation of software reliability engineering techniques in practice. Reliability is obviously related to many characteristics of the software product and development process. This *Handbook* intends to address all its aspects in a quantitative way.

The book is designed for practitioners or researchers at all levels of competency, from novice to expert. It is targeted for several large, general groups of people who need information on software reliability engineering. They include:

1. People who need a general understanding of software reliability. These are high-level managers, professional engineers who use software or whose designs interface with software, and people who acquire, purchase, lease, or use software.
2. Software developers, testers, and quality assurance personnel who use and apply software reliability engineering techniques. This also includes practitioners in related disciplines such as system engineering, reliability management, risk analysis, management-decision sciences, and software maintenance.

3. Researchers and students in software engineering, reliability analysis, applied statistics, operations research, and related disciplines, and anyone who wants a deeper understanding of software reliability and its engineering techniques.

Each of the book's individual topics (i.e., chapters) could be considered as a compact, self-contained minibook. However, these topics are presented in relation to the basic principles and practices of software reliability engineering. The approach is to provide framework and a set of techniques for evaluating and improving the engineering of software reliability. It presents specific solutions, obtained mostly from real-world projects and experimental studies, for routine applications. It further highlights promising emerging techniques for research and exploration opportunities.

The book has been thoroughly indexed for your convenience, so that it can serve as a true handbook, and a comprehensive list of references is provided for the purpose of literature search. As a unique value-added feature, this book includes a CD-ROM, which contains 40 published and unpublished software project failure data sets and some of the most advanced software reliability tools for ready application of software reliability techniques and a jump-start on software reliability engineering programs.

This book is also designed to be used as a textbook by students of software engineering or system reliability, either in a classroom or for self-study. Examples, case studies, and problems have been provided throughout the book to illustrate the concepts and to walk through the techniques. A *Solution Manual* is available from the editor with solutions to some of the exercises.

What is finally presented here is the work of celebrated international experts contributing their most advanced knowledge and practices on specific reliability-related topics. The development team of this book wants to thank our colleagues who provided continuous encouragement and thorough review of the chapters of the book. They are Jean Arlat, Phillip Babcock, Farokh B. Bastani, Brian Beckman, Justin Biddle, James Bieman, Harry S. Burns, Sid Dalal, Chris Dale, Adrian Dolinsky, George Finelli, Amrit Goel, Jack Goldberg, Myron Hecht, Walter Heimerdinger, Yu-Yun Ho, Yennun Huang, Robert Jackson, Mohamed Kaaniche, Kalai Kalaichelvan, Rick Karcich, Ted Keller, Elaine Keramidas, Chandra Kintala, Sy-Yen Kuo, Ming Y. Lai, Alice Lee, Haim Levendel, Yi-Bing Lin, Peng Lu, Richard E. Machol, Suku Nair, Mits Ohba, Gardner Patton, Hoang Pham, Francesca Saglietti, Norm Schneidewind, Robert Sherman, David Siefert, Pradip Srimani, Mark Sullivan, Robert Swarz, K.C. Tai, Yoshi Tohma, Randy Van Buren, C.W. Vowell, Anneliese von Mayrhauser, Chris J. Walter, Yi-Ming Wang, Pramod Warty, Chuck Weinstock, Min Xie, and Jinsong Yu.

We are most appreciative of the organizations and projects that provided funding for the work conducted in some of the book chapters. They are the Advanced Research Projects Agency, the ESPRIT Basic Research Action on Predictably Dependable Computing Systems, the ESPRIT programme as part of the PDCS1 and PDCS2 projects, the EU Environment programme as part of the SHIP project, IBM at Pough-keepsie, New York, the Illinois Computer Laboratory for Aerospace Systems and Software (ICLASS), National Aeronautics and Space Administration (NASA), NASA AMES Research Center, Office of Naval Research, Tandem Computers Incorporated, the U.K. EPSRC as part of the DATUM project, and the U.S. Air Force Operational Test and Evaluation Center (AFOTEC).

I also want to particularly thank Al Aho and Rich DeMillo for writing forewords to this book. Their comments are helpful and rewarding. I am greatly thankful to Karen Newcomb of NASA COSMIC and Lilliam Valdez-Diaz of AT&T for permission to include CASRE, SoftRel, and AT&T SRE Toolkit in this book. My appreciation goes to Jean Glasser, Marjorie Spencer, John Wyzaiek, and Suzanne Rapcavage, editing and production supervisors at McGraw-Hill during different stages of this book. Midge Haramis's assistance is also acknowledged. The invaluable guidance and help of Christine Furry at North Market Street Graphics during many revision, editing, and production cycles have also made this book project much easier than it would have been.

Finally, I want to thank my wife Felicia, to whom this book is dedicated.

Michael R. Lyu Murray Hill, New Jersey