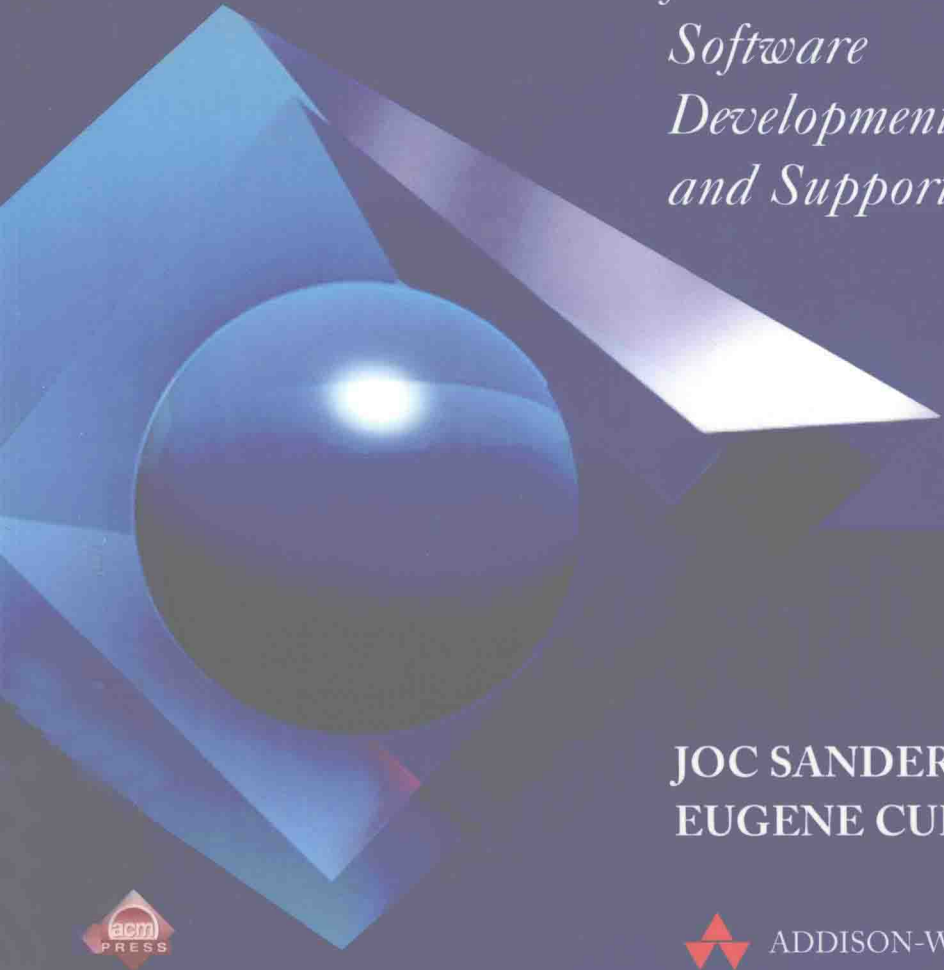
Software Quality

A Framework for Success in Software Development and Support



JOC SANDERS
EUGENE CURRAN

acm PRESS

ADDISON-WESLEY

Software Quality

A Framework for Success in Software Development and Support

JOC SANDERS

EUGENE CURRAN

Centre for Software Engineering, Dublin

Addison-Wesley

Harlow, England • Reading, Massachusetts • Menlo Park, California
New York • Don Mills, Ontario • Amsterdam • Bonn • Sydney • Singapore
Tokyo • Madrid • San Juan • Milan • Mexico City • Seoul • Taipei

Addison Wesley Longman Limited
Edinburgh Gate
Harlow
Essex CM20 2JE
United Kingdom

Cover designed by Arthur op den Brouw, Reading
Printed in Malaysia, PP

First printed in 1994. Reprinted 1995, 1997 and 1998

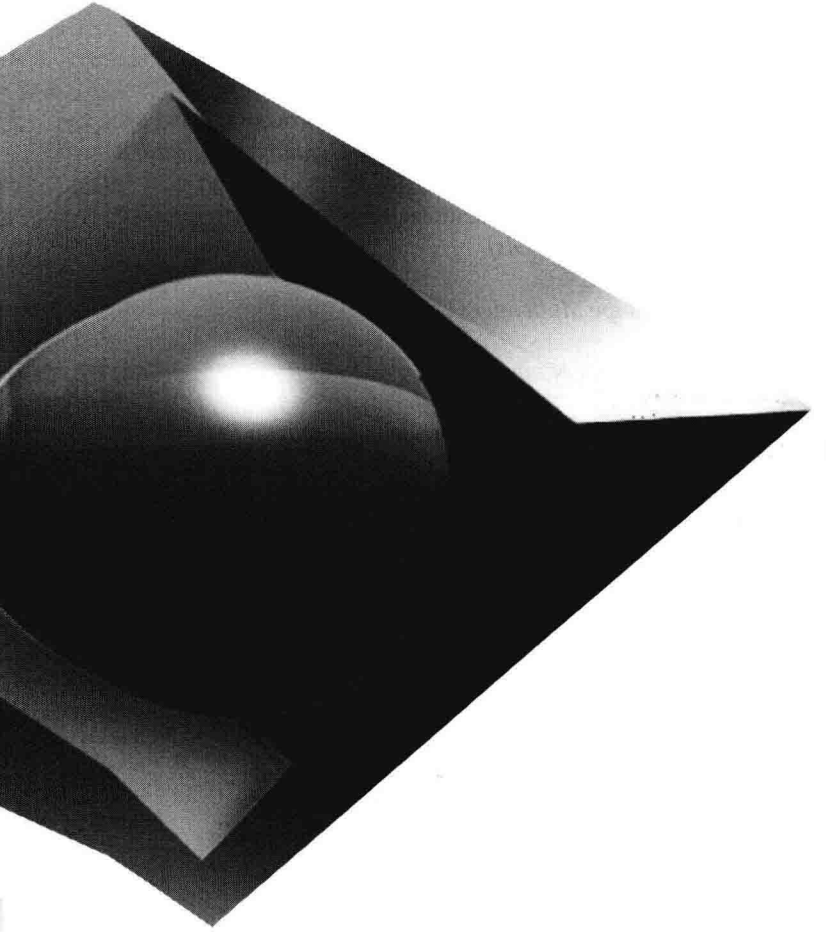
ISBN 0–201–63198–9

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library.

Library of Congress Cataloguing-in-Publication Data is available

Software Quality

A Framework for Success in Software Development and Support



ACM PRESS BOOKS

Editor-in-Chief	**Peter Wegner**	Brown University
International Editor (Europe)	**Dines Bjørner**	Technical University of Denmark

SELECTED TITLES

Object-Oriented Reuse, Concurrency and Distribution
Colin Atkinson

Advances in Database Programming Languages *Francois Bançilhon and Peter Buneman (Eds)*

Algebraic Specification *J.A. Bergstra, J. Heering and P. Klint (Eds)*

Software Reusability (Volume 1: Concepts and Models)
Ted Biggerstaff and Alan Perlis (Eds)

Software Reusability (Volume 2: Applications and Experience)
Ted Biggerstaff and Alan Perlis (Eds)

Object-Oriented Software Engineering: A Use Case Driven Approach
Ivar Jacobson, Magnus Christerson, Patrik Jonnson and Gunnar Övergaard

Object-Oriented Concepts, Databases and Applications
Won Kim and Frederick H. Lochovsky (Eds)

Distributed Systems (2nd edn) *Sape Mullender (Ed)*

Computing: A Human Activity *Peter Naur*

The Oberon System: User Guide and Programmer's Manual *Martin Reiser*

Programming in Oberon: Steps Beyond Pascal and Modula *Martin Reiser and Niklaus Wirth*

The Programmer's Apprentice *Charles Rich and Richard C. Waters*

User Interface Design *Harold Thimbleby*

Project Oberon: The Design of an Operating System and Compiler *Niklaus Wirth and Jürg Gutknecht*

Foreword

Led by theorists such as Shewart, Deming and Juran, practitioners at Toyota, Honda, Xerox, Ford and many other firms around the world have changed our expectations for quality – both as consumers and producers. As consumers we have come to regard defects as unacceptable. We expect products to work 'out of the box' and to give long and reliable service. We also expect them to anticipate and meet our real needs, to be easy to use, and to be flexible for change and modification (as in modular stereo, camera and computer systems). Products that do not measure up to these standards are considered 'second class' or 'cut-rate' and can only demand cut-rate prices. They will not be considered competitive in a global marketplace.

As producers, our relation to quality has also changed. Our old theory was that defects are inevitable, and that a certain number should be allowed to get to the customer in order to maximize profitability! We now realise that there is no inherent limit to the level of quality that can be achieved. However, we have also learned that the secret to continuously increasing quality is not more stringent testing but a continuously improving quality process. In other words, the new rule is 'build quality in' instead of 'test defects out'. But quality is a challenge that requires a commitment from the entire firm, and a detailed plan of action. Fortunately, much has been learned over the last 30 years about how to build a 'quality system' across a wide variety of businesses.

Not too surprisingly, the lessons learned initially in the manufacturing sector, and which are now being applied to many service and white collar activities, are only beginning to be applied to software development. We have a long history of resisting the analysis and redesign of our own processes, even as we analyze and automate the processes of our brethren in other professions. We still accept defects as part and parcel of the product. Rather than issuing a warranty that our products are defect-free, we include a disclaimer disavowing ourselves from responsibility. Surely something is wrong here.

Fortunately, the European Community is actively encouraging the institutionalization of software quality by promoting the ISO 9000 family of standards, which describes what is required of a satisfactory quality system. ISO 9000 is widely recognized in Europe and is rapidly gaining acceptance in the US and many other parts of the world. It promises to become the global standard by which the government, industrial and private consumers can judge the fitness of suppliers. It can also serve as the blueprint by which suppliers can achieve the quality level they desire. Happily, ISO 9000 contains components specifically targeted for software development, and these are the topic of *Software Quality: A Framework for Success in Software Development and Support*.

In this clear, concise and highly-informative text, Joc Sanders and Eugene Curran provide a 'nuts-and-bolts' guide for managers who might ask the question, 'Why bother with ISO 9000 anyway?' Starting with a bottom-line business rationale, *Framework for Success* explains why quality will be a key competitive factor in the coming years – not just in Europe, but around the world.

They map out the definition of quality in operational terms, explain the many business benefits it can produce, and lay the responsibility for achieving it squarely at the feet of top management. Fortunately for wary managers, Sanders and Curran don't stop there. *Framework for Success* also explains how to implement a Quality System throughout the company, from assessing the organization, right through establishing procedures, support activities, certification and training. Besides explaining how each of these topics affect software quality, the book provides a useful summary of ISO 9000 itself, and relates ISO 9000 to essential quality characteristics and good engineering principles.

It's time for the software industry to come out of the 'middle ages' and establish industrial-strength processes every bit as robust, reliable and economical as those now in use in manufacturing and service sectors. *Software Quality: A Framework for Success in Software Development and Support* is an excellent place to start understanding how you can employ ISO 9000 to reach your quality goals, and to achieve certification under an international standard at the same time.

Gene Forte
Executive Editor
CASE OUTLOOK

Preface

Why this book is the way it is

This book does not pretend to be a comprehensive, academic text, telling you all that you could ever want to know about software quality, and more. If it did, it would be much bigger, and you would probably find it difficult to sift the important grain from the detailed chaff! Instead, we have written a *guidebook* with the clear objective of meeting the *practical* needs of busy software engineers and their managers, working in real-world businesses. We believe these needs are for a concise, readable book, with separate parts addressed to two audiences:

Part 1 is a Manager's Guide to Implementing a Quality System. It gives high-level guidance on why and how to implement a Quality System in a software organization. It also explains how to seek certification for it, and gives up-to-date information about the ISO 9000 international standards for quality systems and alternatives to them. It is for managers who are planning or initiating a software quality programme, but it will also be of interest to practising software engineers wanting an introduction to software quality management.

Part 2 is a Software Engineer's Guide to Best Practice. It presents a guide to current best practice, drawn from a variety of current international standards on software engineering, and also addresses software support services. It can be used as a checklist during a software quality programme, both to assess actual practice and to define the objectives of proposed new practices.

We have considered the needs of large, medium and small organizations, without compromising the requirements of the international quality standards applicable to software, namely ISO 9001, ISO 9000-3, and ISO 9004-2. And we believe the guidance we have given is relevant to all types of software organizations, from commercial software houses, to management information systems departments in user companies, to the developers of real-time and embedded systems.

We hope that many companies will be able to use our book as a basis from which to develop their own quality system by themselves. Others may need expert advice and assistance as well. Our book may not contain all the answers you need, but it will set you off in the right direction.

The underlying philosophy

We have found it useful when starting to read a book, or when deciding whether to buy it, to have some idea of the authors' underlying approach and philosophy. It helps the reader to evaluate what is likely to be relevant to him or her. So we feel it is proper for us to declare some of the propositions and beliefs which we have arrived at over the years, as working software managers and engineers ourselves. We summarize them as follows:

- Quality is the key to success in the software business, as it is in every other.
- The cheapest way to improve software productivity is to improve software quality.
- The quality of software support is as important as the quality of the software product: the support environment must be engineered as carefully as the software itself.
- To achieve software quality, people and culture are as important as technology – if not more so.
- The only way to improve software quality reliably is to improve the software process (which includes personnel, facilities, equipment, technology and methodology).
- Process improvement is usually unsuccessful unless top management demonstrate genuine commitment and leadership.
- Quality and process improvement are an unrelenting endeavour: it is always possible to do it a little better, a little faster, a little cheaper.
- An ISO 9000-compliant quality system is a good early target for many software organizations, but not for all.
- A software organization's quality system must be tailored to its specific needs and circumstances or it will not be both effective and efficient.
- An effective software quality system uses good software engineering practices, based on the following principles:

  Quality principles
  - Try to prevent defects from being introduced in the first place
  - Ensure defects that get in are detected and corrected as early as

possible
- Establish and eliminate the causes as well as the symptoms of defects
- Independently audit work for compliance with standards and procedures

Management principles
- Define roles and responsibilities
- Plan the work
- Track progress against the plans and take corrective action where necessary
- Progressively refine the plans

Engineering principles
- Analyze the problem before developing the solution
- Break complex problems into several less complex ones
- Ensure the subproblems knit together by controlling their relationships.

Acknowledgements

The first draft of this book was written as part of an Irish national initiative, co-ordinated by the National Software Directorate, with support from the European Community Structural Funds. Without their support, and the support of the Centre for Software Engineering for which we both work, this book would not have been possible.

Our thanks are due to the very many individuals, who have helped us meet our objective by reviewing early drafts and providing a wealth of valuable comments. They are too numerous to mention by name, but include many colleagues in the Irish software industry, and the staff of Addison-Wesley, our publishers. Such merits as this book displays are in large part due to them, while its defects are of course our own!

Lastly, we are very conscious of the enormous debt that we each owe to those who have gone before us. Some of these are acknowledged in the bibliography in Appendix B, but we recall here also all those friends, teachers and colleagues from whom we have learned everything that we know.

Joc Sanders
Eugene Curran
Centre for Software Engineering
Dublin, Ireland

Contents

Part 1

Manager's Guide to Implementing a Quality System

Part 1 of this book gives guidance for managers of software organizations on why and how to implement a quality system, and how to seek certification for it.

Chapter 1 defines what we mean by software quality, and what a quality system is. Chapter 2 deals with the human and cultural issues which are so vital for success. Chapter 3 explains how to implement a quality system, using a straight-forward five-step model. Chapter 4 discusses how to demonstrate the effectiveness of the quality system either by obtaining ISO 9000 certification, or using other sources.

1
Defining Software Quality

1.1 Why Bother With Quality?

Why bother with quality? Because quality is critical for survival and success. The market for software is increasingly a global one, and your organization will not succeed in that market unless you produce, and are seen to produce, quality products and services. If you do not do so, your organization may not even survive. This is the first message of the chapter. It applies as much to software development and support as to any other product or service.

The word 'quality' means different things to different people. We give a formal definition later and discuss its implications, but in essence, quality means satisfying customers. A happy customer will do repeat business.

There are several reasons why you should be concerned with quality:

- quality is now a competitive issue;
- quality is essential for survival;
- quality is essential for international marketing;
- quality is cost-effective;
- quality retains customers and increases profits;
- quality is the hallmark of world-class businesses.

1.1.1 Quality is Now a Competitive Issue

Software used to be a technical business, in which functionality was the key determinant of success. Today, you can no longer rely on the functionality of your products to win the day. Your competitors can match your functionality relatively quickly and easily. The only way to differentiate your product from those of your competitors, beyond the short term, is by its quality, and the quality of support that goes with it.

As the software market matures, customers want to be assured of quality. They no longer accept your claims at face value, but expect you to demonstrate quality. Certification to international quality

standards is becoming a prerequisite for getting business; not to have certification will become a competitive disadvantage.

This applies to in-house information systems departments as well as to commercial software organizations. Internal customers also want quality assurance, and question increasingly whether work should be carried out in-house or outsourced by external suppliers.

1.1.2 Quality is Essential for Survival

Customers are demanding demonstrable quality. If you cannot deliver it, your ability to survive in a highly competitive and rapidly changing market is in doubt. More and more large organizations are deciding to reduce the number of suppliers they use, often by as much as 90%. In the drive to improve their own quality, they want to work closely with their chosen suppliers, whom they treat as business partners. They often use quality certification as a way of selecting suppliers.

1.1.3 Quality is Essential for International Marketing

The market for software is rapidly becoming global. The ability to demonstrate quality gives even a small company credibility to enter an export market.

The Single European Market came into being on 1 January 1993, and the European Economic Area Agreement (EEA) became effective in early 1993. The EEA countries (the European Community and seven EFTA states – Austria, Finland, Iceland, Liechtenstein, Norway, Sweden and Switzerland) represent the world's largest trading bloc, accounting for 40% of world trade, 30% of world production and 380m citizens. Government and multinational buyers in Europe are already using certification to internationally recognized standards as a criterion for shortlisting suppliers. Similar developments are taking place in other world regions – for instance the North American Free Trade Area (NAFTA).

It works both ways – your home market is vulnerable to quality foreign imports unless you can compete on quality.

1.1.4 Quality is Cost-effective

An effective quality system leads to increased productivity and permanently reduced costs, because it enables management to reduce defect correction costs by emphasizing prevention.

Everyone in the software industry knows that the cost of correcting defects in software late in development can be orders of