

Software Quality Engineering: A Total Technical and Management Approach

MICHAEL S. DEUTSCH and RONALD R. WILLIS

Prentice Hall, Englewood Cliffs, NJ, 1988. 317pp.

ISBN 0 13 823204 0

In all areas of business, customers are becoming more discerning, and consequently more demanding, on the issue of quality. For some products a bad component could lead to loss of life or serious injury, whilst at the other end of the scale the results would be inconvenience or annoyance. The same range of consequences relates to software production, where there is increasing emphasis on quality control.

The software development life-cycle covers all stages from system specification to delivery of the finished product, with the associated support and maintenance. It is recognized that it is no longer sufficient to leave 'quality' to a quality assurance function that checks products immediately prior to delivery. Quality attributes need to be built in at every stage of the development process. This is the thrust of this comprehensive book, which aims to fill in the gaps in the procedures and skills required to develop high-quality software systems.

The authors 'treat quality from a total technical and management viewpoint', and they go on to define software quality engineering as the 'marriage of modern software engineering technology and the traditional rigour of quality assurance process methods'.

The structure of the book reflects this conjunction, with the major sections covering: specifying and achieving quality; verification and testing. Every step in the software development cycle is addressed, with detailed descriptions of how quality can be improved.

Good quality is not left as a vague desirable goal, to be achieved if possible. It is rigorously defined, so that the purchaser of a software system can weigh up functional, performance and quality requirements against cost. Four levels of quality are identified together, at each stage, with the procedures that have to be gone through and the standards to be achieved to attain each level.

The authors, for whom unfortunately we are supplied with no biographical details, are clearly on top of their subject. There are many references to other workers in the field of software engineering, together with their approaches and methodologies. However, I cannot say I found it an easy book to read. I would normally welcome lists, tables and diagrams, but here we are swamped with them. The authors seem to have fallen between two stools—it is neither a descriptive text nor a reference book, although it comes much closer to being the latter.

The strengths of the book lie in the detailed way in which the stages of the software development cycles are described. For anyone wishing to learn about software development and how good quality can be achieved, the book would serve as a valuable reference. Not all aspects are comprehensively covered; integration and system testing for large systems could have been given more detailed attention. The section on 'Management Aspects of Software Quality' similarly lacks the rigour of the chapters dealing with the nitty gritty of software engineering. In particular, the cost models were difficult to follow.

The authors identify four categories of potential readership: software engineers, software managers, system engineers and procuring agencies. The concepts and practical aspects of quality software engineering are certainly required to be known and understood by the first three groups. For them the book would be a valuable reference tome, although I feel it would serve better as the background material to a training course built around its subject structure. As for procurement agencies, the 'shopping lists' of quality factors and attributes are certainly comprehensive, and the book itself would give them an understanding of the role of quality in the development process.

All in all, this is a book full of detailed and practical suggestions for introducing quality into software engineering. It will serve best as a reference book for all those involved in specifying, developing and acquiring large software systems.

MICHAEL J. NORMAN