A superficial book.

"Techniques of Program Structure and Design.", Edward Yourdon, Prentice Hall. 1975

It is not my habit to comment in written form on all the junk and trash that meets my eyes, but sometimes I must. Edward Yourdon's book presents such a case, because the book is not just superficial, it is dangerously so. It is a misleading text, against which potential readers should be warned. The fact that the book is poorly written is something the uninitiated but careful reader can discover for himself; the uninitiated reader, however, can not discover from this book alone that, measured against today's state of the art, it is hopelessly old-fashioned, and betrays a singular lack of insight in the subject matter it claims to cover.

It is poorly written: even the English of the title is faulty! "Techniques of Design" is a proper construct, but "Techniques of Program Structure" is just a meaningless concatenation of words. Another example of poor English do we find on page 247 "...until it has been demonstrated to everyone's satisfaction that the corrected bug does work correctly.". If it "does not work correctly" --in the author's parlance-- how can we then refer to "the corrected bug"? Furthermore I find it somewhat strange that after its correction --i.e. its removal!-- the bug is still supposed to be there and to be "doing" something. Presumably the author should have written "...until it has been demonstrated to everyone's satisfaction that the bug has been successfully removed.". But even after that repair, the sentence is not very helpful, for what is meant by "everyone's satisfaction"? My guess is that from the unspecified environment the author refers to, I, for instance, should be excluded. The quotation is typical for the author's approach: in spite of the fact that the title mentions "techniques" the author skirts technical issues completely, and seems to measure quality primarily indirectly by the degree of "satisfaction" given to the environment. The book is misleading in that it suggests that such a nontechnical, sociological approach suffices.

The next quotation from page 246 is, I am afraid, fairly indicative for the book's quality:

"7.1.7 Glitch

The word "glitch" is one of those unofficial terms that has gained popularity in the vocabulary of the industrial/academic computer communities of Cambridge and Berkely. Though never defined precisely, a glitch is usually thought of as an unforeseen feature of a computer program (or, in most cases, the specification and the design of the program) that renders it inefficient, unelegant, or awkward to use. Note that there is a difference between a bug and a glitch. A bug implies that the program is not working to specifications, while a glitch implies that the specification itself (or some property of the program outside the scope of the specification) is awkward and clumsy."

I am --I am happy to say-- unfamiliar with the "industrial/academic computer communities" alluded to, that are here accused of such sloppy use of the term "glitch", a precisely defined term which in that meaning is officially reserved for a very well-known phenomenon that is to be expected to occur when synchronizing independently clocked systems. (Stronger: neither in Berkely, nor in Cambridge, USA, have I ever heard the term used by people in any other than its technical meaning.) But suppose for a moment that the author were not wrong in this paragraph: does the paragraph then carry any helpful information? I could hardly find any....

An example of hilarious nonsense can be found on page 281 where "it seems that the number [...] of bugs increases exponentially with the number of instructions in

the module." This tells us that by making a module large enough we can achieve that the number of bugs exceeds the number of instructions in the module, nay, even exceeds the number of bits needed to store these instructions! The frightening thing about this piece of nonsense, however, is that, in spite of an earlier chapter on "Modular Programming" --of more than 40 pages-- the author still thinks that he can meaningfully speal of "the number of bugs in a module". If he had used that earlier chapter to develop a meaningful concept "module", he would have discovered that the "number of bugs in a single module" is either zero or one, i.e. the module is either correct or not. In the latter case, couting their number is as meaningless as counting the number of misspellings in the names of the well-known authors in computing science "Ivan Yourdon" or "Niklaus Hoare".

One final quotation --page 9-- "Of course, it is important (sic!) to point out that comments are not an end unto themselves." This sentence may give us a glimpse of the intended readership: those who still think that comments are an end unto themselves.

* * *

We shall not blame the author for the publisher's blurb on the jacket that describes this book as "A clearly written, logically organized, and easy-to-understand discussion....". But I am amazed and alarmed, when I see this book recommended in a Prentice-Hall folder as follows.

"I certainly enjoyed reading this excellent treatise. The author is obviously an authority on the subject but so are many others. However, the rare quality of his exposition is a synergistic combination of humor, elegance of style, breadth of coverage, diversity of programming languages and applications, organization, and lucidity. What more can I say? [...]"
Carl Hammer, Ph.D., Director, Computer Sciences, Univac, Adjunct Professor, The American University, Washington, D.C.

and

"Best I've seen on the topic."
Professor H.R.Knitter, University of Wisconsin, Madison.

Publishers are supposed to be willing to print anything they think they can sell --I don't think that that is the full truth, but they are supposed to be that way. Professors at our Universities are supposed to be able not to mistake a totally unscientific work for something worth their recommendation, but, apparently, that is not the full truth either.....

Plataanstraat 5
NL-4565 NUENEN
The Netherlands

prof.dr.Edsger W.Dijkstra
Burroughs Research Fellow