



Review of "Software Engineering With Java"

Reviewed by [Adrian P. O'Riordan](#)

Software Engineering With Java by Stephen Schach ISBN: 0256241678 Published: by Richard Irwin in 1997

Writing a book on software engineering has become an onerous task. Recent years have seen a proliferation of techniques and methods, as well as further developments and refinements of established practices. One of Stephen Schach's previous books, *Classical and Object-Oriented Software Engineering*, aimed to strike a balance between the two main *modus operandi* mentioned in the title: the tried and tested structured methods familiar to thousands of software engineers and the more modern object-oriented (OO) methods which have blossomed recently. Any example code is given in a subset of C++.

Software Engineering With Java, is a Java version of the above-mentioned book. The Java angle adds to the book. The bulk of the book is largely language independent, with an epilogue demonstrating how Java embodies the principles of software engineering. The text is aimed at senior undergraduates or first year graduates. The book is also suitable for practitioners who want to keep up to date with the latest developments. The language and style of presentation is clear and precise. The text is logically laid out and comprehensive.

He states in the preface that a textbook exclusively devoted to the object-oriented paradigm may be justified. Many such good books exist. Schach then goes on to reason that it is important to cover the structured methods because a) they are still in widespread use, and b) systems designed in that manner still need to be maintained. What isn't usually emphasised is that despite the paradigm shift associated with moving to object-oriented software engineering, there is also commonality between the

new and the old, and the book helps to make this explicit.

Schach achieves success at balancing the treatment of structured and OO methods, avoiding the trap of saying too little about a broad range of topics, instead concentrating on practical and pragmatic aspects of the software engineering process and development lifecycle; presenting material on the structured and object-oriented paradigm in a logical manner as parts in the software development lifecycle. All phases of the software process are covered: requirements, specification, analysis, planning, design, implementation, integration, and maintenance. Important topics such as software maintenance, which are often overlooked in introductory software engineering textbooks or only given a cursory treatment, are given the coverage justified by the economic costs of neglecting what is often perceived as a less interesting side of software production.

The chapter which introduces objects is especially good. It shows how the object-oriented paradigm can be viewed as a progressive development of the theory of modularity. Notions such as coupling, and cohesion, familiar to practitioners of structured methods, are given an OO slant in this discussion. There is a more conventional presentation of important ideas such as encapsulation, ADTs, information hiding, and dynamic binding.

At the end of each chapter there are a series of problems which are ideal either as student assignments or for home study. I particularly liked the problems that involve readings of the software engineering literature, allowing students to form opinions on many well-known and often discussed papers such as Brooks' "No Silver Bullet." A detailed case study is also included with an implementation in Java in the appendices.

Indicative of the style is the fact that material on issues such as software quality is presented from the beginning in an integrated manner. Strategies for improving software quality, such as the Capability Maturity Model and ISO 9000, are detailed as early as Chapter 3. Software testing also has its own chapter, which includes material on program correctness proofs. This is a topic frequently absent from software engineering textbooks. Formal methods, such as finite state machines and Z, are discussed as part of software specification.

Overall, I highly recommend this book for both those planning on doing research in the field of software engineering and aspiring and practicing software engineers.

Biography

Adrian P. O'Riordan (a.oriordan@cs.ucc.ie) is a Ph.D. student at University College Cork, Ireland. His interests include software engineering, information retrieval, and empirical AI.