<<面向对象软件工程>>

图书基本信息

书名：<<面向对象软件工程>>

13位ISBN编号：9787111265269

10位ISBN编号：7111265262

出版时间：2009-3

出版时间：机械工业出版社

作者：Stephen R.Schach

页数：558

<<面向对象软件工程>>

前言

The wheel has turned full circle.In 1988, I wrote a textbook entitled Software Engineering. Virtually the only mention of the object-oriented paradigm in that book was one section that described object-oriented design.By 1994, the object-oriented paradigm was starting to gain acceptance in the software industry, so I wrote a textbook called Classical and Object-Oriented Software Engineering. Six years later, however, the object-oriented paradigm had become more important than the classical paradigm. To reflect this change,I switched the order of the two topics in the title of the textbook I wrote in 2000,and called it Object-Oriented and Classical Software Engineering.Nowadays, use of the classical paradigm is largely restricted to maintaining legacy software. Students learn C++ or Java as their first programming language,and object-oriented languages are used in subsequent computer science and computer engineering courses. Students expect that, when they graduate, they will work for a company that uses the object-oriented paradigm. The object-oriented paradigm has all but squeezed out the classical paradigm. And that is why I have written a textbook entitled Object-Oriented Software Engineering.

<<面向对象软件工程>>

内容概要

本书从面向对象范型出发对软件工程进行重新演绎。
全面、系统、清晰地介绍了面向对象软件工程的基本概念、原理、方法和工具，通过实例说明了面向对象软件开发的整个过程。
　本书分为两个部分：第一部分介绍了面向对象软件工程的基本理论；第二部分以工作流的形式介绍了软件生命周期。

作者简介

（美）沙赫（Stephen R.Schach）1972年获魏兹曼科学院物理学理科硕士学位，1973年获开普敦大学应用数学博士学位。
目前任教于美国范德比尔特大学计算机科学系。
他著有多部有关软件工程、面向对象软件工程、面向对象系统分析与设计的教材。
他还在国际上广泛讲授软件工程方面的

<<面向对象软件工程>>

书籍目录

章节摘录

As explained in Section 3.13, without measurements （or metrics） it is impossible to dettect problems early in the software process, before they get out of hand. In this way, metrics can serve as an early warning system for potential problems. A wide variety of metrics can be used. For example, lines of code （LOC） is one way of measuring the size of a product （see Section 9.2.l）. If LOC measurements are taken at regular intervals, they provide a measure of how fast the project is progressing. In addition, the number of faults per 1000 lines of code is a measure of software quality. After all, it is of little use if a programmer consistently turns out 2000 lines of code a month but half of them have to be thrown away because they are unacceptable. Accordingly, LOC in isolation is not a very meaningful metric.Once the product has been installed on the client's computer, a metric such as mean time between failures provides management an indication of its reliability. If a certain product fails every other day, its quality is clearly lower than that of a similar product that on average runs for 9 months without a failure.Certain metrics can be applied throughout the software process. For example, for each workflow, we can measure the effort in person-months （I person-month is the-amount of work done by one person in 1 month）. Staff turnover is another important metric. High turnover adversely affects current projects because it takes time for a new employee to learn the relevant facts about the project （see Section 4.1）. In addition, new employees may have to be trained in aspects of the software process; if new employees are less educated in software engineering than the individuals they replace, then the process as a whole may suffer. Of course, cost is an essential metric that must also be monitored continually throughout the entire process.A number of different metrics are described in this book. Some are product metrics;they measure some aspect of the product itself, such as its size or its reliability. Others are process metrics used by the developers to deduce information about the software process. A typical metric of this kind is the efficiency of fault detection during development,that is, the ratio of the number of faults detected during development to the total number of faults detected in the product over its lifetime.

<<面向对象软件工程>>

编辑推荐

《面向对象软件工程 (英文版)》从面向对象范型出发对软件工程进行重新演绎.全面、系统、清晰地介绍了面向对象软件工程的基本概念、原理、方法和工具.通过实例说明了面向对象软件开发的整个过程。

《面向对象软件工程 (英文版)》分为两个部分：第一部分介绍了面向对象软件工程的基本理论；第二部分以工作流的形式介绍了软件生命周期。

包括面向对象生命周期模型、面向对象分析、面向对象设计。

以及面向对象软件的测试和维护。

讨论了文档、维护、复用、可移植性、测试和CASEI具等的重要性。

包括了能力成熟度模型（CMM）和人员能力成熟度模型（P · CMM）的内容。

与语言无关。

实例代码对于C++和Java语言背景的读者同样清晰。

包括600余篇当前热点研究文章、经典文献和书籍的参考文献。

包含2个用于说明完整软件生命周期的运行实例，还有7个较小的实例，分别用于突出说明特定的主题。

基于统一过程、Java和C++语言的完整源码可从作者网站（www.mhhe.com/schach）下载。

包括5种类型的习题。

分别是概念理解、项目分析、课程设计、论文研读和实例修改。

<<面向对象软件工程>>

版权说明