

<<软件测试>>

图书基本信息

书名：<<软件测试>>

13位ISBN编号：9787111388630

10位ISBN编号：7111388631

出版时间：2012-10-1

出版时间：机械工业出版社华章公司

作者：（爱尔兰）布朗 等著

页数：190

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## &lt;&lt;软件测试&gt;&gt;

## 内容概要

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自1998年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与pearson, mcgraw-hill, elsevier, mit, john wiley & sons, cengage等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出andrew s. tanenbaum, bjarne stroustrup, brain w. kernighan, dennis ritche, jim gray, afred v. aho, john e. hopcroft, jeffrey d. ullman, abraham silberschatz, william stallings, donald e. knuth, john l. hennessy, larry l. peterson等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

## 作者简介

作者：（爱尔兰）布朗（Stephen Brown）（爱尔兰）Joe Timoney（爱尔兰）Tom Lysaght 叶德仕（Deshi Ye）

## &lt;&lt;软件测试&gt;&gt;

## 书籍目录

## 前言

## 1 introduction

- 1 . 1 the software industry
- 1 . 2 software testing and quality
- 1 . 3 errors . faults and failures
  - 1 . 3 . 1 software faults
  - 1 . 3 . 2 software failures
  - 1 . 3 . 3 need for testing
- 1 . 4 the role of sdecifications
- 1 . 5 overview of testing
  - 1 . 5 . 1 testing in the development process
  - 1 . 5 . 2 test automation
- 1 . 6 the theory of testing
  - 1 . 6 . 1 exhaustive testing example
  - 1 . 6 . 2 implications
- 1 . 7 when to finish testing
- 1 . 8 notes on book structure

## 2 principles of software testing

- .2 . 1 dynamic and static verification
  - 2 . 1 . 1 static verification
  - 2 . 1 . 2 dynamic verification
- 2 . 2 black-box and white . box testing
  - 2 . 2 . 1 errors of “ omission ” and “ commission ”
- 2 . 3 test approaches
  - 2 . 3 . 1 black-box testing
  - 2 . 3 . 2 white-box testing
  - 2 . 3 . 3 fault insertion
- 2 . 4 test activities
  - 2 . 4 . 1 analysis outputs
  - 2 . 4 . 2 test cases
  - 2 . 4 . 3 test data
  - 2 . 4 . 4 test code for test proceduresl
- 2 . 5 analysis of software specifications
  - 2 . 5 . 1 parameters
  - 2 . 5 . 2 parameter ranges
  - 2 . 5 . 3 equivalence partitions
  - 2 . 5 . 4 boundary values
  - 2 . 5 . 5 combinations of values
  - 2 . 5 . 6 sequences of values
- 2 . 6 analysis of software components
  - 2 . 6 . 1 control flow graphs
  - 2 . 6 . 2 decisions and conditions
  - 2 . 6 . 3 paths
  - 2 . 6 . 4 data-flow testing
  - 2 . 6 . 5 ranking

## &lt;&lt;软件测试&gt;&gt;

- 2 . 7 analysis of targets for fault insertion
  - 2 . 7 . 1 offutt ' s 5 sufficient mutations
- 2 . 8 test artefacts
- 3 unit testing
  - 3 . 1 introduction
  - 3 . 2 usage
  - 3 . 3 black-box techniques
    - 3 . 3 . 1 equivalence partitioning(ep)
    - 3 . 3 . 2 boundary value analysis fbval
    - 3 . 3 . 3 testing combinations of inputs(ci)
    - 3 . 3 . 4 testing sequences of inputs(si)or state
    - 3 . 3 . 5 random input data fridl
    - 3 . 3 . 6 error guessing fegl
  - 3 . 4 white . box techniques
    - 3 . 4 . 1 statement coverage fscl
    - 3 . 4 . 2 branch coverage fbcl
    - 3 . 4 . 3 condition coverage fccl
    - 3 . 4 . 4 decision condition coverage fdcl
    - 3 . 4 . 5 multiple condition coverage(mcc)
    - 3 . 4 . 6 path coverage fpci
    - 3 . 4 . 7 d—u pair coverage(dup)
  - 3 . 5 fault insertion
    - 3 . 5 . 1 strong mutation testing(smt)
- 4 unit testing examples
- 5 static v\_erification
- 6 testing object-oriented software
- 7 integration and system testing
- 8 software test automation
- 9 testing in the software process
- 10 advanced testing issues
- a terminology
- b exercises
- select bibliography

## 章节摘录

版权页：插图： 7.1.7 Conclusion Overall, when it comes to selecting an integration testing strategy for a software application a number of things should be considered. These include the number of stubs and drivers to be produced, the location of the critical modules in the system, which layers will be available for testing, and what order the layers are to be implemented in. Integration testing is difficult as it does require synchronization between testers and developers in the production, combination and testing of modules. Also, faults found during integration testing can be harder to resolve, and finally, if design features need to be changed they may result in conflicts with the customer.

7.2 System Testing This means testing of the system as a whole and is almost invariably done using Blackbox tests. System testing can be broken down into a number of categories that may or may not be carried out on the product, as their necessity depends on the type of application it is. These are listed and explained below. It will be noticed that at times some of these testing activities overlap in terms of what they measure. This reflects that not all tests may be applicable to the one product.

7.2.1 System Test Categories

Conformance Testing Used to verify that the system conforms to a set of published standards. Many standards will have a published suite of conformance tests, or have a selected authority to run these tests. This is particularly important for communications software, as a software system must correctly inter-operate with other implementations.

Documentation Testing Used to verify that the documentation (in printed form, online, help, or prompts) is sufficient for the software to be installed and operated efficiently. Typically a full installation is performed, and then the different system functions are executed, exactly as documented. Responses are checked against the documented responses.

Ergonomic Testing Used to verify the ease-of-use of the system. This can be either automated (verifying font sizes, information placement on the screen, use of colours, or the speed of progress by users of different experience levels), or manual (based on feedback forms completed by users).

## <<软件测试>>

### 编辑推荐

《软件测试:原理与实践(英文版)》主要是面向软件测试的本科课程来设计相关内容,但书中讲述的理论和技巧辅以许多实际的例子来说明,这些实用的内容对于软件专业人员也很有吸引力。因为软件测试是现代软件工程项目管理的一个非常关键的要素,其重要性日渐显现,正在引起软件产业乃至IT产业的充分重视.....

#### 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>