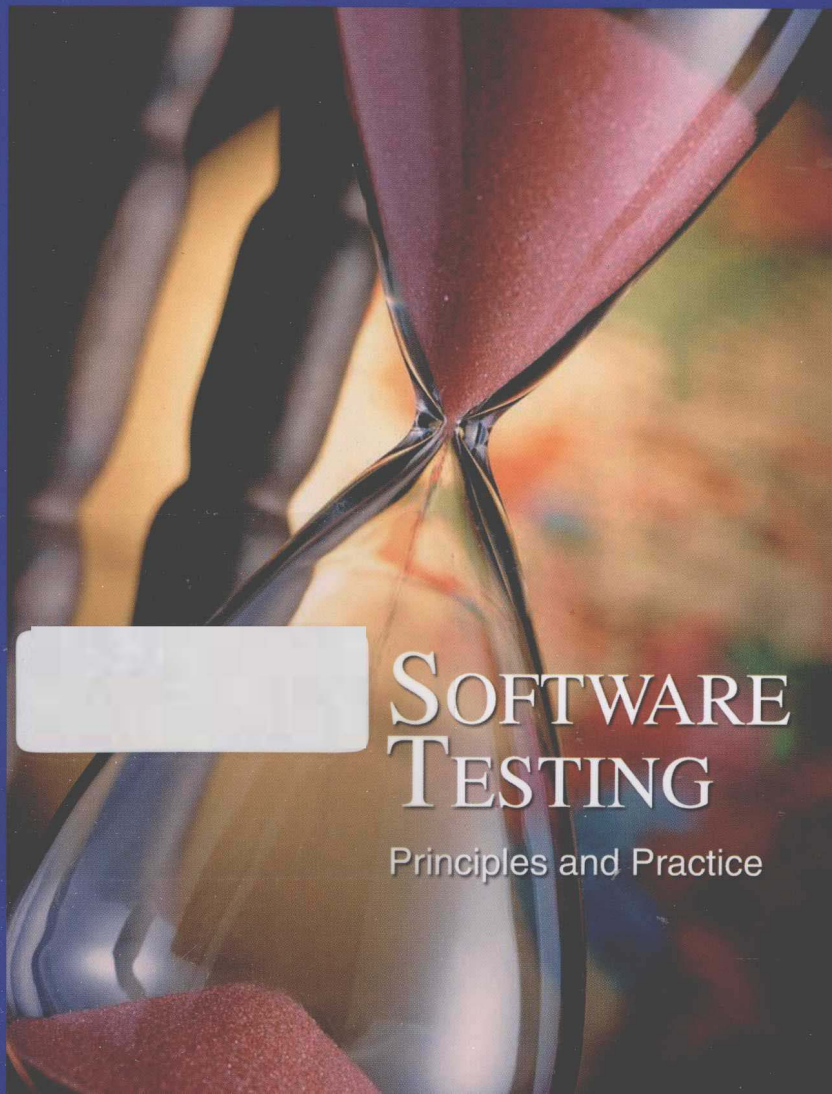
HZ BOOKS
华章教育

经 典 原 版 书 库

软件测试
原理与实践

（爱尔兰）Stephen Brown Joe Timoney Tom Lysaght
（中 国）Deshi Ye（叶德仕） 著

（英文版）



SOFTWARE TESTING
Principles and Practice

机械工业出版社
China Machine Press

经 典 原 版 书 库

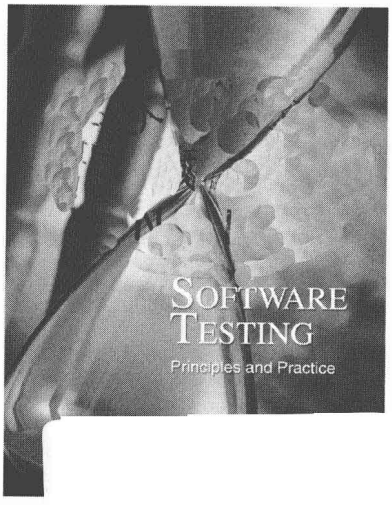
软件测试

原理与实践

（英文版）

Software Testing

Principles and Practice



（爱尔兰）Stephen Brown Joe Timoney Tom Lysaght
（中 国）Deshi Ye（叶德仕） 著

机械工业出版社
China Machine Press

本书版权登记号：图字：01-2012-3795

图书在版编目（CIP）数据

软件测试：原理与实践（英文版）/（爱尔兰）布朗（Brown，S.）等著. —北京：机械工业出版社，2012.6

（经典原版书库）

书名原文：Software Testing: Principles and Practice

ISBN 978-7-111-38863-0

I. 软… II. 布… III. 软件－测试－英文 IV. TP311.5

中国版本图书馆 CIP 数据核字（2012）第 130806 号

机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码 100037）

责任编辑：迟振春

北京瑞德印刷有限公司印刷

2012 年 9 月第 1 版第 1 次印刷

170mm×242mm • 13 印张

标准书号：ISBN 978-7-111-38863-0

定价：35.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：（010）88378991；88361066

购书热线：（010）68326294；88379649；68995259

投稿热线：（010）88379604

读者信箱：hzjsj@hzbook.com

出版者的话

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 Ritchie，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。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

华章网站：www.hzbook.com

电子邮件：hzjsj@hzbook.com

联系电话：(010) 88379604

联系地址：北京市西城区百万庄南街 1 号

邮政编码：100037

华章教育

华章科技图书出版中心

Preface

This book is based on a series of lectures given at the National University of Ireland, Maynooth and Zhejiang University. It provides a textbook for a number of courses, describing the fundamentals of software testing. The material has been developed over the past ten years, and reflects both the experiences from 20 years in industry from one of the authors, and the authors' joint experiences in lecturing.

There is no one standard textbook on software testing, and this book is the result of many years of extracting and interpreting test techniques from a wide and varied number of sources. These include testing classics such as The Art of Software Testing by Myers, Software Testing by Roper, and Testing Object-Oriented Systems by Binder; standard textbooks such as Software Engineering by Pressman and Ince, and Software Engineering by Sommerville; software process books such as Software Testing in the Real World by Kit and extreme Programming explained by Beck; and ISO and IEEE standards related to software quality and testing.

Software testing is a challenging task – it is as important for businesses and government as it is for research institutions. It is still as much an art as a science: there are no accepted standards or norms for applying the various techniques, and interpretation is required. There is no well established research on the effectiveness of different approaches. The techniques are easy to understand, but generally difficult to apply to real-world software. By providing extensive worked examples, this book aims to provide a solid basis for both understanding, and applying, various test techniques

前　言

本书的内容基于爱尔兰国立大学梅努斯和浙江大学的一系列课程讲稿。该书覆盖了软件测试的基本原理，可以作为许多课程的参考教材。本书的内容历经十年发展，融合了其中一位作者近二十年的工业界经验及全部作者的教学经验。

在软件测试领域，目前还没有统一的标准教科书，而本书是通过对各种不同的软件测试技术进行多年的提炼、阐释而形成的。这些测试技术的来源包括：一些经典的测试书籍，如 Myers 的《软件测试的艺术》、Roper 的《软件测试》、Binder 的《面向对象系统测试》；标准的软件工程教材，如 Pressman 和 Ince 的《软件工程》、Sommerville 的《软件工程》；软件过程类书籍，如 Kit 的《现实世界中的软件测试》、Beck 的《极限编程》；软件质量及测试相关的 ISO 和 IEEE 标准。

软件测试是一项具有挑战性的任务，它对科研机构、企业及政府具有同等的重要性。软件测试既是一门科学也是一门艺术。针对如何应用不同的软件测试技术，目前还没有广泛认同的标准，因此对各种软件测试技术的阐释是必需的。针对各种不同技术的有效性，目前还没有成熟的研究成果。总的来说，各种测试技术容易理解，但如何将其应用到软件产品中是困难的。本书旨在通过广泛丰富的实例，为读者理解和应用各种不同的软件测试技术提供一个坚实的基础。

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 The Software Industry

The software industry has come a long way since its beginnings in the 1950's. The independent software industry was essentially born in 1969 when IBM announced that they would stop treating their software as a free add-on to its computers, and instead would sell software and hardware as separate products. This opened up the market to external companies that could produce and sell software for IBM machines.

Software products for mass consumption arrived in 1981 with the arrival of PC-based software packages. Another dramatic boost came in the 1990's with the arrival of the World Wide Web, and in the 2000's with mobile devices. In 2010 the Top 500 companies in the global software industry had revenues of $492 billion. The industry is extremely dynamic and continually undergoing rapid change as new innovations appear. Unlike some other industries, for example transportation, it is still in many ways an immature industry. It does not, in general, have a set of quality standards that have been gained through years of hard-won experience.

Numerous examples exist of the results of failures in software quality and the costs it can incur. Famous incidents include the failure of the European Space Agency's Ariane 5 rocket, the Therac-25 radiation therapy machine, and the loss of the Mars Climate observer in 1999. A study by the US Department of Commerce's National Institute of Standards and Technology in 2002 estimated that the cost of faulty software to the US economy was up to $59.5 billion per year.

However, many participants in the industry do apply quality models and measures to the processes through which their software is produced. Software Testing is an important part of the Software Quality assurance process, and is an important discipline within Software Engineering. It has an important role to play throughout the software development lifecycle, whether being used in a Verification and Validation context, or as part of an actual test-driven software development process such as Extreme programming.

Software Engineering as a discipline grew out of the "Software Crisis". The term Software Crisis was first used at the end of the 1960's but it really began to have meaning through the 1970's as the software industry was growing. This reflected the increasing size and complexity of software projects combined with the lack of formal procedures for managing such projects.

This organizational poverty resulted in a number of problems:

- Projects were running over-budget
- Projects were running over-time
- The Software products were of low quality
- The Software products often did not meet their requirements
- Projects were chaotic
- Software maintenance was very difficult

If the software industry was to keep growing, and the use of software was to become more widespread, this situation could not continue. The solution was to be in formalizing the roles and responsibilities of software engineering personnel. These software engineers would plan and document in detail the goals of each software project and how it was to be carried out, they would manage the process via which the software code would be created, and they would ensure that the end result had attributes to show that it was a quality product. This relationship between quality management and software engineering meant that software testing would be integrated into its field of influence. Moreover, the field of software testing was also going to have to change if the industry wanted to get over the "Software Crisis".

While the difference between debugging a program and testing a program was recognized by the 1970's, it was only from this time on that testing began to take a significant role in the production of software. It was to change from being an activity that happened at the end of the product cycle, to check that the product worked, to an activity that takes place throughout each stage of development, catching faults as early as possible. A number of studies comparing the cost of early vs late defect detection have all reached the same conclusion: the earlier the defect is caught, the less the cost of fixing it.

The progressive improvement of software engineering practices has led to a significant improvement in software quality. The short-term benefits of software testing to the business include improving the performance, interoperability and conformance of the software products produced. In the longer term, testing reduces the future costs, and builds customer confidence.

Nowadays, many software development processes have integrated software testing within their activities. Furthermore, processes such as "Test Driven Development" use testing to lead the code development.

1.2 Software Testing and Quality

Proper software testing procedures reduce the risks associated with software development. This is because modern software programs are very complex, having thousands of lines of code. And they are often attempting to solve a problem that has been defined in very abstract terms, described as a vague set of requirements lacking in exactness and detail. Quality problems are further compounded by external pressures on developers, from business owners, imposing strict deadlines and budgets in order to reduce the time to market and associated production costs. However, these pressures can result in inadequate software testing procedures, leading to reduced quality. Poor quality leads to more failures, increased development costs, and delays in attaining product stability. More severe outcomes for a business can be a loss in reputation, combined with market share and even having to face legal claims.