

Software  
Testing

软件测试丛书

# 软件测试技术大全

——测试基础 流行工具 项目实战

Software Testing Guide  
Fundamentals, Tools and Practice

陈能技 编著

- 介绍了国内外先进的测试技术和测试理念，微软公司的测试方法，RUP中的测试过程，AEP的理念等
- 详细讲述主流测试工具的使用技术和技巧，如LoadRunner、TestComplete、TestDirector和QTP等
- 来自测试实践中的典型案例，如C/S结构测试、B/S结构测试、手机测试、游戏测试、界面测试、数据库测试、Web Services测试等，突出实用性和实战性

 人民邮电出版社  
POSTS & TELECOM PRESS

Software  
Testing

软件测试丛书

TP311.5/237

2008

# 软件测试技术大全

——测试基础 流行工具 项目实战

Software Testing Guide  
Fundamentals, Tools and Practice

陈能技 编著

人民邮电出版社

北京

## 图书在版编目 (CIP) 数据

软件测试技术大全：测试基础 流行工具 项目实战 / 陈能技编著. —北京：人民邮电出版社，2008.6  
ISBN 978-7-115-17891-6

I. 软… II. 陈… III. 软件—测试 IV. TP311.5

中国版本图书馆 CIP 数据核字 (2008) 第 043202 号

## 内 容 提 要

本书在编写过程中考虑到测试人员的需求，尤其是测试新手对各种知识的需求，由浅入深，循序渐进地介绍了各种测试理论知识和实践经验；详细讲述了主流测试工具的使用，例如 LoadRunner、TestComplete、TestDirector 等；还介绍了各种常用的开源测试工具，为期待引入开源测试工具的测试人员提供参考；另外，结合项目实践，介绍了各种辅助测试工具的开发，例如每日构建框架的开发、AEP 系统的搭建、性能测试框架的搭建、正交表测试用例自动生成工具的设计以及数据库比较工具的制作等。

本书包括了丰富的实战案例和流行测试工具的使用技术和技巧，通过阅读本书，不但可以让新手快速掌握软件测试的理论知识，更可以提高读者的实战技能。

本书在每章的最后，针对测试中可能遇到的各种问题，结合作者的实践经验给出了分析和解答，并准备了模拟面试的问答题及其参考答案，以方便那些希望进入测试行业的人员学习。

---

### 软件测试技术大全——测试基础 流行工具 项目实战

---

◆ 编 著 陈能技

责任编辑 张 涛

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号

邮编 100061 电子函件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京鸿佳印刷厂印刷

新华书店总店北京发行所经销

◆ 开本：787×1092 1/16

印张：32

字数：783 千字

2008 年 6 月第 1 版

印数：1—4 000 册

2008 年 6 月北京第 1 次印刷

---

ISBN 978-7-115-17891-6/TP

---

定价：59.00 元

读者服务热线：(010)67132692 印装质量热线：(010)67129223

反盗版热线：(010)67171154

# 前 言

软件测试是一门涉及广泛知识的学科,软件测试行业在国内方兴未艾,测试工程师急需掌握各种测试技术和工具,测试团队的管理也亟待完善和提高。为此,作者精心编写了本书,书中结合作者多年的测试实践经验,为广大测试人员介绍了各种先进的测试技术、主流的测试工具的使用和测试理念。这是一本测试人员提高测试水平、完善知识结构和扩展测试知识面的必备参考书。

## 本书的内容安排

本书分为4篇,共20章,从软件测试的基本概念开始,逐步介绍了一个完整的测试过程所经历各个阶段,并结合目前测试行业中流行的各种实用技术和常用工具,讲述了如何进行各种类型的软件测试。最后根据作者的经验讲解如何营造一个良好的学习环境,让测试人员的水平得以不断地提高。

### 第一篇(第1章至第3章) 软件测试的基础。

讲述了软件测试的基础知识,包括软件测试的起源和发展、测试行业的现状以及优秀测试工程师应该具备的素质,为希望进入测试领域的人员提供一些基础知识。测试新手通过这3章内容的学习,能为发展打下较坚实的基础,并能认识到自己目前的不足及需要努力的方向。

### 第二篇(第4章至第8章) 软件测试基本理论。

讲述了与软件测试相关的各种知识,包括软件工程、软件测试的目的与原则、软件测试的各种方法论、软件测试的具体过程及软件测试的质量度量方法,这部分的内容是一名初级测试人员通往测试工程师必备的知识。

### 第三篇(第9章至第18章) 实用软件测试技术与工具。

具体讲述了各种实用软件测试技术的应用,以及目前主流的测试工具的使用。另外还介绍了单元测试、自动化功能测试、性能测试、探索性测试、用户界面测试等测试案例的开展和管理方法。这部分的内容是作者多年从事软件测试和测试管理工作的经验总结,其中重点介绍了一些辅助测试工具的开发,是测试人员综合利用测试技术和测试方法进行各种测试的重要参考,也是一名普通测试工程师通往高级测试工程师需要掌握的核心知识。

### 第四篇(第19章至第20章) 软件测试的学习和研究。

主要介绍测试人员的发展和提高途径,以及测试团队管理的技巧。测试人员需要找到自己的发展方向,清楚自己的缺点,与其他测试人员一起营造一个共享的交流和学习环境。

本书基本上涵盖了软件测试的各个方面的知识,从测试设计到测试用例,从测试执行到测试管理,从测试的基本理论到测试的实用技术,从测试工具的使用到测试工具的开发,以及各种常用的测试用例设计方法和开源测试工具在测试项目中的使用。

## 本书的特点

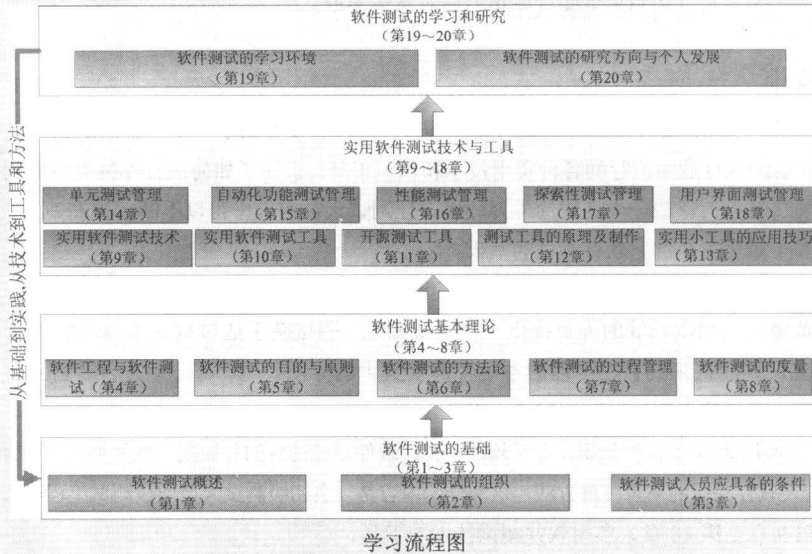
本书的特点主要体现在以下几个方面:

- 本书的编排采用循序渐进的方式,适合初级、中级学者逐步掌握软件测试的基本方法,软件测试设计和管理的精髓;
- 本书在介绍各种测试方法和技术时,结合了许多浅显易懂的例子,方便读者自己进行实践和演练,在介绍软件测试工具的开发时更是提供了丰富完整的开发示例代码,读者可直接使用,或者根据自己的实际

情况进行调整：

- 本书还介绍了目前软件测试领域各种先进的前沿技术和理论以及国外先进的测试方法和技术，方便读者借鉴；
- 本书结合作者多年的团队管理和新人培训经验，在每章的最后还特地指出初级测试人员可能存在的疑惑和误解，并且有针对性地进行解析，为测试新手摆脱“困境”进行答疑解惑，而且给出的模拟面试问答也为希望进入测试行业的人员提供了参考。

## 本书的学习流程图



## 适合阅读本书的读者

- 希望进入测试行业的新手；
- 迫切希望提高个人测试技能和水平的初级测试人员；
- 具备一定的测试理论知识但是缺乏实践的测试工程师；
- 目前正在考虑引入测试工具或正在使用测试工具的测试人员；
- 希望了解各种开源测试工具的测试人员；
- 希望了解测试工具开发过程和开发技术，希望自己动手开发测试工具的测试人员；
- 希望了解大型软件测试团队的测试理念和测试方法的测试人员；
- 希望提高团队凝聚力和加强测试人员学习能力的测试管理者。

本书由陈能技组织编写，同时参与编写和代码测试的人员有陈水峰、慈元龙、关蔼婷、贺宇、胡立实、姜磊、李来春、李争亭、刘吉万、柳玲、罗栋、罗玉霞、秦辉、陈杰、陈冠军、项宇峰、唐敏、唐智峰、王安平、王成喜、王淑敏、谢马远、张丹、张迪妮、钟蜀明、竺东、祝庆林等，在此对他们的辛勤工作表示诚挚的感谢。

本书全部源代码下载地址为：[www.3Atesting.com](http://www.3Atesting.com)。

由于编写时间仓促，加之水平有限，书中难免存有不妥和错误之处，恳请广大读者批评指正，联系邮箱为：[zhangtao@ptpress.com.cn](mailto:zhangtao@ptpress.com.cn)。

编者

2008年4月



# 目 录

## 第一篇 软件测试的基础

|                                   |                                    |
|-----------------------------------|------------------------------------|
| <b>第 1 章 软件测试概述</b> ..... 1       | <b>2.3 软件测试团队的建设</b> ..... 17      |
| 1.1 软件测试的起源..... 2                | 2.3.1 学习型团队的组建..... 17             |
| 1.2 软件测试的发展..... 3                | 2.3.2 测试规范..... 19                 |
| 1.3 软件测试行业的现状和前景..... 6           | 2.3.3 部门制度..... 20                 |
| 1.4 小结..... 8                     | <b>2.4 小结</b> ..... 20             |
| 1.5 新手入门须知..... 8                 | <b>2.5 新手入门须知</b> ..... 21         |
| 1.6 模拟面试问答..... 8                 | <b>2.6 模拟面试问答</b> ..... 21         |
| <b>第 2 章 软件测试的组织</b> ..... 10     | <b>第 3 章 软件测试人员应具备的条件</b> ..... 23 |
| 2.1 测试的组织形式..... 11               | 3.1 测试人员的素质要求..... 24              |
| 2.2 融入测试组织..... 14                | 3.2 测试人员的技能要求..... 28              |
| 2.2.1 根据开发模式定位<br>自己的测试角色..... 14 | 3.3 小结..... 36                     |
| 2.2.2 如何融入一个项目团队..... 16          | 3.4 新手入门须知..... 36                 |
|                                   | 3.5 模拟面试问答..... 37                 |

## 第二篇 软件测试基本理论

|                                    |                                     |
|------------------------------------|-------------------------------------|
| <b>第 4 章 软件工程与软件测试</b> ..... 38    | <b>4.5 小结</b> ..... 50              |
| 4.1 软件工程简介..... 39                 | <b>4.6 新手入门须知</b> ..... 50          |
| 4.2 软件开发模式..... 41                 | <b>4.7 模拟面试问答</b> ..... 51          |
| 4.2.1 常见的软件开发模式..... 41            | <b>第 5 章 软件测试的目的与原则</b> ..... 53    |
| 4.2.2 软件开发模式的发展..... 42            | 5.1 软件测试的目的..... 54                 |
| 4.3 不同软件开发模式下的<br>软件测试..... 43     | 5.1.1 软件测试与建立软件<br>信心的关系..... 54    |
| 4.3.1 CMM 和 ISO 中的<br>软件测试..... 44 | 5.1.2 软件测试的两面性..... 54              |
| 4.3.2 敏捷开发中的软件测试..... 45           | 5.1.3 软件测试的验证与确认..... 55            |
| 4.3.3 QA 与 QC 的区别..... 46          | 5.1.4 测试是一种服务..... 56               |
| 4.4 配置管理与软件测试..... 47              | 5.2 软件测试应该遵循的原则..... 56             |
| 4.4.1 什么是配置管理..... 47              | 5.2.1 Good enough 原则..... 57        |
| 4.4.2 软件配置管理的目的..... 47            | 5.2.2 Pareto 原则..... 57             |
| 4.4.3 软件配置管理的<br>基本任务..... 47      | 5.2.3 尽早开展测试..... 58                |
| 4.4.4 配置管理自动化..... 49              | 5.2.4 在发现较多错误的地方<br>投入更多的测试..... 58 |
|                                    | 5.2.5 同化效应..... 58                  |

|              |                   |           |        |                           |     |
|--------------|-------------------|-----------|--------|---------------------------|-----|
| 5.3          | 小结                | 59        | 7.4.7  | 使用正交试验法设计<br>测试用例         | 91  |
| 5.4          | 新手入门须知            | 59        | 7.4.8  | 使用均匀试验法设计<br>测试用例         | 93  |
| 5.5          | 模拟面试问答            | 59        | 7.4.9  | 使用组合覆盖法设计<br>测试用例         | 93  |
| <b>第 6 章</b> | <b>软件测试的方法论</b>   | <b>61</b> | 7.4.10 | 使用分类树方法设计<br>测试用例         | 95  |
| 6.1          | 软件测试的不同学派         | 62        | 7.4.11 | 测试用例设计的自动化                | 97  |
| 6.2          | 软件测试的方法应用         | 64        | 7.5    | 测试的执行                     | 100 |
| 6.3          | IBM 公司的软件测试方法     | 66        | 7.5.1  | 测试用例的合理选择                 | 100 |
| 6.3.1        | 回归测试              | 66        | 7.5.2  | 测试的分工与资源利用                | 100 |
| 6.3.2        | 测试的度量             | 66        | 7.5.3  | 测试环境的搭建                   | 101 |
| 6.3.3        | 用例驱动              | 67        | 7.5.4  | BVT 测试与冒烟测试               | 102 |
| 6.3.4        | RUP 对软件测试的分类      | 68        | 7.5.5  | 每日构建的基本流程                 | 102 |
| 6.3.5        | RUP 对测试阶段的划分      | 69        | 7.6    | 测试记录与跟踪                   | 103 |
| 6.4          | 自动错误预防方法          | 69        | 7.6.1  | Bug 的质量衡量                 | 104 |
| 6.4.1        | AEP 的基本概念         | 70        | 7.6.2  | 如何记录一个合格的<br>Bug          | 104 |
| 6.4.2        | 实现软件自动错误预防的<br>方法 | 70        | 7.6.3  | Bug 报告应该注意的<br>问题         | 106 |
| 6.5          | 小结                | 72        | 7.6.4  | 跟踪一个 Bug 的生命<br>周期        | 106 |
| 6.6          | 新手入门须知            | 72        | 7.6.5  | Bug 评审应注意的<br>问题          | 107 |
| 6.7          | 模拟面试问答            | 74        | 7.6.6  | 基于 TestDirector 的<br>缺陷管理 | 108 |
| <b>第 7 章</b> | <b>软件测试的过程管理</b>  | <b>75</b> | 7.7    | 回归测试                      | 108 |
| 7.1          | 软件测试的各个阶段         | 76        | 7.7.1  | 为什么会回归                    | 109 |
| 7.2          | 测试需求              | 77        | 7.7.2  | 回归测试的难度                   | 109 |
| 7.2.1        | 需求规格说明书的<br>检查要点  | 77        | 7.7.3  | 基于风险的回归测试                 | 109 |
| 7.2.2        | 需求规格说明书的<br>检查步骤  | 77        | 7.8    | 测试总结和报告                   | 111 |
| 7.2.3        | 通过编写测试用例来<br>检查需求 | 80        | 7.8.1  | 缺陷分类报告                    | 111 |
| 7.3          | 测试计划              | 82        | 7.8.2  | 缺陷趋势报告                    | 113 |
| 7.3.1        | 制定测试计划的必要性        | 82        | 7.8.3  | 典型缺陷与 Bug 模式              | 114 |
| 7.3.2        | 测试计划与测试过程的<br>关系  | 82        | 7.8.4  | 测试中的 PDCA 循环              | 115 |
| 7.4          | 测试设计及测试用例         | 85        | 7.8.5  | 客观全面的测试报告                 | 116 |
| 7.4.1        | 等价类划分法            | 85        | 7.8.6  | 实用测试经验的总结                 | 116 |
| 7.4.2        | 边界值分析法            | 86        | 7.9    | 小结                        | 116 |
| 7.4.3        | 基本路径分析法           | 88        |        |                           |     |
| 7.4.4        | 因果图法              | 89        |        |                           |     |
| 7.4.5        | 场景设计法             | 90        |        |                           |     |
| 7.4.6        | 错误猜测法             | 91        |        |                           |     |

|              |                   |            |       |                    |     |
|--------------|-------------------|------------|-------|--------------------|-----|
| 7.10         | 新手入门须知            | 117        | 8.2.1 | 度量 Bug 的数量         | 124 |
| 7.11         | 模拟面试问答            | 117        | 8.2.2 | 加权法度量缺陷            | 125 |
| <b>第 8 章</b> | <b>软件测试的度量</b>    | <b>120</b> | 8.2.3 | Bug 的定性评估          | 126 |
| 8.1          | 软件测试度量的目的         | 121        | 8.2.4 | Bug 综合评价模型         | 127 |
| 8.1.1        | 度量的目的             | 121        | 8.2.5 | 测试覆盖率统计            | 127 |
| 8.1.2        | 测试度量的难度           | 122        | 8.2.6 | 考核测试人员的硬指标和<br>软指标 | 130 |
| 8.1.3        | 测试人员工作质量的<br>衡量   | 123        | 8.3   | 小结                 | 132 |
| 8.2          | 软件测试的度量方法及其<br>应用 | 124        | 8.4   | 新手入门须知             | 132 |
|              |                   |            | 8.5   | 模拟面试问答             | 133 |

### 第三篇 实用软件测试技术与工具

|              |                     |            |               |                     |            |
|--------------|---------------------|------------|---------------|---------------------|------------|
| <b>第 9 章</b> | <b>实用软件测试技术</b>     | <b>135</b> | 9.3.8         | Web Services 的测试    | 168        |
| 9.1          | 软件测试技术的发展           | 136        | 9.3.9         | 内存泄漏测试              | 169        |
| 9.2          | 软件测试技术              | 137        | 9.3.10        | 报表测试                | 172        |
| 9.2.1        | 黑盒测试与白盒测试           | 137        | 9.3.11        | 联机帮助和用户手册的<br>测试    | 173        |
| 9.2.2        | 手工测试与自动化测试          | 138        | 9.3.12        | 缺乏工具支持的性能<br>测试     | 175        |
| 9.2.3        | 探索性测试               | 140        | 9.3.13        | 本地化测试与国际化<br>测试     | 179        |
| 9.2.4        | 单元测试                | 142        | 9.3.14        | 可访问性测试              | 180        |
| 9.2.5        | 单元级别的性能测试           | 143        | 9.4           | 小结                  | 181        |
| 9.2.6        | 数据库性能检查             | 146        | 9.5           | 新手入门须知              | 182        |
| 9.2.7        | 软件的“极限考验”——<br>压力测试 | 146        | 9.6           | 模拟面试问答              | 182        |
| 9.2.8        | 软件的容量测试             | 147        | <b>第 10 章</b> | <b>实用软件测试工具</b>     | <b>185</b> |
| 9.2.9        | 安全性测试               | 151        | 10.1          | 软件测试工具的作用           | 186        |
| 9.2.10       | 安装测试                | 152        | 10.2          | 软件测试工具的分类           | 186        |
| 9.2.11       | 环境测试                | 154        | 10.2.1        | 按测试工具的用途<br>分类      | 186        |
| 9.3          | 实用软件测试技术的<br>综合应用   | 155        | 10.2.2        | 按测试工具的收费方式<br>分类    | 187        |
| 9.3.1        | 跟踪法测试               | 155        | 10.3          | 如何在测试组中引入测试<br>工具   | 188        |
| 9.3.2        | C/S 结构软件系统的<br>测试   | 158        | 10.3.1        | 测试工具的选型             | 188        |
| 9.3.3        | B/S 结构软件系统的<br>测试   | 159        | 10.3.2        | 测试工具的试用             | 188        |
| 9.3.4        | 手机应用软件的测试           | 160        | 10.3.3        | 测试工具的使用培训           | 189        |
| 9.3.5        | 游戏软件系统的测试           | 161        | 10.3.4        | 测试工具的使用             | 190        |
| 9.3.6        | 界面交互及用户体验<br>测试     | 163        | 10.4          | 测试管理工具 TestDirector | 190        |
| 9.3.7        | 数据库测试               | 164        |               |                     |            |



- 10.4.1 测试管理工具与缺陷管理工具的区别 ..... 191
- 10.4.2 测试管理功能之间的关联 ..... 197
- 10.4.3 利用 TestDirector 生成测试报告 ..... 199
- 10.4.4 TestDirector 数据库的备份和恢复 ..... 200
- 10.4.5 TestDirector 的个性化定制 ..... 203
- 10.4.6 TestDirector 的使用技巧 ..... 208
- 10.5 自动化功能测试工具
  - TestComplete ..... 210
  - 10.5.1 脚本语言的选择 ..... 210
  - 10.5.2 编写脚本的基本步骤 ..... 211
  - 10.5.3 命令行运行方式 ..... 215
  - 10.5.4 TestComplete 如何模拟用户动作 ..... 216
  - 10.5.5 TestComplete 的 Stores 对象的使用 ..... 219
  - 10.5.6 TestComplete 的数据驱动测试方法 ..... 223
  - 10.5.7 TestComplete 的对象驱动测试方法 ..... 225
  - 10.5.8 使用 TestComplete 测试 Open 的应用程序 ..... 227
  - 10.5.9 使用 TestComplete 测试分布式应用程序 ..... 228
  - 10.5.10 TestComplete 的一些常用技巧 ..... 231
  - 10.5.11 TestComplete 的一些常用内建函数 ..... 233
- 10.6 自动化功能测试工具 QTP ..... 235
  - 10.6.1 插件管理 ..... 235
  - 10.6.2 基本测试过程 ..... 235
  - 10.6.3 创建测试 ..... 236
  - 10.6.4 运行测试 ..... 238
  - 10.6.5 分析结果 ..... 239
  - 10.6.6 关键字驱动测试方法 ..... 240
  - 10.6.7 什么时候使用关键字驱动测试方法 ..... 240
- 10.7 性能测试工具 LoadRunner ..... 241
  - 10.7.1 协议的选择与录制 ..... 242
  - 10.7.2 脚本参数化和添加事务 ..... 246
  - 10.7.3 性能参数的选择和监视 ..... 251
  - 10.7.4 性能测试报告与性能瓶颈分析 ..... 253
- 10.8 内存问题测试工具 BoundsChecker ..... 255
  - 10.8.1 在 Delphi 项目中设置编译选项 ..... 255
  - 10.8.2 BoundsChecker 使用简介 ..... 256
  - 10.8.3 如何判断内存问题 ..... 257
- 10.9 微软的单元测试工具 MSTest ..... 259
  - 10.9.1 建立单元测试项目 ..... 259
  - 10.9.2 巧用 NMock 对象 ..... 261
  - 10.9.3 执行单元测试 ..... 262
- 10.10 测试用例设计工具 CTE XL ..... 263
  - 10.10.1 分类树设计 ..... 263
  - 10.10.2 选择组合测试用例 ..... 264
  - 10.10.3 测试用例的自动生成 ..... 265
  - 10.10.4 统计分析和报告 ..... 266
- 10.11 小结 ..... 268
- 10.12 新手入门须知 ..... 268
- 10.13 模拟面试问答 ..... 268
- 第 11 章 开源测试工具 ..... 271**
  - 11.1 开源测试工具简介 ..... 272
    - 11.1.1 开源的背景 ..... 272
    - 11.1.2 开源测试工具的发展现状 ..... 272
  - 11.2 常用开源测试工具简介——测试管理类 ..... 273
    - 11.2.1 Bugzilla Bug 管理系统 ..... 273
    - 11.2.2 Mantis 缺陷跟踪系统 ..... 274

- 11.2.3 BugFree 测试管理工具… 276
- 11.2.4 综合比较… 276
- 11.3 常用开源测试工具简介——单元测试类… 277
  - 11.3.1 NUnit-针对.NET 开发的单元测试框架… 277
  - 11.3.2 NMock 单元测试方法… 278
  - 11.3.3 NUnitForms——界面层代码测试… 279
- 11.4 常用开源测试工具简介——性能测试类… 280
  - 11.4.1 OpenSTA——Web 性能测试工具… 280
  - 11.4.2 TestMaker——性能测试工具… 281
  - 11.4.3 DBMonster——生成大量数据库数据的工具… 282
- 11.5 常用开源测试工具简介——自动化功能测试类… 284
  - 11.5.1 Abbot Java GUI Test Framework——自动化测试框架… 284
  - 11.5.2 SharpRobo——功能测试工具和录制工具… 285
  - 11.5.3 Samie——IE 自动化测试… 287
- 11.6 如何在测试组中引入开源测试工具… 288
  - 11.6.1 开源测试工具的成本考虑… 288
  - 11.6.2 引入开源测试工具的步骤… 289
  - 11.6.3 引入开源测试工具可能碰到的问题… 290
- 11.7 小结… 291
- 11.8 新手入门须知… 291
- 11.9 模拟面试问答… 291
- 第 12 章 测试工具的原理及制作… 294**
  - 12.1 自制测试工具的优势… 295
  - 12.2 辅助工具的制作… 296
    - 12.2.1 测试工具的开发策划… 296
    - 12.2.2 测试语言的选择… 296
    - 12.2.3 测试工具开发的各种实现技术… 297
  - 12.3 使用 Windows 脚本辅助测试… 299
    - 12.3.1 使用 JScript 进行简单的 GUI 自动化测试… 299
    - 12.3.2 使用 JScript 检查注册表… 300
    - 12.3.3 使用 JScript 的 FileSystem Object 对象处理文件… 300
    - 12.3.4 使用 JScript 操作 Excel… 301
    - 12.3.5 在 JScript 中运行应用程序… 302
    - 12.3.6 在 JScript 中使用 WMI… 303
    - 12.3.7 在 JScript 中访问网络… 304
    - 12.3.8 在 JScript 中使用正则表达式… 305
    - 12.3.9 使用 JScript 发送邮件… 306
    - 12.3.10 JScript 脚本的调试方法… 306
  - 12.4 利用 Windows 脚本搭建一个每日构建框架… 307
    - 12.4.1 每日构建框架的基本要素… 307
    - 12.4.2 使用 Windows 任务计划来定时启动脚本… 313
    - 12.4.3 每日构建框架的扩展 1——单元测试… 314
    - 12.4.4 每日构建框架的扩展 2——自动化功能测试… 319
    - 12.4.5 每日构建框架的扩展 3——每日缺陷简报… 321
    - 12.4.6 每日构建框架的扩展 4——每日配置管理简报… 327
    - 12.4.7 每日构建框架的扩展 5——每日里程碑预报… 330
    - 12.4.8 每日构建框架的其他

- 扩展思路 ..... 332
- 12.5 利用 Windows 脚本整合一个自动错误预防系统 ..... 335
  - 12.5.1 轻量级的 AEP 框架 ..... 336
  - 12.5.2 整合 FxCop ..... 336
  - 12.5.3 整合 SQL BPA ..... 337
- 12.6 设计一个性能测试框架 ..... 338
  - 12.6.1 性能测试的基本原理 ..... 338
  - 12.6.2 Controller 的简单设计 ..... 339
  - 12.6.3 Agent 的简单设计 ..... 340
  - 12.6.4 虚拟用户的产生 ..... 343
- 12.7 正交表测试用例自动生成工具的设计 ..... 344
  - 12.7.1 正交表类的设计 ..... 345
  - 12.7.2 加载正交表文件 ..... 346
  - 12.7.3 解释输入 ..... 348
  - 12.7.4 查找正交表 ..... 349
  - 12.7.5 改进方向 ..... 356
- 12.8 数据库比较工具的制作 ..... 356
  - 12.8.1 “三库”问题 ..... 356
  - 12.8.2 数据库比较工具的设计 ..... 357
- 12.9 Oracle 的 SQL 语句跟踪工具的制作 ..... 361
  - 12.9.1 设置 Oracle 的 SQL 跟踪参数 ..... 361
  - 12.9.2 打开 SQL 跟踪 ..... 361
  - 12.9.3 关闭 SQL 跟踪 ..... 362
  - 12.9.4 改进方向 ..... 363
- 12.10 一个简单的猴子测试工具的制作 ..... 363
  - 12.10.1 猴子测试工具应该具备的功能 ..... 364
  - 12.10.2 完整的猴子测试工具 ..... 373
  - 12.10.3 猴子测试工具的扩展 ..... 379
- 12.11 测试覆盖率辅助管理工具的制作 ..... 380
  - 12.11.1 测试覆盖率管理 ..... 380
  - 12.11.2 数据覆盖率管理 ..... 383
  - 12.11.3 测试覆盖率统计的自动化 ..... 384
  - 12.11.4 测试覆盖率对测试管理的意义 ..... 385
  - 12.11.5 测试覆盖率辅助管理工具的设计 ..... 385
  - 12.11.6 测试覆盖率辅助管理工具的使用 ..... 389
- 12.12 小结 ..... 391
- 12.13 新手入门须知 ..... 391
- 12.14 模拟面试问答 ..... 391
- 第 13 章 实用小工具的应用技巧 ..... 393**
  - 13.1 巧用 Windows 自带的小工具 ..... 394
    - 13.1.1 Windows 的任务管理器 ..... 394
    - 13.1.2 Perfmon 的性能监控 ..... 396
    - 13.1.3 NetStat 的网络监视 ..... 398
  - 13.2 一些免费小工具的妙用 ..... 400
    - 13.2.1 SQL Server 数据库的 SQL 事件探查器 ..... 400
    - 13.2.2 Visual Studio 开发工具的 Spy++ ..... 401
    - 13.2.3 Visual SourceSafe 的文件比较器 ..... 402
  - 13.3 小结 ..... 403
  - 13.4 新手入门须知 ..... 403
  - 13.5 模拟面试问答 ..... 404
- 第 14 章 单元测试管理 ..... 405**
  - 14.1 单元测试范围管理 ..... 406
    - 14.1.1 单元测试的分类 ..... 406
    - 14.1.2 单元测试的效果 ..... 408
    - 14.1.3 单元测试的范围 ..... 408
  - 14.2 单元测试的过程管理 ..... 408
    - 14.2.1 单元测试过程策划 ..... 408
    - 14.2.2 单元测试管理规范 ..... 410
    - 14.2.3 单元测试与每日构建的结合 ..... 412
  - 14.3 单元测试的质量度量 ..... 414
    - 14.3.1 单元测试覆盖率 ..... 414

|                            |     |                           |     |
|----------------------------|-----|---------------------------|-----|
| 14.3.2 单元测试评审              | 415 | 16.2.2 性能测试的资源安排          | 441 |
| 14.4 小结                    | 416 | 16.2.3 性能参数的监控            | 443 |
| 14.5 新手入门须知                | 416 | 16.2.4 性能测试报告与<br>瓶颈分析    | 444 |
| 14.6 模拟面试问答                | 416 | 16.3 性能测试的质量度量            | 446 |
| <b>第 15 章 自动化功能测试管理</b>    | 418 | 16.3.1 性能测试是否全面           | 446 |
| 15.1 是否应该进行自动化<br>功能测试     | 419 | 16.3.2 性能测试的投入与<br>回报     | 448 |
| 15.1.1 “猴子”测试技术            | 419 | 16.4 小结                   | 448 |
| 15.1.2 自动化功能测试的<br>原则      | 420 | 16.5 新手入门须知               | 449 |
| 15.2 自动化功能测试的过程策划          | 421 | 16.6 模拟面试问答               | 449 |
| 15.2.1 自动化测试的时机            | 421 | <b>第 17 章 探索性测试管理</b>     | 451 |
| 15.2.2 自动化功能测试的<br>设计      | 422 | 17.1 探索性测试的必要性            | 452 |
| 15.2.3 自动化功能测试的<br>编码      | 424 | 17.1.1 探索性测试的原理           | 452 |
| 15.2.4 自动化功能测试的<br>验证      | 426 | 17.1.2 探索性测试与即兴<br>测试的区别  | 452 |
| 15.2.5 自动化功能测试的<br>管理规范    | 426 | 17.1.3 探索性测试的意义           | 453 |
| 15.2.6 自动化功能测试与<br>每日构建的结合 | 428 | 17.2 如何进行探索性测试            | 453 |
| 15.3 自动化功能测试的质量度量          | 429 | 17.2.1 优秀探索性测试人员的<br>基本素质 | 453 |
| 15.3.1 自动化功能测试的<br>局限性     | 429 | 17.2.2 测试就是向程序提问          | 454 |
| 15.3.2 自动化功能测试的<br>覆盖率     | 429 | 17.3 探索性测试的过程管理和度量        | 455 |
| 15.3.3 增强软件可测试性            | 430 | 17.3.1 测试组长是“教练”          | 455 |
| 15.3.4 可测性设计               | 430 | 17.3.2 基于探索任务的测试<br>计划    | 455 |
| 15.3.5 敏捷的自动化测试            | 433 | 17.3.3 探索性测试的“碰头<br>会议”   | 456 |
| 15.4 小结                    | 435 | 17.4 小结                   | 458 |
| 15.5 新手入门须知                | 436 | 17.5 新手入门须知               | 459 |
| 15.6 模拟面试问答                | 436 | 17.6 模拟面试问答               | 459 |
| <b>第 16 章 性能测试管理</b>       | 438 | <b>第 18 章 用户界面测试管理</b>    | 460 |
| 16.1 性能测试的需求管理             | 439 | 18.1 用户界面测试的必要性           | 461 |
| 16.1.1 性能测试的成本             | 439 | 18.2 如何进行用户界面测试           | 461 |
| 16.1.2 性能指标的确定             | 439 | 18.2.1 用户界面测试的时机          | 461 |
| 16.2 性能测试的过程管理             | 440 | 18.2.2 用户界面测试的要点          | 462 |
| 16.2.1 性能测试的方案选择           | 440 | 18.3 用户界面测试原则             | 463 |
|                            |     | 18.3.1 亲和力                | 464 |
|                            |     | 18.3.2 协助                 | 465 |
|                            |     | 18.3.3 有效                 | 465 |

|            |     |             |     |
|------------|-----|-------------|-----|
| 18.3.4 鼓励  | 466 | 18.3.10 简单  | 469 |
| 18.3.5 熟悉  | 467 | 18.3.11 支持  | 470 |
| 18.3.6 明显  | 468 | 18.3.12 多样性 | 470 |
| 18.3.7 个性化 | 468 | 18.4 小结     | 471 |
| 18.3.8 安全  | 468 | 18.5 新手入门须知 | 471 |
| 18.3.9 满意  | 469 | 18.6 模拟面试问答 | 471 |

## 第四篇 软件测试的学习和研究

|                         |     |                              |     |
|-------------------------|-----|------------------------------|-----|
| <b>第 19 章 软件测试的学习环境</b> | 473 | 19.5 新手入门须知                  | 489 |
| 19.1 学习氛围的建立            | 474 | 19.6 模拟面试问答                  | 489 |
| 19.1.1 培训导师制度           | 474 | <b>第 20 章 软件测试的研究方向与个人发展</b> | 491 |
| 19.1.2 读书会              | 477 | 20.1 软件测试角色与其他项目角色的可转换性      | 492 |
| 19.1.3 找个师傅学习软件测试       | 478 | 20.1.1 转向售前                  | 492 |
| 19.2 软件测试经验的总结          | 479 | 20.1.2 转向售后                  | 493 |
| 19.2.1 测试知识库的建立         | 479 | 20.1.3 转向开发                  | 494 |
| 19.2.2 办一份内部期刊          | 480 | 20.1.4 转向 QA                 | 495 |
| 19.2.3 测试管理经验的总结        | 481 | 20.2 测试人员的发展路线               | 496 |
| 19.3 软件测试的交流            | 484 | 20.2.1 管理路线                  | 497 |
| 19.3.1 日常的交流            | 485 | 20.2.2 技术路线                  | 498 |
| 19.3.2 专门的交流            | 485 | 20.3 软件测试的研究方向               | 499 |
| 19.3.3 与开发人员的交流         | 486 | 20.3.1 软件测试中的数学              | 499 |
| 19.3.4 与管理层的交流          | 488 | 20.3.2 软件测试工具设计              | 500 |
| 19.3.5 外部交流             | 488 | 20.4 小结                      | 500 |
| 19.4 小结                 | 489 |                              |     |

## 第1章

# 软件测试概述

图 1-1 Hopper 关于飞虫事件的玩笑



有人把软件产品与药品并称为世界上两种无法根除自身的缺陷，但却被允许公开合法销售的产品。人们明知道药物不可能百分之百治疗疾病，而且肯定存在一些副作用，但还是会购买。软件产品也一样，人们也知道软件不是百分之百可靠，但还是越来越依赖它们。

在其他行业，如果产品存在明显的不可预测的缺陷，市场、用户和法律都会做出强烈的反响。软件产品的缺陷难以根除，但是可以通过加强软件测试来控制质量，通过修正缺陷来提高软件产品的质量。

软件测试行业是一个新兴的行业，尤其是在国内。称之为“行业”是因为测试已经不是以前单纯地依附在软件开发过程中的一种可有可无的角色，而是发展到了足以成为专门的行业。

软件测试开始得到越来越多人的重视。第三方测试、测试外包的出现，测试培训、咨询、考证的红火，软件测试网站的增多，软件测试专门杂志的出现，种种迹象表明，在国外早已是一种专门学科的软件测试，在国内已开始步入可以称之为“行业”的时期。

本章从测试的起源开始介绍，重点描述测试的几个发展阶段，最后分析目前的软件测试现状并展望软件测试的前景。

## 1.1 软件测试的起源

通常称之为 Bug 的软件缺陷是伴随着软件出现的，而软件测试同样是伴随着软件的出现而出现，并且随着软件的 Bug 日益增多，造成了日益严重的质量事故。因此，人们“对抗”Bug 的态度日益坚决，使得软件测试不断地得到加强、重视和持续发展。

### 1. 第一个 Bug 的故事

故事发生在 1945 年 9 月的某一天，在一间老式建筑的窗户外面飞进来一只飞蛾，Hopper 正埋头工作在一台名为 Mark II 的计算机前，没有注意到这只即将造就历史事件的飞蛾。这台计算机使用了大量的继电器（电子机械装置，那时还没有使用晶体管）。

突然，Mark II 死机了。Hopper 试了很多次还是不能启动，Hopper 开始用各种方法查找问题，看问题究竟出现在哪里，最后 Hopper 确定是某个电路板的继电器出错了。Hopper 观察这个出错的继电器，惊奇地发现一只飞蛾躺在里面。Hopper 小心地用镊子将飞蛾夹出来，用透明胶布贴到“事件记录本”中，写上“第一个发现虫子的实例”。

Hopper 的事件记录本，连同那只飞蛾，现在都陈列在美国历史博物馆中。如图 1.1 所示的照片就是那只飞蛾以及 Hopper 的记录。

### 2. 几个导致严重错误的 Bug

软件的 Bug 事件发生了大半个世纪后，没有表明要停止的迹象，而是愈演愈烈。最近发生的几起 Bug 事件则让人们痛苦不堪。

(1) 2007 年 6 月，某热门的在线股票选购竞赛系统软件中存在一个缺陷，此缺陷可以导致不公平的竞争，从而获取高额

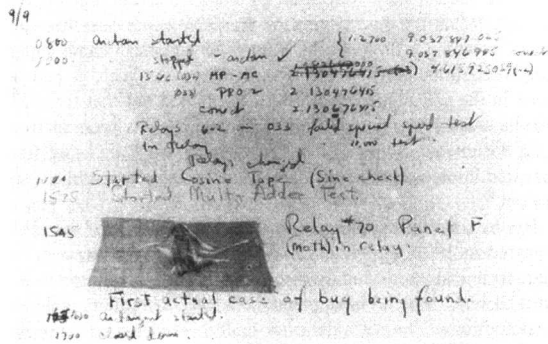


图 1.1 Hopper 关于飞蛾事件的记录

的竞赛奖金。调查结果发现，原本前 5 位的优胜者都要取消资格。

(2) 2007 年 4 月，某软件缺陷导致某地铁系统的列车火灾，软件未能检测和防止剩余动力在设备中的使用，导致列车过热而起火。

(3) 2007 年 3 月，成千上万台某种型号的医疗设备被召回，只是为了修正一个软件错误。据报道，这个错误将会导致不能可靠地指示什么时候供给设备的可用功率过低。

类似的报告数不胜数，据美国国家标准和技术协会 (National Institute of Standards and Technology, NIST) 在 2002 年公布的一项关于软件缺陷引起的经济损失的报告中的数据表明，由于软件缺陷造成的美国经济损失高达 595 亿美元。

### 3. 软件测试的起源

早在我国周朝的时候，就有叫“人”、“氏”的工官，相当于职能工长，他们懂技术，直接负责管理制作器物的工匠，以保证器物制造的质量。因此这些人堪称历史上最早的测试员。

1961 年，一个简单的软件错误导致美国大力神洲际导弹助推器的毁灭。这个简单但是昂贵的错误，致使美国空军强制要求在以后所有的关键发射任务中，都必须进行独立的验证。从此建立了软件的验证和确认方法论。软件测试也从那时候开始兴起。

从上面的几个故事中可以看出，生活中存在这样或那样的错误，促使人们提出相应的管理措施，由此看来，测试多少有点包含对产品制造者进行管理的含义。虽然可能不是直接地管理，而是间接地通过检查产品质量来对制造者进行管理。

人类从很早的时候就已经知道，不能自己检查自己的工作产品，必须由其他人来检查，以保证公平、公正和客观性。但是在软件这个人类的新事物开始出现时，人们似乎并没有意识到这条规律的重要性。

随着不断发生的错误导致很多严重问题的出现，人们开始反省，知道仅靠制造者本身对自己的产品进行检查和验证存在很大的弊端，因此引入了独立的检验者。

然而，软件测试已经历了大半个世纪的发展，软件缺陷仍然大量存在。一方面是因为软件越来越复杂，另一方面与软件测试技术的发展缓慢也有一定的关系。

## 1.2 软件测试的发展

随着软件行业的发展，软件测试也在不断地发展，软件测试大概经历了如图 1.2 所示的几个重要阶段。

### 1. 软件调试

早期软件的复杂度相对较低，软件规模也比较小，因此软件错误大部分在开发人员的调试阶段就发现并解决了，这个阶段的测试就等同于调试。

现在，大部分开发工具都集成了调试工具，调试已成为开发工作中不可或缺的一部分。甚至测试脚本的开发工具也会集成基本的调试功能，如图 1.3 所示的就是 TestComplete 的调试工具栏。

开发人员对自己的程序进行简单的测试，这个阶段处于软件测试的原始阶段。

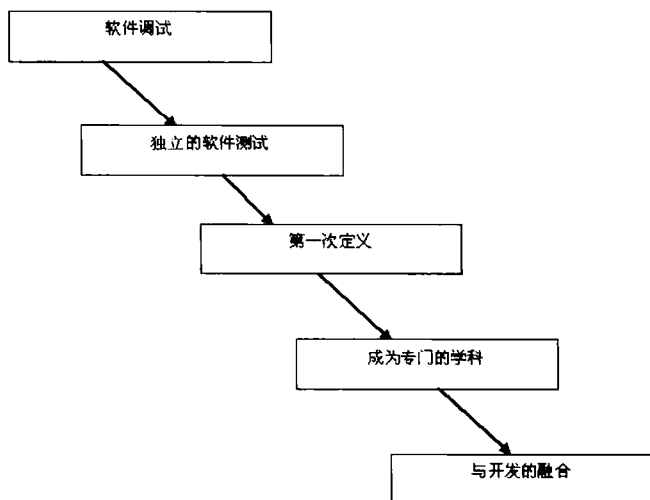


图 1.2 软件测试发展的几个阶段

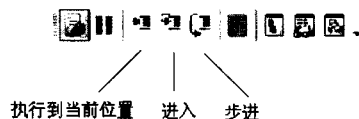


图 1.3 TestComplete 的调试工具栏

## 2. 独立的软件测试

在 20 世纪 50 年代和 60 年代，人们开始意识到仅仅依靠调试还不够，必须引入一个独立的测试组织来进行软件测试。

这个阶段的测试绝大部分是在产品完成后进行的，因此测试的力度、时间都非常有限，软件交付后还是存在大量的问题。

这个阶段没有形成任何测试方法理论，主要靠对错误猜测和经验推断。也没有对软件测试的定位和软件测试的真正含义进行深入的思考。

## 3. 软件测试的第一次定义

1973 年，Bill Hetzel 博士给出了软件测试的第一个定义：“软件测试就是对程序能够按预期的要求运行建立起一种信心”。

1983 年，Bill Hetzel 博士对这个定义进行了修订，即：“软件测试就是以评价一个程序或系统的品质或能力为目的的一项活动”。

因此这个阶段的形成对软件测试的认识就是：软件测试是用于验证软件产品是否是正确工作的、符合要求的。

但是同一时期，Glenford J. Myers 则认为，软件测试不应该专注于验证软件是工作的，而应该将验证软件是不工作的作为重点，他提出的软件测试定义是“测试是以发现错误为目的而运行的程序或系统的执行过程”。

## 4. 软件测试成为专门的学科

20 世纪 80 年代后，软件行业飞速发展，软件规模越来越大，复杂度越来越高。人们对软件的质量开始重视。软件测试的理论和技術都得到了快速的发展。人们开始把软件测试作为软件质量保证的重要手段，用于评价软件的质量。

1982 年在美国北卡罗来纳大学召开了首次软件测试的正式技术会议，软件测试理论开始迅速发展，开始出现各种软件测试方法和技术。

1983 年，电气与电子工程师协会（Institute of Electrical and Electronics Engineers, IEEE）对软件测试做了如下定义：