

软件工程师丛书

软件测试自动化技术 与实例详解

[美] Mark Fewster & Dorothy Graham 著

舒智勇 包晓露 焦 跃 等译


Addison
Wesley



电子工业出版社
Publishing House of Electronics Industry
URL: <http://www.phei.com.cn>

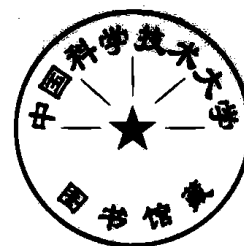
软件工程师丛书

软件测试自动化技术 与实例详解

Software Test Automation

[美] Mark Fewster & Dorothy Graham 著

舒智勇 包晓露 焦 跃 等译



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 提 要

软件测试是软件生命周期的一个重要阶段, 软件测试, 特别是测试自动化技术是当前国际软件界最有争议, 亟待发展的技术。

本书从测试自动化概念开始, 全面说明了如何将手工测试过程自动化, 自动化测试的脚本技术, 验证, 比较及自动化, 测试敏感性, 测试件结构等。并且介绍了如何选择和使用自动化测试工具, 给出了多个应用实例。最后讲解了第三代 ISS 自动化测试系统和第五代方法, 提供了一个自动测试过程的实例。

本书无论对软件开发人员, 系统管理人员都有很高的实用价值, 大专院校计算机专业的师生也可通过阅读本书学习软件测试自动化的有关理论和技术。

Authorized translation from the English language edition published by Addison-Wesley Copyright ©1999.

SIMPLIFIED CHINESE language edition published by Publishing House of Electronics Industry. Copyright 1999.

本书中文简体专有翻译出版权由美国艾维朗公司授予电子工业出版社, 并可在全球出版发行。该专有出版权受法律保护。

图书在版编目 (CIP) 数据

软件测试自动化技术与实例详解/(美)菲尔斯特(Fewster, M.). (美)格雷厄姆(Graham, D.)著; 舒志勇等译. —北京: 电子工业出版社, 1999.12

(软件工程师丛书) ISBN 7-5053-5732-8

I. 软... II. ①菲...②格...③舒... III. 自动化技术—应用—软件质量—测试, IV. TP311.5

中国版本图书馆 CIP 数据核字(1999)第 74652 号

丛 书 名: 软件工程师丛书

书 名: 软件测试自动化技术与实例详解

原 书 名: **Softwar Test Automation**

著 者: [美] Mark Fewster & Dorothy Graham

译 者: 舒智勇 包晓露 焦 跃 等

责任编辑: 寇国华

特约编辑: 杨宝珍

印 刷 者: 北京天竺颖华印刷厂

出版发行: 电子工业出版社 URL:<http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 33.75 字数: 690 千字

版 次: 2000 年 1 月第 1 版 2000 年 1 月第 1 次印刷

印 数: 5000 册

定 价: 56.00 元

书 号: ISBN 7-5053-5732-8
TP • 2956

著作权合同登记号 图字: 01-1999-3116

凡购买电子工业出版社的图书, 如有缺页、倒页、脱页、所附磁盘或光盘有问题者, 请向购买书店调换。

若书店售缺, 请与本社发行部联系调换。电话 68279077

出版说明

近年来，随着计算机技术，特别是 Internet 网络的高速发展，计算机应用已经深入到人类社会的各个行业，各个领域，甚至千家万户。信息化社会、网络时代已经离我们不远了。

然而，计算机技术的发展，计算机应用的普及和深入，引发了对计算机软件的要求更广、更精、更加简便，以解决层出不穷的实际应用的问题。这样一来，非但没有解决十余年前专家们曾经预见的软件危机，反而使这一危机愈演愈烈。在世界范围内，软件开发人员的匮乏，高水平的操作系统、开发语言及应用软件的不足，各种补丁程序满天飞，已经成为计算机应用继续发展的一大障碍。

在我国，软件危机也同样存在，解决这一问题是目前刻不容缓的大事。

更值得引起我们重视的是，中国软件业要走自主创新之路。国际几大软件公司雄厚的技术和资金优势，已经垄断了重要的软件领域，并不断地推出新版本。后来者既要学习借鉴他人的技术，又绝不能永远跟着他人走下去。只是热衷于在他人的新版本上做一些表层开发工作是短视的，长此下去会逐渐丧失自主开发软件的能力。因此，在软件的学习和应用上，要学会掌握核心技术，即软件开发的思路和基本方法，并根据实际工作中提出的问题开发有自主知识产权的创新软件。

正是基于这种形势和认识，为发展我国的计算机软件开发事业尽一份出版工作者的责任，我们推出了这套《软件工程师丛书》。

这套丛书是为所有软件工程师和学习软件开发的计算机用户编写的，内容涵盖计算机软件开发的方方面面，其中既有国内作者编著的书籍，也有从国外精选引进的外版书。

为保证丛书的质量，我们选择的作者都是工作在计算机应用第一线，具有丰富软件开发和应用经验的学者、专家和高级工程师，外版书的译者都有多年计算机图书的翻译经验。

我们出版这套丛书的指导思想是帮助软件开发人员提高技术水平，解决他们在软件开发和应用过程中遇到的各种问题。

这套丛书大致可分为四种类型：

一是实例型，如《Visual Basic 6.0 中文版编程实例详解》，通过大量有用的实例说明如何使用某种流行语言开发自己的应用系统。

二是实用型，如《Windows 2000/NT 疑难问题详解》，回答在应用某种操作系统中遇到的各种疑难问题。

三是技术型，如《软件测试自动化技术和实例详解》，介绍有关软件测试技术及其在实践中的应用。

四是手册型，如《Visual C++6.0 类库大全》，是软件工程师的必备手册，可从中随时查阅所需的内容。

我们相信这套丛书对软件工程师和学习软件开发、应用的读者会有所帮助，我们希望听到读者宝贵的建议和意见。同时，希望更多的作者和我们联系，出版更多更好的书籍，来充实这套软件工程师丛书。

我们曾经努力，我们正在努力，我们仍将努力。

電子工業出版社

译者的话

随着计算机应用日益普及和深化，现代软件的规模越来越庞大。以前用手工作坊式方法开发出来的许多大型软件，由于没有进行软件质量管理，因此几乎无法维护，致使项目报废，造成大量人力、物力浪费。如何提高软件质量，保证软件安全性是一个涉及面广、难度很大的课题。超高质量软件的开发技术将是打开 21 世纪高技术市场的钥匙，而软件测试则是软件质量保证中的关键技术。一个大型应用软件系统在开发过程中以及研制完成后，如何检验系统是否可靠、是否达到预期目标，成为人们越来越关注的问题。

软件测试一般分为手工测试和自动化测试，我国在软件测试方面起步较晚，并且主要在项目组内部进行手工测试，近几年才逐渐引起重视。但如何进行软件测试，特别是如何进行自动化软件测试，则是项目管理层和软件测试人员急需了解的内容。目前测试执行工具已经非常流行，使用较多的是捕获回放工具，我国也已逐步引入一些捕获回放工具，如译者单位就购买了 SQA 自动化测试工具。但许多单位和机构的测试自动化设计缺乏科学的方法，因此效果并不理想。

本书是介绍软件测试方面的专门书籍，重点介绍了如何进行自动化软件测试，如何正确选择自动测试工具，以在中、大型的项目测试中节省时间和金钱，取得效益。在本书的第二部分还给出了软件测试方面的专题研究以及多年从事软件测试人员（其中包括 Microsoft 的软件测试人员）亲身的经验和教训，相信对我国的软件测试工作会有较大的益处。

本书既适用于软件测试人员、软件测试工具的开发人员，也适用于需要进行软件测试的大中型项目的管理人员以及测试工具的使用人员，也可以作为计算机专业的大中专院校的学生参考书。

参加本书翻译工作的有舒智勇、包晓露、焦跃、熊斌斌、张燕、陈剑、张少斌、芮莹、李岩、赵红梅、谈学超等。

翻译若有不妥之处，敬请读者批评指正。

译者

一九九九年十二月

前言

本书内容

本书介绍了如何构造或建立一个自动测试机制，使得通过使用测试执行工具在中型或大型规模的自动测试中可获得持续的利益，对在一个组织内正确地选择工具实现自动测试提供实际的指导。

本书还收集了一些专题研究和用户文章，总结了在自动化测试中成功与失败的经验，可为用户提供一些有益的帮助。

本书的对象

本书的宗旨是为准备进行自动化测试的人士提供帮助，使其少走弯路。

本书的对象包括：

- 计划购置测试执行自动化工具者。
- 已经购置了测试执行自动化工具，但存在问题或收益不大者。
- 在组织内部负责测试自动化的人员。
- 自主开发测试执行自动化工具者。
- 希望测试自动化确实能够提供利益的技术管理者。
- 期望能够很好地实现测试自动化的测试者和测试管理者。
- 管理咨询人员和测试咨询人员。
- 测试工具销售商。

为何阅读本书？

如果准备进行自动化测试，本书将提供很好地帮助。本书的适用场合如下：

- 如果在小范围内进行自动化测试，可以选择阅读其中有用的指导内容。
- 如果进行数百个自动化测试，本书将提供有效的帮助。
- 如果进行数千个自动化测试，若按本书的指导实施，将获得长远的利益。

在下列情况下不需要阅读本书：

- 打算永远进行手工测试。
- 仅希望很少的自动化测试，如数十个。
- 测试仅运行一次以后不再需要。
- 不在乎测试自动化的时间及开销比手工测试还要长、还要大。

自动化测试越多，从本书获得的利益就越多。

无序测试

本书有一个类似的主题，即结构化程序设计。在使用结构化技术以前，缺乏软件工程的科学方法，似乎只要了解编程语言以及如何编译就可以了。这种“意大利面条式的代码”引发了一系列严重问题，即难以理解、难以测试以及维护成本高。经过不断地摸索才有了科学的方法，即在代码实现之前进行需求分析和软件设计。这种结构化方法的好处是大大提高软件开发的可维护性，可理解性以及可测试性。

目前，测试执行工具已经非常流行，特别是捕获回放工具，但许多组织的测试自动化设计缺乏科学的方法。测试执行工具最大的好处是将枯燥但又必须做的任务自动化。但如果仅知道脚本语言以及如何使用测试执行工具，仍会导致“意大利面条式的测试”的难以理解，难以维护，因而抵消了自动测试的优势。本书的主题“测试件的设计技术”相当于二十多年前的“软件设计技术”。

本书的内容

本书不包括目前市场上某个工具的任何信息(第二部分专题研究中提到的信息例外)。这样做有两个原因，第一，本书是关于目前或不久的将来使用测试执行工具的有关技术。因此本书给出一般原则而不针对任何特定工具；第二，测试执行工具市场变换莫测，因此特定信息很快就会过时。

本书不包括测试技术，即如何设计好的测试事例。尽管这是一个重要的专题，但不属本书的范畴。

本书讨论测试执行自动化，而不是测试输入的自动生成，第1章将说明这样做的理由。

如何阅读本书

浏览某一章节时不用阅读前面所有的章节。本书分为两大部分，第一部分为技术性的指导，有助于构造有效的测试自动化机制。第二部分为测试自动化的专题研究，介绍不同组织的成功经验，还包括一些用户合理建议的文章。第二部分的任何章节可以独立阅读，其中的引言介绍了专题研究的主要内容。

对于本书中使用的术语，会有不同的理解。本书后面的“词汇”中给出了这些术语的定义。

下表根据用户目标给出如何阅读本书的建议。

目标	首先阅读的章节
购置测试执行工具	1, 2, 10
想知道为什么测试自动化失败的管理者	1, 2, 7, 8
使用测试执行工具自动化测试即将要编写脚本	2, 3, 4, 5(然后 6, 7, 8, 9)
关心从测试或测试自动化中获得的益处	1, 8
说服组织内的其他人员使用已有的工具	1, 11
希望了解其他人是如何测试的	第二部分任何一章

(续)

目标	首先阅读的章节
希望知道为什么测试自动化存在问题	12
在自动测试的维护中遇到问题	7, 3, 5
希望得到其他人的建议	22, 24, 26, 27, 28
对“失败”故事感兴趣	19, 13

第一部分概述：执行自动测试的技术

第一部分的每一章包括一个小结，概括这一章节的主要内容。下面给出每一章的重点：

第 1 章为本书内容的一般介绍。讨论测试以及测试科学与本书主题的区别，并说明了为什么本书主要的焦点为测试执行自动化。

如果没有体会到兴高采烈的购置与实际使用的效果之间的差距，那么第 2 章将使你增长见识。本章通过手工测试一个简单的应用程序(Scribble)，然后使用捕获回放工具的基本功能。

第 3-9 章为本书的技术内容部分。第 3 章介绍 5 种不同的脚本技术，第 4 章讨论自动化比较技术，第 5 章描述实用的测试件体系结构，第 6 章介绍设置和清除自动化的过程，第 7 章为测试件的维护，第 8 章讨论测试与自动化体系质量的度量，第 9 章为其他一些重要专题介绍。

第 10 章介绍工具的评估和选择，第 11 章为组织内的工具实现。

致谢

感谢下列人士为本书初稿提出了宝贵意见：

Yohann Agalawatte、Steve Allott、Stale Amland、Hans Buwalda、Greg Daich、Peter Danson、Noeman Fenton、Dave Gelperin、Paul Gerrard、David Hedley、Bill Hetzel、Herb Isenberg、Ed Kid、David Law、ShariLawrence-Pfleeger、Aiden Magill、Pat McGee、Geoff Quentin、David Ramsay、Krystyna Rogers、Hans Schaefer、Melanie Smith、Graham Titterington、Otto Vinter、Jelena Vujatov、Stuart Walker、Robin Webb 以及 Kerry Zallar。

特别要感谢专题研究的作者在百忙之中准备材料。感谢 Keiron Marsden 为第 12 章及时带来的资料，Marnie Hutcheson 提供了有益的观点和课程材料，Clive Bates、Peter Oakley、Paul Herzlich、Simon Mills、Steve Allott、Bob Bartlett、Susan Windsor 及 Lloyd Roden 提供了各 UK 界自动的内容。特别感谢 Marnie Hutcheson 及 Stale Amland 向所有的作者如实地说明了遇到的问题及一些经验教训。感谢两组作者给出了有效的可维护的自动化测试的经验：Hans Buwalda 和 Iris Pinkster 为“Action Words”方法做出了贡献，Graham Freeburn 和 Graham Dwyer 及 Jim Thomson 为“RadSTAR”方法做的贡献。特别感谢最后三章的 US 作者：Linda Hayes 友好地允许从其“测试自动化手册”中重新组织许多节的内容，Chip Groder 多年来在建立

GUI 系统的有效测试自动化方面有独到的见解，Angela Smale 根据其对 Microsoft 不同应用的自动测试经验，给出了有益的建议。感谢 Roger Graham 在我们写“Scribble”应用举例时给予的帮助。还要感谢 Sally Mortimore 编辑在过去的三年中热心和耐心的支持。

此外，感谢那些参与我们的有关测试自动化教程并提出建议的人士。

参考文献如下：

CAST Report ,Cambridge Market Intelligence,1995(第 10 章和第 11 章)

The Journal of Software Testing,Verification and Reliability,1991, Sigma Press, 现由 Wiley 出版(第 12 章)

Proceedings of the International Conference on Testing Computer Software,1991(第 12 章)

Proceedings of the EuroSTAR conference,1993(第 7 章)。

如果在致谢部分有所遗漏，我们表示诚挚的歉意。

马克.福斯特 多罗西.格林汉姆

1999 年 5 月

<http://www.grove.co.vk>

目 录

第 1 章 测试自动化概念	1
1.1 引言	2
1.2 测试与测试自动化概念不同	2
1.2.1 测试	2
1.2.2 测试自动化	3
1.3 V 模型	4
1.4 支持生命周期测试的工具	5
1.5 测试自动化的承诺	6
1.6 测试自动化普遍存在的问题	7
1.7 测试活动	9
1.7.1 标识测试条件	10
1.7.2 设计测试事例	10
1.7.3 建立测试事例	12
1.7.4 执行测试事例	12
1.7.5 将测试结果与期望输出比较	12
1.8 自动测试设计	13
1.8.1 适合自动化的活动	13
1.8.2 自动测试事例设计	14
1.9 软件测试自动化的限制	17
1.9.1 不能取代手工测试	17
1.9.2 手工测试比自动测试发现的缺陷更多	18
1.9.3 对测试质量的依赖性极大	18
1.9.4 测试自动化不能提高有效性	18
1.9.5 测试自动化可能会制约软件开发	18
1.9.6 工具本身并不具有想象力	18
小结	19
第 2 章 捕获回放并非测试自动化	21
2.1 应用举例: Scribble	22

2.1.1	测试事例举例：在 Scribble 中编辑一个列表	22
2.1.2	测试 Scribble 列表	22
2.1.3	测试输入	23
2.2	如何将手工测试过程自动化?	29
2.2.1	Ad hoc 测试：非脚本化	30
2.2.2	模糊的手工脚本	31
2.2.3	详细的手工脚本	32
2.3	测试执行自动化：输入	34
2.3.1	测试输入自动化	35
2.3.2	回放录制脚本	35
2.3.3	自动脚本有别于手工脚本	35
2.3.4	使用自动化测试输入的益处	37
2.3.5	录制手工测试的缺点	38
2.3.6	建议：不要仅采用录制的方式进行自动化测试	39
2.3.7	需要手工验证的自动执行	39
2.4	自动测试结果比较	40
2.4.1	确定何时比较测试事例的输出	40
2.4.2	比较大范围?	40
2.4.3	动态比较	43
2.4.4	执行后比较	43
2.4.5	自动比较信息必须手工检查	45
2.4.6	自动化测试比较十分重要	46
2.5	测试自动化的下一个步骤	46
2.5.1	为什么第二次测试失败	47
2.5.2	其他验证需求	50
2.5.3	验证修改文件或数据库的方法	51
2.5.4	所有的文件哪里去了?	51
2.6	不可自动实现自动化测试过程	52
	小结	52

第 3 章 脚本技术

3.1	引言	56
3.1.1	与编程的相同点	56
3.1.2	脚本的共性问题	56
3.1.3	测试事例的设计和实现	60
3.1.4	有关脚本文档的建议	61
3.2	脚本技术	63
3.2.1	线性脚本	63

3.2.2 结构化脚本	65
3.2.3 共享脚本	66
3.2.4 数据驱动脚本	70
3.2.5 关键字驱动脚本	74
3.3 脚本预处理	77
3.3.1 脚本预处理功能	77
小结	81

第4章 自动比较

4.1 验证、比较及自动化	86
4.1.1 比较验证	86
4.1.2 计划比较和随意比较	86
4.1.3 预测输出及验证实际输出	87
4.1.4 自动比较的原因	87
4.1.5 比较的内容	88
4.1.6 自动比较的局限性	88
4.2 比较器的用途	89
4.2.1 可比较的内容	89
4.2.2 比较器告诉用户的内容	89
4.2.3 比较器不能告诉用户的内容	90
4.3 动态比较	90
4.3.1 定义	90
4.3.2 支持和实现工具	90
4.3.3 测试事例智能	91
4.3.4 复杂性与维护费用成正比	91
4.4 执行后比较	91
4.4.1 定义	91
4.4.2 支持工具	92
4.4.3 比较顺序和结构	92
4.4.4 主动与被动的执行后比较	93
4.4.5 实现执行后比较	94
4.5 简单比较	97
4.6 复杂比较	97
4.6.1 需要复杂比较的原因	97
4.6.2 简单屏蔽	98
4.6.3 屏蔽的搜索技术	99
4.6.4 使用正则表达式的搜索技术	100
4.6.5 实现复杂比较	100

4.7	测试敏感性.....	101
4.7.1	敏感与健壮的测试.....	101
4.7.2	敏感测试与健壮测试之间的权衡.....	101
4.7.3	冗余性.....	102
4.7.4	测试敏感性的策略.....	103
4.8	比较不同类型的输出.....	104
4.8.1	基于磁盘的输出.....	104
4.8.2	基于屏幕的输出.....	107
4.8.3	其他输出类型.....	110
4.9	比较过滤器.....	110
4.9.1	建立执行后比较工具的实用方法: 比较处理过程.....	110
4.9.2	过滤器的定义.....	111
4.9.3	实现比较处理过程.....	112
4.9.4	过滤器的优缺点.....	112
4.9.5	过滤器事例.....	114
4.9.6	连续用多个过滤器.....	115
4.9.7	标准化比较.....	119
4.9.8	产生预期输出.....	119
4.10	比较准则.....	120
4.10.1	保持简单.....	120
4.10.2	编制比较的文档.....	120
4.10.3	尽可能标准化.....	120
4.10.4	分割和驾御.....	120
4.10.5	记住效率.....	120
4.10.6	避免比较位图.....	121
4.10.7	敏感和健壮测试间平衡的目标.....	121
	小结.....	121
第 5 章	测试件结构.....	123
5.1	测试件结构的定义.....	124
5.1.1	术语.....	124
5.2	要解决的关键问题.....	125
5.2.1	规模.....	125
5.2.2	重用.....	126
5.2.3	多版本.....	127
5.2.4	独立于平台和环境.....	128
5.3	方法.....	128
5.3.1	介绍.....	128

5.3.2	基本概念.....	129
5.3.3	测试件组.....	131
5.3.4	测试集.....	136
5.3.5	测试件库.....	137
5.3.6	测试结果.....	140
5.3.7	物理实现.....	141
5.3.8	测试工具的接口.....	150
5.4	是否过度.....	150
	小结.....	151
第 6 章	自动化前后的处理.....	153
6.1	什么是前处理和后处理?.....	154
6.1.1	前处理.....	154
6.1.2	后处理.....	154
6.1.3	为什么使用这些术语?.....	154
6.1.4	为什么要自动化前处理和后处理?.....	155
6.1.5	每次存储和恢复时都要设置吗?.....	156
6.2	前处理和后处理.....	156
6.2.1	前处理任务.....	156
6.2.2	后处理任务.....	157
6.2.3	不同阶段的前处理和后处理.....	158
6.3	测试事例执行后应该发生什么?.....	160
6.3.1	正常结束之后.....	160
6.3.2	非正常终止之后.....	160
6.4	执行中的问题.....	161
6.4.1	例子测试事例.....	161
6.4.2	脚本.....	161
6.4.3	使用命令行文件.....	164
6.4.4	数据驱动的方法.....	164
6.4.5	关键字驱动的方法.....	164
6.4.6	怎样使用测试事例定义文件.....	165
	小结.....	166
第 7 章	建立可维护的测试.....	167
7.1	维护过程中存在的问题.....	168
7.2	测试维护的属性.....	168
7.2.1	测试事例的数量.....	169

7.2.2	测试数据的数量	170
7.2.3	测试数据的格式	170
7.2.4	运行测试事例的时间	171
7.2.5	测试事例的调试能力	172
7.2.6	测试之间相互依赖的关系	172
7.2.7	命名习惯	173
7.2.8	测试复杂度	173
7.2.9	测试文档(管理)	173
7.2.10	其他属性	174
7.3	陷阱	174
7.3.1	工具容易令你出错	174
7.3.2	方法越简单就越容易造成维护的高成本	174
7.3.3	一时的心血来潮	175
7.3.4	投入所得的回报	175
7.4	策略和措施	175
7.4.1	策略	175
7.4.2	措施	175
	小结	176

第8章 度量 179

8.1	为什么测量测试过程和测试自动化?	180
8.1.1	投入回报	180
8.1.2	评估选择、对比选择、监控改进	182
8.1.3	对问题的预警和预测	183
8.1.4	基准与标准的对立或竞争	183
8.2	可测量什么?	183
8.2.1	Gilb 的法律	183
8.2.2	举例说明什么可以度量	184
8.2.3	有用的度量	184
8.3	测试和测试自动化的目的	185
8.3.1	测试的目的	185
8.3.2	测试自动化的目的	185
8.3.3	可达到的目的	187
8.4	软件测试的属性	187
8.4.1	测量测试效果	187
8.4.2	全面测量测试过程	193
8.5	测试自动化的属性	194
8.5.1	可维护性	195

8.5.2 效率.....	195
8.5.3 可靠性.....	197
8.5.4 兼容性.....	198
8.5.5 可用性.....	198
8.5.6 健壮性.....	198
8.5.7 可移植性.....	199
8.6 最好的测试自动化体制	199
8.7 应该度量所有这些否?	201
小结	202

第9章 其他问题..... 203

9.1 应首先被自动化的测试	204
9.1.1 它值得自动化否?	204
9.1.2 要执行测试的类型.....	204
9.1.3 应该首先自动化的测试.....	205
9.1.4 不要一口气自动化太多测试.....	206
9.1.5 寻找“可快速制胜的测试”	206
9.2 选择何时运行的测试	206
9.2.1 怎样选择测试子集.....	206
9.2.2 实现一个“测试选择器”	207
9.3 测试执行的顺序	208
9.3.1 测试分析负担.....	208
9.3.2 建立逻辑层次使分析时间最短	209
9.3.3 运行测试的其他必要条件	210
9.3.4 分布测试.....	210
9.4 测试状态.....	210
9.4.1 成功或失败	210
9.4.2 工具不能判断成功或失败	211
9.4.3 已知的未修复错误.....	211
9.4.4 可能的解决方法	212
9.4.5 测试状态: 预料之中的失败.....	213
9.4.6 测试状态: 未知	213
9.4.7 测试状态小结.....	216
9.4.8 更详细的失败状态.....	216
9.5 为可(自动化)测试性设计软件	217
9.6 同步	218
9.7 监视自动化测试的进展	218