

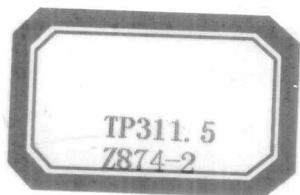
完美测试

软件测试系列最佳实践



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

朱少民 主编



郑州大学 *04010805293W*

完美测试

——软件测试系列最佳实践

朱少民 主编

史 玮	陈 剑	刘 涛	傅 健	黄超男	编著
姜文武	李 茜	李 曾	徐雪梅	张丽丽	



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

TP311.5
7874-2

内 容 简 介

每个人都怀有梦想或理想,测试人员也不例外,希望将自己的工作做得比较完美。本书力求通过一些典型案例告诉大家什么是完美测试,又如何做到完美测试。在给出的例子中,不仅包括功能测试、功能的异常测试、不同平台的功能测试和一些崩溃问题的处理,而且包括国际化测试、性能测试、用户体验测试、Accessibility 测试等,并用较大的篇幅讨论了自动化测试。

为了达到完美测试,建立良好的测试体系、使产品具有可测试性以及缺陷预防等更为重要,对此,本书也做了讨论。最后,本书还展示了软件测试之美,使读者可以领会到软件测试的距离之美、空间之美、技巧之美、辩证之美以及贯穿测试过程的平衡之美。

虽然本书还很难覆盖完美测试应具备的各种方法和实践,但目的在于抛砖引玉,使读者能通过最有效的手段(包括方法、技术和工具)完成所有必要的测试,实现事先所要求的需求和代码的测试覆盖率,最终能准确地给出软件产品一个完整的质量评估,使测试达到相对完美的水平。

本书适合于软件测试工程师、产品管理和管理人员阅读,也适合于大专院校计算机相关专业师生作为参考用书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

完美测试:软件测试系列最佳实践 / 朱少民主编. —北京:电子工业出版社, 2012.3
ISBN 978-7-121-16078-3

I. ①完… II. ①朱… III. ①软件—测试 IV. ①TP311.5

中国版本图书馆 CIP 数据核字(2012)第 026630 号

策划编辑:张春雨

责任编辑:刘 舫

特约编辑:顾慧芳

印 刷:北京丰源印刷厂

装 订:三河市鹏成印业有限公司

出版发行:电子工业出版社

北京市海淀区万寿路173信箱

邮编:100036

开 本:787×980 1/16

印张:22.5

字数:489千字

印 次:2012年3月第1次印刷

印 数:4000册

定价:59.00元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至zlt@phei.com.cn,盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线:(010) 88258888

序

受本书主编朱少民先生的委托已有时日，无奈平日工作繁忙没有余暇。此时在去往中国的飞机上，不再有电话、邮件、微博等的干扰，相对而言，或许这段时光非常适合完成这样的事情。这次去中国，是要参加思科中国研发中心的“春晚”，而本书正与思科（合肥）研发中心有着很深的联系。少民曾是思科-网迅（中国）软件有限公司的资深测试总监（网迅，即 WebEx，于 2007 年被思科收购），现在同济大学软件学院任教。这本《完美测试》，是他和思科（合肥）研发中心的九位同事、思科（苏州）研发中心的一位同事共同创造的成果。这是思科文化所提倡的，同时也是我为思科中国研发团队深感骄傲之处。本书的出版将承载思科中国研发团队更多的知识共享，进而服务社会、贡献 IT 界的社会责任感。

去年，我作为思科全球副总裁全面管理思科协作技术集团的研发团队，而少民由于个人职业发展原因已于更早的时间离开思科，因此我们没有一起工作过。在之后的日子里，我更多地通过其他同事与各种报道了解到一些他的情况：他在 2000 年加入网迅，拥有优秀的软件开发与项目管理经验。但那时，国内软件测试才刚刚起步，业界对软件测试还缺乏足够重视，即使是软件从业人员对于专业的软件测试同样知之甚少。然而，他出任合肥网迅软件公司软件测试经理之后，迅速补充相关知识，同时在美国接受一年有余的在职培训，旋即打造出一支优秀的测试团队。在过去的十多年中，这支团队圆满完成来自美国总部交付的各种测试任务，而 WebEx 产品最终靠质量与创新在其所在领域全球市场获得第一份额。

然而，他本人并不满足于现状，一直期望从更高的层面来把握软件质量管理与测试工作，这种意识促成他不断把实践上升为理论，再将理论应用于实践，并保持和业界分享，最终在短短数年便成为国内最具知名度的软件测试专家之一。少民对于软件测试的把握，早期着重全局，举凡测试知识，皆能梳理充分而精致，所以他的著作可以作为大学教科书反复再版，其中早期的《软件测试方法与技术》，已入选“十一五”国家级规划教材，为软件测试领域教材的佼佼者。稍后的《全程软件测试》则是这种全局观念的集大成者，它提倡全过程的软件测试，即在项目开始就引入软件测试，从需求评审、设计评审到验收测试、部署验证，直至在线测试，将质量保证贯穿整个软件生命周期。之后，他还对于软件测试的热点与新的测试方法倾注心力，《轻轻松松自动化》是对近几

年自动化测试和开源软件热潮的一个响应，而今《完美测试》则尝试以案例教学的方式，给人以更多的软件测试实际工作的切身体验。这次《完美测试》的阵容更强，不仅有自动化测试方面的好手，而且还有可达性、易用性、探索式测试等各个方面的资深测试工程师的参与。《完美测试》不仅涉及前端测试，而且包含后端测试；不仅介绍传统的测试流程，而且还着重讨论了敏捷测试流程。

软件的生命力最终取决于软件质量，但是我们更要时刻关注软件质量内涵的发展与变化。早期，人们只是简单地关注软件功能，只要功能运行正常就可以交付使用；后来，人们逐渐开始关注性能、安全性、可靠性和可用性。而在互联网时代，人们除了关注功能、性能、可靠性之外，更特别关注安全性和易用性，尤其是用户体验。互联网产品的用户（如 Google、Facebook、腾讯的 QQ 等）不只是成千上万，而是几千万，甚至几个亿。面对这种海量用户，完全无法实施传统培训。此时，只有靠产品的优秀设计来提高产品的易用性，让用户获得良好的体验，甚至愉悦的心情，产品才能使用户最终满意。这正是我们经常说的，产品的质量就是用户的满意度。此外，随着社会文明的不断进步，弱势群体也越来越得到重视。与此相应，软件设计和实现开始充分考虑残疾人士的使用，并形成了一套关于软件的 Accessibility 标准。而《完美测试》在易用性、Accessibility 测试方面也有很好的经验分享，这应是本书的一个亮点。

互联网的发展对于软件开发的另一冲击是软件产品发布的频率。在互联网时代，软件服务竞争日趋激烈，促使企业要及时响应用户的变化，快速完成需求的实现和变更，及时推出有竞争力的产品。这也就是敏捷方法流行的主要原因，不论是在国外还是在国内，许多公司都在积极推行敏捷开发流程及其实践。本书则在敏捷测试上不仅有思想、流程上的指导，而且还有具体的案例示范。

《完美测试》以最佳实践的形式勾画与传递软件测试的知识和经验，以实际案例来反映全局的软件测试技术、方法。陆游所谓“纸上得来终觉浅，绝知此事要躬行”，那么这本书算是往“躬行”层面更进一步。本书的内容都是作者们实际工作中的体会与总结。我很乐见这样的工作经验能够铺陈成章，分享给更多有志于软件工业的读者。众所周知，中国的软件业比之硅谷还有一定差距，随着软件外包业在中国的持续繁荣，这种差异必然逐渐缩小，而这项工作跟每位软件从业者息息相关，这是我们的机会，也是我们的责任。

最后，希望更多的读者从本书受益。

许良杰， 思科公司全球副总裁

前言

——从完美说起

当写本书的时候，出于一种心愿，出于一种理想，取书名为《完美测试》。当真正开始写作时，又觉得是一件很困难的事。首先如何理解什么是完美的测试？然后又如何把测试工作做得完美？于是在微博上发出帖子，问大家如何理解什么是完美测试，但得到的答案也是不一样的。

- **@张定勇_darren**：值得思考的问题。我的理解，软件测试是建立对产品信心的过程，将产品引发最终用户损失的风险降低到一个“可接受”的程度。因此，“完美”只能是相对的，或者说测试永远达不到完美的程度（这也是测试的魅力所在）。比如我们都知道的一个基本事实是，产品的缺陷是不可能 100% 消除的。
- **@小马老矣**：简单地讲，满足用户需要就可以了。从实践来看，重在测试需求分析。根据软件需求，先从功能角度，分析所有的功能点及非功能需求，并根据实际应用分级；其次是从技术角度，分析可能对功能产生影响的因素，根据风险高低分级。以此产生测试用例，并妥善安排测试计划。
- **@Eagle_Yu**：我心中“完美测试”的结果是产品推出之后，所有使用该产品的客户都认为它是一款设计合理、性能稳定并且没有什么 Bug 被发现的产品。
（从实现完美测试角度看）我们测试人员希望产品初期 PM、RD、QA、UED 能通力合作：即产品经理能明确规划出用户的需求，用户体验设计能设计出符合用户操作习惯的用户界面和性能，开发人员能分析出将要做的产品技术上是否可行、减少正式开发过程中因为技术达不到而做的返工，品质保证能明确测试范围。而且，测试计划达到测试范围明确、阶段目标清楚、风险分析准确、可执行性高。测试案例达到覆盖面广、优先级权重准确、可维护性强。测试人员经验丰富、时间充裕。同时，开发部门开发能力强、产品高内聚、松耦合。
- **@狂奔的小溪流**：要做到完美测试，就是在规定的时间内通过各种测试方法把软件各个方面（如功能、性能……）的质量状态全面地体现出来。
- **@赵枚**：一个自证明的系统！当这个系统交付的时候，已经完成了其各部分的功能验证。

那我个人的理解又是如何呢？我认为“完美测试”是通过最有效的手段（包括方法、技术和工具）完成所有必要的测试，达到事先所要求的功能需求和非功能需求的测试覆盖率、代码的测试覆盖率，最终能准确地给出软件产品一个完整的质量评估。因此，最有效的手段、必要的测试、测试覆盖率、准确又完整的质量评估是构成“完美测试”的关键要素。

- 1) **最有效的手段：**即以最短的时间或最少的资源来有效地完成给定的测试任务。
自动化测试方法是一种有效的方法，但不一定是最有效的，需要具体问题具体分析。自动化测试方法的投入产出比（ROI）在某些测试场合低于手工测试的ROI，例如一次性的软件项目（而不是产品的系列开发），这时自动化测试方法就往往没有手工测试效率高。适合用自动化测试方法就一定要用自动化测试，该用手工测试的（如软件产品的易用性测试等）就一定用手工测试。
- 2) **必要的测试：**没有执行多余的测试任务，也没有漏掉该执行的测试任务。不需要的测试功能模块、不需要测试的特性、不需要测试的代码等能被隔离出来，它们会被排除在测试范围之外；而且能确定代码改动的影响范围，能够确定测试任务的优先级，从而根据有限的资源和时间，完成所需要执行的测试任务。如果时间非常有限，就完成优先级最高的那部分测试；如果时间稍微宽松些，就完成更多的测试任务，包括优先级较高的那部分。如果有足够的时间，那就完成所有任务的测试。
- 3) **测试覆盖率：**没有覆盖率的衡量，就不能确定测试的效果、不能确定完成了多少应有的测试任务；没有覆盖率的衡量，就很难知道测试什么时候可以结束；没有覆盖率的衡量，就难以对测试工作进行评估，甚至难以控制测试的进度。测试覆盖率包括两方面：实际用户需求验证的覆盖度和被测试代码的覆盖度。
- 4) **准确又完整的质量评估：**在完成必要的测试之后，并有能力发现测试范围中的软件缺陷，然后根据所发现的缺陷，以及前面所参与的需求评审、设计评审和代码评审的结果，针对待发布软件的各种质量特性做出准确的评估，从而能够全面地、准确地了解软件产品的质量，从而能快速地对软件产品能否发布做出正确的判断，对如何持续改进软件质量也能提出明确的举措。

那么，又如何将测试工作做到完美的程度呢？不外乎从人、流程、技术等方面去考虑。人是决定的因素，又是不够稳定的因素，把每个测试工程师都打造成优秀的测试工程师，几乎不可能。但是，我们打造一支优秀的测试团队是完全可能的。即团队的每个成员都有特长，在某个方面有很强的能力、形成互补，这样，从团队来看没有弱项。基于一支优秀的测试团队，我们就能建立或引入良好的测试流程，拥有良好的测试方法、

技能，最终交出一份满意的答卷。

谈到测试流程，首先还得考虑整个组织的研发流程，测试流程也只能算研发流程的一部分，需要和产品开发、项目管理流程和谐共处。如果开发采用 IBM 的统一过程模型（RUP），测试流程可以采用迭代的 W 模型；如果开发采用敏捷开发，测试也得采用敏捷测试的流程。即使选用 W 模型或敏捷测试模型，如何细化流程、如何引入最佳实践等问题还是需要仔细考量的，针对不同的团队、不同的项目类型需要定制、剪裁软件测试流程。例如，互联网创新产品的开发，对软件缺陷有一定的容忍度，软件测试强调协作、灵活和效率，比如某些对于缺陷零容忍度的关键系统（如金融系统、交通自动控制系统等）对缺陷是零容忍度，在测试流程上要严格、规范得多。

从技术上看，构建覆盖整个测试流程的自动化测试框架是重中之重。总体上看，基于自动化测试的方法能够提高测试的效率，也能更好地衡量测试覆盖率，但是自动化测试工具只是工具，很难进行测试需求的分析和设计，测试人员自身的智慧、技术能力依旧是最重要的。他们需要理解软件系统架构的设计、深刻了解软件实现的技术环节，包括如何让开发人员保证软件的可测试性，从而找到有效的测试方法，并能在技术上保证测试的顺利进行，获得高效的生产力。

从“什么是完美测试”到“如何做到完美测试”，当然不是上面几段文字就能说清楚的，甚至难以用一本书把它说清楚。一方面，大家对“什么是完美测试”会有不同的看法；另一方面，每个方面（团队、流程、技术）要说清楚，都需要单独用一两本书完成，加起来，可能需要四五本书才能完全阐述清楚。所以，笔者对写好这本书，一直没有把握，怀着诚惶诚恐的心情，但必须迈出第一步，触发测试人员的不断思考、创新和提高。另外，由于刚到一个新单位工作，有很多事要做，心情也不够平静，对本书的写作也有较大的影响，所以到后来，不得不把如下一些已经写了几页初稿、内容很好的章节删去。

- 让单元测试无处不在。
- 模糊测试——距离之美。
- 安全无底——但测试有方。
- 回归测试——从有到无。
- 测试过程之美——立体之美。
- 测试的三重境界。
- 如何构建一流的测试团队。

希望将来这些内容可以在本书再版时放进来，让大家看完本书后还怀有一丝期盼。

如果说“软件测试是艺术”，多数人都不会同意，他们觉得软件测试是一项技术工作，更乐意去讨论软件测试的自动化测试技术，讨论性能测试、安全性测试的技巧。虽然 Glenford J. Myres 写了一本《软件测试的艺术》，但许多人仍很难从软件测试中获得艺术的欣赏。后来，又有一本叫《测试之美》的书出版了，让我们能欣赏到一些测试的技巧之美，也能获得一些真知灼见、有价值的建议或者寓有挑战性的想法，距离软件测试艺术又前进了一大步，开始领略软件测试之美。

写《完美测试》还有另外一个含义，就是想将测试之美表现出来。因为为数不少的测试人员，特别是测试领域之外的 IT 人，总觉得软件测试缺乏技术含量，是一份枯燥而且重复性很强的工作，测试跟“美”是没有任何关系的。为了改变这种看法，笔者想通过写一本书来展示测试之美，虽然之前已经有一本《测试之美》，觉得还不过瘾。但这本书做得也不够好，多数测试工程师因为是理工科出身、长期从事技术工作，那么仅有的几个艺术细胞早就被磨灭了，所以本书的多数作者还难以写出真正的测试之美，还需要更多的观察和更长时间的修炼。

最后，尽管本书没有完成全部使命，但还是带领大家完成了一趟快速的艺术之旅，在软件测试领域充分地欣赏各种测试之美，包括软件测试的距离之美、空间之美、技巧之美和辩证之美，更重要的是贯穿全部测试过程的平衡之美。当然，本书更多的篇幅还是在讨论如何进行更彻底的测试，从各个方面保证软件产品的质量。这里的“完美测试”，更多倾向于前面所叙述的那部分思想，即如何全面地、完整地、充分地完成测试，将测试工作做到相对完美的程度。

这就是本书写作的初衷和思路，但现在仅仅跨出第一步，算是敏捷方法的前一两个迭代，还需要根据读者的反馈，不断进行修改，不断迭代，争取在不远的将来推出本书的第 2 版、第 3 版，以回馈读者。

朱少民

2011 年秋于上海同济大学嘉定校区

目 录

第 1 章 软件测试的金字塔体系	1
1.1 从 1 个中心到 5 个要素	3
1.2 5 个工作面	5
1.3 8 组关系	6
1.4 13 项原则	8
1.5 21 个关键域	11
1.6 34 个方法	15



朱少民（第 1、2、3、4、18、21 章以及前言、后记的作者）

同济大学软件学院教授，Certified Scrum Master、CSTQB 资深专家和中国科技大学软件学院教指委委员。曾任网迅（中国）软件有限公司 QA 高级总监，创建并领导过几百人的、国际化的软件测试团队。从事软件开发、测试、QA 和过程改进等工作近二十年，在软件工程领域有很高的造诣，在软件测试流程改进、自动化方法和测试管理等方面进行了大量探索和实践，提倡“全过程软件测试”和“缺陷预防”等先进的软件工程思想，先后出版专著《全程软件测试》、《软件测试》、《软件工程导论》和主编《软件测试方法和技术》、《软件项目管理》、《软件质量保证和管理》、《软件过程管理》等多部高等学校精品教材。并先后获得青岛市、合肥市、安徽省、机械工业部等多项科技进步奖，也是国内第一批获得国家人事部认证的“高级程序员”。

个人微博：<http://weibo.com/KerryZhu>

第 2 章 测试架构从何而来	17
2.1 什么是测试架构	18
2.2 测试领域架构	21
2.3 自动化测试架构之说	25
2.3.1 为何要建立自动化测试架构	25
2.3.2 解决什么问题	26
2.3.3 软件开发框架的启发	30
2.3.4 测试自动化框架的基本构成	31
2.4 谁能成为测试架构师	34
第 3 章 如何让缺陷无处藏身	38
3.1 什么是软件可测试性	39
3.2 SOCK 模型和 James Bach 的观点	41
3.3 TDD 和代码的可测试性	43

3.4 设计的可测试性	48
3.5 需求的可测试性	51
第 4 章 可以像这样设计测试用例吗	53
4.1 从需求到测试用例	53
4.2 基于流程图设计测试用例	56
4.3 基于 UML 视图的测试用例设计	61
4.4 小结	65
第 5 章 从虚拟测试环境到一键部署	67
5.1 虚拟出更多的机器	67
5.2 虚拟的疑问	70
5.3 另一种把资源用到极致的方法	71
5.4 一键部署	73
史玮 (第 5、6 章作者) 思科 (Cisco WebEx) 软件测试资深工程师, 6 年自动化测试及相关工具开发经验。最近几年都被公司评为最优秀、最有价值工程师。由他主导开发的一款大规模自动化测试工具为公司节约了上百万美金的硬件成本, 加上超过 180 万的人工测试时间, 使他获得了思科全球包括先锋大奖在内的多项大奖。曾参与《轻轻松松自动化测试》一书的编写工作。 个人微博: http://weibo.com/2341323404	
第 6 章 客户端的 GUI 测试自动化	79
6.1 初识自动化测试	79
6.2 困惑	80
6.3 建议	81
6.4 三类标准控件的不同解决办法	82
6.4.1 标准控件	83
6.4.2 自定义控件	84
6.4.3 自定义控件库	84
6.5 微软的 UIA 和 MSAA	85
6.5.1 MSAA	85
6.5.2 UIA	86
6.5.3 Windows Automation API 3.0	88
6.6 和开发人员合作的好处	88
第 7 章 后台自动化测试	90
7.1 什么是后台测试	90
7.1.1 后台测试的特点	90
7.1.2 后台测试的自动化	91

7.2 后台自动化测试的统一脚本控制	92
7.2.1 自动化测试框架	93
7.2.2 自动化测试脚本的分层实现	93
7.3 后台自动化测试实例	95
7.3.1 测试工具树形图	95
7.3.2 基于 STAF 框架的 Python 脚本	97
7.4 后台大规模性能测试	102
7.4.1 测试工具的管理	103
7.4.2 同步及异步控制模式	103
7.4.3 测试逻辑的同步执行问题	104
7.4.4 测试结果的收集	106
7.5 小结	107



陈剑 (第 7 章作者)

测试工具开发组组长。中国科学技术大学计算机系硕士,最近 4 年一直从事自动化测试框架及测试工具的研究与开发,该测试框架已经成为公司自动化测试的通用平台,同时为公司的大规模性能测试提供解决方案,具有很好的扩展性,实现了模拟数十万人同时在线的大规模性能测试,并很好解决了大规模测试中的工具控制、结果收集等问题。

第 8 章 高亢之龙——JMeter 后台自动化测试	108
8.1 潜龙勿用,见龙在田	109
8.2 终日乾乾,或跃于渊	113
8.3 飞龙在天	117
8.4 亢龙有悔	121
8.5 小结	123



刘涛 (第 8、19 章作者)

北京航空航天大学明星教授张军的学生,曾参与神舟飞船的测试,和嫦娥一号用 490N 发动机阀门测试系统的研发。现任职于思科 WebEx 测试部门,专攻后台自动化测试。擅长工具:JMeter、Ant 等。

第 9 章 重读探索式软件测试	124
9.1 概念及目标	124
9.2 关键要素	125
9.3 常见的认识误区	127
9.4 探索思维“十六”计	128
9.5 如何应用探索式测试	136
9.6 机遇与挑战	138
9.7 小结	139



傅健 (第 9、11 章作者)

吉林大学硕士, 国家认证网络工程师、信息系统项目管理师。发表教育信息化、软件测试相关论文十二篇, 在校期间获吉林大学第二十四届研究生精英杯学术成果奖; 在思科公司工作两年多, 先后参与 Cisco WebEx 的移动项目及文档管理系统(DMS)等多个项目的测试, 其中参与的 BlackBerry Meeting 项目测试工作获得 2010 思科测试革新会议(CTIC)二等奖。现主要专注于“云存储”领域质量保证工作。

个人微博: <http://weibo.com/fujianthinking>

第 10 章 完美的功能测试	140
10.1 测试准备	141
10.2 测试计划	145
10.3 功能测试用例的设计	147
10.4 用例执行的窍门	149
10.5 功能测试的三步曲	150
10.6 查找遗漏问题的七大招	152
10.7 基于组件元素的通用测试点	154
10.8 基于系统行为和操作习惯的测试点	156



黄超男 (第 10、12 章作者)

合肥工业大学软件工程硕士, 主要研究方向是软件测试、配置管理, 特别是在自动化测试工具上进行了比较多的探索和研究, 发表了多篇软件测试相关的论文。2008 年 1 月加入 Cisco WebEx, 作为一名测试工程师, 主要从事 Client 端的功能测试, 特别是在 Mac 平台上的测试, 并负责 Cisco WebEx Client 的多个项目, 积累了丰富的软件测试和管理经验, 还参与了《软件测试方法与技术实践指南》一书的编写工作。

第 11 章 异常测试的方法与技巧	158
11.1 异常测试定义与范围	159
11.2 技巧之一: 改(以彼之道, 还施彼身)	160
11.2.1 更改“自身”	160
11.2.2 更改“相关”	161
11.3 技巧之二: 借(他山之石, 可以攻玉)	162
11.3.1 借 Code Coverage 发现异常	162
11.3.2 借 JUnit+JMockit 测试异常	166
11.3.3 借 iptables 创造异常	167
11.4 异常测试其他技巧	169
11.5 小结	170
第 12 章 如何在苹果上找虫	171
12.1 造成 Mac 系统崩溃的常见操作	171
12.2 通过崩溃日志来定位问题	172

12.3	通过 Console 来定位崩溃问题.....	176
12.4	通过 Shark 工具来定位 Freeze 问题.....	176
第 13 章	穿越时空的国际化和本地化测试	182
13.1	案例背景	183
13.2	时区及夏令时	186
13.3	有关时区的应用测试	188
13.4	本地化测试的方方面面.....	190
13.4.1	语言标识.....	190
13.4.2	文化生活.....	191
13.4.3	书写习惯.....	192
13.5	本地化测试的主要方法.....	193
13.6	本地化测试的常见问题.....	194
13.6.1	人力资源短缺.....	194
13.6.2	开发人员、翻译组、资源文件维护者与测试人间之间的协作.....	195
13.6.3	基础工作的缺乏.....	196



姜文武 (第 13、16 章作者)

兰州大学学士, 软件测试资深工程师, 从事软件测试工作已逾十年, 注重软件测试理论、方法和流程的探讨, 并实际负责或参与大量具体的软件测试项目, 特别是基于 SaaS 平台的企业级网络会议系统的测试, 具有 Web、客户端、服务器端等各方面的丰富的测试经验。曾参与《轻轻松松自动化测试》一书的编写工作。

第 14 章	用户体验的享受	198
14.1	最终用户特点	198
14.1.1	绝大多数的中间用户	199
14.1.2	不同用户需求.....	200
14.2	生活中的用户体验	203
14.2.1	用户体验的复杂性.....	203
14.2.2	视觉反馈.....	204
14.2.3	听觉反馈——声音.....	211
14.3	UE 测试策略	213
14.3.1	少就是多.....	214
14.3.2	工具就放在手边.....	215
14.3.3	提供非模态的反馈.....	215
14.3.4	提供符合情景的信息.....	216
14.3.5	提供直接的操作和图形输入	217
14.3.6	不要用对话框来报告常态内容	218
14.3.7	提供选择而不是提问	218
14.3.8	优化响应能力以调节延迟时间	219
14.3.9	多平台多浏览器支持.....	220

14.4 小结	221
---------------	-----



李茜 (第 14 章作者)

中国科学技术大学软件工程硕士, 现任职思科网讯 (中国) 软件有限公司合肥分公司软件测试工程师, 从事软件测试工作近 4 年, 注重前端 UI 与用户体验的分析研究工作。先后参与并负责基于 Web、Client 等多个大型测试项目, 具有丰富的测试经验, 对软件缺陷具有敏锐的洞察力。

第 15 章 可能没听说过的测试——Accessibility 测试	223
--	-----

15.1 测试的范围和要点	224
15.2 测试工具	226
15.2.1 SOAtest	226
15.2.2 JAWS	228
15.2.3 ACTF aDesigner	229
15.3 Accessibility 测试注意事项	231
15.3.1 规范快捷键, 尽量与系统操作键统一	231
15.3.2 应尽早制定 Accessibility Checklist	232
15.3.3 注意对其他任务的影响	232
15.3.4 Restore Case 要定义明确	232
15.3.5 应从用户使用习惯的角度去考虑测试需求	233



李曾 (第 15 章作者)

思科 (Cisco WebEx) 资深软件测试工程师, 从事软件测试工作 7 年。负责或参与了 URL API、Accessibility、Page/Client UI 等前端测试领域的测试。具有多年的用户体验、流程管理等测试经验。关注于自动化测试方法和测试用例设计的研究, 负责并成功发布了多个软件测试项目。

第 16 章 谁说了算? ——代码覆盖率	234
----------------------------	-----

16.1 衡量代码覆盖率软件的引入	235
16.2 C/C++代码覆盖率测试工具 BullseyeCoverage	237
16.3 开源测试代码覆盖率工具 EMMA	239
16.4 代码覆盖率报告的分析	240
16.5 代码覆盖率工具的相关运用	242
16.6 代码覆盖率工具应用的局限性	243
16.7 小结	244

第 17 章 Bug 的“美丽人生”	246
--------------------------	-----

17.1 报告新的 Bug	246
17.1.1 发现 Bug	247
17.1.2 隔离分析 Bug	248
17.1.3 提交一个新的 Bug	251

17.2	处理已经提交的 Bug	254
17.3	验证 Bug	256
17.4	案例说明	256



徐雪梅 (第 17 章作者)

2006 年毕业于北京交通大学信息与计算科学系。分别就职于博彦科技武汉分公司、思科网迅合肥分公司, 有 4 年多软件测试经验。主要从事于嵌入式固件测试、Windows 客户端以及跨平台功能测试, 并参与重要测试项目的管理, 对测试及项目管理工作有良好的理解和经验。

第 18 章	从亡羊补牢到未雨绸缪	260
18.1	防止“病从口入”	261
18.2	缺陷 RCA 三步曲	262
18.2.1	缺陷的分类统计分析	263
18.2.2	深度分析找到根本原因	266
18.2.3	找到解决问题的办法	268
第 19 章	云中漫步——后端测试流程管理	269
19.1	欢迎来到“云”的世界	271
19.1.1	为什么要进行后台测试	271
19.1.2	Shindig 框架的部署	273
19.2	你不是她, 但你处处为她着想	276
19.2.1	以用户行为为核心	276
19.2.2	以用户数据为核心	277
19.2.3	“虫虫”哪里走	278
19.3	进入云, 必须做好聪明又狂野的准备	279
19.3.1	其徐如林——周密计划, 才能有条不紊	280
19.3.2	其疾如风——快速执行	281
19.3.3	侵掠如火——干吧, 我们就是这个来的	282
19.3.4	不动如山——牢固占领, “云”上一分钟, “云”下十年功	283
19.4	小结	285
第 20 章	敏捷测试——速度之美	286
20.1	敏捷开发和传统瀑布模型比较	286
20.2	快速迭代、发布可工作的软件	288
20.2.1	面对面交流和简洁的文档	290
20.2.2	基于用户故事的测试计划	292
20.2.3	面向敏捷开发团队的高效测试用例	294
20.2.4	持续集成和持续测试框架 Hudson	295
20.2.5	有效管理和跟踪测试任务	297
20.2.6	快速跟踪和解决缺陷	300

20.3 持续回顾和调整	304
20.4 小结：速度之美的真正含义	307



张丽丽（第20章作者）

1999年毕业于安徽大学计算科学与技术系。思科网迅合肥分公司软件测试资深工程师，系统自动化项目测试组组长，Certified Scrum Master，思科敏捷委员会成员，多次获得思科网迅最佳实践奖。2010年成功领导项目组从瀑布模型向敏捷模型的转型，使所在项目组成为中国部第一个成功使用敏捷模型的团队，为提高生产力和质量做出了突出贡献，同时积极组织整个中国部敏捷培训，并分享最佳实践。

个人微博：<http://weibo.com/lukezhang66>

第21章 再谈敏捷测试的方法和实践	310
21.1 测试驱动开发与敏捷测试	310
21.2 敏捷测试自动化	313
21.2.1 自动化测试策略	313
21.2.2 自动化测试框架	314
21.2.3 敏捷测试工具	315
21.3 敏捷测试及其管理	316
21.3.1 敏捷测试流程的优化	316
21.3.2 新功能的测试和回归测试策略	317
21.3.3 基于需求测试和基于风险测试的平衡	318
21.4 有关敏捷测试的思考	319
21.4.1 探索式测试的地位	320
21.4.2 测试人员在敏捷方法中的价值	321
21.5 小结	322
后记——从“软件测试艺术”讲座品味完美	323
空间之美	324
距离之美	324
技巧之美	325
辩证之美	326
白盒测试方法 vs. 黑盒测试方法	327
静态测试 vs. 动态测试	328
被动测试 vs. 主动测试	329
手工测试 vs. 自动化测试	330
计划测试 vs. 探索式测试	332
新功能测试 vs. 回归测试	332
附录A 敏捷方法的相关概念	334
附录B 软件测试的详细分类	336
参考文献	342