

<<探索式软件测试>>

图书基本信息

书名：<<探索式软件测试>>

13位ISBN编号：9787302223849

10位ISBN编号：730222384X

出版时间：2010年4月

出版时间：清华大学出版社

作者：James A. Whittaker

页数：230

译者：方敏;张胜,钟颂东;郭艳春

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>

<<探索式软件测试>>

前言

“用户购买功能的同时也在忍受缺陷。

”——史考特·沃兹沃思(Scott Wadsworth) 任何一个使用过电脑的人都知道软件故障。

从最初的第一个程序到最新的现代应用程序，软件从来没有完美过。

将来这一切也多半不会改变。

不只是软件开发错综复杂和开发人员容易犯错的天性，还有硬件、操作系统、运行时环境、驱动程序、平台、数据库等不断变化，这些加在一起使软件开发任务成为全人类最让人称奇的专业技能之一。

不过，仅仅令人称奇是不够的，正如本书第1章指出的那样，这个世界还要求软件拥有高质量。

显然，关心质量不只是软件测试人员的事。

我们应该用正确的方式构建软件，将可靠性、安全性和高性能等作为系统设计的一部分来考虑，而不是到开发末期才想起它们。

然而，说到理解软件缺陷的本质，测试人员总是站在前沿。

如果没有测试人员在测试的最前沿发挥他们的洞察力、技术以及应变措施，使这样的可能性变为现实，软件质量的全面解决方案差不多算是“镜中花，水中月”。

谈论软件质量的方法有很多，感兴趣的听众也有很多。

本书是为软件测试人员而写的，写的是一种我认为比其他任何缺陷都重要的特殊缺陷：即逃过所有各种检测手段而最终存在于发布产品中的缺陷。

任何一个软件公司发布的产品都有缺陷。

缺陷是怎么引入的？

为什么没有在代码审核、单元测试、静态分析或其他面向开发人员的活动中把它们找出来？

为什么自动化测试没有找出它们？

那些缺陷有些什么特质使其能逃过手工测试？

什么是找出产品缺陷的最好方法？

本书针对的正是最后一个问题。

在第2章“手工测试”中，我提出了一个观点：因为用户是在使用软件过程中找到这些缺陷的，所以我们的测试人员也应该通过使用软件来找到它们。

无论使用自动化测试和单元测试，还是其他一些手段，都难以接触到这些缺陷。

无论测试人员怎么实现自动化测试，即使全部都自动化，这些缺陷还是会处处作怪，并在产品中屡屡重现从而伤害最终用户。

问题在于很多现代化手工测试实践都缺乏目的性，随机性强且重复性强。

有些人可能还会加上一条：手工测试无聊透顶。

本书试图为手工测试流程提供一些指导、技术和规划。

在第3章“局部探索式测试法”中，针对测试人员在运行任何一个测试用例时都需要做出很多细微的战术层面决定，我给出了详尽的指导建议。

测试人员必须决定对于某个特定的输入字段应该使用什么输入值，或者给应用程序使用的文件提供什么数据。

在测试过程中，必须做出许多这样的小决定。

在缺乏指导的情况下，这些决定常常是未经分析且不是最优化的。

在向一个文本框内输入一个数时，选择整数4难道就胜过整数400么？

应该用长度为32字节的字符串还是长度为256字节的字符串？

选择一个而不选另一个是有一定道理的，这一切都取决于处理该输入的软件的具体情况。

鉴于测试人员每天都要做出数百次这样的小决定，在这里提供有效的指导建议显得至关重要。

在第4章“全局探索式测试法”中，针对测试人员在编制测试计划和测试用例设计时需要考虑哪些广泛的战略性问题，我也给出了一些指导建议。

这些技术都基于“漫游测试”(tour)概念，如同一个导游带领旅游团队参观大都市中一系列著名景点一样，这种漫游测试法指出的路线可以指导测试人员如何探索软件的方方面面。

<<探索式软件测试>>

这里的探索并不一定是随机的或者漫无目的的。

本书所记录的方法已经成为微软和谷歌的许多测试人员日常工作的一部分。

诸如“地标测试法”(landmark tour)和“极限测试法”(intellectual's tour)等词汇已经列入了手工测试人员的标准词汇表中。

测试技术以前确实被称作“漫游”，但是用整个旅游业来隐喻软件测试，并在测试实际发布的应用程序时，大规模使用这些隐喻的名称，还属于本书的一个创举。

全局探索式测试法对于制定完整的测试策略给出了指导建议。

例如，如何创建一组特性覆盖率(feature coverage)较高的测试用例？

如何确定是否要在一个单独的测试用例中使用多个特性？

如何创建一个完整的测试用例套件(test case suite)，从而使软件尽可能地满负荷工作以能找到更多重要的缺陷？

这些都是设计测试用例和保证测试套件质量时必须解决的重大问题。

在第5章“混合探索测试技术”中，通过把探索式测试和传统的脚本或基于场景的测试技术相结合，进一步扩展了漫游的概念。

我们将讨论如何修改各种端到端场景(end-to-end scenario)、测试脚本(test script)或用户故事(user story)，来创造更多的变化情况，以激发传统静态测试技术查找缺陷的潜力。

在第6章“探索式测试的实际应用”中，来自微软不同产品组的五位客串作者提供了他们使用漫游技术后得到的经验报告。

这些作者和他们的团队在真实的开发环境中，把漫游方法应用在真实的软件上。

他们记录了各自是如何使用漫游、修改漫游甚至创建自己的漫游的。

这些内容来自于使用漫游法测试重要的关键软件产品的测试人员，属于真正的第一手资料。

最后，我用两章内容总结前面各章所讨论的内容。

在第7章“漫游测试的棘手问题”中，描述了我认为的测试中最困难的几个问题，以及如何将那些具有高度针对性的探索式测试方法融入一个更广泛的解决方案中。

在第8章“软件测试的未来”中，我更进一步讨论在未来几年中，诸如虚拟化、可视化甚至电视游戏之类的技术将如何改变测试的面貌。

附录包括我对测试职业生涯的看法，收集了我以前一些深受读者喜爱的文章(加入了一些新的注解)，其中一些文章已经无法在其他地方看到了。

写这本书对我来说是一种享受，我希望你阅读本书也是一种享受。

<<探索式软件测试>>

内容概要

谈论软件质量的方法有很多，感兴趣的听众也有很多。

本书是为软件测试人员而写的，写的是一种我认为比其他任何缺陷都重要的特殊缺陷：即逃过所有各种检测手段而最终存在于发布产品中的缺陷。

任何一个软件公司发布的产品都有缺陷。

缺陷是怎么引入的？

为什么没有在代码审核、单元测试、静态分析或其他面向开发人员的活动中把它们找出来？

为什么自动化测试没有找出它们？

那些缺陷有些什么特质使其能逃过手工测试？

什么是找出产品缺陷的最好方法？

本书针对的正是最后一个问题。

在第2章“手工测试”中，我提出了一个观点：因为用户是在使用软件过程中找到这些缺陷的，所以我们的测试人员也应该通过使用软件来找到它们。

无论使用自动化测试和单元测试，还是其他一些手段，都难以接触到这些缺陷。

无论测试人员怎么实现自动化测试，即使全部都自动化，这些缺陷还是会处处作怪，并在产品中屡屡重现从而伤害最终用户。

问题在于很多现代化手工测试实践都缺乏目的性，随机性强且重复性强。

有些人可能还会加上一条：手工测试无聊透顶。

本书试图为手工测试流程提供一些指导、技术和规划。

在第3章“局部探索式测试法”中，针对测试人员在运行任何一个测试用例时都需要做出很多细微的战术层面决定，我给出了详尽的指导建议。

测试人员必须决定对于某个特定的输入字段应该使用什么输入值，或者给应用程序使用的文件提供什么数据。

在测试过程中，必须做出许多这样的小决定。

在缺乏指导的情况下，这些决定常常是未经分析且不是最优化的。

在向一个文本框内输入一个数时，选择整数4难道就胜过整数400么？

应该用长度为32字节的字符串还是长度为256字节的字符串？

选择一个而不选另一个是有一定道理的，这一切都取决于处理该输入的软件的具体情况。

鉴于测试人员每天都要做出数百次这样的小决定，在这里提供有效的指导建议显得至关重要。

在第4章“全局探索式测试法”中，针对测试人员在编制测试计划和测试用例设计时需要考虑哪些广泛的战略性问题，我也给出了一些指导建议。

这些技术都基于“漫游测试”（tour）概念，如同一个导游带领旅游团队参观大都市中一系列著名景点一样，这种漫游测试法指出的路线可以指导测试人员如何探索软件的方方面面。

这里的探索并不一定是随机的或者漫无目的的。

本书所记录的方法已经成为微软和谷歌的许多测试人员日常工作的一部分。

诸如“地标测试法”（landmark tour）和“极限测试法”（intellectual's tour）等词汇已经列入了手工测试人员的标准词汇表中。

测试技术以前确实被称作“漫游”，但是用整个旅游业来隐喻软件测试，并在测试实际发布的应用程序时，大规模使用这些隐喻的名称，还属于本书的一个创举。

全局探索式测试法对于制定完整的测试策略给出了指导建议。

例如，如何创建一组特性覆盖率（feature coverage）较高的测试用例？

如何确定是否要在一个单独的测试用例中使用多个特性？

如何创建一个完整的测试用例套件（test case suite），从而使软件尽可能地满负荷工作以便能找到更多重要的缺陷？

这些都是设计测试用例和保证测试套件质量时必须解决的重大问题。

在第5章“混合探索测试技术”中，通过把探索式测试和传统的脚本或基于场景的测试技术相结

<<探索式软件测试>>

合，进一步扩展了漫游的概念。

我们将讨论如何修改各种端到端场景（end-to-end scenario）、测试脚本（test script）或用户故事（user story），来创造更多的变化情况，以激发传统静态测试技术查找缺陷的潜力。

在第6章“探索式测试的实际应用”中，来自微软不同产品组的五位客串作者提供了他们使用漫游技术后得到的经验报告。

这些作者和他们的团队在真实的开发环境中，把漫游方法应用在真实的软件上。

他们记录了各自是如何使用漫游、修改漫游甚至创建自己的漫游的。

这些内容来自于使用漫游法测试重要的关键软件产品的测试人员，属于真正的第一手资料。

最后，我用两章内容总结前面各章所讨论的内容。

在第7章“漫游测试的棘手问题”中，描述了我认为的测试中最困难的几个问题，以及如何将那些具有高度针对性的探索式测试方法融入一个更广泛的解决方案中。

在第8章“软件测试的未来”中，我更进一步讨论在未来几年中，诸如虚拟化、可视化甚至电视游戏之类的技术将如何改变测试的面貌。

附录包括我对测试职业生涯的看法，收集了我以前一些深受读者喜爱的文章（加入了一些新的注解），其中一些文章已经无法在其他地方看到了。

写这本书对我来说是一种享受，我希望你阅读本书也是一种享受。

<<探索式软件测试>>

作者简介

James A. Whittaker, 近日已加入谷歌担任测试工程主管, 他曾在微软担任Visual Studio Team System架构师, 负责为微软测试业务主导产品策略, 并领导内部团队应用探索式软件测试。

Whittaker博士曾在佛罗里达理工学院担任计算机科学教授一职。

在校期间, 他被The Journal of Systems and Software授予“首席学者”称号, 并领导一个研究团队创建了许多领先的测试工具和技术, 包括备受称赞的运行时错误注入工具Holodeck。

Whittaker博士还著有《如何攻破软件》、《如何破坏软件安全》和《如何破坏网络软件》。

他发表过50+有关软件开发和安全的同级评审论文。

他持有安全测试和安全防御技术方面多项发明的专利。

译者简介: 方敏, 现任微软亚洲工程院UIS项目首席测试部门主管, 拥有20年软件测试管理和开发的丰富经验, 曾参加过微软多项重大产品和技术的研制, 包括UIS, Windows Server / Client / Security, SQL Server, Exchange Server, MSN, COM+Services, Windows Media和微软内部IT工具等。

方敏曾在清华大学获得电子工程学士和硕士学位, 在美国新墨西哥技术学院获得计算机硕士学位。

张胜, 现任微软总部高级软件开发测试主管, 拥有10余年软件开发测试和团队管理经验, 参与Visual Studio, SQL Server和Office Live的开发测试与发布, 现主管Office Communications Server本地化软件开发测试工作。

张胜拥有复旦大学计算机系硕士和学士学位。

<<探索式软件测试>>

书籍目录

第1章 软件质量 1 软件的魔力 1 软件失效 4 小结 9 练习题 9 第2章 手工测试 11 软件缺陷的根源 11 缺陷预防和检测 12 缺陷预防 12 缺陷检测 13 手工测试 15 手工测试中使用脚本 16 探索式测试 16 小结 21 练习题 21 第3章 局部探索式测试法 23 想不想测试软件？
23 测试就是有所变，有所不变 25 用户输入 26 状态 36 软件状态的基本知识 36 如何测试软件状态 37 代码路径 39 用户数据 39 运行环境 41 小结 41 练习题 42 第4章 全局探索式测试法 45 探索软件 45 旅游者比喻 47 漫游测试 49 商业区测试类型 51 历史区测试类型 58 娱乐区测试类型 60 旅游区测试类型 63 旅馆区测试类型 66 破旧区测试类型 68 漫游测试法实战 70 小结 72 练习题 72 第5章 混合探索式测试技术 73 场景和探索 73 使用基于场景的探索式测试 75 通过场景操作引入变化 76 插入步骤 76 删除步骤 77 替换步骤 77 重复步骤 78 替换数据 78 替换环境 78 通过漫游测试引入变化 80 卖点测试法 80 地标测试法 81 极限测试法 81 深巷测试法 81 强迫症测试法 81 通宵测试法 81 破坏测试法 82 收藏家测试法 82 超模测试法 82 配角测试法 82 取消测试法 83 混票测试法 83 小结 83 练习题 83 第6章 实践中的探索式测试 85 漫游测试 85 Dynamics AX客户端的漫游 86 有用的探索漫游 87 收藏家测试法和收集缺陷 89 漫游测试提示 92 利用漫游查找隐错 94 测试用例管理解决方案的测试 94 取消测试法 95 破坏测试法 96 快递测试法 97 测一送一测试法 98 在Windows Mobile设备中的漫游实践 98 我的测试方法和哲学 99 漫游测试法找到的有趣缺陷 101 破坏测试法实例 102 超模测试法实例 103 Windows媒体播放器的漫游测试 实践 105 Windows 媒体播放器 105 遍历测试法 106 超模测试法 108 极限测试法 109 与WMP相关的25个“假如”类型的问题 109 极限测试法：边界之旅 110 停车场测试法及其在 Visual Studio Team System测试版的应用 112 Sprint中的测试 112 停车场测试法 114 漫游测试中的测试规划与管理 115 定义地貌 115 旅行计划 116 让漫游测试运转起来 118 漫游结果的分析 118 判断：里程碑和发布 119 在实践中 119 小结 120 练习题 120 第7章 漫游与测试中的棘手问题 121 软件测试的五个棘手问题 121 漫无目的 122 重复性 124 暂时性 126 单调性 127 健忘 128 小结 130 练习题 130 第8章 软件测试的未来 131 欢迎来到未来世界 131 测试人员的专有提示显示 132 测试百科 134 测试用例的重用 135 测试原子和测试分子 136 虚拟化的测试资产 137 可视化 138 未来的测试 141 发布之后的测试 142 小结 143 练习题 144 附录1 经营成功的测试职业生涯 145 你是如何开始做测试工作的？
145 回到未来 146 上山 147 巅峰 149 下山 150 附录2 JW的专业博客摘录 151 教我一些东西吧 151 软件诚律 151 测试错误代码 157 真正的职业测试人员，请上前一步 160 我找到的一些常见的共同特性(无特别顺序) 161 建议总结 162 三击不中出局，是新的打击手上场的时候了 163 正式方法 164 工具 164 流程改进 165 第四种提案 166 软件测试是艺术、技巧或学科？
166 恢复对软件行业的尊重 169 事与愿违的过去 170 寻找更好的方法 171 分析安全漏洞和质量问题的流程 171 附录3 JW微软博客修订版 175 加入博客圈 175 2008年7月 176 开篇 176 PEST(泡吧与软件测试) 177 测试人员评估 179 预防与治疗(一) 181 用户与John 182 手工测试人员的赞歌 182 预防与治疗(二) 185 欧洲，你好！
186 测试赋 187 预防与测试(三) 189 回到测试 190 2008年8月 192 预防与治疗(四) 192 如果微软擅长测试，为什么软件依然糟糕呢？
194 预防与治疗(五) 197 自由式探索式测试 198 基于场景的探索式测试 198 基于策略的探索式测试 198 基于反馈的探索式测试 199 软件测试的未来(一) 199 软件测试的未来(二) 201 2008年9月 203 测试认证 203 软件测试的未来(三) 205 软件测试的未来(四) 207 软件测试的未来(五) 208 2008年10月 210 软件测试的未来(六) 210 软件测试的未来(七) 212 软件测试的未来(八) 214 谈到谷歌 216 再议手工测试与自动化测试 216 2008年11月 218 不再需要测试人员？
218 让测试人员继续测试 219 2008年12月 220 谷歌与微软的开发 测试比例之争 220 2009年1月 221 Zune的问题 221 解释探索式测试 223 (未来的)测试用例重用 224 测试用例重用(续) 226 休假归来 227 鼯鼠和受感染的花生 228

<<探索式软件测试>>

章节摘录

插图：已经有很多文章阐述了漫无目的生活会带来哪些坏处，但盲目的测试和现代测试实践中的无目标性，导致了这样一个问题。

软件测试不是简单地拿起来就做的事情。

它要求有计划、有准备、有策略和有多变的战术，这是成功进行软件测试的前提。

但是，太多的软件企业因偏爱“想干就干”而忽略适当的准备。

测试工作非常重要，所以决不能如此随便地对待它。

我在佛罗里达理工学院担任教授的时候，我教过软件测试课程。

有一个学期，注册这门课的学生多得超乎我的想象。

我决定做一个试验，想把少数随意选择这门功课的学生吓走。

在开学的第一天，我在计算机房上课，我指导学生们确定要测试的应用程序，两人为一个测试小组。

我没有给学生们完成这个测试任务提供进一步的指导。

作为鼓励，我答应他们，如果我对他们采用的技术留下深刻的印象，他们就会被留在我的班上。

如果他们不能满足要求，我就认为他们自动放弃（我不是有意要赶他们走，但是这种要求达到了我想要的效果）。

我在实验室里走来走去，无形中对学生们产生了压力，有时我会与一组学生攀谈，要求他们解释如何找到软件错误。

每当我提出这样的问题，就会得到类似的答案，如“不太清楚，我们只希望它会出错。”

最后，一些聪明的学生意识到这种回答并不奏效，他们在随后的回答中渐渐地开始包含一些策略。

事实上，我清楚地记得有两个学生说“我们将要对所有的文本输入框输入较长的字符串，希望能够找到一些不检查字符串长度的地方。”

我当即接受他们成为我的第一组学生，并鼓励他们继续做下去。

太好了！

虽然这不是最好的或者最重要的策略，但它至少是一种策略，有助于减少测试的无目标性。

软件测试人员运行测试程序时，经常缺乏策略或没有指定明确的目标。

在他们进行手工测试时，他们漫无目的地测试应用程序。

在他们实现测试自动化时，仅仅由于他们熟悉怎样编写测试自动化，就去这么做了，而不考虑这样的自动化是否能发现有价值的缺陷，是否经得起时间的考验，是否值得付出维护费用。

<<探索式软件测试>>

后记

人非圣贤，孰能无过。

一语道破人类容易犯错的天性，也为人们提供了堂而皇之逃避责任的借口。

但随着科学技术的发展，软件在金融、医学、航天、汽车等行业全方位的渗透，导致我们越来越难借助于这样的托词，原谅自己的失误，因为一个微不足道的疏忽或者失误，放过软件中存在的漏洞、缺陷和隐错，就会导致无可挽回的损失甚至灾难性的后果。

上世纪末在美国，因为两套软件采用的度量单位不一致（一个公制单位N/s，一个英制单位b/s，后者是前者的4倍），导致美国国家航空航天局（NASA）几亿美元的损失，使火星气候轨道飞行器在最后的关键时刻坠入火星，而不是顺利进入轨道按照既定设计绕火星运行。

此类事例在计算机一统天下的今天并不罕见，同时也为我们敲响了警钟，软件测试到了不得不高度重视的程度。

反观软件测试行业，可用的测试方法与10年前却没有多大差别。

要想提高软件质量，测试先行，测试创新，已经到了刻不容缓的地步。

探索式测试（Explory tory Testing，也称探索性测试）是一种软件测试方法，最先是CemKaner在1983年提出的。

这是一种强调个人自由与责任的测试方法，让独立测试人员可以借由不断的学习来改善测试的规划与测试的执行，而在测试的过程中也会同时改善测试案例达到相辅相成的效果。

在Nortel和微软的很多项目中，都采用了这一新颖、有趣和富有创意的测试方法。

<<探索式软件测试>>

媒体关注与评论

“好书！”

Whittaker讲述的概念有创意、巧妙、令人印象深刻。

他真正懂得如何鼓励工程师们以不同的角度考虑软件测试。

”——谷歌公司测试工程总监Patrick Copeland “James把美妙的手工测试方法学提升到了极致。

漫游不仅概念正确，而且实际应用得很好，我们已经在所有测试人员的内部培训课程中分享了漫游的概念。

如果你想把手工测试过程带到21世纪，请阅读这本书吧。

”——微软公司卓越测试总监Alan Page “1990年，我开始与James在IBM一起共事。

他早在当时就鼓励测试人员和开发人员大胆创新。

通过这本书，他把对软件质量的热情提升到了全新的高度。

请阅读这本书吧，见证你自己成为一个更好的测试工程师。

James是这方面的权威，这个星球上的软件测试工程师和开发工程师，无论他们是真正关心软件质量，或者只是想在自己的日常工作中增加更多的乐趣，都应该阅读这本书。

”——Cisco Systems公司高级工程主管Kaushal K. Agrawal “James Whittaker 是测试领域中一位真正有远见的人士。

Utest和我们的QA专业全球社区经常向James咨询以获得他的灵感、他对测试未来趋势的解读和他的各种测试理念。

现在，他终于为大家写出了这本书，我们这个行业会由于这本书而变得更有智慧。

”——uTest公司CEO和合伙人Doron Reuveni “只有像James Whittaker这样的人才会把旅游和软件测试以小说的形式结合起来，也只有James才能把它做到这么天衣无缝。

漫游方法不仅提供了令人难忘的有效思考模式，还把适当的结构和组织与广泛的探索和创造结合在一起。

bug们，小心啦！

”——谷歌公司Alberto Savoia “James是软件测试主题最出色的演讲者之一，读他的书就像聆听他的演讲。

如果你希望增加测试知识并成为更优秀的测试人员，这本书就是为你而写的。

”——TCL集团公司主席和合伙人Stewart Noakes “我从事探索性测试已经有段时间了，James的漫游测试法给我所使用的各种方法起了名字，定义了测试重点，更重要的是给了我实际的指导。

这本书将会使探索式测试的教学和应用变得更方便。

”——iMeta Technologies公司高级测试顾问Rob Lambert “我为这项工作感到非常激动，它概念新潮却又合情合理。

连我这样的普通读者都能轻松地理解和使用，而不用首先去学习一些华而不实的过时理论。

在阅读本书时，我也从不需要借助字典。

测试领域长久以来就一直期盼着这样的创新之作，我由衷地认为本书在这方面走在行业最前沿。

”——Netjets公司QA部门经理Linda Wilkinson

<<探索式软件测试>>

编辑推荐

如何发现和修复被常规软件测试忽略的关键软件缺陷？

在《探索式软件测试》中，享誉业界的软件测试专家James Whittaker揭示了当下最严重、隐藏最深的软件错误的真正诱因，并介绍了如何利用功能强大的探索式测试技术来找到并纠正这些错误。先后就职于谷歌、微软和其他顶尖软件公司的James Whittaker，在软件测试的前沿阵地拥有近20年的丰富经验，他为传统的手工测试引入了可重复、规范、可传授和特别高效的新过程。

Whittaker定义了针对单个测试人员的简单技术和针对大规模测试团队的复杂技术。

他还引入了一个混合策略，将探索式概念引入传统脚本测试。

在《探索式软件测试》中，可以体会到如何在恰当的时机使用这些方法，如何成功地充分应用这些方法。

简洁、诙谐和可行，《探索式软件测试》引入的这些技术已经经过上市软件的测试人员广泛应用，人们在实际测试过程中深受这些方法的启发，成功实现了预期目标。

《探索式软件测试》是为测试人员、QA专家、开发人员、程序经理和架构师所写的。

《探索式软件测试》涉及以下重要问题：为什么自动化测试无法消除所有缺陷，如何才能让这些缺陷无处遁形？

哪些技术可帮助我不断发现和消除致命错误？

如何更高效地进行手工测试，增加些许轻松和愉悦的感觉？

对于每个项目，如何确定最高效的高级测试策略？

在我无法进行全部测试时，哪些输入是必须测试的？

哪些测试用例能提供最理想的特性覆盖率？

在结合使用探索测试和传统脚本或场景测试时，如何才能获得理想效果？

如何体现来自开发过程的反馈意见，代码更改吗？

<<探索式软件测试>>

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>