HZ BOOKS

软 件 工 程 技 术 丛 书

测试系列

软件测试过程管理

（原书第2版）

Managing the Testing Process

(Second Edition)

（美） Rex Black 著 龚波 但静培 林生 周志全 等译

机械工业出版社
China Machine Press

软 件 工 程 技 术 丛 书

测试系列

软件测试过程管理

（原书第2版）

Managing the Testing Process

(Second Edition)

（美） Rex Black 著 龚波 但静培 林生 周志全 等译

机械工业出版社
China Machine Press

本书全面论述了软件测试管理的全过程，介绍了管理大型和小型项目中硬件和软件所需要的工具和资源，阐述了如何开发关键工具。这些工具简单、有效并符合行业标准，确保读者能够掌握和了解最新最好的测试管理工具，并能够帮助读者应用相关工具和技术来管理资源，从而能够成功地管理测试项目。

本书还提供了很多作者亲自做的测试试验，从而使读者能够有效地将测试理论与实践相结合，更加深入地理解测试过程。

Rex Black: Managing the Testing Process (Second Edition) (ISBN: 0-471-22398-0).
Authorized translation from the English language edition published by John Wiley & Sons, Inc.

本书版权登记号：图字：01-2002-4801

图书在版编目（CIP）数据

软件测试过程管理（原书第2版）/（美）布莱克（Black, R.）著；龚波等译. —北京：机械工业出版社，2003.10
（软件工程技术丛书　测试系列）
书名原文：Managing the Testing Process (Second Edition)
ISBN 7-111-12747-1

Ⅰ.软…　Ⅱ.①布…②龚…　Ⅲ.软件—测试—管理　Ⅳ.TP311.5

中国版本图书馆CIP数据核字（2003）第068448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：杨　文
北京瑞德印刷有限公司印刷·新华书店北京发行所发行
2003年10月第1版第1次印刷
787 mm × 1092 mm　1/16 · 26.25印张
印数：0 001-5 000册
定价：48.00元

前　言

你在负责管理一个计算机硬件或软件测试项目吗？恭喜你！也许你刚从测试工程提升或从开发小组的另一个部门调过来，或者也许你已经做了一段时间的测试项目了。不管你是测试经理、开发经理、技术或项目负责人，还是对公司测试和质量保证项目负有一定责任的人，你都可能在寻找一些关于怎样管理测试项目的方法。

本书对你可能会有所帮助。本书在1999年出版的第1版，在头两年销售量就超过10 000册。我收到很多读者的电子邮件，一些人想问我一两个问题，一些人对我撰写此书表示了感谢，还有一些人指出了其中的错误。读者们在不同的出版物和网站上写了评论意见，大部分是肯定的意见，但也有一些有助于改进的建议。我非常高兴本书得到大家的认可，也感谢所有阅读本书第1版的读者，特别是那些建议我如何改进第2版的读者。

本书包含了我从编程和系统管理转向测试管理时希望学习到的知识。本书会告诉你怎样开发一些必要的工具以应用于你的测试项目，还提供一些帮助你获得和使用必需资源的技术。如果你掌握了这些基本工具，将这些技术运用到你的资源管理过程中，合理分配精力，你就能成功地管理测试项目。你甚至还会做得更加出色，这可能让你同我一样成为测试项目经理。

本书重点

出于很多种原因，我编写了本书。首先，在所谓的“软件危机”中，很多项目在交付日期、预算和质量上的实际结果和预期相差甚远，特别是编写和测试软件的个人、高级项目负责人和用户之间的评价相差甚远。同样，计算机硬件开发项目经常忽略了关键进度安排和质量里程碑。作为项目风险管理策略的一个完整部分，有效的测试和明确的结果交流都大有裨益。

其次，我发觉关于软件测试和硬件测试方面的文献之间存在差别。我读过一些如何设计和实施测试用例的基础知识方面的书籍，同样也读过描述经验丰富的高级项目经理如何运用诸如能力成熟度模型（CMM）、ISO9000、全面质量管理、软件质量度量等概念和工具来提高产品质量的书籍。然而，我相信像我们这样的测试经理需要这样一本书，它将向我们阐述测试项目管理中像砖和水泥一样的基本工具和技术。

在本书中提供的技巧和工具会帮助你计划、构建和执行一个结构化的测试操作。不同于所有普通、特定且无用的测试项目，结构化的测试操作是有计划的、可重复的且编制成文档的，但仍在适当的地方保持了创造性和灵活性。在本书中学到的东西让你能够开发用于理解测试产生的大量数据含义的模型，从而让你能够有效地管理软件或硬件开发项目中经常是混乱的、无序的且由变化驱动的那些领域。本书还讲述了如何组建一个有效且高效的测试组织。

最后，我要重点谈谈开发和维护环境中的测试管理问题。因为以下两个相关主题在其他书里介绍得很多，在此我就不讨论了。

基本项目管理工具，比如工作分解结构、甘特图、状态报告和人员管理技巧。在从事管理

工作时，这些工具对你很有帮助，因此我建议你应参考一些项目管理方面的书籍，比如在附录B中参考文献里面列出来的书目。你也可以参考现在项目管理方面大量优秀的培训课程。

计算机硬件产品测试。如果你的业务范围包括这种测试类型，我推荐你参考W.Edwards Deming、Kaoru Ishikawa和J.M.Juran编写的关于统计质量控制方面的优秀书籍，还有Patrick O'Connor编写的关于可靠性工程方面的书籍；详细内容请参看附录B中的参考文献。

从把“黄金代码”原封不动地拷贝到发布介质上的意义来说，软件产品不需要测试。但是，硬件和软件产品经常包含一些小的修订和维护发布版本。你可以利用本书中描述的技术来管理那些包含这种发布版本的较小的测试项目。

本书对于软件和硬件测试的不同之处有详细的描述，因此，乍一看以为本书分为两个方面阐述。然而，我发现从**测试项目管理**的角度看，这两个方面测试的区别没有从**测试技术**角度看重要。意思很清楚：硬件测试软件，软件测试硬件。这样你就可以用相似的技术管理硬件开发和软件开发项目的测试了。

规范还是食谱

在我刚开始担任测试工程师及测试项目经理的时候，我对测试一无所知。无知会让你意识不到在隧道洞中看到的亮光实际上是正在飞速驶来的火车。“那会有多么艰难呢?”我这么想，“测试仅仅就是算出可能出现什么差错，并进行试验验证罢了。”

但是我很快发现自己错了，出现以上看法错误在于以下三个关键方面：

- “算出可能出现什么差错，并进行试验验证”的任务的确很困难，也就是设计出好的测试用例很困难。在近十年里，涌现出了许多关于测试用例工程的好书。但是，虽然在我学生时代或以前，Boris Beizer、Bill Hetzel、Cem Kaner和Glenford Myers都出版了相关主题的书，我的教授却没有教我测试方面的知识。随着软件工程的下半世纪的开始，这种状况最终开始改变。但是，大多数成长中的软件工程师仍然很少学习有关测试方面的知识。
- 测试不是在真空中进行的。相反，它是整个项目的一部分，因此测试必须和实际项目需求相符，而不是黑客的异想天开。简而言之，测试项目需要对测试项目进行管理。
- “测试太难了。”这种想法的盛行只会增加测试专家面临的困难。一旦我们从痛苦的经历中明白了测试到底有多难，我们有时候就觉得好像是命中注定要一个项目接一个项目地反复解释为什么这个测试任务需要花费这么多时间和金钱。

以上几个方面暗含了很多复杂的因素。其中最重要的因素之一就是组织测试过程的成熟度等级可能发生较大变化：测试可能是一个可重复的、可度量的过程的一部分，或者对一个混乱项目的亡羊补牢。另外，动机因素（也就是管理为什么会干扰测试的原因）在重点和强度上都可能发生改变。因为害怕重复近期的失败项目而萌发测试动机的测试经理与尽可能生产出好的产品的经理对待测试的态度是不同的，而两种动机都不同于那些被迫进行测试但并不重视测试的人的动机。最后，测试和项目的其他部分紧密相关，因此测试经理经常遭到外界的各种影响。当测试范围和强度的改变在项目中引起较大变动时，这些影响并不总是良性的。

上述因素使得为发一个关于“如何”计划和执行测试项目的指南非常困难。正如学者们可

能说的那样，测试项目管理不是简单开发一个规范。“理解以下观点你就能理解这个领域。”这句话不适合测试这个领域，并且，测试规范的开发不是本书要完成的任务。

你需要一份规范来正确管理测试项目吗？我认为不需要。相反，打个比方：我是个技艺精湛的厨子，一个业余主厨。我永远不可能成为世界级的名厨，但是我照样能为我的家人做出不错的正餐。我成功地准备了丰盛的感恩节大餐，有些是在汽车旅馆的小厨房里准备的。在学校里为了生活需要，我掌握了烹制家常饭菜的方法。在这个过程中，我学会了如何从食谱里面找出烹制方法，如何应急，怎样快速地准备各种配料，以及怎样掌握好正餐和一系列小吃的时间，然后凭着记忆烹制食物。

对你的测试组织所需要的管理，一顿价格合理的家常饭菜就是很好的类比。因此，本书就可以作为测试项目经理的“食谱”，它描述了需要的基本工具，并帮助你组合和调配适当的元素。

需要的工具

我的测试管理方法中包含五种基本工具：

详尽的测试计划。一份详细的测试计划就像一个水晶球，让你可以预见和预防潜在的危机。这样的计划阐述了测试范围、质量风险管理、测试策略、人员配备、资源、硬件补给、配置管理、安排进度、阶段、主要里程碑、阶段过渡，以及做预算。

良好的测试系统。好的测试系统能有效地确定可能损害上市产品或降低内部用户认可度的错误。它的内部和外部保持一致，易学易用，且建立在一套设计良好和兼容的工具之上。我用术语“良好测试系统体系结构”来描述这种系统的特性。*体系结构*一词让测试小组内的成员培养对测试开发的全局和结构化的观点。对于管理来说，创造良好测试系统包含开发结构优良、持久耐用的产品。

基本状态的错误跟踪数据库。在测试过程中，你和勇敢的测试小组成员将发现很多 bug（又称为问题）、缺陷、错误、问题、过失，以及其他不容易用纸记录的内容。试图在你的大脑里或一个文档中记住所有错误会立即造成灾难，因为你不能在测试小组内和程序员、其他开发小组同行或项目管理小组进行有效的交流，而这样并不有助于提高产品质量。你需要一个方法来跟踪每个错误从开始到结束过程中的一系列状态。我将告诉你怎样安装和使用一个有效和简单的数据库来实现这个目的。这个数据库还可以以信息图表的方式总结错误，告诉管理部门关于项目测试完成情况、产品稳定性、系统转变时间、麻烦的子系统和根本原因。

全面的测试跟踪电子表格。除了跟踪错误之外，你还需要跟踪每个测试用例的状态。当使用某特定硬件时，操作系统会失效吗？按照特定格式保存文件的时间是否太长？哪个版本的软件或者硬件不能完成重要的测试？使用单个电子表格的多个简单工作表就可以跟踪每个测试用例的结果，使你获得回答各种问题的细节内容。详细的工作表组成一个汇总的工作表，使你可以了解整体情况。测试用例进度如何？多少测试用例被阻塞？所有测试用例到底花费多长时间才能完成？

简单的变更管理数据库。多少次你曾经纳闷：“进度为什么与计划出现如此大的偏差？”诸如硬件或者软件交付日期推迟、丢失阻塞测试用例的特性、无效的测试资源，以及其他看起来

微不足道的变更都可能会对项目测试实施产生负面影响。如果测试实施得晚，整个项目完成时间就会推迟。你不能防止测试延迟情况的发生，但是可以跟踪测试延迟情况，以便提前了解可能发生的延迟，并有效地解决问题。本书提供一个简单、高效的数据库，使你可以及时跟踪风险，防止出现问题。

本书向你展示了如何开发这五种基本工具，把这五种基本工具应用到测试项目中，以及如何获取和使用必需的资源。我已经在 PC 中普遍安装的 Microsoft Office 套件（Excel、Word、Access 和 Project）中实现了它们。你也可以很容易地使用其他办公自动化应用程序，因为我并没有使用 Office 套件的高级功能。

需要的资源

让我们继续使用烹饪的比喻，为了做出一盘菜，你需要其他调料或者资源。在测试烹饪书中，我将演示如何组装资源以便实现测试项目。这些资源包括以下内容：

实用的测试试验室。工作舒适和安全的地方，配有优秀人才和计算机设备。该实验室需要通过多种方式与开发小组、管理部门，以及外界进行交互和沟通。你必须保证实验室装备足够的软件和硬件以保证测试人员高效率地工作，同时还应该及时更新软件和硬件。请记住，这是个测试试验室，需要让工程师很轻松地跟踪有关系统配置的关键信息。

测试工程师和技术人员。你需要能够努力工作的优秀的团队。寻找优秀的测试工程师比寻找优秀的开发工程师更难。你怎样识别初露头角的测试天才和那些会使测试变得一团糟的庸才呢？有时，识别两种人的难度超出你的想像。一旦团队组建完成，就可以真正地开始工作了。你怎样激励团队努力工作呢？怎样消除影响士气的因素呢？

承包人和顾问。作为测试经理，你有可能借用外部资源，按小时雇佣技术高手，在项目结束时终止合作。我将帮助你区分普通的高技术临时工人，了解他们从事这种工作的原因，并消除他们的顾虑。你何时需要承包人？承包人关心什么事情？你应该尝试固定与一些合作愉快的承包人合作吗？你怎样判断何时需要顾问？

外部测试试验室和供应商。在有些案例中，有必要不在自己的测试试验室中做某种测试，如当自己的测试试验室的测试任务很重时。合理使用外部资源提供的技能、基础设施和设备有助于节省时间和成本。这些实验室和供应商到底能够为你做什么？你怎样使用他们来缩小测试项目的规模，而不会影响测试覆盖率等方面？你怎样把这些过程和结果映射到自己的测试项目？

当然，在使用这些资源之前，你需要对此进行合理的调配。在前面你也许已经了解到，管理部门对测试需要的资金和设备的要求也许很精确。我提供一些有助于你尽快获得所需资源的建议。

相关环境

我已经在许多大型和小型项目中使用过这些工具和技术。这些概念可以很容易扩展和缩小，尽管大型项目也许需要购买自动化方式的某些工具。在这种情况下，我所描述的工具可以充当你准备购买或者构建自动化工具的原型和需求。

在分布式项目中，这些概念也可以扩展。在旅馆房间和机场大厅里，我曾经在膝上电脑中使用这些工具来同时管理多个项目。我曾经使用这些工具来测试市场驱动的终端用户系统和内部信息技术项目。尽管背景不同，但是我认为可以改编这些概念，使之应用于多种环境下。

这些工具符合工业标准，与领先的软件和硬件供应商的最佳测试管理实践和工具保持一致。我使用这些工具来整理我对项目的认识，开发有效的测试计划和测试包，在动态高技术开发环境中执行计划，并跟踪、分析和向项目经理展示结果。同样，我对测试资源管理的建议也来自各种成功和失败的经验。

由于环境原因，我在本版的最后新增加一章，讨论测试过程与整体开发或者维护过程保持一致的重要性。这涉及诸如组织环境、测试的原因和经济因素、系统开发的生命周期和方法论，以及过程成熟度模型等问题。

使用本书

本书中所有内容都不是基于科学真理、闭门造车、学术研究，或甚至是灵感。它只是一些关于我曾经在管理大量测试项目时从事的和将继续从事的工作的内容。你可以照搬这些方法或者做一些修改。也许你会发现所有的方法或其中的一部分非常有用。

同样，这不是一本描述测试方法、测试理论或开发过程艺术的书。本书是有关硬件测试和软件测试的实践书。对于开发过程（最佳实践或贵公司的实践），我只是假设你作为测试经理而介入到一个开发项目，并花大量的时间进行必要的测试开发。最后一章讲述了我所见过的和参与过的不同的开发过程。我还讨论了开发周期的选择是如何影响测试的。

当然，在一定程度上，我不能只谈论测试管理，而不谈论测试技术。因为硬件测试技术和软件测试技术有区别，所以你可能会发现我使用的一些术语可能对你的运用造成困惑、矛盾或不清楚。本书提供了术语表，如果你是一个软件测试者，它能帮助你解开硬件测试示例的疑团，反之亦然。最后，测试经理通常既是技术带头人也是经理，因此你务必在特殊测试项目中理解和使用最佳实践，尤其是在采用测试技术的过程中。附录 B 中有一个书目，如果需要，那些书可以帮助你重新学习这些主题。

本书基于我的实际经验，其中好坏经验兼有。失败的教训是为了帮助你避免一些我曾犯过的错误。我试图让这种讨论保持轻松风趣。在附录 B 的参考文献中列出的书目中你可以找到本书内容的理论支持。

我发现对照例子学习的效果最好，因此我还在书中包含了很多例子。由于我描述的工具应用于软件测试和硬件测试，因此很多例子我都以以下两个假设项目之一为基础：

- 大多数软件例子包含基于 Java 的字处理包 SpeedyWriter 的开发，它由 Software Cafeteria 有限公司编写。SpeedyWriter 拥有字处理器的所有功能，还有网络文件加锁、Web 集成和公钥加密等功能。SpeedyWriter 包含了其他供应商提供的各种 JavaBeans。
- 大多数硬件例子参照服务器 DataRocket 的开发，它由 Winged Bytes 公司开发。DataRocket 提供强大的、高容量的文件和应用服务，也是局域网用户的 Web 主机。它运行多个操作系统。借助于第三方软件，Winged Bytes 计划集成一个 U.S. 制的局域网网卡和台湾的 SCSI 控制器。

至于书中讨论的工具，你可以在网址 www.rexblackconsulting.com 找到相关例子。我为第 2 版花了大量时间改进这些模板。我还增加了一些实际项目的案例研究。在描述这些工具用途的章节中，我用了简短的篇幅指导你如何使用和学习这些模板和案例研究。这样，你就可以凭借自己的力量利用这些资源成功地实现那些工具了。我在书中用了一些图来描述部分工具，并在本版中改善了图的可读性。但是，工作表和图表的屏幕快照内容还是很有限。因此，在阅读不同的章节时，你也许想从网站打开并验证相应的案例研究和模板以加深对工具工作机制的理解。

请注意本书中提供的工具是可用的，但是也包含了很少量的“虚构数据”。不应该用这种数据来推导对错误汇总、缺陷密集度、主要质量风险的任何粗略估计，或者任何其他项目采用的衡量尺度的经验。我开发这些工具主要是为了证实一些想法，所以它们不具有你所期望的如同商业产品一样复杂的自动功能。如果你打算在自己的项目中使用这些工具，那么就要花足够时间和精力来使它们适合于你的项目环境。

考虑到你们在实际项目中使用这些工具前需要做练习，我在每章的最后都安排了练习。这些都是本版中新添加的内容，也使得本书更加适合作为关于测试、软件工程或软件项目管理方面课程的测试管理教材（练习解答请参见网站 www.rexblackconsulting.com）。假定测试越来越多地被明智的项目经理看作是项目风险管理策略的一个关键部分，那么相关内容的资料作为学校或认证课程的部分教材是非常有意义的。

最后，你应当知道测试不是时间和资源充足的领域。我发现至关重要的是把重点放在对项目经理来说真正有价值的测试上。过去，我经常都是因故终止错误步骤，因为我总是在一些不重要的细节上花费时间而忽略了重要的环节。这些经历教会我辨别并关注那些少数重点，而忽略多数琐碎的细节。书中介绍的工具和技术同样可以帮助你做到这一点。许许多多计算机测试组织维持不了几年就垮台了。本书可以帮你避免重蹈覆辙。

虽然测试管理的成功不仅仅简单地取决于一份工作，这一点是很清楚的，但是对不同的人就有不同的含义。在日常工作中，我保持平和的心态，减缓压力，从实际管理测试中不断提升专家形象，在我的知识范围内衡量成功的价值，而不是通过对在混乱环境中引起的无休止的危机做出反应来衡量。我希望我的这些工具和想法将帮助你成为一个成功的测试专家。

第 2 版的改动和新增内容

对于本书第 1 版的读者和正犹豫着是否要购买第 2 版的读者，我下面将简要说明第 2 版中的一些改动和新增内容：

- 最后一章是新加的，其中讨论了测试过程与整体开发或维护过程相适应的重要性。我阐述了组织环境、测试的原因和经济因素、系统开发的生命周期和方法论，以及过程成熟度模型。
- 为书中讨论的工具和技术添加了一些案例研究。这些案例研究源自真实项目，令人高兴的是它们不受非公开协议的限制。本书第 2 版还包含了一些背景信息帮助你理解这些案例研究文档。
- 增加了模板和例子的数量，并改善了一些工具的格式。还添加了一些新的度量。模板

里包含了生成这些度量的工具。第 1 版中的一些模板有些小错误。(第 1 版的读者和一些测试专家发现并给我指出了那些错误。)我已经改正了那些错误。

- 第 1 版中的一些图片很小且字体太小。第 2 版中我修改并扩大了图片，这样就提高了可读性。在最后的编辑中我还改正了一些放错地方的图片。
- 除了案例研究，我还增加了一些练习，其中有些是我以前在三日制“测试过程管理”课程中采用的练习，有些是特别为第 2 版创作的。这些练习可以用于自学、图书俱乐部或课堂教育。(California Polytechnic State University 的 Patricia McQuaid 教授把第 1 版选作软件测试课程的教材。)为了确保这些练习的实用性，我和软件工程学术界的著名人士共同做出了努力。
- 包含模板、案例研究和练习的支持文件和其他工具没有用易损并难以更新的 CD-ROM 提供，而是发布在网站 www.rexblackconsulting.com 上。我将继续更新和改正这些模板，这样从某种意义上来讲，本书也将不断地得到完善。
- 最后，从我编写第 1 版以来，根据管理测试项目人员面对的挑战，我对第 2 版做了一些小的变动。每次当我将此书作为两日制和三日制“管理测试过程”课程的教材时，至少有一个学生告诉我说：“令人惊讶的是我们这里讲到的每个问题在我的项目中都发生了。”然而，我已经学习了一些新技巧并开阔了思想。例如，我在第 1 版中批判探索性测试，但是通过和一些成功实践者的认真交流和自己尝试，我了解到探索性测试是一项有用的技术。

如果你阅读了第 1 版，欣赏它，并觉得它有用，那么我想这些改动和新增加的内容会使第 2 版对你更加有用。

参与本书翻译的人员除封面署名外还有：张雅升、王华、詹荣开、汪洲、尹建军等。

目　录

第 1 章
确定测试的重点内容：测试项目的基础

测试工作应把握重点。我们的工作很容易做过头。对于重要的软件和硬件，需要做很多的测试。即使将重点放在质量要“足够好”的测试上，你仍会发现这样的测试成本太高或很难断定“足够好”对客户和用户有何意义[1]。在开发测试系统，包括测试件、测试环境和测试过程，而且在将工作交给测试小组之前，要算出可能测试什么，然后是应该测试什么，最后是能够测试什么。确定这些问题的答案有助于制定测试计划和把握测试重点。

测试的可能是那些在测试组织内还未被测试的领域。我所参与的每个项目，有些测试工作量都是由我的职责范围以外的组织完成的。测试其他小组已经测试的内容是没有价值的，是在浪费时间和金钱，而且会带来麻烦。

应该测试的是那些直接使客户和用户对质量感到不满的未测试内容。人们经常使用有错误的软件和计算机，并还感到满意；这也许是因为他们从没遇到错误或这些错误不会严重地妨碍他们的工作。测试工作应该集中在发现关键的缺陷，这些缺陷会限制人们使用产品的能力。

能够测试的是那些未测试的、让有限资源得到最有效利用的关键领域。可以测试应该测试的所有内容吗？不可能，因为有时间进度和经费预算的限制[2]。对于大多数项目，必须根据有限的信息，在很紧的进度安排下做出艰难的选择。还需要把测试项目报告给经理，从而获得需要的资源和时间。

1.1 可能测试什么：扩大的测试工作量

在我喜欢的开发项目中，测试无处不在。对此，我的意思是大量的测试工作在独立的测试小组之外仍旧存在。此外，测试很早就开始了。这种安排不仅有技术上的意义，而且可以管理测试小组的工作量。这部分从两方面检验非正式测试小组如何从事测试工作：一方面是测试的聚焦程度，即粒度；另一方面是不同测试阶段进行的测试类型。或许公司内的其他组织能够（或正在）协助你的测试工作。

1.1.1 从显微镜到望远镜：测试粒度

测试粒度（granularity）指测试重点的精细或粗糙程度。精细的测试实例允许测试者检测低

---

[1] 据我所知，James Bach 首次把“足够好”这个词语应用到软件质量中。你可以在 www.satisfice.com 查寻他在这方面的观点。

[2] 有关实现“彻底”测试的困难的讨论，请参考 Boris Beizer 的著作《软件测试技术》（Software Testing Techniques）、Cem Kaner 等人的著作《测试计算机软件》（Testing Computer Software）或者 Glenford Myer 的著作《软件测试艺术》（The Art of Software Testing），这些著作给出了“彻底”的定义。

级细节，通常是在系统的内部；而粗糙的测试实例为测试者提供总体系统性能的信息。可以认为测试粒度是沿着结构化（白盒）到行为（黑盒和实时）测试的频谱顺序进行的，如图 1-1 所示。

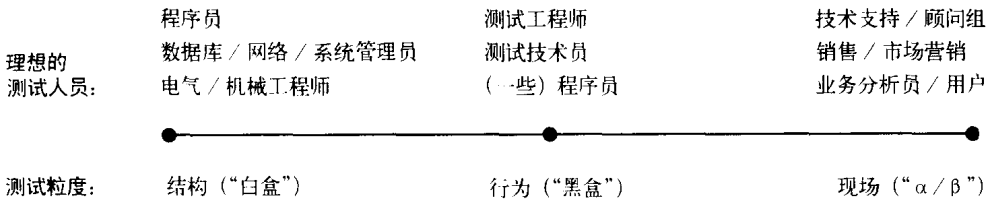

图 1-1 测试粒度频谱和所有者

1．结构（白盒）测试

结构测试（又称"白盒测试"和"玻璃盒测试"）可以查找低级操作中的错误，如发生在代码级别、数据库模式、芯片、配件和接口级别以下的错误。这些结构测试基于系统运行方式进行。例如，结构测试可能揭示存储用户配置数据库的用户名空间有 80 个字符，而实际上这个字段只允许输入 40 个字符。

结构测试需要知道系统的细节内容。对于软件，测试者亲自检查代码和数据结构来进行结构测试；硬件结构测试是把芯片的规格说明同示波器或电压表的读数进行对比。因此结构测试非常适合于开发领域。对于测试人员来说，至少对于那些不知道底层细节和没有编程或工程经验的人员来说，结构测试是很困难的。

进行结构测试还需要具备结构测试技术的知识。不是所有的程序员在基础教育和后期培训中都接受过测试技术培训。这种情况下，让测试小组成员与学科专家的程序员一起工作，这样可以促进成功地完成结构测试。这个人有助于培训程序员能够找出在结构级发生错误的技术。

2．行为（黑盒）测试

行为测试（又称"黑盒"测试）通常用于查找高级操作、特性级、操作手册和用户场景等级别中的错误。功能测试基于系统应该做什么。如果 DataRocket 网桥建立 1Gb 以太网连接，但是吞吐量仅仅是 10Mbps，那么黑盒网络性能测试就可以找出这个错误。

行为测试包含对应用领域、解决的业务问题和系统任务的详尽理解。最好让懂得系统设计的测试人员进行行为测试，这样他们就能找出设计方面的错误。例如，在 Java 虚拟机上运行用 Java 实现的程序出现严重的性能局限。

除了应用领域和被测系统的一些技术问题之外，行为测试人员必须知道找出这些错误的最有效的特别行为测试技术。虽然一些行为测试确实要根据一般的用户场景，但还是设计了很多测试来检验极限值、接口、边界和错误条件。这些都是错误容易发生的地方，且行为测试主要用来搜索缺陷，就像结构测试一样。优秀的行为测试人员使用脚本、需求、文档和测试技巧来辅助查找错误。简单地只考虑系统或认为系统在一般条件下工作不是行为测试的有效技术。好的行为测试和好的结构测试一样，它是结构化的、有组织的，通常是测试人员查找可疑的系统弱点，并尽力找出错误而创造的重复条件顺序。行为测试是大多数独立测试组织最主要的测试技术。