



软件工程实践丛书

有效软件测试

[美] Elfriede Dustin 著 新语译

Effective Software Testing

提高软件测试的50条建议
50 Specific Ways to Improve Your Testing



清华大学出版社

软件工程实践丛书

有效软件测试

[美] Elfriede Dustin 著

新语译

清华大学出版社

北京

内 容 简 介

本书的内容涵盖了从有关过程和管理的内容到技术方面的话题。书中内容并不局限于任何特定的技术和应用程序平台。本书适用于质量保证人员、软件测试人员、测试组长和测试经理等读者阅读,也可供项目经理和软件开发人员参考。

Simplified Chinese edition copyright © 2003 by **PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.**

Original English language title from Proprietor's edition of the Work.

Original English language title: **Effective Software Testing, 1st Edition by Elfriede Dustin,**
Copyright © 2003

EISBN: 0-201-79429-2

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Pearson Education, Inc.

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macao).

本书中文简体翻译版由 Pearson Education 授权给清华大学出版社在中国境内(不包括中国香港、澳门特别行政区)出版发行。

北京市版权局著作权合同登记号 图字: 01-2003-2088

本书封面贴有 Pearson Education(培生教育出版集团)激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

有效软件测试 / [美] 达斯汀著; 新语译. — 北京: 清华大学出版社, 2003

(软件工程实践丛书)

书名原文: Effective Software Testing

ISBN 7-302-06945-X

I. 有… II. ①达… ②新… III. 软件—测试 IV. TP311.5

中国版本图书馆 CIP 数据核字 (2003) 第 064036 号

出 版 者: 清华大学出版社

地 址: 北京清华大学学研大厦

<http://www.tup.com.cn>

邮 编: 100084

社 总 机: 010-62770175

客 户 服 务: 010-62776969

文稿编辑: 刘伟琴

封面设计: 立日新设计公司

印 刷 者: 清华大学印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 170×230 印 张: 12.5 字 数: 234 千字

版 次: 2003 年 8 月第 1 版 2003 年 8 月第 1 次印刷

书 号: ISBN 7-302-06945-X/TP·5132

印 数: 1~4000

定 价: 25.00 元

译者序

我曾经在国内一家知名的软件企业工作了8年，在此期间一直是核心项目组的成员之一。我们负责开发一个专业性很强、技术含量很高的产品，这一点和本书中提到的财务系统有些接近。虽然我并没有真正从事过软件测试工作，但是作为一名开发人员，经常需要和测试人员打交道，所以我有机会感受到了测试工作乃至整个软件工程环境的变化。

这家企业的软件工程环境的发展大体可以分为3个阶段：

精英软件工程环境

我刚进入这家公司时，计算机工业在中国处于超高速发展期，这个行业的待遇和福利相对较高。而且当时计算机还没有现在这么普及，计算机行业对于普通的大学毕业生来说充满了神秘感和吸引力。

我们的前身是一所国内著名大学的校办产业，创业者基本上都是这所学校的职工。当时有相当部分的员工一个人有两个身份，既是公司的职员，又是学校内某个部门的职工。因此在项目开发中不可避免地继承了一些学校办事的风格，例如：重技术、轻管理。

我刚进公司的时候，企业仍然保持着强劲的上升势头，虽然规模不大，但是效益很好。职工的进取心也很强。

当时我们公司的主要用户是一些大客户，这些大客户的共性很强（从后来的结果来看，他们对产品的功能需求非常片面，所以对开发人员和测试人员的行业知识要求不高）。并且当时卖方市场的竞争不激烈，用户对产品质量的要求也不是太高。

由于以上的大环境、小环境和其他一些因素，当时的软件工程环境有以下一些突出特点：

- 人员素质很高。无论是开发人员还是测试人员大都是名牌大学的优秀毕业生。

- 开发人员和测试人员都不是行业专家。
- 校园文化对企业文化影响至深，现代企业制度不完善，制度的执行也缺乏力度。

当时负责我们项目测试工作的测试人员一共有 3 位，个个都是以一当十的强人。虽然他们缺乏基本的行业背景，对计算机技术也缺乏了解，但是由于他们的素质很高，学习能力很强，再加上当时对行业知识和计算机技术的要求不高，所以他们很快就掌握了测试工作所需的各种知识和技术。

虽然当时制度有漏洞，但是由于企业的小环境很好，职工大都责任感很强，所以不完善的制度并没有过多地影响开发、测试工作。

测试人员的高素质特别体现在当他们再次从事同一产品的后续版本的测试工作时，尽管他们并没有正式的渠道了解软件技术、软件结构，但是通过与开发人员的非正式交流和经验，他们已经能够猜到哪些部分容易出问题，并且一旦某个问题暴露了以后，他们立刻会想到哪些地方也可能出现类似的问题。因此测试效率比较高。

从现在的观点来看，当时的软件工程环境的最大问题是非“结构化”的。但是职工出色的个人能力和敬业精神在很大程度上弥补了制度上的不足。

标准化的软件工程环境

随着时间的推移，企业的扩张，第一代测试人员由于种种原因，有的走上了领导岗位，有的转向其他性质的工作（可能是厌倦了测试工作吧），公司迎来了第二批测试人员。要求也不是那么高了，他们的专业素质和个人素质与第一代测试人员相比有所下降。由于这些人毕业时间较晚，所以他们的知识结构和第一代职工有一定的差异，但对软件工程有了比较系统的认识。

企业为了发展，需要拓展产品的生存空间。与此同时，国外的产品开始进入国内市场，市场竞争日趋激烈。我们的产品在新市场上遇到了挫折，用户不再总是唱颂歌，而是开始有了不少的抱怨。在这种背景下，我们公司出现了第一次有关软件工程环境的激烈争论，争论的主题是：谁应该为软件产品的质量负责？

整个事件的导火索是在一次产品会议中，一位技术骨干对测试负责人说：“产品的质量出了问题，全部是测试部门的责任。”而测试负责人显然对软件工程的理论有一定的了解，所以当然对这个在软件工程领域中常见的错误观点给予了猛烈的反驳。会议以后，

争论的方式改成了邮件，并且相继有多人加入了讨论的行列。在争论渐渐平息之后，公司开发部门的总负责人认为这个讨论很好，于是把经过多人多次回复的邮件转发给所有开发人员和测试人员，当时邮件的总长度可能已经超过了一万字。

此时企业的规模几乎达到了顶峰，在经营上也开始出现了一些不好的苗头：经过了多年的高速发展，增长的速度已经放缓，员工创业的激情逐渐消退；市场竞争开始加剧，公司的利润率开始下降。各种内忧外患使得公司高层忧心忡忡，他们迫切地想找到一条企业发展的长久之计。与此同时，国家开始推广现代企业制度，并且在政策上给予了强有力的支持。最终公司高层决定实施 ISO 9000 标准。

ISO 9000 的实施在客观上极大地提高了测试部门的地位，系统测试的重要程度也达到了一个前所未有的高度。产品的测试报告可以决定一个项目的命运，可以一票否决一个产品的发行。

此阶段软件工程环境的特点是：

- 初步建立了标准化的软件工程环境。
- 开发人员和测试人员的素质整体上有所下降。
- 由于市场的拓展，相关人员在行业知识方面的欠缺变得突出。

ISO 9000 质量体系虽然在公司内建立起来了，但是由于观念上的问题，核心的开发人员都不愿意从事质量保证工作，所以建立的 ISO 9000 体系本身和实施过程都非常教条，开发人员对 ISO 9000 认可程度不高。此外，测试部门的权利过大，造成了开发人员和测试人员的矛盾。当测试人员的决定影响了公司或者部门的产品策略时，也造成了测试人员和市场人员的矛盾。

由于测试人员和开发人员分别属于不同的部门，他们之间的交流不多，因此测试工作的有效性很成问题。另外，测试人员的基本素质和行业知识方面的欠缺渐渐成为影响测试工作质量的重要因素。

总而言之，虽然公司在软件质量保证方面下了很大的力气，但是标准化的软件工程环境并没有起到应有的作用，所以此阶段开发的软件产品的质量也没有多大的提高。

务实的软件工程环境

到了这个阶段，公司的内部和外部环境更加恶劣。为了重新在市场上取得优势，市场人员要求开发部门对来自市场上的信息快速做出反应，因此各个开发部门的压力越来越大。在这种情况下，开发人员和市场人员对测试部门“轻易”否决他们负责的产品怨言很多，认为根本无法实施他们的产品战略。

为了扭转这种不利的局面，公司高层决定调整公司构架。调整的重点之一是把开发人员、测试人员合并到各个事业部，这种调整在客观上提高了市场人员的权利，但是削弱了测试部门的权利。测试部门重新回归到一个从属的地位。在这个阶段也发生了一起标志性的事件：有一位坚持原则的测试经理，坚决不同意发行一个他认为问题较多的产品。产品经理和开发组长万般无奈，只好找到上级领导向这位测试经理施压。在压力下，这位测试经理最终还是在产品的质量合格证上签了字。从此以后，所有的测试评审工作均由一个评审委员会完成，不再是一票否决，也不再由测试经理独立完成。而测试经理们也学乖了，他们总是把产品的遗留问题和对产品的担忧一一列出，不再对产品是否应该发行发表正式的意见（事实上，在新的公司构架下，他们的影响力实在有限）。

随着公司员工正常的新陈代谢和质量体系潜移默化的影响，软件工程的理念逐渐深入人心，开发人员不再对软件工程环境所涉及的各种规章制度有过多的抵触情绪，并且开始从中受益。

测试部门的领导已经意识到了行业知识的重要性，并且在员工培训和招聘工作中力争弥补这个欠缺。此阶段招收的第三代测试人员开始向相关专业的毕业生倾斜。

由于公司经营状况不佳，所以公司高层对费用的控制越来越严格。他们认识到一旦到了测试阶段，开发费用实际上已经花得差不多了，到这个时候再决定枪毙一个项目对公司的风险太大。因此公司高层开始考虑对整个软件生命周期进行控制，他们决定实施更符合软件企业的 CMM 规范。

此阶段的软件工程环境的特点是：

- 软件工程环境逐步完善，大家对软件工程环境的建设从理想主义转向务实，对整个软件生命周期各个阶段的理解也逐渐加深。
- 重视测试人员的行业背景。
- 对市场人员的意见空前重视。

到了这个阶段，公司终于开始从结构化的软件工程环境中受益了。虽然公司的内部和外部环境没有起色，开发人员和测试人员的基本素质也呈下降的趋势，测试部门不再具有否决一个产品发行的权利，但是由于整个公司的软件工程环境不断完善，所以产品的质量还是有所上升。

然而，此阶段软件工程环境的缺点也是非常明显的。主要问题是：

- 由于企业经营状况不佳，开发资源和测试资源空前紧张。
- 经过多年的积累，测试用例库的规模增长很快。同时，由于市场领域的不断扩展，需要测试的内容越来越多，测试的周期越来越长。
- 测试人员的行业知识不再成为问题，但是他们的基本素质和技术背景却成了制约测试工作有效性的主要原因。

怎样阅读本书

从上面的例子可以看出，这家公司用了多年时间才把软件开发工作初步纳入正轨。可见，一个好的想法在付诸实施以前，还应该考虑许多非技术的因素。在下面的观点中，有些在本书中已经提到了，有些是我多年实际工作的总结，与想在本企业中掀起一场质量保证革命或者改良运动的各位朋友分享。

- 任何软件工程环境的改进活动，其目标、实施策略必须和企业或者组织文化的精神一致。
- 必须获得公司高层、部门高层的支持。这一点甚至是 CMM 实施过程中的第一要素。
- 任何工作都是由人完成的，高素质的人才可以弥补制度上的缺欠，而且有些想法只有高素质的人才能成功实施。
- 软件工程环境的完善程度，取决于公司或者部门的投入。在改进活动中一定要量力而行，不要追求根本不可能实现的目标。

虽然本书是一本实用性很强的书籍，但是本书还是给出了大量测试生命周期中的原则。我认为，深刻地理解这些原则对读者的帮助会更大，因为它们适用的期限会远比具体的方法长得多。例如：测试人员应该及早介入软件开发周期中的需求阶段，这条原则

至少 30 年不会动摇。我希望本书的读者都能以这种态度阅读本书。

致 谢

我要感谢 Jane 和 Run 为本书作了认真细致的校对工作。他们的意见为本书增色不少。

我更要感谢清华大学出版社的尤晓东先生和刘伟琴女士。感谢尤晓东先生给了我这次宝贵的学习机会，感谢刘伟琴女士为本书所作的具体工作。

最后，我还要特别感谢本书的读者，如果你们能从本书中汲取一点有用的营养，得到一些帮助，那么我将会感到无限欣慰。这是我从事翻译工作的初衷。

由于时间仓促以及水平有限，错误之处在所难免，敬请读者批评指正。

前 言

在大多数软件开发组织中，测试工作是应用程序的最后一道“质量关”，它允许或者拒绝应用程序从理想的软件工程环境进入真实的现实世界。伴随着这个角色的是巨大的责任：应用程序的成功、甚至可能是组织成功的赌注全部押在软件产品的质量上。

众多小规模测试任务必须由测试组来完成和管理，事实上，由于测试任务的数量过多，所以测试工作关注的只是测试一个应用软件的功能，很少顾及测试工作需要的外围工作。获得合适的测试数据、应用程序需求和构架的可测试性、适当的测试过程标准和文档，以及硬件和设备等诸如此类的问题即便不是没有考虑，也往往会到项目生命周期的后期才予以考虑。对于规模较大的项目，单纯使用测试脚本和测试工具是不够的，大多数有经验的软件测试人员都会认同这一点。

一般情况下，我们只有通过具体实践才能熟知使测试工作从头到尾都获得成功所需要的各个环节的有关知识。在项目生命周期中，如果某些任务完成得早，那么测试工作的效率就会高得多，这是一条有价值的经验。当然，在我们意识到这一点时，当前的项目通常已不可能从这条经验中获益了。

本书提供了一些实践经验和关键理念，组织可以利用这些实践和理念成功地实现有效的测试工作。本书的目的是为读者提供各种经过精心挑选的技术和建议，软件从业人员能够直接用它们来改进产品，同时也能避免损失惨重的过失和疏忽。本书详述了 50 条最好的软件测试实践，分为 10 部分进行讲述，这 10 部分大致和软件生命周期的各个阶段相对应。这种结构本身就阐述了软件测试工作中一个关键理念：为了获得最大的有效性，测试工作必须完全融入软件开发过程。我们必须避免把测试工作当成“项目流程”中一个独立的步骤（在软件开发周期的最后阶段），这是一个很常见的错误。

本书的内容涵盖了从有关过程和管理的内容（例如：管理变化的需求和测试组的构成）到技术方面的话题（例如：提升系统可测试性的方法和把单元测试融入开发过程）。虽然在需要的地方本书中也出现了一些伪代码，但是本书的内容并不局限于任何特定的技术和应用程序平台。

但是，还有很多测试工作以外的因素也会强烈地影响项目的成败，认识到这一点非常重要。虽然包含测试工作在内的完整软件开发过程会确保与软件工程有关的工作成功，但是所有项目还要处理有关商业用例、预算、时间表和组织文化方面的问题。在某些情况下，这些问题会和高效的工程环境的需要发生冲突。应用本书中的建议的前提是：为

了测试工作的成功，组织能够适应测试工作的需要，并且为这些需要提供支持。

本书的组织结构

本书由 50 个独立的条目组成，它们覆盖了 10 个重要的方面。这些经过挑选的最优实践出现的顺序和系统开发生命周期的各个阶段的先后顺序一致。

读者可以一条接一条、一章接一章地顺序阅读本书。当需要获得或者了解有关一个特殊问题的信息时，也可以直接跳到特定的条目。虽然在每一章中会引用其他章节或者其他书籍的内容，它们会有助于向读者提供更多的信息，但是每一章的内容基本上是自成体系的。

第 1 章描述了测试工作在需求阶段需要考虑的问题。在需求阶段，包括测试组代表在内的所有涉众必须参与需求工作，并且必须收到需求变更通知，这是非常重要的。此外，对于任何大型项目来说，基于需求开发测试用例都是一个最基本的理念。测试组参与此阶段工作的重要性怎么强调都不过分，只有在这个阶段才能获得对系统和它的需求的全面理解。

第 2 章描述了测试规划活动，其中包括：对测试工作目标的了解方法，确定测试策略的方法，以及有关数据、环境和软件本身需要考虑的问题。在软件生命周期中，规划工作必须及早开始，这是因为我们需要为成功地实施测试工作预留时间。及早规划使得我们可以对测试进度和预算进行估计，并且使之获得批准和加入整个软件开发计划。我们必须不断地监控这些估计，并且和实际情况进行比较，这样就可以根据需要对它们进行修正，并且实现预期的目标。

第 3 章主要讨论测试组的人员构成。所有成功的测试工作的核心是执行它的人。一个成功的测试组需要同时具备技术和行业两方面的知识，还要有结构化和简明的角色与职责划分。为了确保测试工作成功完成，在整个测试过程中，必须不断地评估每个测试组成员的有效性。

第 4 章讨论了有关测试系统构架方面的考虑。为了保证系统本身是可以测试的、能够进行灰箱测试和有效进行缺陷诊断，考虑这些因素是非常重要的，但是它们经常被忽视。

第 5 章详细描述了如何有效地设计和开发测试过程，其中包括在测试创建和文档化方面需要考虑的问题，还讨论了最有效的测试技术。随着时间的推移和系统开发迭代的

继续，需求和设计会不断精化，因此测试过程也要不断精化，它们需要加入新的和修改后的需求以及系统功能。

第6章讨论了在整个测试策略中，开发人员进行单元测试所扮演的角色。在实现阶段中，单元测试会显著地提高软件质量。如果全面地执行了单元测试，以后的测试阶段会更成功。但是，基于对问题的了解的、随意的单元测试和基于系统需求的、结构化的、可重复的单元测试是有区别的。

第7章讲解了有关自动测试工具的问题，其中包括：在项目中使用恰当的工具类型、有关定制开发还是购买的决定和为组织选择恰当的工具需要考虑的因素。本章描述了多种类型的测试工具，它们可以用于开发生命周期的各个阶段。此外，本章还讲到了开发定制工具方面的问题。

第8章讨论了为自动测试选择最佳实践方面的问题。本章描述了如何正确地运用记录/回放工具、自制测试工具和回归测试。

第9章提供了测试一个应用软件非功能性方面的信息。如果满足了应用程序的非功能性需求（包括性能、安全性、可使用性、兼容性和并发性测试），那么会提升应用程序的整体质量。

第10章提供了测试执行的管理策略，其中包括：追踪测试过程执行和缺陷生命周期的正确方法以及收集用于估计测试进程的度量。

本书面向的读者

本书面向的读者包括：质量保证人员、软件测试人员、测试负责人和测试经理。本书中提供的大部分信息对于项目经理和软件开发人员也是有价值的，他们可以利用这些信息改进软件项目的质量。

致 谢

感谢所有在本书撰写过程中向我提供帮助和支持的软件专业人士，其中包括参加了本人讲授的自动软件测试(Automated Software Testing)、高质量的Web系统(Quality Web Systems)和有效测试管理(Effective Test Management)这几门课程的所有学生，各个公

司中协助本人完成各种测试工作的合作者，还有本人各种著作的合著者。他们提出的问题、见解、反馈和建议直接或者间接地增加了本书内容的价值。特别感谢 Douglas McDiarmid 为本书做出的杰出贡献。他提供的信息极大地丰富了本书的内容，同时也极大地提升了本书素材的整体质量。

还要感谢下列人员，他们提供了非常有价值的反馈。他们是：Joe Strazzere, Gerald Harrington, Karl Wieggers, Ross Collard, Bob Binder, Wayne Pagot, Bruce Katz, Larry Fellows, Steve Paulovich 和 Tim Van Tongeren。

还要感谢 Addison-Wesley 的工作人员对本书的支持，特别是 Debbie Lafferty, Mike Hendrickson, John Fuller, Chris Guzikowski 和 Elizabeth Ryan。

最后，还要感谢 Eric Brown，他为本书设计了有趣的封面。

Elfriede Dustin

目 录

第 1 章 需求阶段	1
第 1 条：测试人员及早介入.....	2
第 2 条：验证需求.....	4
第 3 条：需求就绪后马上设计测试过程.....	8
第 4 条：确保需求变化的传达.....	11
第 5 条：注意在现存系统上进行开发和测试.....	14
第 2 章 编制测试计划	17
第 6 条：了解手头的任务和相关的测试目标.....	18
第 7 条：考虑风险.....	22
第 8 条：根据功能优先级安排测试工作.....	28
第 9 条：牢记软件方面的问题.....	30
第 10 条：获得有效的测试数据.....	32
第 11 条：规划测试环境.....	36
第 12 条：估计测试准备和执行所需的时间.....	38
第 3 章 测试组	47
第 13 条：定义角色和职责.....	48
第 14 条：测试技巧、行业知识和经验三者缺一不可.....	55
第 15 条：评估测试人员的有效性.....	57
第 4 章 系统构架	67
第 16 条：了解系统构架和基本组件.....	68
第 17 条：确认系统的可测试性.....	70
第 18 条：使用日志增加系统的可测试性.....	71
第 19 条：验证系统支持调试和发行两种执行模式.....	74
第 5 章 测试设计和测试文档	77
第 20 条：分而治之.....	78
第 21 条：使用测试过程模板和其他测试设计标准.....	82
第 22 条：根据需求得到有效的测试用例.....	86
第 23 条：把测试过程当作“动态”的文档.....	89
第 24 条：利用系统设计和系统原型.....	91

第 25 条: 设计测试用例场景时采用经过检验的测试技术	92
第 26 条: 在测试过程中避免包含限制和详细的数据元素	96
第 27 条: 运用探索性测试	98
第 6 章 单元测试	101
第 28 条: 用结构化的开发方法来支持有效的单元测试	103
第 29 条: 在实现之前或者与实现同时开发单元测试	108
第 30 条: 使单元测试的执行成为生成过程的一部分	111
第 7 章 自动测试工具	115
第 31 条: 了解各类测试支持工具	116
第 32 条: 自主生成一个工具	120
第 33 条: 了解自动测试工具对测试工作的影响	122
第 34 条: 关注组织的需要	126
第 35 条: 在应用程序的原型上对工具进行测试	129
第 8 章 自动测试: 选择最好的实践	131
第 36 条: 不要过分依赖记录/回放工具	132
第 37 条: 必要时自制开发一个测试工具	134
第 38 条: 使用经过考验的测试脚本开发技术	138
第 39 条: 尽量使回归测试自动化	142
第 40 条: 实现自动生成和烟雾测试	146
第 9 章 非功能性测试	149
第 41 条: 不要事后才考虑到非功能性测试	150
第 42 条: 用产品级数据库进行性能测试	153
第 43 条: 为预期受众定制可使用性测试	155
第 44 条: 特定需求和整个系统都需要考虑安全性	157
第 45 条: 研究系统对并发性测试计划的实现	159
第 46 条: 为兼容性测试建立高效的环境	163
第 10 章 管理测试的执行	165
第 47 条: 明确定义测试执行周期的开始和结束	166
第 48 条: 隔离测试环境和开发环境	168
第 49 条: 实现缺陷追踪生命周期	170
第 50 条: 追踪测试工作的执行	175
术 语 表	179

第1章 需求阶段

最有效的测试工作应该始于项目的开始阶段，远远早于程序代码的编写阶段。首先需要检验的是需求文档，只有如此，在项目的后续阶段测试工作才能专注于保证应用程序代码的质量。在项目生命周期的早期——详细设计和编码工作之前，消除需求工作中的缺陷能够使昂贵的返工工作降到最低。

软件应用程序或者系统的需求说明书必须非常详细地描述它的功能。与提供需求的人员进行交流是确定需求的工作中最具挑战性的工作之一。每条需求必须阐述得准确和明确，这样它的读者对需求的理解才会完全相同。

如果使用了一致的方法来撰写需求文档，那么任何需求收集人员就有可能有效地参与需求过程。一旦某条需求浮出了水面，那么通过详细咨询相关人员这条需求就可以被测试和澄清。我们可以利用各种各样的需求测试来保证每条需求是恰当的，并且大家对它的含义有相同的理解。

第 1 条：测试人员及早介入

测试人员需要从项目生命周期之初就开始介入，这样他们才能准确地理解测试的对象并且和其他涉众一起生成可测试的需求。

缺陷预防是指在各种错误遗留到后续开发阶段之前，运用各种技术和过程来发现和避免这些错误。缺陷预防工作在需求阶段的效率最高，此时修正缺陷所需的改动很小：需要改动的仅仅是需求文档，可能还需要相应地修改此阶段制定的测试计划。如果测试人员（和其他涉众一起）从开发生命周期之初就开始参与项目，那么他们就能够协同发现遗漏、矛盾、含糊的地方和一些其他问题，这些问题可能会影响项目需求的可测试性、正确性以及其它测试质量。

对于一条需求，如果可以设计出一个过程来执行所测试的功能，若输出结果是可以预先知道的，并且能够通过编程或者人工方式加以验证，那么我们就称这条需求是可测试的。

测试人员需要彻底地了解产品，只有这样他们才可能设计出更出色和更全面的测试计划、测试设计、测试过程和测试用例。测试组及早介入，就可以避免在项目生命周期中的后续阶段对产品的功能行为不理解。此外，测试组及早介入，还可以了解到应用程序的哪些方面对最终用户来说最关键以及哪些元素的风险最大。这样测试组就可以首先把精力集中在应用程序中最重要的部分，避免对不经常使用的部分过度测试而对重要的部分又测试不充分。

有些组织不是促使测试人员及早介入，而是把测试人员完全当作需求和其他软件开发工作产品的客户，当软件交给测试人员的时候，才要求他们学习应用程序的使用及行业背景知识。这种做法对于小项目还可以接受，但是在复杂的环境中，如果测试人员在需求、分析、设计甚至是部分实现阶段之后才接触到应用程序，那么就不能指望他们发现应用程序中所有的重要缺陷。测试人员所要掌握的不仅仅只是软件的“输入和输出”，他们还需要掌握更深入的知识，而这种知识只有了解撰写产品功能说明书的思考过程才能获得。这种理解不仅能够提高测试人员开发的测试过程的质量和深度，而且测试人员也可以针对需求提出反馈意见。

在生命周期中发现缺陷越早，那么修正缺陷的代价就越小。表 1.1 列出了在项目生