



软件测试 过程改进

Software Testing in the Real World
Improving the Process

Edward Kit 著 李新华 陈丽容 马立群 等译



机械工业出版社
China Machine Press



中信出版社
CITIC PUBLISHING HOUSE



软件工程技术丛书

测试系列

Edward Kit 著

李新华 陈丽容 马立群 等译

软件测试 过程改进

Software Testing in the Real World
Improving the Process



机械工业出版社
China Machine Press



中信出版社
CITIC PUBLISHING HOUSE

本书系统全面地介绍了软件测试方法,为改进软件测试过程提供了一套行之有效的办法。内容包括软件测试成熟度、改进测试过程的基本框架、验证及确认测试、测试工具、测试管理技术等。附录中给出了软件工程和测试的相关标准、验证及确认测试审查清单示例、测试工具选择等。书中包含了许多测试技巧和指南,帮助组织内部制定战略,改进测试过程。书中所讨论的技术对于规模不同、过程成熟度各异的组织都适用。

本书结构清晰、内容丰富,适于软件测试人员、产品开发人员、管理人员、质量保证人员、系统分析人员、工具开发人员以及其他与软件测试有关的工作人员阅读,也可作为高等院校计算机专业师生的参考书。

Edward Kit: Software Testing in the Real World: Improving the Process (ISBN 0-201-87756-2).

Copyright © 1995 by Pearson Education Limited.

This translation of Software Testing in the Real World is published by arrangement with Pearson Education Limited.

本书中文简体字版由英国 Pearson Education 培生教育出版集团授权出版。

本书版权登记号: 图字: 01-2002-0858

图书在版编目 (CIP) 数据

软件测试过程改进/基特 (Kit, E.) 著; 李新华等译. —北京: 机械工业出版社, 2003.3
(软件工程技术丛书-测试系列)

书名原文: Software Testing in the Real World: Improving the Process
ISBN 7-111-11635-6

I. 软… II. ①基…②李… III. 软件-测试 IV. TP311.5

中国版本图书馆 CIP 数据核字 (2003) 第 009748 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: 邱李华 刘立卿

北京牛山世兴印刷厂印刷 · 新华书店北京发行所发行

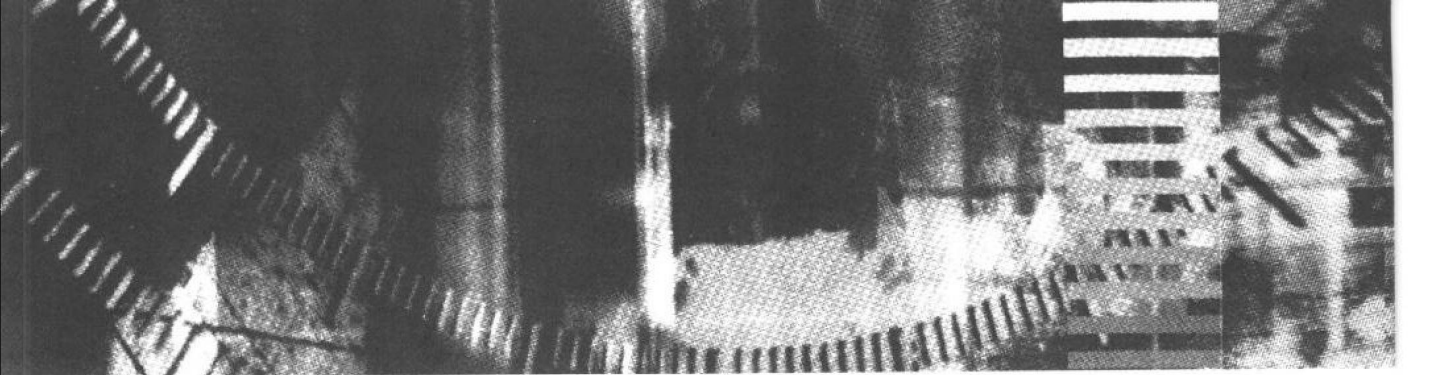
2003 年 4 月第 1 版第 1 次印刷

787mm×1092mm 1/16 · 13.25 印张

印数: 0 001-5 000 册

定价: 29.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换



序

多年来，我有幸与多位测试领域的著名专家共事过，而 Ed Kit 更是其中的佼佼者。他是一位软件专家、测试专家与质量管理专家，也是一位优秀的研讨会组织者和咨询专家。他在许多实践方面对测试技术的发展做出了卓越的贡献。我本人与 Ed Kit 相处、共事十载，可谓受益匪浅。本书是 Ed Kit 对测试领域的又一重要贡献，使测试界的同仁可以进一步分享他的真知灼见。

本书汇集了 Ed Kit 从事测试工作的亲身经验，强调测试的过程和改进办法。书中既有实践经验，也有不少好的窍门，当你阅读本书时，可以从实际应用深入到基本思想，快速浏览潜在的改进过程（从第 3 章“白纸方法”开始），书中提供了许多技巧和练习，并且包含了八个附录，附录中包括了与测试相关的标准概要以及测试资源和测试工具的参考。多数读者还将发现，本书包括了许多易于理解和应用的实用思想。

第二部分主要是为新手或在初创的组织机构内工作的人员设计的。其中的章节对测试的内涵进行了阐述并对过程的改进提供了框架。尽管不少内容看起来是很平常的“常识”，但真正要将其付诸实施却一点都不平常。本书非常有助于从实用的角度对软件测试做出改进。

本书简明易懂，非常适合那些想对软件测试做出改进的人员阅读。

许多读者，尤其是业内的开发人员、测试人员和管理人员应该为本书的出版感到高兴，不妨“测试”一下本书。不少读者也许读过出版已达十年之久的拙作《*The Complete Guide to Software Testing*》。本书在许多方面实际是对拙作的更新，不少方面我一直也想做出修改，只可惜时间有限、力不从心。我还想了解各位读者的兴趣所在以及成功的测试经验。如能分享各位的测试结果，当不胜感谢。

Bill Hetzel[⊙]

软件质量工程公司总裁

[⊙] Bill Hetzel 博士是软件测试学科的先驱，1972 年世界首次软件测试会议的召集人。

软件测试。

它既令人兴奋，也令人烦恼。

既令人羡慕不已，也令人望而却步。

要想在预算内按时交付高质量的软件，测试是必不可少的。

本书是专为测试过程的改进而撰写的，面向测试专家、产品开发人员、管理人员、质量保证专家、维护人员、项目管理人员、审计人员、系统分析人员、工具开发人员以及其他与软件测试有关的工作人员。通过阅读本书可以进一步了解测试工作，从而使开发人员和非测试领域的专业工作者进一步提高专业水平。

为了了解测试并对测试工作有所改进，必须从广义的角度来观察测试过程，即应该将人员、方法、工具、测试和领导艺术集成到测试软件产品的过程中。本人从事测试实践、教学和咨询工作 20 余年，发现人们为了改进测试过程反复提到以下一些关键性问题：

方法问题

- 哪些是测试的主要方法，我们什么时候用到这些方法，怎样实施这些方法？
- 怎样避免缺陷转移并提高发现缺陷的能力？
- 怎样理解并利用有关的测试标准和术语？
- 怎样生成有意义的测试计划及其他重要的测试文件？
- 怎样发现并确定软件测试过程中需要改进的地方？

领导艺术问题

- 怎样组织软件测试工作，从而使其可以度量、可以控制？
- 从何处可以寻求有关软件测试的帮助？
- 怎样确保在中、长期内改进工作的可支持性？
- 怎样组织测试工作才能高效地协调人们之间的相互关系？
- 怎样扬长避短？

工具和技术问题

- 使测试过程自动化的最重要的时机是什么？
- 怎样利用度量手段了解并控制软件测试过程？
- 怎样挑选测试工具和销售商并有效地利用测试工具？
- 怎样将风险分析技术用做测试决策的基础？

根据我的经验，在方法、领导艺术和技术方面采取一种平衡的策略是改进测试最有效的方法。

大多数的软件开发组织由于软件过程的不成熟而蒙受损失。因此，专业工作者和管理人员通常会问：“我们应该怎么办呢？”。本书为有效地改进软件测试过程提供了一个工具箱。既然是工具箱，就不是简单地从 A 到 Z 教人们如何去做的某个方法，而是一系列可供选择的技术，可以单独使用也可综合采纳，帮助人们制订并实现各类改进目标。

第一部分是用于明确方向的。这一部分介绍了软件测试中的六个基本问题，软件测试的历史以及改进测试过程的一个简单、实用的方法。从一般到具体，对软件工程和测试这些在新生事物不断涌现的时代应运而生的新兴学科进行了阐述。

第二部分讨论根据实际情况确定具体目标，并说明这些目标与各种正式的与不太正式的测试定义、测试目的关系，以及与对于专业工作者的日常工作来说非常重要的各类现行标准之间的关系。

介绍了测试过程的基本形式，阐明了成本和风险方面的一些重要的现实问题，这些问题涉及到决策过程中的测试对象、时间和方法。这一部分还对验证和确认这些测试活动所涉及的有关方面进行了探讨，从而为进一步深入地探讨验证和确认做好了准备。对规划制订、软件工程成熟度目标、配置管理、标准和工具等问题进行了阐述和定义。

第三部分介绍基本的验证和确认任务，包括对测试成本的规划与控制，对资源的充分利用等。阐述了如何发挥工具在各种测试活动中的作用，说明了工具的分类，并就如何获得工具提供了指南。探讨了如何合理并建设性地利用度量，并指出了应如何正确地利用度量结果。

第四部分介绍涉及测试工作的组织和管理问题，这是测试成功与否的决定因素。概括了当今一些公司的最优方法，然后集中讨论了如何在中、短期内不断完善测试工作。

本书的正文尽量包括许多定义、范例和参考文献。附录包含了各种补充材料（由于太多，无法收集在本书的正文内）。这些材料有范例清单、练习、计划制订文件和范例工作文件。还收集了正式标准、规范及其他一些有用的信息源，如期刊、工具、会议、时事通讯、用户群和加注的参考书目。

Ed Kit

1995 年 9 月于加州圣何塞

序 前言

第一部分 软件测试过程成熟度

第 1 章 软件测试的六个要点..... 3

- 1.1 要点 1: 测试过程的质量决定测试工作的成败 3
- 1.2 要点 2: 使用早期软件生存周期测试技术可避免缺陷转移 3
- 1.3 要点 3: 软件测试工具的时代已经到来 4
- 1.4 要点 4: 改进测试过程必须有专人负责 4
- 1.5 要点 5: 测试是一个专业技术学科, 要求富有经验的专门技术人员 4
- 1.6 要点 6: 培养破旧立新的、积极的精神 5

第 2 章 技术与实践 6

- 2.1 一门新兴学科的短暂而丰富的历史 6
- 2.2 我们的现状如何 8
- 2.3 测试应该如何定位 8
- 2.4 参考文献 9

第 3 章 白纸方法 10

第二部分 测试过程改进框架

第 4 章 树立注重实际的观点..... 15

- 4.1 我们的目的是什么 15
- 4.2 关于错误你知道多少 15
 - 4.2.1 什么和为什么 16
 - 4.2.2 错误在哪里 16
 - 4.2.3 错误的成本 16
- 4.3 关于测试的一些定义 17
- 4.4 优秀测试人员的测试态度 18
 - 4.4.1 测试者寻找错误 19
 - 4.4.2 测试是破旧立新 19
 - 4.4.3 测试人员找出错误, 对事不对人 19
 - 4.4.4 测试提高产品价值 20
- 4.5 测试人员怎样工作 20
- 4.6 现在能做什么 20
- 4.7 参考文献 21

第 5 章 重要选择: 测试什么、何时测试、怎样测试 22

- 5.1 风险及风险管理 22
- 5.2 尽早开始测试 24
- 5.3 测试过程的基本形式: 验证和确认 25
- 5.4 测试、开发生存周期及合同 25
- 5.5 有效测试 27
- 5.6 测试的效益 27
- 5.7 现在能做什么 28
- 5.8 参考文献 28

第 6 章 重要方法: 测试的框架 29

6.1 计划	29
6.1.1 验证计划中要考虑的因素	30
6.1.2 确认计划中要考虑的因素	30
6.2 软件工程成熟度和 SEI	30
6.2.1 SEI 过程成熟度等级	31
6.2.2 过程成熟度是怎样影响测试的	32
6.3 配置管理	33
6.3.1 什么是配置管理	34
6.3.2 在 CM 方面测试所关心的问题	34
6.4 标准	35
6.4.1 IEEE/ANSI 标准	36
6.4.2 ISO 9000、SPICE 及其他标准	37
6.5 正式文档	38
6.6 测试件	38
6.7 度量	39
6.8 工具	40
6.9 现在能做什么	41
6.10 参考文献	41

第三部分 测试方法

第 7 章 验证测试	45
7.1 验证的基本方法	45
7.1.1 验证的组织形式	46
7.1.2 审查: 关键元素和阶段	47
7.1.3 走查	47
7.1.4 走查: 关键元素	47
7.1.5 伙伴检查	48
7.2 发挥验证的作用	48
7.2.1 验证要做什么	48
7.2.2 审查单: 验证工具	49
7.3 验证不同阶段的文档	50
7.3.1 验证需求	50
7.3.2 好的需求规格说明的特性	50
7.3.3 验证功能设计	52
7.3.4 验证内部设计	53
7.3.5 验证代码	54
7.4 从验证中获取最大收益	55

7.4.1 作者	55
7.4.2 开发小组	56
7.4.3 审查小组	56
7.4.4 高收益的验证	56
7.5 验证的三个成功因素	57
7.5.1 成功因素 1: 过程责任人	57
7.5.2 成功因素 2: 管理部门的支持	57
7.5.3 成功因素 3: 培训	57
7.6 建议	57
7.7 参考文献	58
第 8 章 确认测试	59
8.1 确认概述	59
8.1.1 覆盖	60
8.1.2 基本测试策略	61
8.1.3 确认任务与测试覆盖	61
8.1.4 测试基础	61
8.1.5 确认策略	62
8.2 确认方法	63
8.2.1 基于功能测试的黑盒方法	63
8.2.2 基于内部测试的白盒方法	68
8.3 确认活动	71
8.3.1 低层测试	72
8.3.2 高层测试	73
8.3.3 再测试	80
8.3.4 累进测试和回归测试	80
8.3.5 测试执行的筹划	81
8.3.6 测试点	82
8.4 确认测试的推荐策略	82
8.5 参考文献	83
第 9 章 控制确认成本	84
9.1 使测试实施成本最小化	84
9.1.1 预运行启动成本	84
9.1.2 执行成本	85
9.1.3 后运行成本	85
9.1.4 降低测试实施成本的建议	85
9.2 降低测试的维护成本	86
9.3 降低确认测试件开发成本	87

9.4 测试件库	88
9.5 建议	89
第 10 章 测试任务、可交付文件及其在 生存周期中的对应阶段	90
10.1 主测试计划	90
10.2 验证测试任务和可交付文件	92
10.2.1 制定验证测试计划	92
10.2.2 验证执行	93
10.3 确认测试任务和可交付文件	94
10.3.1 制定确认测试计划	95
10.3.2 测试结构设计	96
10.3.3 测试件开发——设计和实施 细节	97
10.3.4 测试执行	100
10.3.5 测试评估	102
10.3.6 何时停止	102
10.4 用户手册	104
10.5 产品发布标准	104
10.6 IEEE/ANSI 测试文件概述	105
10.6.1 测试计划和规格说明的文件 结构	105
10.6.2 测试报告的文件结构	106
10.7 把任务和可交付文件对应到生存 周期	106
10.7.1 概念阶段	106
10.7.2 需求阶段	107
10.7.3 功能设计阶段	107
10.7.4 内部设计阶段	107
10.7.5 编码阶段	108
10.7.6 集成和测试阶段	108
10.7.7 运行/维护阶段	109
10.8 参考文献	109
第 11 章 软件测试工具	110
11.1 测试工具的分类	110
11.1.1 评审与审查工具	111
11.1.2 制定测试计划的工具	111
11.1.3 测试设计和开发工具	112

11.1.4 测试执行和评估工具	113
11.1.5 软件测试支持工具	115
11.2 工具采购	115
11.3 参考文献	116
第 12 章 度量	117
12.1 通过度量获得答案	117
12.2 有用的度量	118
12.2.1 度量复杂度	118
12.2.2 度量验证效率	119
12.2.3 度量测试覆盖	119
12.2.4 度量/跟踪测试执行状态	119
12.2.5 度量/跟踪事故报告	119
12.2.6 基于事故报告的测试度量	120
12.3 其他的相关度量	120
12.4 建议	121
12.5 参考文献	121

第四部分 测试管理技术

第 13 章 测试的组织方法	125
13.1 测试的组织和改组	125
13.2 结构设计因素	127
13.3 测试功能的组织方法	127
13.3.1 方案 1: 测试是各人的责任	127
13.3.2 方案 2: 测试是各小组的 责任	128
13.3.3 方案 3: 用专用资源进行 测试	129
13.3.4 方案 4: 将测试组织置于 QA 中	130
13.3.5 方案 5: 隶属于开发的测试 组织	130
13.3.6 方案 6: 集中管理的测试 组织	131
13.3.7 方案 7: 设有测试技术中心的 集中式测试组织	132
13.4 选择正确的方案	133
13.5 参考文献	133

第 14 章 目前的做法、发展趋势和

挑战 135

14.1 图形用户界面：有哪些新东西 135

14.2 应用测试 135

14.3 测试人员与开发人员的比例 136

14.4 软件度量和实践基准研究 136

14.5 参考文献 138

第 15 章 获得可持续收益 139

15.1 实现收益 139

15.2 获得帮助 139

15.2.1 软件测试书籍和业务通讯 140

15.2.2 咨询和培训服务 140

15.2.3 软件测试研讨会 140

15.3 后续工作 141

15.4 参考文献 141

第五部分 附 录

附录 A 软件工程和测试标准 145

附录 B 验证审查单 150

附录 C 验证练习 167

附录 D 确认练习（答案） 181

附录 E 参考书目（包括软件测试工具
一览表） 184附录 F 信息资源：会议、期刊、通讯、
DOD 规范 188

附录 G 专用工具和工具选择 191

附录 H 改进实施示范清单 194

第一部分

软件测试过程成熟度

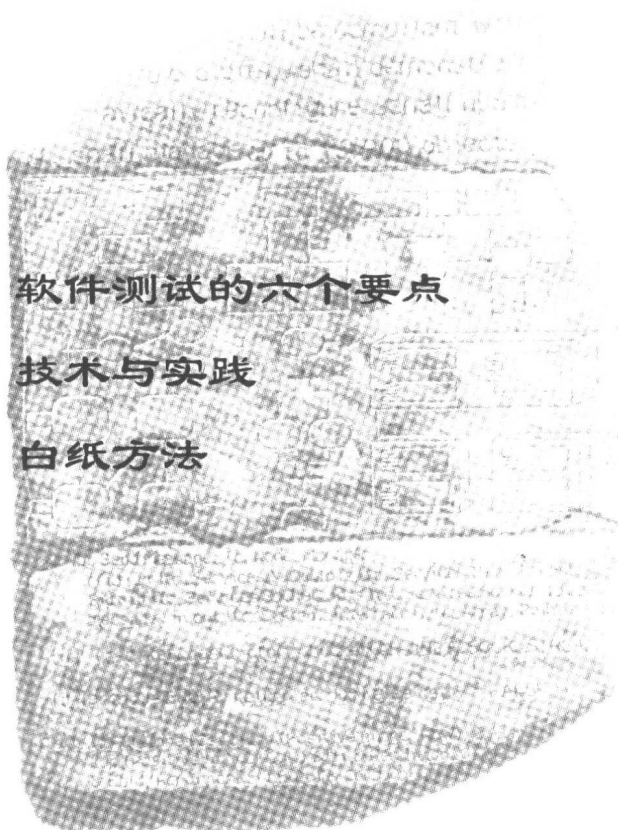
“追求永无止境”

——H.James Harrington

第1章 软件测试的六个要点

第2章 技术与实践

第3章 白纸方法



软件测试的六个要点

本章讲述设计新的或更新旧的软件测试过程的基础。利用软件测试过程，可集成人员、方法、度量、工具和设备对软件产品进行测试。作为本书基础的软件测试过程的要点如下：

- (1) 测试过程的质量决定测试工作的成败。
- (2) 使用早期软件生存期测试技术可避免缺陷转移。
- (3) 软件测试工具的时代已经到来。
- (4) 改进测试过程必须有专人负责。
- (5) 测试是一个专业技术学科，要求富有经验的专门技术人员。
- (6) 培养破旧立新的、积极的合作精神。

1.1 要点 1：测试过程的质量决定测试工作的成败

从根本上说，软件开发过程的质量决定了软件系统的质量，同样的，测试过程的质量决定软件测试的质量和有效性。

测试工作有它自身的周期。测试过程从产品的需求阶段开始，此后，与整个开发过程并行开展，换句话说，伴随着开发过程的每一个阶段，都有一个重要的测试活动。

在一个开发过程不成熟的组织中工作的测试小组将比其他的小组忍受更多的痛苦，但是无论开发机构的成熟状态如何，测试小组能够而且也应该致力于改进小组自身的内部过程。不成熟的测试组织中不成熟的测试过程将导致没有收益的、混乱的、令人沮丧的环境，生产出质量低下、不令人满意的产品。对不成熟测试组织中的测试过程进行富有成效的改进，将促进整个组织开发过程的改进。

1.2 要点 2：使用早期软件生存周期测试技术可避免缺陷转移

有一半以上的错误是在需求阶段引入的。如果错误在它被引入的阶段即被发现，并且对程序进行有效的测试，避免错误从任何一个开发阶段转移到它的后继阶段，这样，错误成本将降到最小。

尽管许多人意识到了这一点，但在实际中，我们常常缺乏手段，无法在错误被引入的阶段立即发现它，而是拖到很后面，常常要拖到功能和系统测试阶段，此时我们已经进入了“混乱带（The Chaos Zone）”。如果我们不能利用那些已得到证明了的可应用到早期开发过程中的测试技术，那么，很可能就失去了提高测试效率的最佳时机。

例如，我们应该学会对诸如需求一类的关键文件进行评审。即使是一个不成熟的组织也能够进行有效的审查，在至少二十年的历史中，审查被不断地证明是一个成功者，成本/收益比一次次地证明（文献已证明）了这一点。

1.3 要点 3：软件测试工具的时代已经到来

经过多年的观察、评估、甚至是等待，现在我们可以说测试工具的时代已经到来。有许许多多、不同种类的工具可供选择，而且他们中间有许多是成熟的、强健的产品。

例如，利用捕获/回放工具，可以完成一天二十四小时的无人参与的测试运行，缩短测试周期，实现测试自动化。对于每一个主要平台都有一类工具：有些工具需要人工干预的更多一些；有些工具更适合客户/服务器类型；有些工具在技术上是精雕细琢的；另一些则是比较简单的。

另一类基本工具是结构覆盖工具，用于确定软件是否已被充分测试，这种工具可特别指出产品的哪部分在测试中已被实际执行到。指望用户成为代码的第一位执行者的想法已经不再能接受了。

掌握获得工具的策略，并采用一个适当的步骤选择工具无疑是很重要的。然而这种步骤往往是基于某些常识，并不需要系统地去做。在获得工具方面，听从独立专家的忠告可能是非常有好处的。

1.4 要点 4：改进测试过程必须有专人负责

当测试小组感到一筹莫展时，就应着手在几个关键地方进行改进了，譬如完善规格说明，强化评审和审查。

管理人员应该指定一名设计师或一个核心小组优先考虑潜在的改进问题，领导测试改进工作，而且必须明确支持他们的工作。虽然这不是什么特别的尖端科学，但却费时耗力。假如我们能在包括有效的测试计划和设计在内的整个测试过程中使用工具，肯定可以获得巨大的帮助。

说到底，软件测试是一个过程，需要有人负责该过程的改进工作，必须有专人计划和管理测试过程的改进进程。

1.5 要点 5：测试是一个专业技术学科，要求富有经验的专门技术人员

软件测试已经成为一种专业——一种职业的选择，是一个可留下足迹的领域。软件测试过程发展十分迅速，形成了一个需要专业人员的学科。今天，任何组织要在软件测试领域获得成功，它必须拥有足够的具有丰富软件测试经验的行家，而且，他们应能从管理者那里得到大力支持。

测试不是一个入门级的工作，也不是其他工作的垫脚石。很多人发现，一旦测试工作真正开展起来，它具有的挑战性将超过产品开发。它应该被组织起来，切实遵守用户至上的宗旨，而不是处于辅助的，容易被产品开发部门或其他组织所左右的地位。

测试应该是独立的，公平的，有组织的，公平分享对产品质量所做贡献的认可和回报。

1.6 要点 6：培养破旧立新的、积极的合作精神

测试要求专门的训练，要求富有创造力。优秀的测试，也就是目前设计和执行成功的测试——发现产品中的缺陷的测试——要求有真正的灵活性，但同时也可以看成是破坏。实际上，以可控的、系统的方式破坏某些东西需要相当大的创造力。优秀的测试者是井井有条地将产品拆开，找出它的弱点，促使它发挥或超水平发挥其性能。

我们的产品能否正常工作？何时会发生故障？我们知道错误总是存在，故障也是难免的，但是，故障在合理的范围内吗？如果考虑产品的临界情况，故障是可以接受的吗？我们将各种重要的可能性都考虑在内了吗？以上这些信息都应该由测试者提供，只有当我们对诸如此类的问题进行了令人满意的探讨后，才能将产品交付用户。

树立正确的“破旧立新”的思想，对于测试成功会产生意义深远的影响。如果我们的目的是指出产品做了不应该做的，没有做应该做的，那么，这样的测试就踏上了成功之路。尽管现在所做的离规范的测试标准还相去甚远，但我们的专业工作人员和管理人员进行“破旧立新”的测试工作中，所取得的成绩确实是令人惊奇的。

成功的测试要求有系统的方法，要求我们特别关注下面几个关键因素：计划、项目和过程控制、风险管理、审查、度量、工具、组织以及专业精神。在整个开发过程中，测试人员为保障产品质量发挥了积极作用，做出了重要贡献，这点是我们牢记在心的。

技术与实践

几个世纪以来，人们在通过各种复杂方法认识和控制客观世界的过程中，始终伴随着痛苦的尝试和惨痛的失败。最终产生了基本的技能、技术指导和最佳实践，并成为特殊领域里专业技术人员的第二本能。

在人类各学科的漫长演化过程中，软件工程还处于萌芽状态。当我们意识到这点时，记住软件开发学科有多么年轻就显得十分重要。直到六十年代后期，“软件工程”一词才开始有了运用基础法则进行系统研究的含义。

此外，技术与实践之间还存在着鸿沟。例如，软件测试已成为大学里软件工程课程的基本成分，工业部门的技术培训正在开展，但这种培训还常常是随意的。已发表的文献（那些真正需要的人不一定能看到）提出了一些前沿的方法，用以更好地开发软件，尤其是达到所期望的软件质量，但是有很多先进方法停留在研究阶段，不能在实际领域里得到检验而有许多研究忽略了现实问题，例如投资的回报。

即使那些得到了很好的检验的方法也没有被工业部门广泛应用，软件系统的开发仍然过分昂贵，充满了严重错误，而且常常滞后，产品维护的费用远高于开发费用。

与此同时，我们的系统变得日益复杂，50%以上的开发工作落在了测试身上。那些负责测试的人员还要为搭建一个切实可行的工作平台，配上适用的测试工具付出极大的努力，为争取一个独立的地位，一定的人力物力资源保证而不懈奋斗。

2.1 一门新兴学科的短暂而丰富的历史

提到“软件工程”一词，总是具有通过努力实现可靠性、功能性、可预测性、节约开支、提高效率的含义，软件测试就涉及到这些方方面面。

在早期的软件开发中，测试的含义比较狭窄，将测试等同于“调试”，纠正软件中已知的故障，常常由开发人员自己完成这部分工作。对测试的投入极少，测试介入也很晚，常常是等到形成代码，产品已基本完成时才进行测试，这种最坏的情况至今依然存在。

直到1957年，软件测试才开始与调试区别开来，作为一种发现软件缺陷的活动。由于一

直存在着“为了使我们看到产品在工作，就得将测试工作往后推一点”的思想，测试仍然是后于开发的活动。潜意识里，我们的目的是使自己确信产品能工作。大学里关于测试的讨论也很少。计算机科学课程中研究数值方法和算法设计，但不包括软件工程，也不包括测试。研究集中在编译、操作系统和数据库，然而它们都不能对研究实际领域里的测试问题提供帮助。

到了 20 世纪 70 年代，尽管对“软件工程”的真正含义还缺乏共识，但这一词条已频繁出现。1972 年在北卡罗来纳大学举行了首届软件测试正式会议，随后，一批专著相继出版（Hetzel, 1973; Myers 1976, 1979）。

Myers 注意到人类实现自我的本性对测试工作的重要影响，将测试定义为“为发现程序错误而执行程序的过程”。他指出如果我们的目的是表明程序没有错误，那么将发现不了几个错误，但如果我们的目的是指出错误的存在，那么将发现许多错误。正确的目标和心态对测试成功有着深远的影响（见第 4 章）。

Myers 及其他几位在七十年代的工作是测试过程发展的里程碑。然而，在实际工业领域，软件测试仍然是可有可无的，在时间和经费紧张的情况下，首先砍掉的项目就是软件测试。测试工作总是开展得太晚，以至于从一开始就没有足够的时间使测试工作得以正常进行。管理人员甚至很愿意压缩测试过程，“因为没有时间测试，不管怎样都得交付产品了”（也就是说，所得到的回报似乎要超过所冒的风险），这话是不是听起来很耳熟，因为这种现象在许多组织仍然普遍存在。

直到上世纪 80 年代早期，“质量”的号角才开始吹响。软件开发人员和测试人员开始坐在一起探讨软件工程和测试问题。制定了各类标准，包括 IEEE（Institute of Electrical and Electronic Engineers）、美国的 ANSI（American National Standard Institute）标准以及 ISO（International Standard Organization）国际标准，这些标准过于庞大，根本无法在日常工作中将它们全部消化。不管怎样，它们中确实包括了一些重要的指南和合同书的底线，提供了非常宝贵的参考。

上世纪 90 年代，测试工具终于盛行起来。人们普遍意识到工具不仅仅是有用的，而且要对今天的软件系统进行充分测试，工具是必不可少的。我们已经看到各种工具在测试中发挥的重要作用。今天几乎所有的软件开发公司都在研制工具，工具已成为测试过程的重要组成部分。但不管怎样，工具毕竟是工具，它们本身是不能解决实际问题的。

开发与测试的发展

	1960	1970	1995
软件规模	小	适中	超大
软件复杂性	低	中等	高
开发队伍规模	小	中等	大
开发方法及标准	特别	适中	复杂
测试方法及标准	特别	早期	正在形成
独立测试组织	很少	有些	许多
测试重要性的认可	很少	有些	重要
测试专业从业人数	很少	很少	许多