TURING 图灵程序设计丛书

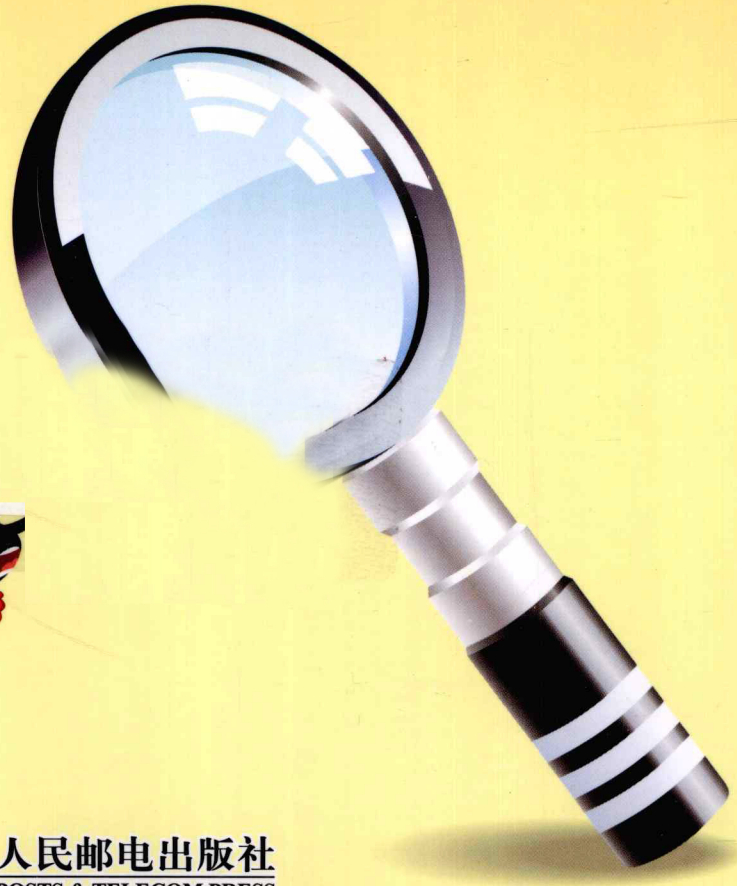
AMACOM

Debugging

The 9 Indispensable Rules for Finding Even the Most Elusive Software and Hardware Problems

调试九法

软硬件错误的排查之道



[美] David J. Agans 著

赵俐 译

人民邮电出版社

POSTS & TELECOM PRESS

TURING 图灵程序设计丛书

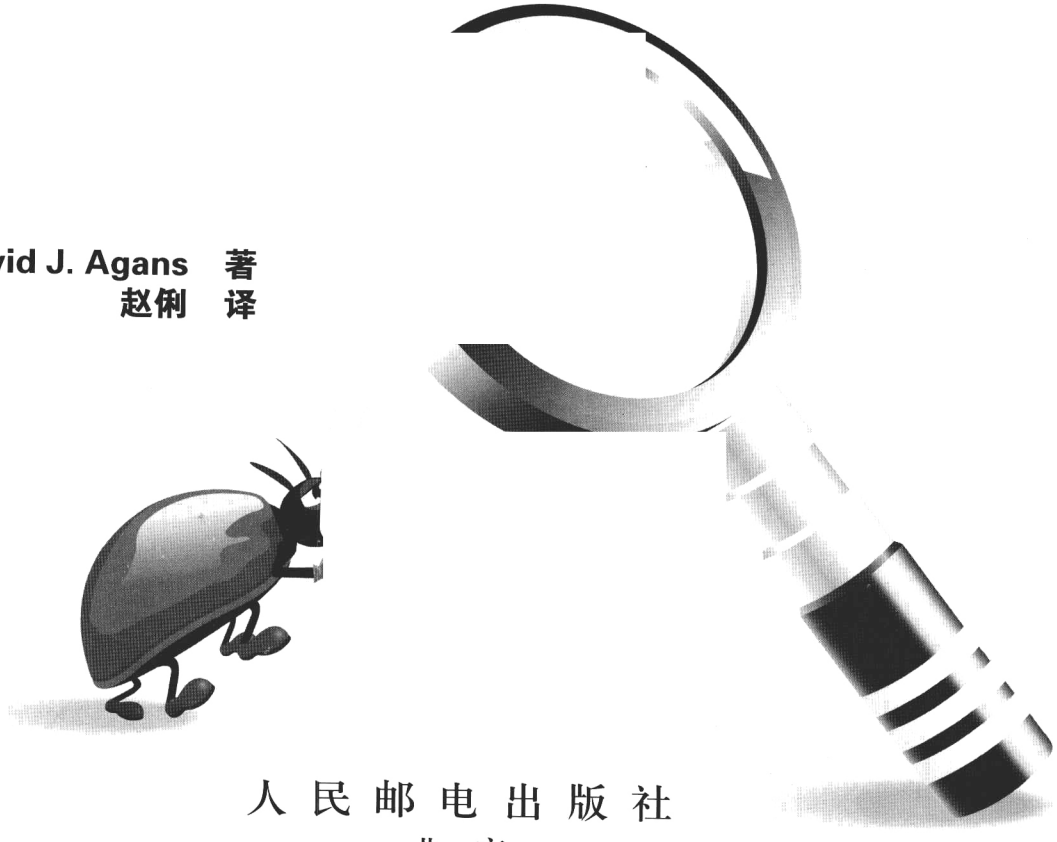
Debugging

The 9 Indispensable Rules for Finding Even the Most Elusive Software and Hardware Problems

调试九法

软硬件错误的排查之道

[美] David J. Agans 著
赵俐 译



人民邮电出版社
北京

图书在版编目（CIP）数据

调试九法 ：软硬件错误的排查之道 / （美）阿甘斯(Agans, D. J.）著 ；赵俐译. — 北京 ：人民邮电出版社，2011.1
（图灵程序设计丛书）
书名原文：Debugging:The 9 Indispensable Rules for Finding Even the Most Elusive Software and Hardware Problems
ISBN 978-7-115-24057-6

Ⅰ. ①调… Ⅱ. ①阿… ②赵… Ⅲ. ①电子计算机—调试 Ⅳ. ①TP306

中国版本图书馆CIP数据核字(2010)第209193号

内 容 提 要

本书主要介绍了调试方面的9条黄金法则，并结合实际的环境讲述了如何合理地运用它们。本书的内容没有针对任何平台、任何语言或者任何工具，讲述的重点是找到出错的原因并修复它们，高效地追踪和解决不易察觉的软硬件问题。

本书适合所有软硬件从业人员阅读。

图灵程序设计丛书

调试九法：软硬件错误的排查之道

◆ 著 [美] David J. Agans
译 赵 俐
责任编辑 傅志红
执行编辑 谢灵芝

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子函件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
北京艺辉印刷有限公司印刷

◆ 开本：800×1000 1/16
印张：9.75
字数：147千字 2011年1月第1版
印数：1-3 000册 2011年1月北京第1次印刷

著作权合同登记号 图字：01-2010-5547号

ISBN 978-7-115-24057-6

定价：35.00元

读者服务热线：(010)51095186 印装质量热线：(010)67129223
反盗版热线：(010)67171154

译 者 序

有人说调试是一门艺术，这不无道理，但本书作者认为它并不仅仅是艺术，更多的是科学，调试人员也不仅仅是艺术家，还是科学工作者。遵循本书所讲的9条规则，就可以把调试艺术转化为科学。

本书翻译到一半的时候，我已经钦佩不已。它绝对称得上调试领域的经典之作，但显然，在某种程度上它并没有引起国内业界的注意。常言道“千里马常有，而伯乐不常有”，虽然用千里马来形容一本书多少有些不恰当，但我确实觉得本书被埋没了，我想我们应该感谢人民邮电出版社图灵公司，把这样一本好书发掘出来，让国人有机会分享这位拥有二十多年实践经验的调试高手的知识和经验。

把书写厚了容易，写薄了却难，我想这一点大家都会认同。作者正是用这么薄薄的一本书讲述了适用于软件、硬件、工程领域的9条基本调试规则。这些规则甚至还适用于我们的日常生活，例如解决汽车和房屋问题。仔细揣摩，我们会学到不少生活知识，这也是阅读本书的一个额外的好处。

本书就像是一碗心灵鸡汤，也像是一坛陈年佳酿，书中所举的一些案例散发着古朴的气息。虽然我没有怀旧情节，但仍感到亲切而自然，有那么一刻，我与作者灵犀相通，仿佛他就站在那里，正在向我微笑，与我倾谈。

作者是个福尔摩斯迷，我想这是不是与他的职业生涯有关呢？在bug面前，他就是一名侦探。每章的开头都会引用福尔摩斯的一句名言，暗合该章的主题，也为本书平添了一层神

秘的色彩。我想要是把“神探狄仁杰”的故事讲给他听，他也一定会非常感兴趣，尽管他可能连狄仁杰是谁都不知道。

在本书翻译的过程中，有些地方我思索良久，有些则需要查询一些相关知识。我觉得提供一点背景资料会有助于理解，因此加了一些脚注，对于那些不甚了解相关背景或知识的人，可能会有一点帮助作用，但有些读者可能很熟悉硬件、软件和工程领域，如果这些内容都是你所熟知的，那么请恕我赘述之过。

最后，本书在翻译的过程中得到了人民邮电出版社图灵公司编辑们的精心指导和宝贵意见，帮我纠正了翻译中的很多错误，使我受益匪浅。由于水平有限，难免还会留有一些错误，恳请读者批评指正。

致　　谢

本书的孕育可追溯至1981年，当时Gould公司的一组测试工程师问我是否能用一篇文档写清楚如何解决硬件产品问题。我听了之后有点不知所措，因为我们的产品是一些由上百块芯片、几个微处理器和无数通信总线组成的主板。我深知没有“魔力配方”，他们必须学会如何调试。我与Mike Bromberg（他一直是我的良师益友）讨论了这件事，我们达成共识——最起码要编写一些通用的调试规则。于是我最后编写了“Ten Debugging Commandments”（调试十诫），这是一页简短的调试规则，测试小组很快把它贴到了墙上。几年后，这份清单已被压缩为一条规则，而且被推广应用到很多软件和系统，但它仍然是本书的核心。因此感谢Mike和那些提出这个请求的基层技术人员。

感谢Doug Currie、Scott Ross、Glen Dash、Dick Morley、Mike Greenberg、Cos Fricano、John Aylesworth（原来提出请求的技术人员之一）、Bob DeSimone和Warren Bayek，他们使得那些富有挑战性的工作变得充满乐趣。一直以来，我都很高兴能够为他们工作并与他们并肩作战，他们带给我无数灵感，帮助我培养了调试技巧，也教会了我幽默。还要感谢三位追求卓越并为我的学习过程带来乐趣的老师，他们是Nick Menutti（虽然这不是诺贝尔奖，但这里让我奉上对你的赞誉）、Ray Fields和Professor Francis F. Lee。还要感谢我一直未曾谋面的几位作者，他们的书对我的创作产生了巨大影响，他们是William Strunk Jr和E. B. White（著有*The Elements of Style*）、Jeff Herman和Deborah Adams（著有*Write the Perfect Book Proposal*）。

感谢包容我28年之久的夏日垒球球队Delt Dawgs，感谢朋友们的审阅和帮助。Charlie Seddon详细审阅了本书并给出很多有益的评论，Bob Siedensticker也审阅了本书并为我提供了案例、主题建议和出版建议，对此我感激不尽。还有几位当时我还不认识的朋友审阅了本书并给我寄来他们的意见，本书的出版离不开他们的帮助。他们是Warren Bayek和Charlie Seddon（上面提到过）、Dick Riley、Bob Oakes、Dave Miller和Terry Simkin教授，感谢他们付出的时间和提出的意见。

感谢Sesame工作室的Tom和Ray Magliozzi。（“侃车[①]”节目的主持人Click和Clack，抑或是Clack和Click[②]？）另外感谢Steve Martin允许我使用他们的故事和笑话，还要感谢Arthur Conan Doyle（柯南道尔爵士）创造了福尔摩斯这个侦探形象，并通过他表达了如此多的妙语，此外感谢Seymour Friedel、Bob McIlvaine和我的兄弟Tom Agans为我讲述妙趣横生的案例。发现和证明这些规则离不开他们提供的例子，感谢所有案例故事的参与者，感谢“英雄和笨蛋[③]”（你们知道我说的是谁）。

与Amacom的编辑们一起工作是一段美妙的经历，而且给了我很多启发。感谢Jacquie Flynn和Jim Bessent，感谢他们的热情和中肯的建议。感谢在出版过程中作出贡献的设计者和才华横溢的人们，正是由于他们的出色工作，才使本书成功出版。

特别感谢我的代理人Jodie Rhodes，感谢他给了我初次写书的机会，让我用别具一格的方式来写这个非同寻常的主题。他了解他的市场，事实也证明他是对的。

感谢我的亲人Dick和Joan Blagbrough给予的支持、鼓励和大量帮助。特别感谢我的女儿Jen和Liz，我要拥抱和亲吻她们，她们非常可爱，也感谢她们给予我的信任。（还要感谢她们每天晚上在用IM聊天和玩游戏的间隙把电脑让给我用。）

最后，我要把永恒的爱和感激献给我的妻子Gail，感谢她鼓励我把这些调试规则写成一

① 侃车，Car Talk，美国国家公众广播电台的一档节目。（如无特殊说明，本书中的脚注均为译者注。）

② Click和Clack，上述“侃车”节目二位主持人Tom和Ray Magliozzi的昵称，指车子运行时发出的咔咔声。

③《英雄和笨蛋》，*heroes and fools*，龙枪系列小说之一，作者在这里借用此名开了个玩笑。

本书，感谢她督促我寻找代理人，感谢她给予我创作的时间和空间。也感谢她校对了大量的初稿，要知道，没有她的校对，这些书稿我甚至不敢拿给别人看。她可以用一个吸尘器点亮吊灯[①]，却完全凭借她自己点亮了我的人生。

Dave Agans

2002年6月

① 第13章中的案例故事有具体的解释。

目　录

第 1 章　简介 …… 1

1.1　本书如何教会你调试 …… 1
1.2　这些规则都很显而易见 …… 2
1.3　本书适用于任何人 …… 3
1.4　本书可用于调试各种问题 …… 3
1.5　本书的主旨不在预防、保证或筛选 …… 4
1.6　调试不仅仅是故障检修 …… 5
1.7　有关案例故事 …… 6
1.8　精彩内容，即将上演 …… 6

第 2 章　总体规则 …… 8

第 3 章　理解系统 …… 10

3.1　阅读手册 …… 12
3.2　逐字逐句阅读整个手册 …… 13
3.3　知道什么是正常的 …… 15
3.4　知道工作流程 …… 16
3.5　了解你的工具 …… 17
3.6　查阅手册 …… 18
3.7　小结 …… 20

第 4 章　制造失败 …… 21

4.1　制造失败 …… 24
4.2　从头开始 …… 24
4.3　引发失败 …… 25
4.4　不要模拟失败 …… 25
4.5　如何处理间歇性 bug …… 27
4.6　如果做了所有尝试之后问题仍然间歇性发生 …… 29
4.6.1　仔细观察失败 …… 29
4.6.2　不要盲目相信统计数据 …… 30
4.6.3　是已修复 bug，还是仅仅由于运气好，它不再发生了 …… 31
4.7　“那不可能发生” …… 33
4.8　永远不要丢掉调试工具 …… 34
4.9　小结 …… 36

第 5 章　不要想，而要看 …… 37

5.1　观察失败 …… 41
5.2　查看细节 …… 43
5.3　问题忽隐忽现 …… 46
5.4　对系统进行插装 …… 46
5.4.1　设计插装工具 …… 46
5.4.2　过后构建插装 …… 48
5.4.3　不要害怕深入研究 …… 50
5.4.4　添加外部插装 …… 51
5.4.5　日常生活中的插装 …… 51
5.5　海森堡测不准原理 …… 52

5.6 猜测只是为了确定搜索的重点目标 53
5.7 小结 54

第 6 章 分而治之 55

6.1 缩小搜索范围 59
6.1.1 确定范围 60
6.1.2 你在哪一侧 61
6.2 插入易于识别的模式 62
6.3 从有问题的支路开始查找问题 63
6.4 修复已知 bug 64
6.5 首先消除噪声干扰 65
6.6 小结 66

第 7 章 一次只改一个地方 67

7.1 使用步枪，而不要用散弹枪 69
7.2 用双手抓住黄铜杆 71
7.3 一次只改变一个测试 72
7.4 与正常系统进行比较 73
7.5 自从上一次能够正常工作以来你更改了什么 74
7.6 小结 77

第 8 章 保持审计跟踪 78

8.1 记下你的每步操作、顺序和结果 80
8.2 魔鬼隐藏在细节中 81
8.3 关联 83
8.4 用于设计的审计跟踪在测试中也非常有用 84
8.5 好记性不如烂笔头 84
8.6 小结 85

第 9 章 检查插头 86

9.1 怀疑自己的假设 88
9.2 从头开始检查 89
9.3 对工具进行测试 90
9.4 小结 92

第 10 章 获得全新观点 93

10.1 寻求帮助 94
10.1.1 获得全新观点 94
10.1.2 询问专家 94
10.1.3 借鉴别人的经验 95
10.2 到哪里寻求帮助 96
10.3 放下面子 97
10.4 报告症状，而不是理论 98
10.5 小结 99

第 11 章 如果你不修复 bug，它将依然存在 101

11.1 检查问题确实已被修复 103
11.2 检查确实是修复措施解决了问题 103
11.3 bug 从来不会自己消失 104
11.4 从根本上解决问题 105
11.5 对过程进行修复 107
11.6 小结 107

第 12 章 通过一个案例讲述所有规则 109

第 13 章 牛刀小试 113

13.1 灯和吸尘器的故事 113
13.2 大量出现的 bug 115
13.3 宽松的限制 119
13.4 识破 bug 123

第 14 章 从帮助台得到的观点 128

14.1 帮助台的限制 130
14.2 规则，帮助台风格 130
14.2.1 理解系统 131
14.2.2 制造失败 132
14.2.3 不要想，而要看 132
14.2.4 分而治之 134
14.2.5 一次只改一个地方 134

14.2.6　保持审计跟踪 …… 135
14.2.7　检查插头 …… 136
14.2.8　获得全新观点 …… 136
14.2.9　如果你不修复 bug，它将依然存在 …… 137
14.3　小结 …… 137

第 15 章　结束语 …… 139

15.1　调试规则网站 …… 139
15.2　如果你是一名工程师 …… 139
15.3　如果你是一名经理 …… 140
15.4　如果你是一名教师 …… 141
15.5　小结 …… 141

第1章

简　介

“你知道，现阶段我非常忙，但我打算在晚年倾力写一本书，把所有侦探艺术都集中写到这本书里。”

——福尔摩斯，《格兰其庄园》

本书告诉你如何快速找到工作中的错误。它很短，也很有趣，因为它必须如此——如果你是一位工程师，你每天都在忙于调试，可能除了看点漫画之外就没时间读别的了。即使你不是工程师，也经常会遇到问题，这时你必须查明如何解决问题。

可能有人从来不需要做调试工作。或许你正忙着赶在公司倒闭之前把通过dot.com IPO[①]发行的股票卖出去，因而只是让你手下的人去查找问题。或许你总是很幸运，你的设计一直未发生问题，或者bug总是很容易找到（尽管这不太可能）。有可能在你和你的所有竞争对手的设计中都有一些很难查找的bug，谁能够最快地修复它们，谁就占据了优势。当你快速找到bug时，不仅能够更快地为客户提供更高质量的产品，而且也能够更早下班回家，与家人一起享受美好的时光。

因此，请把这本书放在你的床头柜上或洗手间里，两周后，你就会成为一位调试高手了。

1.1　本书如何教会你调试

为什么这样一本简短且易读的书会这么有用呢？根据我26年的系统设计和调试经验，我

① dot.com IPO，是指通过互联网公司上市募集资金。IPO，即首次公开募股（Initial Public Offering）。

发现了两件重要的事情（如果你把“从咖啡壶里倒出的第一杯咖啡含有全部的咖啡因”这样显而易见的错误也当成重要的问题，那么在你看来重要的事情就不止两件了）。

(1) 如果查找一个bug花费了大量时间，那么原因可能是忽略了某个最基本的、最重要的规则，一旦应用了那条规则，很快就会找到问题。

(2) 擅于快速调试的人已经深刻理解并应用了这些规则，而那些很难理解或使用这些规则的人则很难找到bug。

我把这些基本规则编写成一个清单，并教给其他工程师，我发现他们的调试技术和速度都提高了。这些规则的的确确起了作用。

1.2　这些规则都很显而易见

当你读到这些规则时，你可能会自言自语地说：“这些都是一些明显的规则啊。”不要着急下结论，这些规则确实都很明显（而且常常是基本的规则），但如何把它们应用于特定的问题就不总是那么显而易见了。而且不要把“显然”和“容易”混淆在一起，这些规则遵守起来并不总是那么容易，因此在解决实际问题时常常被忽略。

关键是记住并应用这些规则。如果这很明显而且容易，那么我就不必总是提醒工程师们应用这些规则，我也不必通过几十个案例故事[①]来说明不遵守这些规则将会发生什么情况。能够自如地运用这些规则的调试人员是凤毛麟角。我喜欢问求职者这样一个问题：“你调试时使用什么拇指规则[②]？”奇怪的是，很多人都回答说：“艺术。”好极了，那么让毕加索来调试我们的图像处理算法吧。事实上，利用简单和艺术的方法未必就能快速找到问题。

本书把这些“明显的”规则收集到一起，帮助你记住它们，知道它们的益处，并掌握如何运用它们，从而帮助你抵挡住“走捷径”的诱惑，因为捷径往往是陷阱。本书将把调试艺

① 案例故事，war story，原指战争故事，后泛指一些给人留下深刻印象的经历，在本书中则是作者举出的一些经典的、有代表性的案例。

② 拇指规则，英文为rule of thumb，又译为“大拇指规则”或“经验法则”，是一种可用于许多情况的简单的经验性的原则。

术转变为一门科学。

即使你已经是一位非常优秀的调试人员，这些规则仍然能够帮助你更上一层楼。当我把本书早期手稿拿给经验丰富的调试人员审阅时，他们不约而同地表示，本书除了教会一两个他们没有用过（但将来会用到）的规则之外，还帮助他们明确意识到了他们在不知不觉中遵守的规则。团队领导者（当然是顶尖的调试人员）指出，本书为团队提供了一种很好的沟通语言，使他们能够把技巧传授给其他成员。

1.3 本书适用于任何人

本书通篇都用“工程师”这个词来指代读者，但即使你不是工程师，这些规则对你也非常有用。当然，如果你正在查找设计中的错误，本书就更有用了，无论你是工程师、程序员、技师、客户支持代表，还是顾问。

如果你不直接参与调试工作，而是负责管理调试人员，那么也可以把这些规则传授给你的工作人员。你甚至不必理解他们所使用的系统和工具的细节，因为本书所讲的都是一些非常基本的规则，因此在读完本书后，即使你是一位“尖发经理[①]”，也能够帮助那些比你聪明得多的团队成员更快地找到问题。

如果你是一位教师，你的学生将会非常喜欢书中的案例故事，这些故事将为他们带来真实世界的体验。当他们走出校门时，将会比那些经验丰富（但没有受过调试培训）的竞争对手们更有优势。

1.4 本书可用于调试各种问题

本书具有很强的通用性，它并不是讲特殊的问题、工具、编程语言或特殊的机器。相反，本书讲的都是通用的技术，它们可以帮助你找到任何问题，无论你使用的是什么机器、语言

① 尖发经理（pointy-haired manager），指发型向上翘起，这是斯科特·亚当斯（Scott Adams）绘制的漫画Dilbert中的一个非常有趣的角色，他管理一家高科技公司的一个部门，但看上去对他下属所做的事情却毫不知情。

和工具。本书讲了一种查找问题的全新方法，例如，它不是告诉你如何在Glitch-O-Matic数字逻辑分析器上设置触发器，而是告诉你为什么必须使用分析器，即使把它挂接到系统需要费很大一番工夫。

本书也适用于修复各类问题。你的系统可能在设计、构建和使用中有错误，或者只是被破坏了，无论是什么情况，这些技术都将帮助你快速找到问题的核心。

本书介绍的方法甚至不仅限于工程领域，虽然它们都是从工程环境中总结出来的。这些方法也可以帮助你查找其他方面的问题，例如汽车、房屋、音响设备、管道，甚至是你的身体（本书中会有例子）。但不可否认的是，有些系统并不适合使用这些技术，例如经济学就不适用，因为它太复杂了。

1.5　本书的主旨不在预防、保证或筛选

虽然本书介绍的方法和系统都是通用的，但它们都紧紧围绕一个重点，那就是查找bug的根源并修复。

本书所讲的并不是像ISO-9000、代码评审或风险管理这样的质量改进过程，这些过程主要强调的都是防止bug的产生。如果你对这方面的内容感兴趣，我可以推荐几本好书，例如*The Tempura Method of Totalitarian Quality Management Processes*和*The Feng Shui Guide to Vermin-Free Homes*。质量保证过程所涉及的技术都很有价值，但它们往往不易实现，即使实现了，系统中仍然会留有一些bug。

一旦有了bug，就必须要检测它们，这项任务一般由质量保证（QA）部门来完成，如果没有这个部门，那么就只能由客户方来做了。本书也不会讨论这个阶段，因为已经有很多资源详细讨论了测试覆盖分析、测试自动化和其他质量保证技术。当你在产品线上检查6 467 826种选项组合时，可以找本这方面的经典书来打发时间，例如*How Do I Test Thee, Let Me Count the Ways*。

迟早会有一种组合失败，这时某位质量保证人员或客户就会起草一份bug报告。接下来，一些经理、工程师、销售人员和客户支持人员可能会召开一次“bug筛选会议（triage

meeting)”，激烈地讨论这个bug的重要性，以及是否需要修复它（何时修复）。虽然这个主题与你的市场、产品和资源密切相关，但本书绝对不会去触及它。但是，当人们决定修复bug时，肯定会看一下bug报告并且想弄清楚 “这究竟是怎么发生的”，这时就到了阅读本书的时候了（参见图1-1）。

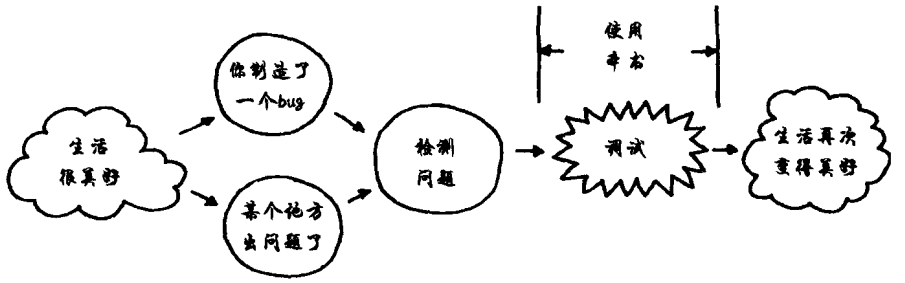


图1-1 何时使用本书

下面的章节将教给你如何准备查找bug，如何挖掘并仔细审查各种线索，以便找到根源，追踪实际问题，并修复它，然后确认你已经修复问题，这样你就可以高高兴兴地回家了。

1.6 调试不仅仅是故障检修

虽然调试和故障检修（troubleshooting）这两个词常常混用，但它们实际上还是有区别的，而且本书作为一本调试书，与其他数以百计的故障检修指南也是存在区别的。调试通常是查明为什么一个设计没有按计划工作，而故障检修通常是在已知设计没有问题的情况下，查明一件产品出了什么问题——可能是某个文件被删除了，某条线断了或者是某个元件出了问题。软件工程师做调试工作，汽车机修工做故障检修工作，而汽车设计师做调试工作（在理想世界中）。医生给病人看病就相当于“故障检修”，医生永远没有调试的机会了。（因为上帝已经完成了调试的任务，他花了一天时间设计了人类并造出原型，又在同一天把他的产品投放到人间，看起来上帝好像很赶时间！我想我们可以原谅他优先处理重要的方面，而给我们留下了像“拇指外翻[①]”和“男性规律性脱发”这样的小bug。）

① 医学术语，拇指由于囊肿胀而外翻，也称为拇囊炎。

本书中所讲的技术既适用于调试，也适用于故障检修。这些技术并不关心问题是怎么产生的，而是告诉你如何找到它。因此，无论是设计出了问题，还是部件有了毛病，都可以利用这些技术来查找。相反，有关故障检修的书只是在部件有问题的情况才有用。它们用几十张表格列出某个系统可能出现的一切症状、问题和修复方法。这些都很有用，它们是该类型的系统过去曾经出现过的一切问题的汇总，并且说明了症状和修复方法。它们把很多人积累的经验提供给故障检修人员，并帮助快速找到已知的问题。但是，当出现了新的、未知的问题时，它们就没有多大作用了。因此，这些书对设计问题几乎没什么作用，因为工程师们都非常有创造力，他们“喜欢”制造新的bug，而不会再用那些旧的bug来考验你。

因此，如果你正在检修一个标准系统的故障，那么不要忽略了规则8，查询一下故障检修指南，看看其中是否已经列出了你的问题。但如果它没有列出来，或者给出的修复方法不起作用，又或者你正在调试世界上第一个数字化的传输系统，因此根本没有故障检修指南，你不必担心，因为本书中所讲的规则将帮你找到新问题的核心。

1.7　有关案例故事

我出生于1954年，是一名美国电子工程师。在问题的讨论中，我所讲述的“案例故事”都是真实的，这些故事是我那个时代的人所熟知的。有些故事可能是你不了解的，因此有些我提到的事情你可能不理解。如果你是一位汽车机修师，可能不知道中断（interrupt）是什么。如果你生于1985年，你可能不知道什么是电唱机。但这没关系，重要的是我要通过这些故事说明的原则，而且我在讨论的过程中会给出足够多的解释，确保让你能够理解这些原则。

你知道，有些细节被我改动了，以便保护个人隐私，特别是保护那些犯了错的人。

1.8　精彩内容，即将上演

本书将介绍9条调试的黄金规则，每章介绍一条。每章的开头将讲述一个案例故事，通过它来说明规则对成功的重要性。然后描述规则并证明它如何应用于前面的故事。我会讨论