

<<完美软件>>

图书基本信息

书名：<<完美软件>>

13位ISBN编号：9787121099311

10位ISBN编号：7121099314

出版时间：2009-12

出版时间：电子工业出版社

作者：Gerald M.Weinberg

页数：345

译者：宋锐

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>

## &lt;&lt;完美软件&gt;&gt;

## 前言

软件测试的目的是什么？

我想大多数人对于这个问题的回答会是“保证交付高质量的软件”。

也许从最终目的上来说是这样的，但是单纯只是软件测试本身是无法为软件的质量提供保证的。软件的质量是从需求分析甚至是刚刚产生软件相关概念的时候就开始产生，受到整个开发过程影响的一个性质，而测试只是用来将这一性质数值化、表面化的过程。

是不是拥有良好的测试过程就能够保证交付高质量的软件呢？

显然是不够的。

要保证软件具有较高的品质，需要管理、开发、测试等多个部门的共同努力，尤其是管理部门如何看待测试，如何利用测试获得的信息是至关重要的。

当我们按照一个预定的过程对软件进行测试，却未能获得预期的结果时，往往会说：“这次测试失败了。

”或者是“这个测试没有通过。

”而实际上正确的说法应该是：“软件在这个测试中失败了。

”或者是“软件没能通过这个测试。

”在许多人的看来，这种说法上的差异也许是微不足道的，或者只是文字游戏。

但是事实上，它反映了大家对于软件测试的心理误区。

人们往往并不能很清楚地区分测试与除错、开发过程甚至是软件本身的关系。

而这一误区正是导致对测试产生不切实际的期望，夸大或者忽视测试的作用的根源。

如果你从事测试已经有很长时间了，就很可能听到过这样一些问题：“你为什么没有发现那个问题？

”“我们为什么要在可以自动化测试的情况下雇佣人来进行测试？

”“一定要让它在下周可用。

”而产生这些问题的原因，往往就在于人们对软件测试的误解。

本书的主要目的，就是要破除这样的一些误解，还软件测试以本来面目。

这本书的预期读者非常广泛，涉及与软件开发有关的所有人，包括但不限于软件开发人员、测试人员、客户、项目经理以及高层管理人员，等等。

本书的作者Gerald M. Weinberg在软件行业浸淫了五十余年，几乎开发过所有类型的软件，出版了四十余本技术书籍，发表过数百篇技术文章。

他对软件项目的管理、设计、开发和测试具有极其丰富的经验，对于与软件开发有关人员的心理尤其有深入的研究。

他的《The Psychology of Computer Programming》（中译本名为《程序开发心理学》）开创了“以人为本”的研究方法，它以其对程序员们的智力、技巧，团队和问题求解能力等方面独特的视角和敏锐的观察经受住了时间的考验，具有深远影响。

而在《完美软件——对软件测试的各种幻想》这本书中，Weinberg采用的是纯散文和故事化的风格，而没有使用花哨的指标和图表。

他深入浅出地讲述了与软件测试有关的各种误解与错觉，可以为行政人员、经理或者开发人员提供有关测试的足够信息，以便他们理解测试所要面对的挑战，进而对测试设置恰当的期望并就这些期望进行清晰的交流。

本书回答的问题包括：为什么在看起来测试只会耽搁时间的时候还要进行测试？

为什么无法一开始就构建正确的软件，从而不需要测试？

需要对所有可能性都进行测试吗？

为什么不对所有可能性都进行测试？

是什么原因导致测试如此困难？

为什么测试需要这么长的时间？

是否有可能构建完美的软件？

## &lt;&lt;完美软件&gt;&gt;

为什么我们不能接受一些缺陷？

本书的入手点更多的是出于心理学而不是计算机技术。

通过对心理学的研究，我们会发现人类本身就是不完美的，自然也不应该期望软件是完美的。

由于除了某些最简单的情况以外，软件的可能路径实际上都是无限的，因此对软件进行穷举测试通常是不可能的。

由于人类容易出错，而且程序中产生缺陷的可能途径也非常多，所以没有任何缺陷的（完美的）程序是一个无法实现的目标。

虽然这不是什么好消息，但是事实就是如此。

受乐观主义和必要性的推动，人们每年仍然会创造数百万行代码，而这些程序中的相当一部分将会被发布到客户群中。

因此，谨慎的开发人员别无选择，只能通过建立测试过程来发现软件中最重要的和最有可能出现的那些缺陷。

正如Weinberg 指出的那样，有些人确实选择只进行很少的测试，有时是依赖于诸如“如果我们运气好的话，就没有人会遇到那个缺陷”的想法。

其他一些人只是将该缺陷重新定义成一个特性就继续别的工作。

这些做法既不合乎道德，也可能导致法律问题。

Weinberg以非常诙谐而含蓄的风格介绍了他遇到过的众多例子，揭示了与软件测试有关的众多常见误解与谬论，还揭露了在测试中常见的各种欺诈方式。

通过阅读本书，应该认识到软件测试是一个与人的心理活动密切相关的过程；测试工作的核心是收集信息，这些信息是关于软件产品、开发过程以及测试过程本身的；要达到“保证软件质量的目的”，不仅仅要进行良好的测试，更重要的是要对测试获得的信息进行合理的利用。

本书是给涉及软件开发的所有人的一本指南，告诉他们：对于软件而言，完美是一个不现实的目标。

因此，本书对软件开发团队中的所有人都是必备的。

## <<完美软件>>

### 内容概要

《完美软件：对软件测试的各种幻想（中英文对照）》是从事软件行业五十余年的Gerald M. Weinberg针对软件测试所写的新作。

他在软件项目的管理、设计、开发和测试方面都具有极其丰富的经验，对于与软件开发有关人员的心理尤其有深入的研究。

在《完美软件：对软件测试的各种幻想（中英文对照）》中，他重点讨论了与软件测试有关的各种心理问题及其表现与应对方法。

作者首先阐述软件测试之所以如此困难的原因——人的思维不是完美的，而软件测试的最终目的就是发现对改善软件产品和软件开发过程有益的信息，故软件测试是一个信息获取的过程。

接着，作者利用丰富的经历和大量的实例，展现了在软件测试中可能会出现各种与人的心理有关的现象、误区、欺诈，以及容易犯下的常见错误等等。

《完美软件：对软件测试的各种幻想（中英文对照）》的重点不是告诉大家要做什么或者说如何做，而更多的是让读者明白在与软件测试相关的活动中会出现某些特定现象的原因。

理解这些与人的心理有关的现象有助于与软件开发有关的所有人之间更好地就软件测试的目的和实现过程进行沟通，从而实现具有更高品质的软件。

## 作者简介

Gerald M.Weinberg, 软件领域最著名的专家之一, 美国计算机名人堂代表人物, Weinberg&Weinberg) 顾问公司的负责人。  
Weinberg精力旺盛、思想活跃, 从20世纪70年代开始, 他总共撰写了40多本书籍和数以百计的论文。在西方国家乃至全球, Weinberg拥有大量忠实的读者群, 他们甚至建有专门的组织和网站, 讨论和交流大师的重要思想。  
可以说, Weinberg近年来的每本新书都是在万众瞩目中推出的。

## 书籍目录

前言第1章 进行测试的原因 11.1 人类不是完美的思考者 31.2 我们要做出有关软件的决定 31.2.1 日记条目1 31.2.2 日记条目2 51.2.3 日记条目3 51.2.4 日记条目4 71.2.5 日记条目5 71.2.6 日记条目6 71.3 决定可能是有风险的 91.4 测试可以提供降低风险的信息 131.5 小结 171.6 常见错误 17第2章 测试无法做的事 212.1 信息不一定有助于降低风险 232.2 也许我们不会使用那些花钱得到的信息 252.3 决定是感性的而不是理性的 272.4 不良的测试也许比不测试更糟 292.5 产品可能尚未准备好接受测试 312.6 小结 332.7 常见错误 33第3章 不对所有可能性进行测试的原因 393.1 可能进行测试的数目是无限的 393.2 测试最多只是采样 433.3 信息的成本可能超过无知的成本 453.4 我们也许可以用较少的测试获取更多的信息 473.5 测试自助餐 473.6 小结 493.7 常见错误 49第4章 测试和除错的区别 531. 通过测试来发现 532. 查明问题 553. 定位 554. 确定重要性 575. 修改 576. 解决问题 577. 通过测试来学习 598. 任务切换 614.1 测试会随着机构的成长发生变化 614.2 以时间限制试探法作为管理法则, 但根据需要进行调整 654.3 小结 674.4 常见缺陷 67第5章 元测试 735.1 我们有说明书, 但是找不到了 755.2 我们的错误太多了, 导致缺陷数据库无法高效运转 755.3 我们没找到任何缺陷, 实际上我们并没有真正地找 775.4 我们修改记录让缺陷看起来没那么严重 775.5 这不是我的组件中的问题, 所以我不记录 795.6 我不知道在测试错误的应用程序 795.7 我们不测试最差的组件, 因为花得时间太长 815.8 我们发现了这么多缺陷, 不会还有更多的 815.9 我们的测试证明程序是正确的 835.10 我们运行了很多测试用例, 根本就看不过来 835.11 如果我们的软件在有三名用户时工作良好, 显然它在有一百名用户时也不会有问题 835.12 我们不希望测试人员知道我们将忽略他们提供的信息 855.13 我没有报告缺陷, 所以开发人员不会对我发脾气 875.14 我们不需要测试它, 因为开发人员非常有水平 875.15 接着说元信息 895.16 小结 895.17 常见错误 91第6章 信息免疫 956.1 我们在生存规则受到威胁的时候会感到害怕 976.2 我们压抑无法接受的事物 996.3 我们让不可接受的事物合理化 1016.4 我们将自己的负面品质投射给其他人 1056.5 我们转移指责从而免除自己的责任 1076.6 我们对自己的不足进行过度补偿 1116.7 我们在觉得失去控制时开始出现强迫 1116.8 小结 1136.9 常见错误 113第7章 如何应对防卫反应 1197.1 确定恐惧 1217.2 使用危机思维 1217.3 实践, 实践, 再实践 1237.4 对自己进行测试 1257.5 小结 1277.6 常见错误 127第8章 良好测试的要素 1298.1 永远无法确切地知道 1298.2 只能根据事实来评估良好性 1318.3 可能希望故意插入一些缺陷 1358.4 对良好性的估算总是统计性的 1358.5 可以对非差性进行估算 1378.6 小结 1398.7 常见错误 139第9章 有关测试的主要误区 1439.1 指责误区 1439.2 穷举测试误区 1459.3 “测试产生质量”误区 1479.4 分解误区 1499.5 合成误区 1519.6 “所有测试都相同”误区 1519.7 “随便哪个笨蛋都可以测试”误区 1559.8 小结 1579.9 常见错误 157第10章 测试不仅仅是敲击键盘 16110.1 毫无目的地敲击键盘是不是测试 16310.2 白手套测试 16510.3 狗食测试 16710.4 对测试人员也要进行测试 17110.5 可能在没有意识到的情况下进行测试 17310.6 演示不是测试 17310.7 小结 17510.8 常见错误 175第11章 信息摄取 18111.1 使用萨提亚交互模型来解析沟通 18111.1.1 摄取 18311.1.2 确定含义 18311.1.3 确定重要性 18511.1.4 做出反应 18511.2 人们听取信息时是有选择性的 18711.3 数据来源会影响到摄取 18711.4 时机也会导致差异 18911.5 人们会出现信息过载 19111.6 减少测试的数量也许可以传递更多的信息 19311.7 寻找测试之外的信息摄取 19311.8 不要混淆理解和摄取 19511.9 使用数据质疑来过滤理解 19711.10 小结 19711.11 常见错误 197第12章 确定含义 20112.1 案例1: 四个缺陷, 五种含义 20112.2 案例2: 四个缺陷, 七种含义 20512.3 案例3: 四个缺陷, 自行确定含义 20712.4 进行解释之前先弄清期望的是什么 20912.5 不知道期望时的做法 21112.6 使用已经获得的信息 21312.7 使用间接信息 21312.8 使用未获得的信息 21512.9 同样的话可能具有不同的含义 21712.10 “相同”可能并不一样 21712.11 某些时候不精确会更好 21912.12 小结 22112.13 常见错误 221第13章 确定重要性 22513.1 不同的人会给同样的信息赋予不同的重要性 22713.2 公共的重要性也许与对个人的不一样 22913.3 重要性依赖于上下文环境 23113.4 不能总是根据金钱来确定重要性 23313.5 不要采用过细的尺度 23713.6 首先解决重要问题 23713.7 听从自己的情绪反应 23913.8 小结 24313.9 常见错误 243第14章 做出反应 24714.1 是运气不好还是管理不善 24714.2 项目最后会赶进度的原因 24914.3 接近项目结束时应如何反应 25314.4 对测试所需时间的估算与现实差距很大的原因 25514.4.1 好天气估算 25514.4.2 不切实际的过程模型 25514.4.3 低质的过程数据 25714.4.4 没有过程数据 26114.5 确定是否已经错过了可以有所改变的时刻 26314.6 小结 26314.7 常见错误 265第15章 避免软件测试变得越发困难 26715.1 情况变得更糟的原因

## &lt;&lt;完美软件&gt;&gt;

26715.2 让系统尽可能小 26915.3 让“系统”模型是可扩展的 27115.4 增量构建有清晰接口的分立组件  
27315.5 减少进入产品的缺陷数目 27515.6 小结 27515.7 常见错误 275第16章 不使用机器进行测试 27916.1  
用机器进行测试总是不够的 27916.1.1 即时评审 28116.2 首先对最差的部分进行评审可以让人了解缺陷  
的严重性 29316.3 事实并不总是能令人信服 29516.4 测试人员是颇有价值的评审者 29516.5 小结 29716.6  
常见错误 297第17章 测试欺诈 30117.1 我们会卖给你一个神奇的工具 30117.2 我们的演示是欺诈 30317.3  
这么多的证明信表明它一定很好 30717.4 我们可以通过定价来欺诈 30717.5 我们的工具会读心术 30917.6  
我们保证你不用做任何事 31317.7 我们一起密谋 31317.8 避免欺诈的方法 31517.9 小结 31517.10 常见错误  
315第18章 忘却型欺诈 31918.1 推迟文档化造成的后果 31918.2 不明确的测试报告就像流沙一样 31918.3  
伪造的测试报告阻止了改进 32118.4 在别的地方进行报复 32318.5 早期的答案可能产生误导 32318.6 “量”  
不是“质”的同义词 32518.7 不要将非测试活动当做测试 32718.8 太整洁了，不可能是真的 32918.9 电  
子表格中的垃圾还是垃圾 33118.10 小结 33118.11 常见错误 331尾声 333尾注 335其他阅读材料 343

## &lt;&lt;完美软件&gt;&gt;

## 章节摘录

产品何时准备好接受测试了？

这个问题并不总是很容易回答，但是可以通过问一些更容易回答的问题来了解一些信息，帮助确定何时尚未准备好接受测试。

对于产品是否存在一个以上的问题.需要通过测试来帮助回答？

如果没有任何问题，就没有理由进行测试。

是否希望知道该问题的答案？

如果不想知道答案，就不要提出问题，也就是说，不要进行测试。

是否只是无聊地对测试结果感到好奇？

如果并不准备对测试结果进行估算或据此采取行动，就不要考虑测试。

如果只是为了满足无聊的好奇心而进行测试，代价就太高了。

不过从另一方面来说，积极的好奇心是有效测试的一个重要方面，因为你永远无法准确地知道自己应该寻找什么。

因此，如果准备花一些时间来进行探索，保持好奇心就有助于让测试物有所值，不过一定要对发现的情况追根究底。

是否能够和测试人员预先就如何算通过测试达成一致？

如果不能就怎样才能算通过测试、怎样不能算通过测试达成一致，那你想通过测试得到什么呢？

是否已经预先就怎样的测试是成功的、怎样又是不成功的达成了一致？

是否认可只要是能够提供新信息的测试，就至少可以被看作是部分成功的？

是否以为测试的结果会替你做出决定？

不可能从纯技术的角度做出商业决策。

当然应该使用测试得到的信息来支持商业决策，但是不要用测试代替商业决策过程。

例如，交付一个未能通过某些测试的系统可能是一个不错的商业决定。

与之相反，交付一个通过了所有测试的系统也可能是一个不良的商业决定。



#### 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>