

大话 软件测试

不扭曲·不变形·不晦涩·不忽悠
明晰软件测试·从哪里来·到哪里去

大鸟和小白为您洞见软件架构底层，
诠释软件测试的设计哲学。

欧立奇 何金池等编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

大话 软件测试

欧立奇 何金池 等 编著



电子工业出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

本书通过小白与大鸟的趣味情景对话形式，用多个小故事、案例、漫画来组织讲解软件测试的方方面面，包括测试需求、测试分类、测试计划以及测试管理等。

本书表现形式虽为“大话”，但内容结构实为严谨。在讲解软件测试的过程中，通过问询式结构，把学习门槛降低，让读者可以更加容易地理解测试的目的、策略、方法以及管理，澄清有关软件测试的常见误解，用一种不扭曲、不变形、不晦涩、不忽悠的表达方式表现测试的真谛，以达到不但授之以“鱼”，还授之以“渔”的目的，引导读者体会软件测试过程中蕴藏的大智慧。

本书适合软件测试、软件开发和软件管理人员以及其他计算机爱好者阅读。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

大话软件测试 / 欧立奇等编著. —北京：电子工业出版社，2014.9
ISBN 978-7-121-24097-3

I. ①大... II. ①欧... III. ①软件—测试 IV. ①TP311.5

中国版本图书馆 CIP 数据核字（2014）第 187934 号



策划编辑：符隆美

责任编辑：徐津平

印 刷：北京季蜂印刷有限公司

装 订：北京季蜂印刷有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编：100036

开 本：720 × 1000 1/16 印张：17.5 字数：336 千字

版 次：2014 年 9 月第 1 版

印 次：2014 年 9 月第 1 次印刷

印 数：4500 册

定 价：45.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zltts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前言

本书通篇都是以情景对话的形式，用多个小故事或案例、漫画来组织讲解软件测试。从软件需求设计说起，在对软件测试做了妥善分类后，将本书的内容分为如下几大方面。

在测试需求方面：如何测试需求，如何审核需求，如何设计文档。

在测试分类方面：详解软件质量模型的6大特性27个子特性的各个检查点，并提供经验和案例，从而使读者能够容易地运用到实际项目环境中；让读者能够清楚地得知，软件测试究竟是测什么？

在测试计划方面：如何写用户故事、测试用例、测试计划，如何进行测试建模，如何制定人力资源的分配计划。

在测试管理方面：如何预知风险，如何写日报，如何与DEV（研发）、PM（项目经理）进行交流，如何进行测试项目的管理，如何利用自动化技术来提高测试的效率等。

本书表现形式虽为“大话”，但内容结构实为严谨。在讲解软件测试的过程中，通过问询式结构，把学习门槛降低，让读者可以更加容易地理解测试的目的、策略、方法以及管理，澄清有关软件测试的常见误解，用一种不扭曲、不变形、不晦涩、不忽悠的表达方式表现测试的真谛，以达到不但授之以“鱼”，还授之以“渔”的目的，引导读者体会软件测试过程中蕴藏的大智慧。

本书不同于其他软件测试书籍的主要特点如下。

细

软件测试由于工作的特殊性，软件测试人员更要具有认真、耐心、细致、敏感等个性元素，涉及的方面比较多，且比较基础，也比较细。以软件安装为例，一个完整的软件安装检查点就包括：1.安装环境检查；2.中断安装的情况；3.回溯的检查；4.可定制化安装；5.安装特定参数/端口依赖；6.安装路径；7.安装介质；

8.安装语言; 9.安装 Shell; 10.安装组件; 11.操作系统兼容性; 12.操作系统语言包; 13.硬件系统兼容性; 14.逻辑安装次序; 15.安装安全性; 16.安装接口; 17.安装结果检查, 等等。

所以本书把这些细小的知识点和检查点做了汇总, 让读者有章可循, 按图索骥就可以轻松测试。本书使用了四级标题, 每一条都是一个实际的案例, 切切实实地解决读者遇到的实际问题。

深

说到测试, 人们首先想到的是: 测试是一种技术。然而事实上, 测试是一种哲学, 一种思想, 思想的背后是一个人的眼界和世界观。一个测试也许能从多方面揭示测试者的素质和看待问题、思考问题的能力。市面上流行的测试书籍在此专业性的分类方面做得不够, 正因为如此, 本书追求的是循循善诱, 讲深讲透, 侧重于软件测试技术的本质理解, 而不仅限于对测试的单纯讲解。

广

市面上流行的软件测试书籍仅对软件功能性测试本身比较侧重, 而忽略软件测试外延的东西: 比如非功能性测试中的易用性测试、性能测试、文档测试。而随着用户对易用性、可靠性要求的与日俱增, 本书对以上诸多方面都给出了详细分析, 并结合大量案例制定出测试方案, 以满足读者需求。

乐

苦逼的团队做不出有爱的产品, 愉悦编程、快乐测试才能使你的职场之路长久。所以, 本书语言幽默诙谐, 并夹杂了一些漫画来摆事实、讲道理, 力求为众多严谨的软件测试书籍添加一抹亮色。

本书不是一本万能书籍, 但肯定是您软件测试/开发/管理工作的好助手、好伙伴。

本书主要由欧立奇、何金池编著, 其他参与编写的人员有刘洋、秦晓东、李启高、马雪、马煜、胥虎军、李富星、牛永洁等。

最后, 感谢本书编写过程中的几位重要人士的支持, 衷心感谢明总、峰总、大胖、小四的鼎力协助。

目录

第 1 部分 软件需求与设计	1
第 1 章 软件需求	2
1.1 从需求的含混性说到软件测试的目的	2
1.2 需求的定义与分类	5
第 2 章 PRD 审核	8
2.1 PRD 分类	8
2.2 软件产品定位	9
2.3 软件产品需求	14
2.4 审核软件产品需求	17
2.5 范围约束	26
第 3 章 用户故事	29
3.1 什么用户故事	29
3.2 用户故事特点	30
3.3 用户故事分解、细化、合并	32
第 4 章 审核 FS	35
4.1 实现的含混性	35
4.2 交付目标	36
4.3 范围约束	37
4.4 假设和依赖	37
4.5 功能描述	38

4.6 审核功能描述 38

4.7 非功能描述 41

第 2 部分 软件功能性测试 43

第 5 章 功能性测试的准确性和合适性 44

5.1 功能性测试概念 44

5.2 功能性测试分类 45

5.3 适合性测试 45

5.4 准确性测试 46

第 6 章 软件功能性测试用户故事 47

6.1 软件功能性测试故事表 47

6.2 执行者/行为/状态/预期结果/检查点 48

第 7 章 软件互操作性测试 72

7.1 互操作性测试定义 72

7.2 兼容性和互操作性的区别 73

7.3 不可互操作的原因 74

7.4 互操作性测试分类 75

第 8 章 安全性测试 81

8.1 软件安全性测试概念 81

8.2 软件安全性测试策略 82

8.3 用户认证安全 84

8.4 系统网络安全性测试 89

8.5 数据库安全性测试 94

第 9 章 全球化测试 99

9.1 全球化测试分类 99

9.2 日期 101

9.3 字符格式 103

9.4 数字格式 104

9.5 输入法编辑器测试 106

9.6 语言敏感信息测试 107

第 3 部分	软件非功能性测试	109
第 10 章	易用性测试	110
10.1	易用性测试分类	110
10.2	易理解性测试	111
10.3	易学习性测试	113
10.4	易操作性测试	114
10.5	UI 测试	121
第 11 章	可靠性测试	126
11.1	容错性测试	126
11.2	可恢复性测试	127
11.3	故障转移测试	130
第 12 章	可移植性测试	131
12.1	可移植性测试定义与分类	131
12.2	用户故事列表	131
12.3	行为/状态/预期结果/检查点	133
第 13 章	性能测试	143
13.1	功能与性能的区别	143
13.2	性能测试指标	144
13.3	获取性能需求	148
13.4	性能测试分类	149
13.5	如何进行性能测试	151
13.6	分析性能瓶颈	152
第 14 章	文档测试	156
14.1	文档测试重要性	156
14.2	文档种类	156
14.3	文档测试检查点	157
第 4 部分	软件项目流程与风险	159
第 15 章	软件项目开发流程	160
15.1	Project Milestone 定义	160
15.2	软件项目的主要阶段	161

15.3	研发周期制定	170
15.4	工作量估计	171
第 16 章	项目风险分析	177
16.1	风险、问题、缺陷的区别	177
16.2	风险分类	177
16.3	风险分析	183
16.4	风险缓解	184
16.5	常见的风险处理措施	186
第 5 部分	测试策略与测试计划	190
第 17 章	测试策略	191
17.1	测试策略的定义和分类	191
17.2	测试重点/测试优先级分析	192
17.3	各时间阶段对应的测试策略	193
17.4	多平台/操作系统/浏览器的测试策略	198
17.5	测试开始和结束的标准	198
17.6	测试环境策略	201
17.7	测试人员指定策略/任务分配策略	203
17.8	测试工具的使用策略	205
17.9	测试报告/进度策略	205
第 18 章	测试计划	206
18.1	测试计划的定义	206
18.2	测试计划的范围	207
18.3	测试用例设计方法	207
18.4	测试用例优先级划分	217
第 6 部分	软件测试管理	219
第 19 章	如何预防 Bug	220
19.1	Bug 和 Defect 的区别	220
19.2	预防 Bug 的意义	221
19.3	干净的代码	222
19.4	代码可复用	224

19.5	代码审核管理机制	227
19.6	做好单元测试	227
19.7	代码重构	228
第 20 章	如何 Log 高质量的 Bug	232
20.1	Bug 报告写给谁看	232
20.2	Bug 模板介绍	232
20.3	常见的 Bug 问题	246
20.4	如何分析 Root Cause	247
20.5	Bug 生命周期	251
20.6	测试报告分析	253
第 21 章	其他 QA 日常工作	255
21.1	日报	255
21.2	如何开会	258
后记	QA 的自我修养	261
第 1 课	QA 到底是做什么的?	261
第 2 课	质疑和思考	262
第 3 课	QA 要具备的技能	264
第 4 课	QA 和 DEV 的关系	265
第 5 课	QA 的主要贡献	266
第 6 课	自动化测试和常规测试的关系	267

第 1 部分 软件需求与设计

若要建一幢数百万元的房子，房主一定会与建房者详细讨论各种细节，他们都明白完工以后的修改会造成损失，都知道变更细节的危害性。软件开发也一样，软件项目中百分之四十至百分之六十的问题都是在需求分析阶段埋下的“祸根”，导致的后果便是开发者开发的软件与用户所想得到的软件存在着巨大期望差异。在大多数的软件系统中，最终用户可能都不清楚他的需求是什么，这是千真万确的。所以，在我们没有问清楚所有的需求之前，先别忙着进行设计和测试。

第 1 章 软件需求

测试的目的应该是验证需求，Bug（预期结果与实际结果之间的差别，即缺陷）是这个过程中的产品而非目标。测试人员应该像工兵一样，在大部队（客户）预期前进的方向上探雷、扫雷，而不需要去关心那些根本没有人会去碰的地雷。衡量一个测试人员应该去衡量他测试了多少需求（测试工作量）。许多测试团队在需求不清晰的情况下，就进行测试，这简直是事倍功半。往往到最后还要返工重来。所以对于测试工程师来说，测试需求是永远放在第一位的。

1.1 从需求的含混性说到软件测试的目的

1.1.1 需求的含混性

小白 大鸟啊，我郁闷得很啊。

大鸟 咋啦？

小白 我们公司参与了一个大项目。当时市场人员把项目签下来的时候，项目组是欢天喜地。项目做了一年多，到了交付的时候，用户却很不满意，当初说好的需求，好多都变了卦。

用户是上帝，最关键的是如果收不到后面的钱，那就算白干了。公司要求项目组加班加点进行修改。我们测试部门也不得不一遍一遍地做着重复劳动，搞得大家是怨声载道。做市场的和做开发的相互指责，然后，大家又一起骂客户刻薄。公司里面弥漫着灰心丧气的气氛。

大鸟 你们是怎么和客户谈需求的？

小白 市场人员先谈单，估计人家需要做什么。然后，我们这边派一个搞技术的人员过去了解需求，拿一些对方的表格和笔记回来。

大鸟 你们去询问需求，然后做了一个文档。你们头脑里的东西，跟客户要的东西，其实是不一样的。但是，大家都认为这样白纸黑字，基本上是一样的。但这里面其实是有差异的。这种差异，有时影响不大，但有时，是致命的。毕竟，文档不是最终的实物。客户永远认为，他是把需求给你讲清楚了了的，如果你做不到，

不是他的责任。而且，你要记住一点，用户只有在见到或使用过实物的时候，他才知道他其实要的是什么东西。否则，如果研发人员没搞清需求就闷头做事，迟早项目会变成“包工队的故事”。

场景漫画：包工队的故事

甲：说一个包工队接一好活儿，盖一个 70 多米高的大烟囱！

乙：还真不错！

甲：起早贪黑把活干完了，人家来一验收，死活不给工钱，还叮当五四被打了一顿！

乙：为啥啊，质量不行？

甲：把图纸拿倒了，人家让挖口井！

小白 “用户只有在见到或使用过实物的时候，他才知道他其实要的是什么东西。”这句话，我严重同意。可是，要按你的说法，那用户和我们之间，就永远不可能存在真正意义上的沟通！那不成了虚无主义了吗？”

大鸟 如果你不了解客户的业务需求，或者不熟悉其行业规则的时候，你需要更严谨的方式来询问和确定需求。否则，那些落在纸上的文字和文字之间，埋藏着数也数不完的陷阱。需求阶段过于匆忙，就会出问题。比如说，客户忙，随便给你找一个表格，到最后就是无尽的麻烦。

小白 客户不都是这样吗？有何问题呢？

大鸟 客户没有问题，而是去问的人要格外小心了。他要搞清楚客户的背景、立场——客户关注自己想要的结果，但不同人关注的东西不同。有些客户希望不要给他们增加过多的工作任务，越傻瓜越省事越好；有些用户关心新系统的性能是否可靠，速度够不够快；而系统管理员，关心的技术和安全。总之，每个人都各怀心事。需求是多种多样的，所以对于用户所提出来的需求都要仔细记录，认真掂量。

小白 如何掂量？

大鸟 多问自己几个问题。

(1) 这是谁的问题？谁是顾客？谁必须被取悦？（Who 的问题）

(2) 他们的问题是什么？来自于哪些方面？基于什么？有什么特殊性？



(What 和 Why 的问题)

(3) 哪些问题是一定要解决的? 哪些可以不解决? 哪些用户根本不在乎?

(Scope 的问题)

(4) 哪些问题是比较简单的? 哪些是比较复杂的? 简单的要不要合并? 复杂的如何分解? (拆分和合并的问题)

很多人去做需求, 以为拿着本子记下来就好了。其实, 很多时候, 都没那么简单。觉得简单, 是因为很多人认为从技术上设计这类的软件, 简直是小菜一碟。但是, 他没有想到的是, 很多问题只是冰山一角, 有些问题暗流汹涌, 有些问题前后矛盾, 有些问题不可实现。用户需求暗藏着一个个陷阱和地雷。

场景: 两个圆的故事

甲用铅笔在纸上画了两个圈。问乙: “你说, 这是啥?”

乙说: “两个圆。”

甲摇摇头: “是两个鸡蛋。”

乙觉得甲很无聊, 不耐烦地说: “好吧, 就算两个鸡蛋。”

甲摇摇头: “不是, 这不是鸡蛋, 这是两个乒乓球。”

乙不知甲在暗示什么, 他在听。

甲说: “我看到的東西, 和你看到的東西, 不一樣。但是, 在紙上, 畫的是同樣的兩個圓。”

1.1.2 软件测试的目的

小白 你上面所说的这些东西, 不是市场人员或者需求分析人员应该考虑的问题吗? 和测试 Team 有什么关系呢?

大鸟 那我问你, 你现在做软件测试, 测试的目的是什么?

小白 让我想想。软件测试的目的是尽可能发现并改正被测试软件中的 Bug (缺陷), 提高软件的可靠性。

大鸟 这个定义听起来很正确, 但用它来指导测试会带来很多问题。比如有的公司, 用发现的 Bug 的数量来衡量测试人员的业绩, 其结果如何呢? 其一, 有一些不够敬业的测试人员会找出一些无关痛痒的 Bug 来充数, 结果许多时间会被浪费在这些无关痛痒的 Bug 上; 其二, 测试人员会花很大力气设计一些复杂的测试用力去发现一些迄今尚未发现的缺陷, 而不关心这些缺陷在实际用户的使用过程当中是否会发生, 从而浪费了大量的宝贵时间。究其根源, 就是因为对测试目的的这种错误理解造成的。为什么这么说呢? 因为软件里 Bug 的数量是无法估计的, 所以如果测试的目的是为了找 Bug, 那么测试工作将变成一项无法完成也无法衡

量进度而且部分无效的工作（因为有些 Bug 在实际的运行过程当中根本不会发生）。

小白 唔……那你看这样说对不对：测试的目的就是为了保证软件质量？

大鸟 这个定义也是看似正确，但实际上，混淆了测试和质量保证工作的边界。软件质量要素有很多，包括可理解性（Understandability）、简洁性（Conciseness）、可移植性（Portability）、一致性（Consistency）、可维护性（Maintainability）、可测量性（Testability）、易用性（Usability）、结构（Structure）、效率（Efficiency）、安全性（Security）等，所以，软件质量保证和测试的关注点其实是不同的。假如说，你保证了软件安全性的高质量，但是用户压根就不关心安全性，那你不是白测吗？

小白 那么测试的目的应该是什么呢？

大鸟 IEEE 对软件测试的定义是：“使用人工或自动手段来运行或测定某个系统的过程，其目的在于检验它是否满足规定的需求或是弄清预期结果与实际结果之间的差别。”所以，简言之，测试的目的应该是验证需求，Bug（预期结果与实际结果之间的差别，即缺陷）是这个过程中的产品而非目标。测试人员应该像工兵一样，在大部队（客户）预期前进的方向上探雷、扫雷，而不需要去关心那些根本没有人会去碰的地雷。衡量一个测试人员应该去衡量他/她测试了多少需求（测试工作量）。我看到许多测试团队在需求不清晰的情况下，就进行测试，这简直是事倍功半。往往到最后还要返工重来。所以对于我们 QA（测试工程师）来说，测试需求是永远放在第一位的。

小白 我一直以为程序完成后了，才开始软件测试。

大鸟 我一直反对软件测试仅存在于程序完成之后。如果只将程序设计阶段后的阶段称之为软件测试的话，需求阶段和设计阶段的缺陷产生的放大效应会加大。这非常不利于保证软件质量。需求缺陷、设计缺陷也是软件缺陷，记住“软件缺陷具有生育能力”。软件测试应该跨越整个软件开发流程。需求测试和设计测试也可以算作软件测试的一种。软件测试应该是一个泛型概念，涵盖整个软件生命周期，这样才能确保周期中的每个阶段经得起考验。

1.2 需求的定义与分类

小白 我明白了，正因为测试的目的是为了验证需求，所以测试部门就该和研发、市场、技术等支持部门通力合作，搞清楚需求到底是什么？大鸟，到底需求的定义是什么呢？

大鸟 我认为需求是这样的：需求是指明了必须实现什么的规格说明。它描述了系统的行为、特性或属性，以及在开发过程中对系统的约束。用户（提出）的需求是用户个人认为在对自身目标的解决方案中，自己不能或者不愿完成的部分。

因为不愿且不能，所以用户的目标可能是含混的、模糊的。好多产品只是有一个潜在的客户群，产品做出来后用户才有反馈。

小白 如何减少需求含混性？

大鸟 当我们发现存在比较大的歧义时，自己通过交流的方式来降低这种歧义。开始的时候，我们会有一个初步的文档。但是在合同签下来后，会有一个相对时间较长的需求的再确认过程，我们会和客户一起来走一个流程。然后，我们会把大家商业需求的结果，转换成最终的设计，用 PPT 把用户故事（包括操作界面和业务流程）都模拟出来，让客户有身临其境的感觉。在正式开工以前，给客户汇报。到此为止，我们并不做任何真正的编码工作。

小白 这样做，还是很花时间，效果如何呢？

大鸟 相对于搞错了需求，重新开发，这是最合算的了。很多人都不愿意这样做，最后人都跑光了。知道项目经理最痛苦的事情是什么吗？项目没完人跑光了（你知道项目经理最最痛苦的事情又是什么吗？人在，项目不让你做了！这种事情，也有不少啊）。

1.2.1 业务需求

小白 那么，如何获得、采集需求？

大鸟 搞清楚用户需求是一个长期不断地获取和澄清的过程。对项目不同阶段所要求的需求粒度（需求分解细化的程度）也各不相同。在项目初始的立项阶段，需求采集的主体是市场人员，市场人员从客户那里得到一些直接的或者潜在的需求，这些需求或为全新的，或为对现有产品的改进期望。市场人员得到此类需求后，及时做初步分析，并书面反馈给 PM（项目经理）。这时市场人员是需求的主要来源，但并不是唯一的来源，技术支持人员、中间商、潜在客户群都会做出贡献。我们把立项阶段得到的需求叫业务需求。

小白 业务需求？

大鸟 业务需求反映的是企业/组织对软件系统的高层次目标。业务需求确定后会得到一个叫做**业务愿景**的东西，其实就是项目成功后会是个什么样子，并在涉众范围内达成一致。而业务需求的确定对之后的用户需求和软件需求起了限定作用，业务需求就是需求过程的宪法，任何需求不得与之相违背。比如说：研发一台电视机，为用户提供看电视的服务。请注意，是服务，是以外部的眼光，全局性地

看这一需要实现的东西。

小白 高层次目标？

大鸟 对，就是高层次的。咱们中国有一个成语“纲举目张”，纲是鱼网上的总绳，比喻事物的主干部分；目是指网眼，比喻事物的从属部分。纲举目张是指抓住事物的关键，就可以带动其他环节。在这里面，业务需求就是那个纲，指用户需求的大方向、大框架，它的输出结果就是 MRD，即市场需求书（Market Requirement Document），或者有的公司称其为商业需求书。

1.2.2 用户需求

小白 懂了，那然后呢？

大鸟 然后我们要在业务需求的基础上进行用户访谈、调查，对用户使用的场景进行整理，从而建立用户角度的需求。用户需求必须能够体现软件系统将给用户带来的业务价值，或用户要求系统必须能完成的任务，即用户能使用系统来做什么。比如说，为了让用户能够看到电视节目，要做到：接受电视信号、显示到屏幕、可以换台等。这些内容都是为业务需求服务的，也是为了能够看到电视节目所必不可少的，即用户需求是业务需求的深入和细化。在这个阶段，我们需要将 MRD 中笼统的“概念化”需求转化为可实现的“专业化”细则，将 MRD 中的内容进行指标化和技术化。此外我们还要挖掘出其中的隐性需求，以便后续引导用户的兴趣和投资点。

此阶段的输出结果是 PRD，即产品需求文档（Product Requirement Document），着重在论述“要做什么”的问题。此文档重点体现产品的功能和各项指标的说明，分别是功能要求、开发要求、兼容性要求、性能要求、扩展要求、产品文档要求、产品外观要求、产品发布要求、产品支持和培训要求、产品其他要求等。其中，在功能要求当中，应该重视主要 Feature（特性）的重点描述。