

Broadview
www.broadview.com.cn

测试实践丛书

51 testing
软件测试网作品系列

软件测试精要

51Testing软件测试网 组编
董杰 编著



電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

每日及時觀看電子月刊書籍

就上溜客安全網 www.176ku.com

软件测试精要

专家推荐

作者以别具一格，另辟蹊径，跳出传统技术人员的思维框架，在高空中以另外一个视角审视测试，提出了许多独特的见解，定会让长期迷失在测试里的技术人员闻到外界飘来的清香，瞥见墙外探进的花枝，窥到世外飘忽的美景。对测试迷茫的人，不妨放松心情，泡上一杯清茶，躺于一张藤椅上，在阳光下打开此书慢慢酌之。

——具有8年测试经验的Nokia高级测试工程师 王东旭

作者简介

董杰，生于四川宜宾，进入软件测试行业多年，先后从事过功能测试、性能测试、压力测试、自动化测试、系统测试、安全测试、解决方案测试、测试需求调研分析、测试策略设计，参与过测试咨询服务。测试过的产品领域有电信类SDH ATM、Ethernet、IP、网络安全、GPRS、IT产品。工作过的企业有自研网络产品的民企，也有Motorola等外企。对测试的心得是：“像老板一样做测试，做测试就是做老板！”

上架建议：软件测试

网上订购：www.dearbook.com.cn
第二书店·第一服务



责任编辑：葛娜
责任美编：李玲



本书贴有激光防伪标志，凡没有防伪标志者，属盗版图书。

ISBN 978-7-121-08028-9



9 787121 080289 >

定价：39.00元

就上溜客安全網 www.176ku.com

测试实践丛书

51 testing

软件测试网作品系列

软件测试精要

51Testing软件测试网 组编 董杰 编著

电子工业出版社

Publishing House of Electronics Industry

每月及时观看电子月刊书籍

北京·BEIJING

就上溜客安全网 www.176ku.com

《51Testing 软件测试网作品系列》

编辑委员会名单

编委:

王威: 具有多年软件开发经验和软件测试工作经验, 51Testing 软件测试培训高级讲师

王琰: 具有丰富的通信终端产品的测试及管理工作经验, 51Testing 软件测试培训高级讲师

王海龙: Mercury 认证 CPC, 对自动化测试有深入的研究和丰富的实战经验, 51Testing 软件测试培训高级讲师

王静兰: ISO9000:2000 质量管理体系内部质量审核员, 具有丰富的软件测试工作经验, 51Testing 软件测试培训高级讲师

朴春龙: Mercury 认证 CPC, 自动化测试专家, 51Testing 软件测试培训高级讲师

吴晓红: 具有多年的软件测试工作经验, 对测试技术与测试流程及测试管理有丰富的经验和深刻的认识, 51Testing 软件测试培训高级讲师

宋锋: 具有多年软件开发和软件测试工作经验, 具有丰富的项目实战经验, 51Testing 软件测试培训高级讲师

陈霁: ISO 内审员, 积累了丰富的测试和管理工作经验, 51Testing 软件测试培训高级讲师

陈文广: 谙熟软件测试流程, 擅长自动化测试和性能测试, 51Testing 软件测试培训高级讲师

周峰: 信息产业部认证系统分析员, 51Testing 软件测试培训高级讲师

周春江: 具有多年通信协议和通信终端设备的测试工作经验, 51Testing 软件测试培训高级讲师

徐林林: 熟悉大型应用软件的开发和测试流程; 熟悉性能测试流程、方法和工具(如 LoadRunner 等), 51Testing 软件测试培训高级讲师

商莉: 多年从事软件开发、软件测试及质量保证方面的管理工作, 51Testing 软件测试培训高级讲师

编辑部成员: 张晓晓 蒋玉鹏

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

测试是一种思想生成活动，而不是计划实现活动。

测试中有科学也有艺术。

测试的流程和规范是科学，测试的思想和方法是艺术。

流程和规范搭起了骨架，思想和方法是肌肉和血液。

请耐心看完本书，书中定有满足你需要的内容！

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

前 言

目前中国市场上关于各种测试流程、测试规范、测试工具的相关书籍非常多，基本上都可以解答广大测试同行们在业务上的问题和困惑。可是关于测试同行们非技术外的困惑，却仍无法从现有书籍中得到满意的答案。就连一些工作了四五年的测试朋友也常问我：“做测试能有什么快乐，有什么激情？”非常多的测试朋友们缺乏做测试的激情，没有动力，充满了职业发展的困惑。同时我发现很多测试人包括我自己一直以来都非常缺乏关于测试意义的交流，大家的经验和心得很少在业内、公司间进行交换和分享。同行们对于测试人生的意义、未来的职业发展太缺乏交流和借鉴了。正由于看不到测试人生的意义，工作没有激情，自己得不到从事测试工作的快乐，加上对测试职业发展未来的迷茫，导致很多测试人员选择了离开这个行业。虽然人各有志，但我还是希望能有更多的测试战士们通过得到适时的激励和肯定，发现测试的人生意义，从测试工作中找到激情和价值，看到未来的希望，用一个健康的心态过着一个自己满意、幸福、有安全感的职业生涯。最好能够每天早上起床投入工作时，能怀着一颗今天又是一个充满创意、成就感的心，而不是一颗充满厌倦、工作重复单调的心，投入到测试工作中。

我希望读者们通过本书能多了解一些测试以外，但与大家的职业发展、生活快乐有关的信息和经验。在未来的职业发展和职业选择中少走些弯路，少一些迷茫，少一些浮躁，看到自己的价值所在。希望通过本书的一点点启发，跳出每天测试工作的细节，反思测试、工作和人生，最终找到自己工作和生活的平衡点、价值与快乐。

本书内容基础是以人的认识客观发展规律为主线来逐渐演进的，使得处于测试各阶段的读者都能在本书中找到对自己有价值的内容。

内容基本主线为：

第一步：端正和树立正确的“测试态度”，掌握“软技能”。

第二步：学习和掌握向高阶测试高手发展的技巧和思想，掌握“硬技能”。

• V •

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

目 录

第 1 章 测试的态度 1

在我们的生活和工作中，一个好的态度将是影响我们是否能够成功，是否能够取得进步的最重要因素。足球运动员有“足球的态度”，我们软件测试人员也应该有“测试的态度”，因此本书将“测试的态度”放在了第 1 章作为本书最重要的内容。先点燃读者心中积极的火焰，然后再带着良好的态度来吸收和了解其他软件测试的相关经验和观点。

1.1 精益求精	2
1.1.1 测试用例设计的精益求精	2
1.1.2 性能测试的精益求精	6
1.1.3 回归测试的精益求精	9
1.1.4 测试脚本开发的精益求精	10
1.1.5 测试工具开发的精益求精	11
1.2 总结与温习的重要性	12
1.3 学习和思考	17
1.4 谦虚好学	19
1.5 工欲善其事，必先利其器	21
1.6 人无远虑，必有近忧	24

第 2 章 测试策略的相关因素 27

本章通过融合中国古代经典的军事哲学思想，来帮助我们掌握取得测试成功的硬实力。软、硬实力皆有后，能让测试人员更有力地挥舞起遨游高空的翅膀。

2.1 测试资源与时间控制	28
---------------------	----

• VIII •

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

2.2	测试的知己知彼	31
2.3	测试效率的优化	35
2.4	测试中技术风险的控制	38
2.5	测试中的金矿	40
2.6	灵活机动的测试	43
第 3 章	自动化测试策略	46
<p>本章将帮助读者对自动化测试建立正确的认识，了解自动化测试实施的策略和实施过程，从中发现自动化测试并不只是进行自动化测试脚本的开发，同样也是一个完整的系统体系。</p>		
3.1	解析自动化测试的理解误区	47
3.2	自动化测试与手工测试的分工	49
3.3	自动化测试实施策略	52
3.4	自动化测试实施过程	58
3.4.1	制定自动化测试方案	58
3.4.2	自动化测试脚本开发	60
3.4.3	自动化测试脚本的执行	62
3.5	自动化测试开发脚本	63
3.5.1	什么是测试脚本语言	63
3.5.2	Tcl 的故事	67
3.5.3	Perl 的故事	70
3.5.4	Python 的故事	74
3.6	自动化测试工具介绍	76
3.6.1	WinRunner	77
3.6.2	Fanfare	79
第 4 章	性能测试与 Troubleshooting	84
<p>本章将向读者展示什么是性能测试，以及性能测试与压力测试之间的关系。性能测试与压力测试是测试工作中对产品系统内部整体了解要求最高的测试阶段，需要测试人员能对产品系统有更全面和深入的认识。</p>		

4.1 性能测试	85
4.1.1 什么是性能测试	85
4.1.2 性能调优	89
4.1.3 压力测试	91
4.2 Troubleshooting	93
4.3 优秀性能测试工具推荐	97
4.3.1 LoadRunner	97
4.3.2 SmartBits	100
4.3.3 IXIA	101
第5章 安全测试技术	103

本章将为读者奉献一个在互联网上广为流传的一个中国黑客高手的故事，来体会测试技术与黑客技术本质上的相似性。同时，本章还会告诉读者产品安全测试应该包含哪些内容，以及如何开展安全性测试。并推荐在一些领域较好的安全性测试工具，以供大家研究了解世界最新安全性测试技术的趋势。

5.1 终极测试人——黑客	104
5.1.1 黑客本质	104
5.1.2 知名的测试黑客的故事	104
5.1.3 安全测试应该包含什么	108
5.2 优秀安全性测试工具推荐	113
5.2.1 针对 Web 安全性的工具	113
5.2.2 针对设备的安全性工具	113
第6章 测试职业发展	118

本章将重点与测试朋友们在非测试技术领域进行一些经验的分享，希望能帮助大家解除心中的一些疑问。本章将从一个测试人的思考角度出发，将如何规划测试从业者的职业发展和职业选择作为非技术疑惑的解惑开始。

6.1 选择——听你的心	119
6.2 职业选择的象限	123
6.3 测试职业种类	125

· X ·

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

6.3.1	普通测试工程师	125
6.3.2	性能测试工程师	127
6.3.3	测试设计专家	128
6.3.4	自动化测试专家	130
6.3.5	测试工具开发专家	131
6.3.6	测试管理者	133
6.4	突破测试的选择	136
6.4.1	市场	137
6.4.2	项目管理	139
6.4.3	售前技术支持	141
6.4.4	售后技术支持	142
6.4.5	技术培训	143
6.4.6	转向开发	144
6.4.7	做生意	146
6.5	外企和民企的选择	147
6.6	大公司和小公司的选择	150
6.7	外包公司和自主产品公司的选择	152
第7章 测试组织架构与测试管理		154
<p>本章将对测试人员在测试团队中所处的价值和地位，通过与公司架构和军队组织架构进行类比，让测试人员直观地感觉到自己所处的位置和价值所在。同时，本章还会针对测试管理的现状和测试新人培训过程中容易疏忽的地方进行一定的经验分享。</p>		
7.1	测试部架构与公司架构的对照	155
7.1.1	功能测试工程师——销售部	155
7.1.2	测试工具开发和自动化测试开发——研发部	155
7.1.3	测试设计工程师——公司的战略规划和市场部	156
7.1.4	解决方案测试工程师和系统测试工程师——销售部中的高级销售人员	156
7.1.5	测试领导者——公司的管理部门	156
7.2	测试部架构与军队组织架构的对照	157

7.2.1	功能测试工程师——一线战斗的步兵	157
7.2.2	测试工具开发——军工厂给战士提供先进的武器	157
7.2.3	测试设计工程师——军队参谋和各级指挥官	158
7.2.4	解决方案测试工程师和系统测试工程师——军中的空军和海军	158
7.2.5	自动化测试脚本开发工程师——军中的炮兵	159
7.2.6	测试领导者——军中的军长和政委	160
7.3	测试团队管理	160
7.4	测试新人的培训建议	168
第8章 测试杂谈		171
<p>本章是把大多数测试人员所感兴趣的测试非技术话题集中起来，当读者在阅读前面几章内容感觉疲倦时，可以直接跳到本章来休息一下，换一下思路来品一些轻松的“测试咖啡”和“测试红牛饮料”。或者当读者在前几章依然没有找到自己心中期望解决的困惑时，可以到本章来试试，看是否能找到自己期望的答案。</p>		
8.1	测试是科学也是艺术	172
8.2	测试人员真正做什么	174
8.3	软件测试的前景	178
8.4	测试人员市场需求现状	180
8.5	测试中的困难	191
8.6	测试经验杂谈	193
8.7	测试工程师的基本素质	196
8.8	谁是测试团队中的核心技术人员	198
8.9	软件测试值得投资吗	201
8.10	软件测试的黄金阶段	203
8.11	软件测试验收基准	206
附录 A 测试职业经历随谈		208
后记		212

第 1 章 测试的态度



知名足球教练米卢曾说过“态度决定一切”。在我们的生活和工作中，一个好的态度将是影响我们是否能够成功，是否能够取得进步的最重要因素。足球运动员有“足球的态度”，我们软件测试人员也应该有“测试的态度”，因此本书将“测试的态度”放在了第 1 章作为本书最重要的内容。先点燃读者心中积极的火焰，然后再带着良好的态度来吸收和了解其他软件测试的相关经验和观点。

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

1.1 精益求精

精益求精不仅仅是一种做事的标准，更是一种做人的态度。无论是测试用例设计、性能测试、回归测试、测试脚本开发还是测试工具开发等任何一个测试的分工，都需要用精益求精的态度来提高每个测试环节的质量，并相应地提高产品的测试质量。



案例

A 是某著名软件公司的软件测试人员，工作的流程非常严谨而明晰，这自然也意味着重复劳动。但枯燥并没有淹没 A 的工作激情，发现一个 bug 会带来很大的成就感，特别是想到每天将会有几百万人通过使用没有这个 bug 的软件准确无误地达到他们的目的，A 就特别有成就感。虽然可能一整天都为一个小功能“循规蹈矩”地反复测试，但是这样的重复却被 A 当做一种重要的积累，在积累中不断地追求精益求精的完美。

正如新东方学校的徐小平在其著作《骑驴找马》中的一句话：“重复做汉堡，就是麦当劳；重复煮咖啡，就是星巴克；重复教托福，就是俞敏洪；重复做好事，就是活雷锋。”重复的过程也是追求精益求精的过程，每一次重复都需要比上一次做得更好、更精彩，你才能不断地进步，并做到极致的成功。

1.1.1 测试用例设计的精益求精

测试用例设计工作本身是一个很难直接定量的工作，也不可能像开发代码一样，只要编译通过可执行就认为完成了任务。那么在测试用例设计过程中，我们如何才能促进测试用例设计水平的提高呢？测试用例的设计和编写基本上只有依靠测试工程师自己精益求精的态度才能保证测试用例设计的质量。测试人员自身精益求精的态度，不但影响着测试用例的设计质量，而且直接影响着测试人员之间测试水平的高低。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

型集成模型 5 级) 都过了的, 这些测试用例也是经过了每一个需要评审的流程才正式进入测试用例库的。那为什么这些外国人写的用例方法如此简单? 因为测试流程和测试规范只关注测试用例的骨架是否完成, 而附着在骨架上的肌肉状况, 则很难由流程来规范和考评。这样有的老外就偷懒, 用一句简单的句子就描述完了一个测试方法, 给后来者带来了极大的痛苦。

据笔者所知, 后来在这两家欧美企业的中国工程师一改他们的老外同事留下的弊病, 在写测试方法时, 尽量做到非常详细的文字描述, 并尽可能地把所有的操作步骤和命令都补充在对应的测试步骤后面。使用这种精益求精的态度写出的测试用例, 完全可以让任何一位测试用例执行者只看一遍测试用例即可完成测试用例的执行。测试用例设计工程师通过自己多付出些时间来完善和丰富测试用例的描述, 不但大大提高了未来测试用例执行的效率, 而且为公司节省了时间和成本。

1.1.2 性能测试的精益求精

在进行性能测试时, 我们需要细致地关注每一个数据的变化, 不放弃任何一个怪异的数据变化是最基本的性能测试工作的态度要求。那么在性能测试中的精益求精可以体现在哪些地方呢? 一个性能测试活动本身大致需要经历如下 4 个阶段。

第 1 阶段: 选择可靠的性能测试工具。

第 2 阶段: 调试及稳定性能测试环境。

第 3 阶段: 正式的性能测试。

第 4 阶段: 统计性能测试结果, 输出性能测试报告。

在**选择可靠的性能测试工具阶段**, 如何做到精益求精? 可能有朋友看到这里会问: “选择工具还需要精益求精? 那么如何精益求精? ”。俗话说: “好的开始是成功的一半”。软件的性能测试非常依赖性能测试工具的长期高负荷运转的稳定性 and 测试数据统计的精确度, 对性能测试工具的选择决定了后续工作的成功与否和成本消耗的代价。就是这样一个对后续工作至关重要的步骤, 却在实际工

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

作中没有得到足够的重视，这个过程很有可能就只是某个性能测试工程师花两三个小时的时间到网上搜索几篇文章，按网上文章推荐的工具来初选，然后根据个人主观的判断就决定了未来所用的性能测试工具。结果，有可能这个性能测试工具在后续的调试稳定测试环境阶段，以及正式测试阶段和统计结果阶段会出现各种各样的奇怪问题，并导致性能测试团队不得不花费数倍的时间来解决这些性能测试工具的问题。

因此，在选择性能测试工具时，建议除了在网上搜索介绍资料外，最好能亲自把所选的几个性能工具进行本地对比测试。在同等环境下，对后续关注的测试数据指标先进行测试观察，然后再将这些性能测试工具的各项性能参数、长期稳定性等关键指标，形成一个表格交由整个测试团队来决策并最终选出未来正式使用的性能测试工具。虽然在工具性能对比测试时，会消耗掉测试人员的部分时间和公司人力成本，但是却能避免以后在错误的道路上越走越远，造成无谓的成本消耗越来越大。

测试团队一致选定了性能测试工具后，负责该工具操作的测试人员，需要继续发扬精益求精的工作态度，去全面、深入地了解 and 掌握该性能测试工具的各类使用方式。笔者曾见过某公司花费重金购买了一个世界顶级的性能测试工具，结果，几年来该公司的性能测试人员只会使用该性能测试工具的少数几个基本功能，其中的大部分功能从未应用过，大大浪费了公司的资产。为什么会出现测试人员对性能测试工具使用不充分的现象呢？原因可能有如下几种可能：由于人的天生惰性，在完成了最基本的性能测试需求后，就不再对性能测试工具的其他功能花时间来了解、操作、学习。当然也可能是由于该工具的学习和使用难度较大，测试工程师在无客观压力的情况下，面对困难退缩了，不愿意继续钻研学习该工具。

所以，我们需要性能测试工程师在性能测试工具上同样能发扬精益求精的精神，在使用性能测试工具时能精益求精地多钻研该工具的其他功能，全面深入地了解该工具的使用特点，最大化地发挥性能测试工具的作用，提高公司资产的利用率。

在**调试和稳定性能测试环境阶段**，性能测试工程师可以在只完成最基本的环境搭建并让大部分设备和软件正常运转起来后，就直接开始正式的性能测试。但

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

是，只是保证大部分设备和软件能够运转起来对于期望开展高质量的性能测试是远远不够的。只有保证整个性能测试环境能够长期稳定地工作，才能真正确保性能测试的效果和效率；否则会在后续统计测试结果阶段，付出很多时间和成本来分析测试结果中的“垃圾数据”。

一个稳定的性能测试环境是执行性能测试和准确统计性能测试结果的发动机。如果发动机不结实、不稳定，时而无动力，时而动力下降，那么驾驭这个测试环境进行性能测试的人将会非常痛苦。在笔者以前的性能测试经历中，就曾经出现过测试环境中的模拟器时而正常工作产生正常的的数据，时而停发数据，时而效率下降，其直接结果就是大大影响了正式性能测试的项目进度，很难得到准确的性能测试效果。

因此，一个追求精益求精的性能测试工程师，应该用尽一切方法，确保性能测试的环境能够非常稳定，仔细地调试性能测试环境中的每个模拟器。如果物理连线环境有问题或设备有缺陷，则一定要事先准备好备用方案，绕开这些问题，来保证性能测试环境的稳定。如果性能测试工程师觉得只是保证性能测试环境的长期稳定还不够体现其精益求精的精神，则可以努力将性能测试环境再改造成一个半自动化测试的环境。一个半自动化测试的性能测试环境将会大大帮助提高性能测试环境的使用和搭建的效率，同时也是性能测试工程师对工作精益求精追求精神的体现。

正式的性能测试阶段，通常是性能测试工程师在所有性能测试工作的各阶段中最轻松也最有空闲时间的阶段。大多数情况下，很多性能测试工程师就觉得该自己休息、喝咖啡、聊天了。请先别忙着完全放松下来，虽然前期的测试准备工作非常辛苦，现在难得有空休息了，是应该短暂休息一下。但是，是否我们还可以更好地利用好这段唯一的休息时间来做一些让我们的工作更精益求精的事呢？例如：为了以后分析、定位问题更快，完成环境参数配置更快，我们是否可以利用这段时间开发一些自动化配置环境参数和自动化分析定位的小工具，每当遇到麻烦时，就可以大大提高解决麻烦的效率，为公司节约时间和人力成本。同时你还可以利用这段时间，多思考是否可以在现有的性能测试方案的基础上，针对性能测试方案再进行改进和优化，创造出更多新的性能测试方案，发现更多隐藏得

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

更深的 bug。

另外，你也可以利用这段较空闲的时间优化性能测试报告的内容，让其图文并茂，能更准确、简洁地展现性能测试的结果。因此，如何充分利用好这难得的大块空闲时间，取决于我们是否有着一颗精益求精的心。只要有一颗精益求精的心，在性能测试的执行阶段也能创造出更大的贡献和价值。

在**统计性能测试结果和输出性能测试报告阶段**，我们依然可以进行精益求精的改进。可以在查看性能测试仪器的统计数据结果时，仔细查看性能测试过程中的每一条 log 信息，从执行的 log 中不放过任何稍纵即逝的异常信息，毕竟每一个详细的 log 信息也是我们性能测试的劳动成果，很有必要充分利用起来。如果你现在还完全用手工和人眼来对大量的 log 信息进行处理，那么你很有必要通过编写自动化 log 分析工具来自动查找异常 log 信息的方式来大大减少工作量，提高 log 分析工作效率，而这也是体现精益求精的方式。

在性能测试报告中，你可以不只是简单地罗列本次测试的几个数据就算完成了任务，还可以加上丰富、详细的历次测试数据的趋势对照，并且图文并茂，以及除测试参数数据外的其他相关数据的变化。相信这样的性能测试报告应该会更让所有人满意，一位能做出这么完整的性能测试报告的性能测试工程师一定会因其精益求精的工作态度得到同事和领导更多的尊重和肯定。

1.1.3 回归测试的精益求精

前面两节谈了有关测试用例设计和性能测试中精益求精的方法，相信大家还是有所收获的。那么接下来我们应该如何进行回归测试的精益求精呢？笔者曾经做过手工回归测试，也做过自动化回归测试，从笔者自己的经验和身边看到的实例证明，在从事看似枯燥的回归测试工作时，我们同样可以调整自己，在回归测试工作中追求精益求精。

在手工回归测试时，我们除了需要按照测试用例的要求，保证每一步都是正常完成测试外，自己还可以利用空闲的时间和资源，尝试小范围地改变测试用例中的参数和方法，来进行一定的探索性测试。这样的行为有时能帮助我们得到意

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

外的惊喜，发现新 bug。毕竟创新是无止境的，没有哪个测试用例是完美的，按照测试用例执行回归测试可以保证 90% 的测试目标，但是靠自己想出的新点子找到的问题，却可以帮助测试团队和公司向最后 10% 的质量目标又前进一步。笔者相信，任何测试经理对于你在完成了本职工作后，还能发现新问题，都会感到非常高兴。能超出预期的员工肯定是一个追求卓越、追求精益求精的员工。

自动化回归测试阶段，与性能测试执行阶段类似，你将会有大量的空闲时间。在这段空闲时间中，最容易想到追求精益求精的方面是，自己编写一些自动化测试脚本结果的分析工具。当出现自动化测试脚本运行失败时，可以大大缩短分析、定位的时间，提高工作效率。同时，你还可以思考是否可以通过改变和优化脚本执行的顺序来大大降低自动化脚本运行失败的概率，从而达到缩短自动化测试脚本运行时间，提高运行效率的效果。

1.1.4 测试脚本开发的精益求精

精益求精的态度在测试脚本的开发中比较容易得到体现。例如：测试脚本优化；测试脚本的信息提示；测试脚本与测试工具的融合；测试脚本文档 4 个领域，都有着足够的空间让我们去发挥，去追求精益求精。

1. 测试脚本优化领域

我们可以不断优化测试脚本，使其能够适应在不同的测试环境中运行。也可以通过优化脚本的测试代码，使其能够有很强的容错性，在有一定干扰或异常的情况下依然成功运行下去，不对测试环境的“纯净性”要求过高。通过使测试脚本的代码松耦合，当被测设备的命令风格发生变化时，只需要对测试脚本做尽可能少的修改，就能适应新的命令风格。

2. 测试脚本的信息提示领域

当测试脚本在运行失败时，能打印出更多的错误信息和更详细的调试分析信息。在脚本运行失败后，通过报错信息直接告诉测试人员是脚本语法错误，还是测试环境错误，或者是某个正常的逻辑处理错误。这些提示信息可以让自动化回

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

归测试执行的工程师高效地分析、定位出脚本失败的原因，越是详细的调试分析信息，越能帮助测试工程师提高工作效率。

3. 测试脚本与测试工具的融合领域

一般情况下我们在进行手工测试时，常常需要借助一些第三方辅助小工具来完成测试。对于这种情况下的自动化测试脚本开发而言，要在测试脚本中完全模拟像手工测试一样的测试效果，脚本开发的难度是比较大的。不过，我们可以从如下几方面入手来实现目标。

(1) 多了解一些开源的小工具来代替目前辅助测试的小工具。通过开源的小工具，可以更容易地开发出适合自己调用的 API，便于集成到自己的测试脚本中。

(2) 在脚本与工具的接口代码处，优化脚本代码使其松耦合，当以后需要更换测试工具时，依然能很容易地驱动新的测试工具。

(3) 测试脚本能够对测试工具发生的异常和错误提供足够详细的 log 信息，便于以后因测试工具的错误而引起脚本运行失败时，可以快速地进行定位，节省分析、定位的时间和成本。

4. 测试脚本文档领域

文档的编写也许是大多数工程师都不愿意从事的工作，但是它却对公司非常重要，公司要实现铁打的营盘、流水的兵，就必须有非常详细、规范的各类文档。虽然，我们可以在测试脚本中通过注释的方式来解释各个测试步骤的脚本代码，但毕竟注释方式所覆盖的信息量太少。我们最好是针对每个测试脚本有一个独立的测试脚本开发文档，告诉后来者这个脚本要干什么；各个测试步骤是怎么实现的；关键参数是什么；用了哪些变量，每个变量的含义；与其他测试脚本的层次关系如何等。

1.1.5 测试工具开发的精益求精

在测试工具开发过程中，对于工具的开发者的不光要实现工具的功能要求，也

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

要像对待一个商业产品一样，对代码规范及工具的功能稳定性、易用性和相关文档做到精益求精。

一个测试工具的开发并不是一次性的。因此，我们需要像对产品开发要求一样，尽量有一个规范的代码结构，以便于后来者进行二次开发和维护。虽然我们很少会利用自己开发的测试工具进行精确的性能指标统计测试，但是我们却常常利用测试工具进行流量的模拟。为了保证测试环境流量大小的稳定性，我们也需要对自己开发的测试工具进行测试，优化测试工具的性能，使测试工具能够在实际应用中尽量状态稳定地工作。因此，我们可以在测试工具开发这一环节进行尽可能精益求精的发挥。往往一个测试工具的稳定性效果就是优秀测试工具开发者追求精益求精、追求卓越的结果，也是与其他普通测试工具开发工程师的最大区别。

在测试工具的易用性方面，通常由于测试工具开发项目时间紧，使得测试工具的使用方式在风格上可能过于偏“开发化”，也就是说，开发者自己可以很容易理解和使用测试工具，而测试工具的用户——测试工程师却比较难全面搞懂所开发的测试工具的使用方式。因此，建议测试工具的开发工程师应在稳定了测试工具后，再从测试工具易用性的角度，来不断完善、改进开发的工具。

在测试工具的相关文档方面，如果开发者自己不希望未来每一个工具的使用者都亲自来向他请教工具如何使用，或是未来进行二次开发时，自己都回忆不起来所有的代码细节，那么测试工具的开发还是最好能在工具的开发过程中，编写一份详细的开发文档，以及在完成测试工具的开发后，开发一个详细的使用手册。这样不但方便了未来的测试工具用户，也大大减轻了测试工具的开发未来维护和支持的工作量。

1.2 总结与温习的重要性

任何人类思维活动，无论是科学、考试、读书、工程、体育、艺术等，总结与温习都是帮助我们充分利用已有成功经验和失败经验，提升自我水平，最后帮

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

助我们在未来取得更大成功的好方法。我们从事任何软件测试活动，无论是测试策略设计、测试用例设计、回归测试、自动化测试开发、测试工具开发，还是测试项目管理等，经常总结与温习同样是非常重要的。总结与温习能帮助我们充分利用已有的经验，让我们的工作和方法更加精益求精，无论对个人技术水平的提升，还是测试项目的质量提高都是非常积极有效的。让我们聚焦到测试的几个主要环节，看看在各个环节应该重点做好哪些方面的总结与温习。

1. 测试策略设计领域

当每一个测试项目成功结束后，我们应该总结这个项目在选择测试弱点突破口时的成功经验、人员选择上的成功经验、优先级选择上的成功经验，以及合理的时间进度计划安排上的成功经验。而当一个测试项目没有达到预期目标时，则必须确定失败的因素有哪些？是项目实施过程中组织协调工作没做好；团队成员中有人影响了集体活动；还是在选择测试弱点时，方向错误导致整个测试无功而返，没有取得预期结果；或是在人员选择、优先级选择、测试工具选择上，进度安排计划存在失误，才导致的效率不高，未达到预期结果。

无论测试项目失败原因是什么，我们都需要找到失误点在哪里。而对于成功的测试项目，我们除了选出决定成功的因素外，还应该选出一两个方面是否在此次实施中，应该可以加强和提高的因素。通过总结与温习，做到胜不骄，败不馁。

2. 测试用例设计领域

当完成了测试用例的设计和 execution 工作后，就可以检验我们设计和评审工作的质量，以及测试用例的成功之处和不足之处了。通过使用这个测试用例发现的 bug 类型和 bug 数量来总结此测试用例设计的收获，并温习我们从这个测试用例设计中获得的经验和技能。通过分类分析找到的 bug 哪些是属于开发人员懒惰、粗心大意造成的；哪些是由于产品设计经验不足，或是差的产品设计/架构造成的；哪些是开发人员错误地理解了需求造成的；哪些是由于代码的复杂度过高造成的。

为什么要先对 bug 产生的原因进行分类？原因是我们可以从中检验出测试用例设计思路的强、弱。如果找到的 bug 大多数都只是粗心大意、懒惰的原因造成

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

的，而其他种类的 bug 发现较少，则说明我们的测试用例在设计的测试方法上过于简单，且对系统内部结构的理解和产品需求的认识还较浅。通过找到本次测试用例设计不足的方向，也就找到了下次进步的方向。其次，我们还需要温习和总结测试用例最重要的输出“bug 描述”是否足够好，我们是否在 bug 报告中用一个简短的篇幅做出了足够准确、详细的描述。比如：

- 报告是否过长或过短；
- 是否有一些不必要的步骤；
- 重现步骤是否精炼、最短；
- 给开发人员描述 bug 是否准确、简练；
- 是否在报告里写了一些不必要的信息。

最后，在测试用例设计方法中是否有些步骤是没有用处而可以省略的；我们是否正确理解了产品/软件规格说明书，以及测试用例和测试场景。

3. 回归测试领域

(1) 我们的测试脚本一次性通过率是否每次都在稳步提高？如果一次性通过率没有提高，是脚本代码不健壮，还是自动化测试工具有问题，或者是测试环境不稳定。无论短板出现在什么地方，我们都需要得到一个结论。如果是脚本代码不健壮，我们则应该针对脚本代码进行优化，优化逻辑判断条件，增强测试代码对异常的接受和处理能力等。如果是自动化测试工具的问题，就需要想办法找出是工具使用效率方面存在问题，还是工具本身有问题，是否需要升级或寻找其他辅助测试工具。如果测试环境不稳定，则需要我们对测试环境经常出问题的地方，采取一定措施确保测试环境的稳定。

(2) 对于脚本运行失败的脚本代码原因和环境原因，通过脚本报错的信息和形式进行统计分类，总结出每种脚本报错信息可能对应的失败因素，并以文档的形式保存下来，以便以后的工作能借鉴前人的劳动成果，提高定位问题的效率。

(3) 对于自动化测试脚本的分类，可按同类测试环境的脚本组合在一起执行，也可按同一优先级的脚本组合一起执行。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

(4) 针对自动化测试脚本执行的先后顺序，运行时间短的脚本可在白天工作时间运行，即使失败了也可快速定位；对于运行时间长的脚本则在晚上运行，以便节省白天可用于定位的时间和成本。

4. 自动化测试开发

在自动化测试开发领域，我们可以从提升脚本开发的效率，以及代码组织结构领域进行总结。提升脚本开发的效率，可以时常评估自己用于脚本开发的平台是否易用，是否扩展性好，是否辅助的测试工具易于集成到自动化测试平台中。如果以上几方面，我们取得了进步，则应该形成好的文档将这些成功之处的原因和工具的使用心得记录下来，以便后来者能够更好地工作；如果以上几方面，我们还存在不足，则应该想办法在薄弱环节进行相关提高。

在代码组织结构方面，如果开发的脚本具有了很强的健壮性，测试环境适应能力强，则可以对这些成功的经验进行总结，在整个脚本开发团队中进行推广。反之，脚本对测试环境的适应能力弱，运行通过率低，就需要大家一起来分析代码出错概率高的代码段，共同对代码结构进行优化。

5. 测试工具开发

目前，在国内测试工具的开发还不是很受大多数公司重视的现状下，一些中小规模的公司可能也就只有一名测试工程师在全职甚至是兼职编写测试小工具来模拟一些测试。通常在这种不利的情况下，测试工具可能会出现由于需求理解得不准确，开发投入资源少，导致测试工具的性能和稳定性不高，甚至有时出现功能实现上发生偏差。而在一些大公司，则由专门的测试工具开发团队全职来进行相关工作。虽然这时投入的资源比较多，但是有些测试工具开发团队却依然会出现开发需求产生偏差的情况，或者对测试工具本身缺乏足够的测试。

出现测试工具开发目标偏差，其主要原因还是大家在进行测试工具开发项目时，不像对待开发公司主营产品一样重视，并严格按照公司的开发规范流程化来工作。有时在测试工具开发的流程中，测试工程师在提出测试工具的功能描述文档时过于简单化，甚至随意化，再加上开发工程师对测试工具的测试对象理解可

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

能不是很深入、很精确，进而共同导致了工具开发工程师开发出的工具经常在功能上达不到预期的目标。

因此，建议大家根据自己所在公司的资源和流程情况，严格按照已有的开发规范和流程，把测试工具的开发项目当成公司主营产品的发展一样来重视。同时，测试工程师应对工具开发工程师进行专项技术辅导培训，让其尽可能多地了解测试工具所测试目标的功能细节。虽然项目开发启动的速度可能会慢一些，但是对于达到测试目标的最终质量还是非常有效的，而且有着规范的测试工具开发文档，也有利于将来测试工具代码的重用、扩展及后期维护。对于测试工具的开发，建议从如下几个方面进行总结：

- 测试工具的功能需求理解准确度；
- 测试工具的可自动化测试的集成扩展性；
- 测试工具的代码日后可维护性及重用性；
- 测试工具自身的质量稳定性。

提示

测试工具成功经验共享：

- 好的测试工具开发工程师对于准备开发的测试目标必定是通过一段时间专项培训和实际测试经验，充分正确理解后才进行开发的。
- 测试开发工程师与测试团队在一起工作，每天都可以针对开发过程中的问题，向测试团队（也就是未来工具的用户）进行请教，求证确认各种问题。
- 开发的测试工具最好是基于 Linux 平台的，并提供命令行接口。这样才能为未来集成到自动化测试中提供便利。
- 测试工具的开发代码用 CVS 之类的版本管理工具进行统一管理，任何变更都要经过工具开发团队和测试工程师团队共同的评审。
- 尽量多地从美国的互联网上寻找开源的测试工具，但不可迷信开源工具的质量。可通过优化修改开源工具，大大提高测试工具开发的效率和质量。

6. 测试项目管理领域

测试项目管理的经验和收获主要在项目计划前期阶段和项目执行过程中。每

个测试项目完成后，无论是手工测试、回归测试还是自动化测试，都可以对项目计划的制定、项目执行过程、项目结果报告进行总结和温习。

在项目的计划执行阶段，我们可以反省此次制定的项目计划是否对如下专题取得新的进步或者有什么不足：

- 对工作范围及要求是否有了准确的确定；
- 项目的后勤保障是否充分保证了项目的成功进行；
- 人力资源及日程安排是否合理；
- 验收标准是否科学。

在项目执行过程中可以对如下方面进行反省：

- 团队成员每日工作报告是否详细、准确；
- bug 定位和 bug 状态是否每日准确更新；
- 如果项目执行状况出现偏差，是否及时发现问题并做出相应的调整；
- 在各阶段的评审会是否效率高效，是否都提出了足够的改进意见。

在项目执行完成后进行如下的反省：

- 测试报告是否准确、清晰；
- 项目计划是否按期完成；
- 如果项目计划有延误，本质原因是什么；
- 是否每个 bug 的描述都简洁、准确；
- bug 的发现趋势，是否在逐渐收敛；
- 是否可以针对项目计划和执行过程提供新的改进建议。

1.3 学习和思考

对于任何一个从事 IT 行业的人来说，学习和思考将伴随着我们的职业生涯。聚焦到我们所有从事测试工作的人，无论是进行功能测试的工程师、进行自动化测试开发或测试工具开发的工程师，还是进行测试项目管理和人员管理的测试管

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

理者，在工作之余，不断地学习和思考无论对于现在的本职工作，还是未来往更高平台发展都是必需的。而学习和思考的范围除了目前所承担的职责外，也应该积极在未来有可能拓展的领域进行学习和思考。

每当我们完成一轮功能测试后，需要思考是否真正正确地理解了产品需求，是否正确理解了测试用例和测试场景，是否真正在测试时把自己当做最终用户，是否有更好的测试方法来发现问题。通过学习其他测试工程师发现 bug 的思路，经过自己独立的思考，将其测试思路应用于所负责的测试中。即便今天你只是一个初级的功能测试工程师，你也可以利用业务时间来学习自动化测试开发的相关知识，或者有空时自己写写小的测试工具，为将来有可能从事自动化测试或测试工具开发做一些储备。

对于很多普通的基层测试工程师，如果你有志走管理路线，则可以在当前被管理时，从自己的直接主管这里去体会他管理艺术的高低、管理过程中的优缺点。好的记下，仔细品味；不好的，也深刻反思不好在什么地方，假设自己处在这样一个管理的角色，会怎样去处理。你除了从自己的主管身上去学习管理技巧外，还可以通过思考寻找其他一些学习的途径。例如：从其他测试组的主管身上去学习，从开发组的一些主管身上去学习，甚至可以从电视上一些管理创业的节目中去学习都是不错的好方法。但是请注意学习本身并非目的，我们的目标应该是学会举一反三，灵活运用知识才是真正的目的。

测试人员如果能长期坚持广泛学习和独立思考，那么至少在测试领域会取得更高水平的提升。

提示

- 我们会比以前更了解软件的功能和内部运转机制，有利于构造出更多的测试方法；
- 我们能比以前设计出更多好的测试用例；
- 我们设计的测试用例能更精简且覆盖范围更广；
- 我们分析定位 bug 会更准确；
- 我们可以对产品的应用和设计有更全面、深入的了解，非常有利于将来发展成为产品经理或产品规划经理。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

因此，无论是某项具体的测试技术，还是管理艺术，我们唯有发挥主观能动性，进行积极、认真地思考，弄清知识的来龙去脉，以及知识的有机联系，才能有利于进行正确的思考。如果学到的东西不经过头脑加工，就好比吃下的食物未经口腔咀嚼、肠胃消化，即便是美味佳肴，也不会被身体吸收一样，非但无益，反而有害。法国作家伏尔泰对此有着十分精辟的论述：“书读得越多而不加思考，你就会觉得你知道得很多。而当你读书思考得越多时，你就会清楚地看到你知识得还很少。”所以，我们在通过阅读学习了大量技术类知识后，还需要通过电视、互联网扩展自己的技术视野和管理眼界，并进行深度思考。我们宁可拿四分精力来吸收，六分精力来思考，也不要拿九分精力来吸收，一分精力来思考。善于思考，就需要有蜜蜂酿蜜的精神，每一克甜美的蜂蜜不知凝聚了那个小生命的多少心血。思考需要我们下苦功夫，以“打破沙锅问到底”的探索精神去钻研，切不可不懂装懂，浅尝辄止。学习和思考是一对螺旋型交替上升的兄弟，在学习的过程中，需要经常停下来进行思考和回味，思考结束后再继续前行学习。学习→思考→学习，周而复始。

如果我们能通过测试工作这一人生阶段性的职业，掌握学习和思考的艺术，形成好的学习和思考习惯，那么无论将来我们是继续从事测试工作，还是转到开发领域，或者转到管理岗位，都会从在这一工作期间养成的学习和思考习惯中而受益匪浅。

最后，总结一下前面讨论的学习和思考的意义。学习是思考的基础，思考是学习的升华。在学习的基础上思考，思考才能深入；在思考的前提下学习，学习才有效果。同时对所学的知识必须结合实际反复运用，知识才能巩固，技能才可纯熟，这就是我们 IT 人在职场发展的必由之路，愿我们大家都能勤于学习，善于思考，不断在测试领域及其他领域扩充自己的竞争力，在未来获得更大的发展和成就。

1.4 谦虚好学

孔子曾教诲我们“三人行，必有我师”。可是要真正做到谦虚好学并不是一件容易的事，特别是对于很多 IT 精英们，或者是自我感觉是团队中牛人的高手，

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

让他们能真正地虚心向周围所有的人学习，特别是职务比他们低、工作年限比他们少的同事学习是非常难的！那么如何才能算是真谦虚？要知世间学问无量无边，唯有眼光放开，学问自然长进，方知自己所懂的实在少之又少。

常言道“任何人都可以成为你的老师”，笔者很认同这个观点。至少每个人的经历和世界都是不一样的，正如世上没有两片一样的叶子这个道理。因此，我们可以从每个人身上听到不同的经历、不同的故事、不同的失败、不同的成功。只要用心去思考，从这些不同中肯定都能得到新的收获为我所用。仔细观察身旁的人，你会发现，无论多么出色的人，也不可能拥有所有的优点；而看上去十分乏味的人，也必然会有长处。无论对于哪一类人，你只需学习他们的长处就可以了；而对于他们身上的不足之处，则可当做教训来警惕自己。

虽然从态度上我们需要向周围的所有人学习，不过总地来说，通过观察所在环境中出类拔萃的人，是最好的学习方法。毕竟在各行各业出类拔萃的人，必有过人的优点。在这样优秀的人物身边，就像站在一面明镜前，照见我们的缺失。此时，善学改过就是最快速的进步方法。

聚焦在我们的测试工作中，对于那些优秀的测试工程师，可以发现如下优点：

- 工作非常执着，对 bug 的定位不轻言放弃；
- 坚持真理，无畏开发工程师和管理层的怀疑；
- 能更灵敏地找到正确的 bug 定位方法；
- 具备非常深厚的产品常识，可深层次地思索 bug 的集结地；
- 爱好学习并广泛地了解业务相关知识；
- 谦虚好学。

从一些好的管理者身上则可以看到如下优点：

- 为人平和，对下属基本不会大吵大闹；
- 维护测试团队的利益和重要性；
- 合理的工作计划，不会让下属无缘无故地加班；
- 主动积极与下属沟通、谈心，关心下属的工作、生活感受；

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

- 对团队的每一个成员能不抛弃、不放弃；
- 为人善良、真诚；
- 尽可能地公平、公正；
- 具备开放的心，接受团队成员提出的新观点和新建议。

1.5 工欲善其事，必先利其器

无论对于从事工程技术类工作的人，还是从事艺术类工作的人，要做好工作，都要先使工具锋利才能事半功倍。正如：搞艺术雕刻的艺术家肯定需要对各种雕刻刀具的特性非常了解，并熟练使用才能让自己的创意和想法得到最好的发挥。“工欲善其事，必先利其器。居是邦也，事其大夫之贤者，友其士之仁者。”其含义不仅仅指要掌握好工具的使用，更表达了要做好一件事，好的准备工作是非常重要的前提。

几乎每一个从事软件测试的工作者，都会用到有关的测试工具。这些测试工具能帮助我们实现一些靠人力无法实现的工作，或者能大大减轻人力工作的强度和难度，并提高测试执行的效率，缩短测试执行的时间，保证项目的计划。因此，无论是功能测试工程师、回归测试工程师、自动化测试开发工程师、系统测试工程师，还是测试管理者都需要对自己所在领域的测试工具有足够多的了解，并在有条件的情况下尽可能地亲自去熟悉它，充分发挥测试工具的效用。

例如，从事数据通信产品测试的工程师，对于 Smartbits 和 Ixia 测试设备就必须非常熟悉，如果对这两个工具中的任意一个工具达到非常熟悉的程度，不但可以提高工作效率，更可以帮你构造出更多新的、高质量的测试方案来发现人工难以发现的问题。如果作为数据通信产品的测试工程师连 Smartbits 或 Ixia 一个都没用过，则基本上不可能称之为合格的测试工程师。测试工具对于测试工程师的作用就好比武器对士兵的重要性。功能测试工具就好比步兵手中的枪，不同的工具犹如不同的步枪，有的是冲锋枪，有一定的强度却没有精确度和远度；有的是狙击枪，非常精准，也有一定射程，但却没有强度。而性能测试工具就好比炮兵手中的火炮，火力大，精度高，武器就越好。自动化测试工具对于自动化测试

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

工程师就好比无人驾驶侦察机一样，可以大大节省人力投入，并自动取得准确的情报。

所以，各类测试工程师应利用一切可以利用的条件去熟悉相应领域的测试工具。功能测试工程师通过了解多种功能测试工具的特性，可以不断地创造出新的测试方法，构造出更多自己以前想都不敢想的测试手段。性能测试工程师通过体验不同的性能测试工具，来选出持续火力最猛、精确度最高的性能测试工具，以保证自己工作输出性能数据的正确性。自动化测试工程师则通过了解更多的自动化测试工具特性，将其中适合自己的工具集成到自动化测试方案中，可以大大提高产品的自动化测试率，也可以大大简化一些脚本的开发难度。

只是在强调测试人员必须熟悉和了解自己的测试工具的时候，我们也不可以疏忽了两件很重要的事：

(1) 尽信书不如无书——不能对测试工具过度依赖。

(2) 测试工具外的准备也是非常重要的。

虽然，笔者在前面的叙述中强调了很多测试工具对测试人员的重要性，也做了比较生动的比喻。也许这时就有朋友已马上打开 Google 或 Baidu 搜索测试工具的相关资料，以希望发现最适合自己的“AK47”。但是，笔者还是要做一个善意的提醒：“尽信书不如无书”。工具虽然能帮助我们大大提高战斗力，实现很多以前无法想象的目标。可是，毕竟工具是人创造的，任何工具都有其缺点或弱点。因此，对测试工具太过依赖或太过信任，将会对项目产生技术风险。

为什么会产生技术风险？

首先，任何人的判断都有其一定的客观或主观的偏差——我们做出的选择未必就是最好的选择方案。既然每个工具选择方案都有其不足之处，那么我们必须针对每个工具选择方案的弱点，进行人为的补缺才能保证测试工作的严谨。其次，人都是有惯性或惰性的，当我们已习惯了工具所带来的便利和轻松后，则很容易形成思维的惯性和行为的惯性。从此以后，很容易每次进行功能测试时都只用某个测试工具完成测试而不再进行新的探索性测试了；或者在进行性能测试时，每

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

次都用一组固定的测试配置数据，人为地减少了不同测试数据组合对产品测试的效果，隐藏了许多性能测试的 bug。

所以，我们在重视测试工具使用的同时，还必须克服和避免形成测试工具依赖性或因测试工具使用而形成的测试思维惯性。在充分使用测试工具的基础上，别忘了人的主观能动性的发挥，让测试工程师的大脑来驱动测试工具，而不是测试工具来麻痹人、奴役人。正如从古至今，绝大多数的战争取得胜利的的决定性因素是战略指挥家、战术指挥家、士兵的勇气和智慧。先进的武器能够影响战争的进程，但很难决定战争的命运和结果。因此，无论是战争还是测试工作，人的力量才是最重要和具有决定性的。

本节一开始就谈到“工欲善其事，必先利其器”。其含义不仅仅指我们要掌握好工具的使用，更包含：要做好一件事，准备工作是非常重要的。

对测试工具的了解和熟练使用，是每一个合格测试人员必须事先做好的准备工作。除了准备测试工具，测试人员还需要做好哪些测试前的准备才能保证做好测试工作呢？

**提示**

对于功能测试工程师：

- 充分了解测试功能的特性；
- 充分了解测试功能的实际应用环境和应用数据；
- 充分了解测试功能的系统架构及相关实现技术；
- 充分了解测试功能所在平台或系统的相关功能；
- 充分了解测试功能所在平台或系统的实现架构。

功能测试工程师只有至少将以上 5 个方面的知识准备就绪了，才能设计出好的测试方案，模拟最真实的用户使用场景，进行最深层次的产品实现原理的测试，以及最全面的测试覆盖率。

**提示**

对于自动化测试工程师：

- 充分了解进行自动化测试开发的脚本特性；
- 充分了解自动化测试运行平台的特性；

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

- 尽量了解进行自动化测试转换测试用例的真实含义；
- 多储备一些产品相关手工测试的经验。

自动化测试工程师至少需要掌握以上 4 个方面的知识和经验，才能高效地开发出高容错性、健壮的自动化测试脚本，并准确实现测试用例的真实思想。

提示 对于性能测试工程师：

- 准确了解用户应用环境的真实数据组合；
- 准确深入了解行业标准的数据指标；
- 产品性能测试功能的使用方式；
- 了解用户实际环境的“噪音”类型；
- 深入了解性能相关的软件实现原理。

唯有性能测试工程师具备了以上 5 个基本要求，才能避免构造出在理想环境下的性能测试，并真正在“知己知彼”的状态下，对产品的性能进行有用户真实环境“噪音”的性能测试。

只有当各类测试工程师完成了测试工具的准备工作，以及其他测试工作涉及的知识技能的准备工作后，我们才能真正算是“工欲善其事，必先利其器”。下面就可以开始打一场有准备的测试大仗了！

1.6 人无远虑，必有近忧

孔子有名句“人无远虑，必有近忧”。告诫我们要未雨绸缪，不要老看眼前的事物，而要为远景期望积极奋斗。几乎每一位有着 1 年以上软件测试工作经验的朋友都有这样的体会，当对自己所负责的那部分测试工作熟练后，后续工作相对就轻松了。这时无论是从事功能模块的手工测试，还是自动化脚本开发，人都很容易地形成惰性。有的工程师则会满足眼前的工作职责，每天原地踏步，习惯于干一天混一天。

作为 IT 行业的一员，职业忧患意识是大部分人都需要的。或许有人担心 35

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

岁以后找不到工作；或许有人担心自己目前所拥有的技能未来还能否有市场；或许有人担心，如果离开现在的公司还能找到相同的岗位或更好的岗位吗？所以，职业的忧患意识为我们 IT 人带来了较大的工作压力和生存压力，也正是这些压力促使我们不断学习，不断往前进，不敢放松自己。总的来说，在这种压力下，我们感受到了一种进步的感觉，一种不断进步所带来的安全感。因此，无论从事什么岗位的测试工程师都必须拥有“人无远虑，必有近忧”这样的一个心态，才能不断提升自己。对于仅仅从事过黑盒功能测试的工程师，要不断地多丰富自己的业务相关知识，或者牺牲一些个人休息时间学习一些自动化测试的技术和测试工具开发的相关知识，多多扩展自己在测试领域的技能。如果有心向软件开发方向发展也可以多花些时间和精力，利用空闲时间学习一些软件开发的技能。对于进入测试行业直接就从事自动化测试脚本开发或测试工具开发的朋友，为了写出好的测试脚本、测试工具，应该寻找一些机会到手工测试一线去体会具体实战测试的感受。这样不但可以直接促进脚本开发或工具开发质量的提升，也有利于自己有目的地优化开发工具的选择和对测试的全面了解。

对于少数在国际大公司从事软件测试工作的朋友们，你们是幸运的。因为，可以有机会在一个大平台上体验到欧美全面、先进的测试规范和测试流程，以及自动化测试平台和自动化测试的理念。可是对于许多刚毕业就进入国内中小企业的测试同行，要在一个较高的层次提高自己对软件测试的理念、自动化测试的理念和学习到规范的软件测试流程，将会困难重重。由于国内中小企业开始启动软件测试的时间较晚，投入的资源少，以及公司重视程度的相关因素，在这些企业工作的大部分软件测试工程师很可能会长期停留在较初级的软件测试理念状态中，甚至有可能是错误的理念中。这些理念有可能就让一些其实蛮适合长期从事软件测试工作的人选择离开这个行业，或者一直将就凑合地工作下去。所以，对于在中小企业工作的测试工程师，更应该时刻保持“人无远虑，必有近忧”的心态。

如果在你的价值观中，认为对一个组织的忠诚最重要，或者希望能在一个组织中长期平稳地工作下去，那么恭喜你，你比很多人幸运和幸福。不过，对于很多没那么幸运的朋友，在没找到自己可长期归属的组织前，也许就需要时刻保持一颗居安思危的心态和“人无远虑，必有近忧”的精神状态。很多在不规范的测

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

试环境中工作，或者在一个不重视测试的环境中工作的朋友都有这样的苦闷：虽然，当前的单位给了自己一个起飞的平台，但是由于不少客观的原因限制，在这个平台上自己很难有在测试领域不断上升提高的机会，而自己的水平和眼界也长期停留在原地踏步的状态，无法再体会到测试所带来的成就感，或者工作所带来的激情。

于是有的人开始讨厌测试这份工作，认为其毫无意义，不受公司重视，工作仅仅是简单重复，是一份只需要由一些能力弱的人从事的傻瓜工作。自然有人在工作了一两年后，离开了测试这个行业，并到处对人说：“测试是很简单的工作，测试很不受重视，测试没意义。”从而导致社会上形成了一些对测试的负面认识。因此，目前还在一些不规范的测试环境中工作的朋友，如果你不想与这个组织“结婚”，那么请努力在当前的环境中工作、学习。

当有一段时间，自己发现再也无法提高和提升自己的时候，就可以考虑到一个真正规范的好的测试平台去长长见识了。如果你觉得离开曾培养过你的组织，个人情感上有亏待的感觉，那么你可以悄悄给自己立下一个目标：“出去取经，留洋后，一定再回老东家，将自己在测试领域的所见所闻，毫无保留地奉献出来，帮助有恩于自己的老东家提高测试水平。”这样你心里就不会有内疚感了，不但可以帮助公司“塞翁失马，焉知非福”，免费得到先进的测试理念和测试流程，有利于公司长期地提升产品质量，同时自己也避免了测试理念和水平止步不前。综合来看，对当前的公司、对自己都是一个双赢的结果。

总结：对于所有的测试人员，为了能让自己在未来 5~10 年依然保持强劲的竞争力，就需要一直保持“人无远虑，必有近忧”的心态和精神状态。不轻言满足当前的工作内容，从深度和宽度上积极拓展自己多方面的测试技能。如果所工作的平台，无法再提升自己时，则需要依据自己的价值观来寻找最好的“改进方案”。

第2章 测试策略的相关因素



通过第1章“测试的态度”，帮助我们拥有了测试成功的软实力。而本章将通过融合中国古代经典的军事哲学思想，来帮助我们掌握取得测试成功的硬实力。软、硬实力皆有后，能让测试人员更有力地挥舞起遨游高空的翅膀。

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

2.1 测试资源与时间控制

古人云：“多算胜，少算不胜”。它告诉我们这样一个道理：做任何事之前，必须先脑中盘算清楚才行动，切忌盲目冲动。“多算”与“少算”的关系是越反复思虑，越周密推算，越能赢得胜利；反之就可能大打折扣，甚至招致惨败。因此，我们必须明白，一个“算”字的重要性，即不算不胜，多算必胜。

引申到我们的测试工作中，无论是测试计划和测试策略的制定者，还是测试任务的执行者，都需要在工作中“多算”。对于测试计划和测试策略的制定者，需要针对测试的项目时间、参与测试的人力资源状况、测试所需要的各类测试仪器资源、项目中影响进度的风险预期等诸多客观因素来制定相应的测试计划和测试策略，以保证测试目标的最佳实现。测试策略与测试计划是互相联系、紧密配合的，两者不可互相独立地制定，毕竟我们制定测试计划和测试策略的目的是为了达成最终的测试目标。在实际测试工作中，很多人在制定测试计划时，有意或无意地把测试策略独立开来设计，基本上只按主观意识来制定一个测试任务的时间进度表，而且通常都是按最理想的状态或惯性来制定，是否真的对风险和意外情况留有足够的缓冲是不得而知的，这也是非常多的测试工程师在进行测试任务时，常常大量加班的一个本因。同时，未将测试策略与测试计划紧密联系起来共同制定，也是一个影响最后计划延误、测试质量不高的一个重要因素。首先通过一个简单的军事案例来论证一下计划和策略之间的关联性。



案例

某次战役，某师接到命令必须在 3 小时内渡过黄河，让部队到达预定地点阻击敌人。这时，部队的指挥人员在制定计划和渡江策略时想到了两个方案。

- 第一种方案：也是最习惯的方案，大家按照次序，依次坐船过江，最好是超载坐船，战士尽量快地划过河。
- 第二种方案：马上把船连起来，搭成一个浮桥，大家跑步过江。

在两种方案的取舍讨论中，大家很快评估出了判断依据：

如果天上没有敌人的飞机，且战略目的是让所有人都过江，就用方案二。

如果天上有敌人飞机，且战略目的只是为了用最快的时间让一部分人到达指定地点，抢占有利地形，则用方案一。

最后的结论到底是方案一还是方案二，自然是由军事指挥官们根据当时的具体环境、具体的战略目标来选择了。在这个案例中，渡江的两个方案其实就是两个融合了各不相同渡江策略的渡江计划。大家可以清晰地看出，渡江计划和渡江策略都是为了渡江的最终目的而根据具体的客观条件来一起制定的，两者根本不可能完全独立地计划和设计。

从上面这个简单的案例中，大家可以领悟到一个测试计划的制定和测试策略的设计，都是要围绕这次测试任务的最终测试目标和具体的测试现状来组合设计的。测试策略要依赖测试目标和测试资源，以及测试时间限制，而测试计划的完善和科学也需要结合每次测试任务的测试目标和测试策略。

可以说，测试策略的方案既影响测试计划的制定，测试计划同样也影响着测试策略的制定。计划和策略的选择，最根本的依据还是要围绕着本次测试目标和自己所面对的各种客观环境和客观条件的现状来判断。

为什么目前我们国内的企业加班多，外企加班少，中国的测试工程师们勤奋、努力，但还是达不到美国产品的质量？除了企业文化、工艺、经验积累、技术积累等客观因素外，我们在制定项目的测试计划和测试策略时，是否进行了足够的“多算”来保证胜利，却常常被大家忽略了。在实际工作中，常看到的情况是：一个测试计划和一个测试策略在制定好后，可能会一直延续到5个测试项目、10个测试项目，甚至更多的测试项目中继续执行，而没有任何改变。也许最大的变化就是每个测试项目开始前，仅仅修改测试计划中各路标的日期、时间而已。对于测试计划和测试策略的“少算”，往往导致了我们在测试执行过程中显得时间紧迫，大家忙得热火朝天，加班甚多，但测试目标的实际效果却未必达到我们心目中的理想效果，而仅仅是达到了流程和制度的要求。

虽然，不少企业弘扬加班文化，认为加班体现了员工的敬业和勤奋。但是大家心里明白，通常加班的效率是达不到正常工作时间的效率的。并且由于有一个

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

测试计划的底线时间在那卡着，测试人员在具体的测试执行时，将就凑合的行为肯定也会多起来。因为，他根本就没有足够的时间和精力，甚至客观物质条件来将自己手头上的测试任务 100% 圆满完成。原本可以做到精益求精的质量目标，被迫变成了只有应付流程和制度了，因此最后产品在实际应用中的质量也跟着相应地打了折扣。

现在，大家明白了对于测试计划和测试策略的设计制定应该“多算而谋定”的重要性，也知道了测试计划和测试策略是需要紧密联系、互相影响来制定的。那么对于具体测试任务的执行者，广大的普通测试工程师们是否也需要“多算”而胜呢？答案是：同样需要“多算”。普通测试工程师“多算”的意义，不但对自己非常重要，更对整个测试计划和测试目标是否达到也有着非常重要的意义。普通测试工程师的“少算”，甚至会让测试计划和测试策略设计者的心血白费。

案例

以前有 3 位新同事被分配到一个新的测试项目，测试经理计划根据他们各自的实际情况给他们分配所测试的模块。当测试经理与他们一一进行沟通，询问是否可以按期按质完成目标任务时，3 位同事由于考虑自己是新来的同事，为了体现自己的实力和敬业的工作态度，对分配给自己的测试模块未作任何评估和考虑，一口答应没问题。可当项目正式启动后，这 3 位同事由于对新模块不熟悉，对困难的预计不足，3 人基本上天天加班，但仍然无法完成自己的任务，自己日子过得痛苦，测试计划的制定者（测试经理）也跟着一起痛苦，最后他的测试计划因为这 3 位同事的进度而大受影响。

幸运的是，这个测试计划是按期完成了，但由于这 3 位同事的工作是在长期加班加点的状态下很疲惫地完成的，其所完成的质量大家也是可想而知的。如果最后这个测试计划未能按期完成任务，则不但因为这 3 位同事的“少算”影响了整个测试计划，甚至还有可能由于他们 3 人占据的测试资源而不能释放出来，影响了其他项目的正常开展。

所以，不但是计划和策略的制定者要“多算”，对于基层的计划执行者也需要“多算”。唯有执行者与计划和策略的制定者共同“多算”，才能“多算则胜”。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

虽然广大的软件测试从业者有着许多客观困难等诸多因素，影响我们在短期内无法达到欧美企业的测试水准和产品质量。但我们若能积极地在主观上学习祖先们留下的“多算胜，少算不胜”的道理，从主观上提高测试效率和科学性，同样也可以帮助我们优化测试效果，提高产品的质量。

最后，希望大家在正式开始测试项目前，测试设计者和测试管理者能够“多算”后再来制定测试策略和测试计划。而执行计划的测试工程师们在承接测试任务前，自己心里也要“多算”，真实评估自己能否按计划保质完成工作，切勿打肿脸充胖子。“多算”对自己有利，对整个项目、对测试经理也是非常有利的。

2.2 测试的知己知彼

众所周知，“知己知彼”的思想在军事中和商业活动中已得到了广泛应用，但却很少有人会想到“知己知彼”在软件测试的工作中也同样可以得到广泛应用。在军事中无论是解放战争，还是抗日战争，共产党所领导的人民军队总能在装备落后的情况下胜多败少，其中情报工作做得比对手好，是一个非常重要的因素。

在测试活动中，当我们还在争论是否应该让黑盒测试人员了解和学习产品的设计原理、实现机制和实现方式时，我们是否考虑过是今天产品的质量重要还是未来的安全保密重要。如果没有今天的高质量，公司的产品就会没有销路，公司今天的生存都会有问题，何谈未来的发展，安全保密就更无从谈起了。

测试活动本身的找 bug 工作就可以理解为找目标的弱点。对于测试而言，有两个领域需要测试人员去知彼：

- 第一个领域：测试目标的设计原理、实现方式。
- 第二个领域：测试目标的实际应用环境。

当然不是所有的测试人员都要求对这两个方面去全面知彼，如同只有军事策略和军事计划的制定者才需要全面了解整个战局的情报，而非所有级别的军官和士兵都需要知道全部情报一样。在我们的测试组织中，只需要制定测试策略、测

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

试计划的测试技术骨干能够对这两个方面的情况进行全面地了解即可。技术骨干对这两个领域了解得越准确、越全面，越利于我们设计出成功的测试策略和测试计划，并有利于提高后续测试方案开发的质量和测试执行生产力效率，从而让产品的质量更上一层楼，提高产品销量和品牌价值。

在了解这两个领域相关重要信息的过程中，有如下方法可帮助测试人员用最少的时间和成本达到尽量了解情报的效果。

1. 了解测试目标的设计原理和实现方式

可以让系统设计人员和开发人员定期给测试设计人员进行培训和讲解。在他们的帮助下，测试设计人员可以少走不少弯路，集中精力、尽早、全面、准确地了解测试目标的内部情况。



案例：某即时通信软件服务器和客户端通信测试

首先，通过与开发人员交流、沟通了解到，服务器端是在 UNIX 平台上用 C 语言来实现的，而 C 语言在 UNIX 上较难调试，因此服务器端的软件在程序员的内测中会做得不够充分，服务器端软件出现 bug 的概率要高，所以在服务器端的软件模块要投入大部分的测试资源。同时了解到客户端软件的实现工具是用 Delphi 实现的，考虑到 Pascal 语言与 C 语言在字符串处理上的区别，因此针对客户端软件需要多在字符串转换上进行测试。

其次，通过阅读软件设计文档，了解到客户端与服务器之间自定义的通信协议中各数据结构元素大小，针对这些数据结构元素的大小可以进行溢出测试。假设所定义的传送数据块的字符数组大小为 `char Data[500]`，那么在传送数据块时，就一定要测试数据块大小为 500 个字符和 501 个字符大小的情况。同时，通过了解服务器端的并发处理方式，是异步方式还是同步方式，是多线程还是多进程来处理多个用户的并发连接，就可以针对本产品实现的同步处理方式来制定不同的有针对性的测试策略。

最后，了解到后台存储数据的数据库是哪种数据库后，可通过上网去了解该种数据库已发现的缺陷和性能短板在哪里，然后针对数据库的这些问题结合所测试的项目特点进行测试，保证服务器端实现的软件代码能够很好地屏蔽数据库的短板。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

2. 了解测试目标的实际应用环境领域

可以先让技术支持小组和解决方案组提供所有已知的设备应用方案和设备配置参数。然后，测试设计人员可以在公司资源的支持下利用各种到用户现场与用户交流的机会，从测试和质量的角度听听用户对现在已有的产品问题和未来产品应用时的质量要求。最后，综合公司的已有资源和测试人员自己的第一手资料，来得到测试目标在实际应用环境中的质量要求，以及测试方向的重心。



案例：一个以太网交换机产品的测试

某以太网交换机厂商的产品，在市场应用中常出现各种工作不正常的情况，有些是客户使用环境的问题，有些则是产品对实际环境的适应能力不强的问题。因此，该公司测试部门先依托公司市场部的资源，将市场部已知的各种交换机产品应用的组网方式收集起来，统一进行逐一测试验证。同时，在外派测试调研小组前，先在研发部内部收集产品各功能开发小组对产品质量的调研需求。然后才到不同类型的用户现场与用户一起讨论交流，了解用户在使用以太网交换机的过程中，常遇到哪些问题和故障现象影响了正常的工作；并询问用户对于产品的质量提升，最关注哪些方面，有哪些好的建议。

待测试需求调研小组回来后，首先按照产品的不同应用场合进行分类。然后按照用户最关注的质量点的优先次序排序。最后，把产品最常出现问题的地方和场景数据进行统一管理，以确保在测试设计时尽可能地接近产品出现问题的场景。结果，在此次外出调研后得知，以太网交换机很多时候出现工作不正常的原因，就是由于真实网络环境中的数据混有大量的病毒，而这些病毒也能直接攻击公司交换机本身的资源，导致交换机无法正常运转调度各种功能。同时，用户也反馈 IPv6 之类的功能基本不用，建议减少测试投入力量；IP 路由也主要使用静态路由，建议强化静态路由的测试。用户用得最多的功能就是访问控制列表，且使用到了该功能的很多小功能点，而这些功能点恰好是测试人员所不重视的简单功能。

通过测试调研，该公司以太网交换机测试小组明白了下一步测试工作中的重点，以真正用户的思维角度来重新排序了测试功能的优先顺序，不再是以测试人员的纯技术角度按功能的实现复杂度来划分测试优先级。从此，对于一些实现简单，

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

但用户爱用的小功能提高了重视；相应地，对于一些用户基本不用或只是使用极少基本功能的大模块，减少了测试资源投入。使得整个以太网交换机产品的测试策略既没有忽视用户的真实困难，又将测试资源最科学地使用起来，大大提高了测试效率和测试质量。

前面谈了不少关于“知彼”的内容，相信读者也开始意识到“知彼”对于测试的重要性了。那么现在我们就来谈谈何为测试的“知己”。为什么本节直到现在才开始谈论“知己”的重要性，因为人最难的就是知己，毕竟人各有千秋，各有特长，各有所爱。例如：有的人擅长想和研究，有的人擅长做和实施，有的人擅长沟通，有的人就喜欢埋在代码堆中。在常见的测试团队中，通常有如下 4 种常见的人员特点。

A 类人：擅长做和实施，不擅长想和研究

这类人可能敲键盘时速度飞快，进行设备配置和环境搭建时也速度飞快，分析定位问题时效率也挺不错的。但有可能缺乏足够的创造力和发散思维的特点，构思测试方法时可能就不如 B 类人。

B 类人：擅长想和研究，不擅长做和实施

这类人虽然可能不如 A 类敲键盘快，设备配置和环境搭建时也没 A 类人快，甚至有可能看起来好像没 A 类人勤奋尽力，但却常常能在 A 类人找不到 bug 后，B 类人可以通过新创造出的很多测试方法来找到更多的 bug。

C 类人：喜欢写代码，不喜欢找 bug

这类人对代码是狂热的爱好者，只要让他写代码就兴奋不已，而对测试基本上就无兴趣和积极性。我们可以让他专注于自动化测试脚本的开发和测试工具的开发，让他在自己的爱好中工作。

D 类人：擅长沟通和协调

这类人大家都知道啦，比较适合作为测试项目的 PM 与各部门进行协调，争夺测试资源和测试的权益，或是团队的管理。还有就是从测试的角度思考与产品

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

用户进行沟通，深度挖掘出测试需要的市场信息。

以上4类人是大家比较常见的测试人员类型。现实中有许多测试人员同时兼具两种类型特点，甚至集多种特点于一身。当我们理清了测试人员的分类后，大家对“知己”也应该有了比较清晰的认识。再好的计划，也需要配上强有力的执行才能让计划得到完美实现。作为测试的管理者为了达到测试设计策略的科学性与高效，最好的策略是：先选择具有D类特长的人去收集用户测试需求，接着选择B类人来进行高质量的测试设计。当好的策划和设计产生后，再让A类人和C类人来快速、高效地实施设计好的测试策略和测试方案。只有按照“知己知彼”的思路，才能得到一个高效率、高产值的测试团队，测试人员也才能在测试团队中找到快乐和激情。

2.3 测试效率的优化

任何企业的内部活动、外部活动都是有目的的。为了实现企业的目的，需要消耗人力、物力、财力。因此企业的运营、商业运作必须讲究效率，也就是说，只能“兵贵胜，不贵久”。

行动拖延的时间越长，相应的费用就会增加，时间成本就越高。同时，由于时间上的延迟，失去先机，将丢掉市场，形成连锁反应。在前面一节中，我们曾谈到因为测试工程师在接受任务前，“少算”而导致项目执行质量下降，项目完成工期延误，不但影响了自己的项目，还影响了后续的项目。

在测试工作的各个环节中，我们可以从优化人力、物力、财力的消耗入手，用最少的资源尽早达到相应的测试目标。测试项目中可以提高效率的地方有如下4大类：

- 在测试计划中对测试模块进行优先级划分。
- 在测试计划中分配合适的人做合适的项目。
- 实施自动化测试。
- 提高测试用例设计的科学效率。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

1. 在测试计划中对测试模块进行优先级划分

在制定测试计划时，通过将准备测试的模块进行优先级划分，可以保证我们用最充裕的时间，集中注意力来保证实现最需要的测试目标。避免对所有的测试模块一视同仁，随机选择安排测试的顺序进行测试，并在忙碌中忘记了自己最需要的测试目标。

(1) 测试优先级（针对新模块测试）划分

- 新模块单元内部业务逻辑测试优先第一。
- 新模块与其他模块一比一组合测试优先第二。



案例

假设以太网交换机新开发了一个 RIPv2 协议。

测试优先顺序应该是先对 RIPv2 协议内部的各种状态转化进行测试，覆盖完所有状态转换逻辑路径后，再由近及远地与其他模块进行组合测试。先与 RIPv1 进行互通测试，再与静态路由、OSPF 进行组合测试。

(2) 针对回归测试

- 最基本的功能组合测试优先第一。
- 最常用的功能组合测试优先第二。
- 与最新代码修改有关的功能测试优先第三。



案例

假设以太网交换机的 RIPv2 协议出现了一个 bug，并成功完成了修改。

在完成了该 bug 的验证后，为了证明所做的软件修改没有影响到其他功能的正常使用。因此，我们先对以太网交换机最基本的 VLAN 划分、二层地址学习进行验证；然后再对生成树、静态路由、三层地址学习等常用功能进行验证；最后，再验证该 bug 所在的 RIPv2 协议是否没有受到影响。如果我们不先验证基本功能和常用功能，一旦用户把软件版本换成了这个新版本，出现了基本功能和常用功

能都不能用的问题，那么对 bug 修改所做的所有工作不但毫无意义，而且会带出新的麻烦。

(3) 对特殊市场版本

- 最基本的功能组合测试优先第一。
- 最常用的功能组合测试优先第二。
- 特殊市场版本开发的功能测试优先第三。



案例

假设以太网交换机临时开发了一个 RIPv2 协议的试用版。

我们先对以太网交换机最基本的 VLAN 划分、二层地址学习进行验证；然后对生成树、静态路由、三层地址学习等常用功能进行验证；最后，再验证这个临时开发出的 RIPv2 协议试用版。之所以这样安排，也是与回归测试安排的理由相同。

2. 在测试计划中分配合适的人做合适的项目

- 避免任何版本的测试全用新人。
- 避免任何版本的测试全用牛人。
- 手工测试项目的测试人员定期互换测试模块。

3. 实施自动化测试

在项目测试的过程中，实施自动化测试可以大大提高项目测试的进度。

迅速找出新版本中不稳定的变更，尽可能早地暴露回归部分的错误，减少人力和时间的投入。

- 让需要大量人工进行的测试数据能得到快速地执行。
- 结束长期重复手工测试给测试人员所带来的恶梦。
- 解放出有测试经验的人员，让其有时间从容地从事新测试方案的开发和测试水平的优化提升，间接提升测试效率和效果。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

4. 提高测试用例设计的科学效率

- 避免测试用例中的测试步骤和数据重复、冗余。
- 让测试用例设计的测试方法大部分能转为自动化测试，建议至少 50% 的方法可转为自动化测试。
- 建议有覆盖冗余的测试用例进行合并。
- 建议测试用例的执行时间要在一定范围内，便于任务的合理分配。
- 建议测试用例的可读性要高，通俗易懂能大大减少用例学习执行的时间。
- 为了达到测试的最终目的，测试用例最好是已经通过了开发者、测试者、市场人员的共同评审。
- 少做无意义的测试。漏测是我们很讨厌的，但是过度测试对开发人员和项目进展更可怕。

综上所述，我们可以通过提升以上 4 个方面的效率，来提升整个测试的效率，达到降低人力、物力、财力和时间成本的“兵贵胜，不贵久”的效果。

2.4 测试中技术风险的控制

不打无把握之仗是一切成大事者共同信守的原则，凡成大事者都需要先洞察局势变化，再拿出最有效的方法。本节将与大家一起分享“不打无把握之仗”与做好测试需求收集、新技术可行性分析和自动化测试评估的关系。在“测试资源与时间控制”中，我们讨论过“多算”对制定测试策略和测试计划的重要意义，是侧重于资源和时间的合理评估。在本节中，则侧重讨论测试中技术风险的合理评估。

众所周知，当开发人员准备开发一个从未实现过的技术时，因为没有相应的技术经验，除了评估工作量外，还会非常严肃、认真地评估技术的可行性，充分收集开发需求，判断是否有足够的技术能力攻克这一技术难关。如果选择的技术方案无法达到需求的目标，则会另选新的方案或放弃开发。既然软件测试也是一个技术性很强的岗位，那么当我们遇到了一个从未测试过的一项新技术时，也应

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

该先做好充分的测试需求调查评估，然后再对涉及新技术的测试目标进行技术可行性评估，最后才开始正式地行动。不可逞一时之勇，在未经足够需求收集分析和技术可行性分析的前提下，就轻易地接下新技术的测试任务。

软件需求项是测试需求分析的起点，而测试需求分析中最重要的是要与开发人员、需求分析师、项目经理，甚至市场人员一起来确认准确的测试目标。在做好测试需求的收集和确认工作之后，才能针对测试需求中新冒出来的新技术进行技术风险评估。

在进行新技术风险评估前，先询问自己在现有团队中是否有相关新技术经验的测试人员？如果有，拥有相关经验的人员具备“攻城略地”的能力吗？我们有时会犯这样的错误，将一些需要攻坚的任务交给一些并不真正具备攻坚能力的人，即便他有一定的技能。我们必须正视，人与人各有特点：有的人擅长攻，有的人擅长守，也有的人攻守平衡。擅长攻的人，适合做开创性的工作，具备非常强的学习能力，能用最短路径在最短的时间内寻找或创造出解决方案。但缺点是有可能很难忍受平凡和普通，甚至是重复的简单工作；对新事物非常感兴趣，对重复性工作无法坚持做到尽善尽美。而擅长守的人，做事踏实细致，也很勤奋，做着熟悉的工作时效率很高，经验丰富，掌握的技能也多，但是要独立地进行开创性的技术研究或许并不是他的特长。不过在测试工作中，非常需要大量能守的人员，就像一个足球队一样，防守队员往往是最多的，而前锋队员是最少的。只有后防稳固后，才能有进攻的基础和实力。对于擅长攻的开创性人员，如果他们的实力能达到此次任务的高度，即使目前还不具备相关的技术，也可让其放手一搏。如果能让有相关技术但擅长守的同事来协助，则技术攻坚将事半功倍。

可是现有人员中如果没有擅长研究的攻坚人员怎么办？在下次招募新人时，最好能留出 20% 的名额给虽不擅长守，但擅长攻的人，并在日常的工作中，让其有机会发挥特点。不可只为了有一个听话、易管的团队而在测试团队中只聘用擅长守的成员，当然测试团队中也不可能全是擅长攻的成员，“攻守平衡”将是最佳状态。

作为测试工作中重要的辅助手段——自动化测试，也需要我们在规划和实施

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

前进行足够的技术评估。通常在完成测试需求和新技术评估后，开始编写测试计划和测试用例。假设在测试计划中本次测试自动化率要达到 60%，那么我们应该先对测试用例中具体可转为自动化测试的步骤和方法进行逐一评估，判断现有的自动化测试技术是否能实现，是否有免费的自动化测试工具可以使用，是否需要额外开发自动化测试平台和测试工具，是否团队中有人具备某项具体的自动化测试技术，能解决新自动化测试技术可能会遇到的问题。只有完成以上自动化测试技术的评估后，才能最终确定自动化测试率，定下测试计划的目标和日程安排。

综上所述，为了做到“不打无把握之仗”，测试需求的准备收集和分析、新技术的可测试性技术评估、自动化测试的技术实现能力评估将是我们正式开始这场测试战役的重要风险评估和技术储备工作。如果每一步的技术风险评估准备工作都完成得很好，再配合“测试资源和时间控制”中的“多算”，那么我们将能稳扎稳打地打好每场测试战役，并取得最后的胜利。

2.5 测试中的金矿

本节的核心思想是把握住“势”，把时机抓在手中，把握住稍纵即逝的机会，取得更大的胜利。测试中有一个现象：不少埋藏较深的 bug，正是在偶然地发现了一个异常时，把握住了一个也许是稍纵即逝的机会，深入地挖掘，才能发现。

在寻找 bug 的过程中，我们通常都会经历这样一个规律曲线：刚开始进行功能测试时，bug 数开始逐渐上升，这是第一阶段。这个阶段的 bug 种类比较多，从 P1 到 P5 都有散布，属于遍地开花的局面，感觉不用花太多精力就可以很轻松地找到 bug。

第一阶段结束和第二阶段开始的标志是 bug 数开始慢慢减少了，bug 不再很容易就能找到。这时就需要测试人员不只进行功能测试，而要把系统测试、性能测试、负面测试的武器一起抬出来投入战斗。同时测试人员也需要再多开动脑筋，构造新的前期测试用例未包含的测试方法，依赖在第一阶段中对产品功

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

能和内部原理的熟悉，测试人员可以在这一阶段更好地补充测试用例的方法和策略。

通过新的测试手段和测试武器在第二阶段的使用，结合测试人员更多的思考，在第二阶段常常可以找到一些埋藏较深的 bug。因此，第二阶段的测试工作就很类似于挖金矿的工作，先通过一定的侦察和自己专业的知识经验，找出最容易出现金矿的地方，然后用工具开始坚持往下进行尝试。如果这时发现了一块金子，笔者相信任何掘金人肯定都会更兴奋地坚持往下挖。可是我们在测试时，很多情况刚好相反，当我们在经过一段时间的努力，发现了一个 P1 或 P2 问题时，人很容易就松懈下来。虽然很高兴通过努力又发现了新 bug，却没有在出现 bug 的地方再加把劲，再努力地去试图挖出更多的金矿。

我曾听一位有 15 年以上经验的测试老兵谈过，大部分严重的 bug 都集中在少数地方。原因很可能是在软件开发过程中，这里是属于开发复杂度较大的地方，或者某类异常情况处理没有想到，又或者负责该模块的开发工程师经验不够。因此，当我们在这些地方发现了一个 bug 后，很容易采用同样的测试策略发现更多类似的 bug。所以，我们需要像挖金矿一样，只要找到一个 bug，就更需要继续沿着相同的方向深入下去，以期挖到更多的“金子”。

案例：电信网络设备的测试

因为很多电信网络设备的基本功能主要是数据转发，所以对该类产品进行性能转发功能的测试是非常重要的。通常业内会第一步对该产品进行性能指标的测试，判断是否可以达到产品规格说明书所要求达到的参数。然后，测试工程师会对产品进行长时间的压力测试，这个阶段基本上都会发现比较严重的 bug。同样的测试环境和测试条件，连续跑 2 天流量和连续跑 4 天流量会出现不一样的 bug。

有些测试人员容易犯这样一个简单的逻辑错误：认为同一个连续跑 2 天流量的测试方案，跑 2 次或 3 次的效果是一样的。可是测试经验却告诉我们，事实上同一个性能压力测试用例跑 2 次、3 次有可能发现的 bug 是不一样的。假设该性能转发模块潜伏了 A、B、C、D、E 5 个 bug，很巧合的是，当 A bug 发生了，设备

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

就不会再运转了，因此 B、C、D、E 4 个 bug 就不会被发现。如果这时我们就停止了同一个方案的测试，就会把 B、C、D、E 4 个 bug 遗漏。所以即使是跑同一个测试方案，每次运行时都有可能出现不同的 bug。

举例

寻找以太交换机的自身安全性 bug。

在大多数情况下，很多厂商对以太交换机产品很少会进行网络安全测试。因为以太交换机产品的定义中并不包括网络安全测试，主要针对以太交换机的二层功能和三层功能进行测试。即使少数厂家的以太交换机加入了部分网络安全模块，会针对通过交换机交换的数据进行安全性检测，但还是对以太交换机本身的安全性很少进行测试验证。由于以太交换机的广泛应用，大家会发现以太交换机工作失灵的原因不是我们在测试功能时有遗漏，而是大家从未考虑到以太交换机本身会受到网络病毒的攻击。

有一次在处理市场问题时，我们发现如果持续对以太交换机的网管接口发出半连接的 TCP 连接，将会最终消耗完所有的 TCP 连接内存，导致设备最后重启。正是通过来自市场的实际应用问题，启发了我们对以太交换机的网管软件模块重点进行网络安全测试的想法。于是将常见的网络安全攻击测试，如 UDP、TCP、SNMP 的资源消耗的攻击，以及协议完整性的 Fuzzing 测试应用到了网络软件模块的测试工作中。

在对以太交换机的网管模块深入进行这些安全测试后，我们的确又发现了 10 个以上 P1 级别的严重 bug。虽然我们偶尔挖到一块“金子”，但却在发现 bug 的地方不但没有结束测试，反而进行了更大范围的深度挖掘，以至于最终在这块被大家疏忽的地方发现了更多严重的 bug。

综上所述，在实战测试工作中需要在发现过 bug 的地方比未发现 bug 的地方花更多的精力，来研究出现 bug 的地方和触发原因，把握住这个已找到 bug 端倪的“势”。要抓住产品的弱点不放，争取把握住时机，实现一点突破带动全面突破，将发现 bug 的成就由小变大，取得最大的成绩。

2.6 灵活机动的测试

战场上非常讲究将帅带兵打仗时应该根据不同情况采取不同的战略战术，讲究针对不同的地形使用不同的打法。那么我们的测试工作是否也应该针对不同的客观情况，在不同的测试环境、测试时间、测试人员状况的情况下采用不同的测试策略呢？

在测试活动中，我们需要根据不同的情况采取不同的战略战术。首先要根据不同的测试需求选择对应的测试方案，然后为了保证各种测试方案的高效执行，需要采用科学的方法保护测试人员的创造力，保障强有力的执行力。可以想象，如果测试人员没有创造力，也自然不会有灵活性，当遇到困难时，就不懂得如何高效地解决问题，继续往前走。所以，为了保障灵活机动的测试效果，我们必须同时注意保护测试人员的创造性。

通常针对不同的测试需求会有如下测试方案。

1. 普通新功能测试

对新功能进行基本功能的测试，包括：是否满足产品规格的定义、功能内部逻辑的验证、用户接口测试、与其他模块组合测试。

2. 系统测试

从一个系统的整体视角进行测试，包括：有目标地进行各种功能组合配置和实际系统应用场景的模拟测试。

3. 性能测试

验证某个功能或整个产品是否能满足产品定义的性能指标。

4. 压力测试

验证某个功能或整个产品能否在达到极限或超过极限的情况下，依然能进行

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

正常运行，并且保证性能和功能不出现异常。

5. 回归测试

当有新功能版本发布时，对已经稳定的功能进行的测试。

6. 大配置极限测试

对产品进行尽可能多的配置，验证产品在满足极限配置的状态下，保证所有配置都能正确生效，属于压力测试的一种。

7. 客户真实数据模拟测试

完全按照产品或功能的实际应用环境和数据进行测试。

8. 安全测试

对产品进行安全保护的测试，验证某产品或功能是否可以抵挡住黑客的攻击，保证产品的安全性。

9. 自动化测试

让部分回归测试从重复的手工操作中解放出来，提高了回归测试的效率和准确性。

10. 冒烟测试方案

在开始一个新功能版本测试前，可以先选择部分测试点进行验证。如果出现 bug 太多，则证明该版本还不具备进行全面测试的前提条件。该方案可以帮助测试人员避免将时间和精力浪费在一些质量很差的版本中。

11. 开关测试方案

在开始一个新功能版本测试前，先对某产品或某些功能进行最基本的功能测试。如果这些最基本的功能都失效了，则可以一票否决进行后续大规模测试的必

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

要性。与冒烟测试方案相同，可以帮助测试人员避免将时间和精力浪费在一些质量很差的版本中。

如果把以上这些测试方案比作战术中的不同战术方案，那么测试计划的制定者作为军中主帅，就需要根据实际情况，按照不同的情形制定出合适的战术方案。

当具有了灵活机动的测试战术后，我们应该如何采用科学的方法保护测试人员的创造性，来保障强有力的执行力呢？本书建议采用轮换测试员的测试对象和测试工作的劳逸结合两种方式来保护测试人员的创造力。

(1) 轮换测试员的测试对象

在足球比赛中常有“换帅如换刀”的说法，或许是不同的人有不同的运气，可以给球队带来主观上的好运。客观上由于每个足球教练都有着不同的人生观、价值观和做事的风格，因此当前任教练已证明无法带领球队再进一步时，这时换一个新的教练则至少可以给球队带来一些新鲜的东西，对球队的现状会产生一定的改变。在具体的测试活动中，我们都知道同一个功能，两个水平相当的人来测试，肯定会互相找到对方未能找到的 bug，也正是“换帅如换刀”的道理。如果我们能进行测试工程师定期互换测试，则由于“世上没有同一片叶子”的道理，以及“换帅如换刀”的说法，两个测试工程师测试同一个功能肯定会出现不同的逻辑推理路径和思维发散路径，创建出不同的测试方法，从而找到各自独立发现不了的缺陷。所以，我们应该有计划地在一个测试工程师开始对一组功能测试感到厌倦之前，安排测试工程师互相进行轮换测试。这样既可以发现前期无法发现的问题，又不会因为一直只测试一个功能而使测试工程师感到厌烦。

(2) 测试工作的劳逸结合

测试是一项以脑力劳动为主的工作，好的测试工程师是需要发散思维和创新思维的，再好的跑车也需要定期维护保养才能保持好的驾驶状态。

综上所述，作为测试的管理者要针对不同的情形来制定不同的测试策略和战术，因为每一个灵活的战术都需要依靠测试人员的创造性来保障其执行力。所以，在制定灵活机动的测试战术时，好好保护测试人员的创造性也是同样重要的。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

第3章 自动化测试策略



本章将帮助读者对自动化测试建立正确的认识，了解自动化测试实施的策略和实施过程，从中发现自动化测试并不只是进行自动化测试脚本的开发，同样也是一个完整的系统体系。

資
子
知
庫

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

3.1 解析自动化测试的理解误区

1. 所有测试用例都可以自动化

不是所有的测试用例和测试步骤都可以转化为自动化测试。在自动化测试投入较多的行业，领先企业的自动化测试率有的能达到 80% 左右，但仍有 20% 左右的测试用例需要手工来进行。在国外，通常从开发第一版测试用例时，就同步进行自动化测试脚本的开发，所以自动化测试率普遍比中国企业高。

2. 自动化测试找不到 bug

自动化测试不直接找 bug，而是通过解放有经验的测试工程师的生产力，让其从重复的回归测试中解放出来，从事新的测试方法和测试手段的研究。通过自动化测试解放出测试人员的时间和精力来间接地找到更多、更深层次的新 bug，将产品质量再提高一个档次。

3. 自动化测试一定会马上大量减少测试人员数量

自动化测试不会马上大量减少测试人员数量。因为开展自动化测试初期需要投入一定的人力进行自动化测试脚本开发，并逐渐将自动化测试脚本用于日常的测试中，逐步减少手工测试人员从事重复劳动的时间和人数。为了缩短自动化测试脚本的开发时间，可以考虑将自动化测试脚本的开发工作借助外包的力量来早日实现大规模的自动化测试。

4. 自动化测试能代替手工测试

自动化测试不适合新功能测试，适合对软件质量稳定且经常需要被测试的模块进行投入开发。

5. 只有性能测试才需要自动化

自动化测试不光进行性能测试，更被大量应用于功能测试验证，在国外超过

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

半数的自动化测试脚本都是用于功能验证测试的。

除了以上列举的 5 类常见误区外，还有其他不同的理解误区。自动化测试理解误区的产生，归根到底最本质的原因是由于对自动化测试不现实的期望，也就是期望过高造成的。

如果没有建立一个正确的软件测试自动化的观念，认为测试自动化可以完全代替手工测试，或者认为测试自动化可以发现大量新缺陷，或者不愿在初期投入比较大的开支等，则自动化测试一定会让我们大失所望。

在多数情况下，我们对自动化测试持有过于乐观的态度和过高的期望，期望通过自动化测试方案就能解决目前遇到的所有问题。而同时一些自动化测试工具厂商过于强调其工具的优势、有利的或成功的一面，却忽视了要取得这些成功要付出持久不懈的努力和克服各种困难。因此，也就加深了我们对自动化测试理解的误区。除此以外，实施自动化测试还有如下风险：

- 一旦实施了自动化测试，便不再进行手工测试。
- 因为不再进行探索性测试，而慢慢丧失进行探索测试的能力。
- 如果脚本有问题，但测试脚本标识为 Pass，则很可能会有 bug 被永远掩盖了。

我们应该如何在合适的场合选择自动化测试呢？下面的场合就只适合手工测试，而不适合自动化测试：

- 一个新开发的不稳定功能。
- 一个不够稳定的测试用例。
- 一个需要人来判断和干涉的测试用例。
- 测试结果很难准确预测的测试用例。

而另一些场合则更适合自动化测试，不适合手工测试：

- 运行时间很长的测试用例。
- 运行重复次数较多的测试用例。
- 每轮测试都将进行测试的测试用例。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

- 高度冗余的任务或场景。
- 乏味且人工容易出错的工作。

最后我们应该有这样一个正确观念：自动化测试不是万能的，自动化测试只是测试人员工具箱里的一件利器，它无法取代测试工程师的地位。在不同的情况下，有的自动化测试目标比较容易达到，有的则比较难以达到。尽管如此，自动化测试仍然毫无疑问地具有强大功能，只要选择对了发挥其功效的最佳时机及方式，就能在测试效率和测试彻底性方面使我们获益匪浅。

3.2 自动化测试与手工测试的分工

在软件开发周期中，bug 发现得越晚其修改的代价就越高。我们除了在手动测试领域多花精力和多利用资源，提高手工测试发现 bug 的效率外，还有没有新的方法，可帮助我们较早、较快地发现 bug 呢？

实施自动化测试就可以帮助我们达到这个效果。因为自动化测试可以提高软件测试的速度及精确度和灵活度，使公司能更早发现和改正 bug。而当有限的 IT 资源和紧张的交付时间使得手工测试对于满足业务目标来说过于耗时的时候，如果继续采用手工测试，将会过多地消耗有价值的时间和资源。根据 Aberdeen Group 一个独立行业分析公司的报告，90% 的 IT 项目交付出现延迟，手工测试是其中一个因素。当手工测试需要进行测试操作系统、客户端设备、业务过程和数据集等多种因素的组合时，需要验证功能的测试用例数量会很大，工作量会非常大，测试时间也会过长。这时与手工回归测试花费的时间过长相比，使用自动化测试将会节省大量的时间。

为了改变手工测试的这些短板，人们开始使用自动化测试。不过，正如上一节所谈到的，自动化测试不是完美的，自动化测试同样需要和手工测试互相配合，优势互补，才能发挥出它的优势。我们也不可能在每个部分都采用自动化测试，而是应该寻找能够带来最大回报的部分，把自动化测试集中在关键的业务过程、复杂应用，以及由这些组成的用例方面。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

下面看看自动化测试与手工测试相比所具有的优势：

- 大大缩短回归测试项目的时间，在减少了人力投入的同时，更能保证研发项目能按时发布，甚至能缩短研发周期，提前发布产品。
- 在同样的产品研发时间内，能对产品进行更全面的多次测试，将新引入的问题尽可能多地在产品发布前挖掘出来。
- 能保障回归测试的质量。因为每次自动化回归测试都是保持同一个标准的步骤、环境和测试方法，所以测试结果具有一致性。
- 让更有经验的测试工程师从回归测试中解放出来，专注于新测试方法的研究，来发现更多深层次的产品问题。
- 减少测试工程师人数，降低研发成本。因为实施自动化回归测试后，厂商就不用像以前一样保留非常多的人力来专职进行回归测试了。
- 能避免因测试人力和时间的紧张，而降低了回归测试的质量要求，导致引入了新问题而未被发现。
- 没有手工回归测试中因为测试工作的重复性，导致测试工程师对已测过的功能过于自信，从而测试覆盖面不全，新引入问题没被发现的人为隐患。
- 避免了部分工程师因非主观的疏忽大意而引入了新的问题。

除了以上笔者想到的自动化测试的优势外，在网络上还有其他测试同行总结出的一些自动化测试可以带来的好处。包括：

(1) 快速执行

计算机在执行功能测试脚本的时候比人快得多，因此在有限的的时间里能测试得更多，在给定的时间里更多的应用可以被测试，可以按时完成更多的工程。和人不同，计算机一天工作 24 小时，还包括晚上、周末和假期，它们不会感到无聊或者疲倦，而且它们从不对该做的事情和不该做的事情自作主张。

(2) 提高测试覆盖

用自动化测试的工具对不断变化的应用和环境做回归测试，要比手工测试容易得多。通过整合的数据驱动表单的功能，自动化测试允许开发和测试团队执行

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

计算、操作数据集，以及快速创建多种反复的测试，使得扩大测试覆盖范围。

(3) 提高测试精确度并提早发现更多错误

自动化测试给开发人员提供了一种再现和记录软件缺陷非常容易的方法。这将在所有环境、数据集和业务过程等之间确保功能的正确性，同时对开发过程起到加速作用。

(4) 提供规范化的过程

自动化测试鼓励测试团队规范化他们的过程，以得到更高的一致性和更好的文档记录。

(5) 提高测试的重用性

测试一旦脚本化，开发人员就可以使用和重用这些脚本，可以将脚本添加到测试套件中，以适应应用的变化。没有必要为每个应用的相同功能重新创建脚本。

综上所述，可见自动化测试为整体测试目标带来很多益处，它能够帮助我们改善手工测试过程中的缺陷和不足。

与自动化测试相比，手工测试则有如下一些优势，这是自动化测试无法替代的。

- 大大发挥人的创造力和主动性，能设计出更多的测试方法来发现 bug。
- 更敏锐的洞察力，能从一个稍纵即逝的小异常挖掘出大问题。
- 测试的领域不仅仅局限于回归测试和性能测试。
- 能构建非常复杂的测试场景。
- 能做一些因实施难度很大，而无法转成自动化的测试。
- 一些需要人眼感官判断的测试。
- 手工测试是思想之源，是测试的大脑。手工测试设计的水平高低直接影响着自动化测试的价值。

正因为自动化测试和手工测试的特点不同，所以对于自动化测试工程师和手工测试工程师的技能特长要求也应该不一样。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

手工测试工程师需要很强的产品相关业务知识，思维发散并且具有创新精神。执行手工测试时，通常需要一个有头脑、善于思考、具有观察力的测试工程师，他能比自动化测试更灵活、机动地创造出新的测试方法，更敏锐地发现稍纵即逝的异常情况。

而自动化测试工程师则需要具有较强的编程能力、严谨的逻辑能力和分析定位能力。同时自动化测试工程师又可细分为两类：一类是自动化测试脚本开发工程师，另一类是自动化测试脚本执行工程师。两类自动化测试工程师要求的素质也有所不同，自动化脚本开发工程师专注于编程规范；自动化测试脚本执行工程师则需要具有非常强的快速分析定位能力和严谨的逻辑能力。

通过上面对自动化测试和手工测试特点的描述，大家对自动化测试和手工测试之间不同定位、优势互补的特点有了一定正确的认识。笔者认为，手工测试是整个测试的核心和基础，而自动化测试则具有锦上添花的作用，它无法完全替代手工测试，却可以解决手工测试无法解决的一些问题，间接地帮助手工测试提高效率和质量。二者应该是互相依赖、优势互补、无法完全替代的关系。唯有二者在各自合理的领域进行高效的配合，才能帮助测试团队取得更好的测试绩效。

3.3 自动化测试实施策略

我们在制定自动化测试实施策略时，首先应该考虑其中可能存在的风险。

1. 自动化测试时间不充足

有时根据项目计划的安排，测试人员往往被安排利用自己的个人时间或者项目后期介入自动化测试，使得没有充分的时间进行自动化测试，无法得到真正的关注。

2. 对自动化测试期望过高

有很多好的理由去开展自动化测试工作，诸如自动化测试可以节省时间，使测试更加简单，提高测试的覆盖率，可以让测试人员保持更好的测试主动性。但

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

是人们却经常过高地期望自动化测试实施后马上就能产生的效果目标。不同的公司及其投入自动化测试的资源状况，需要对自动化测试寄予不同的符合自身状况的目标，否则面对的很可能是失望。

3. 缺乏自动化测试实施的经验

因为没有经验，结果初期计划投入的资源太少，导致自动化测试迟迟未能正常启动，无法按计划达到期望的目标。

4. 自动化测试工具更新过于频繁

学习不同自动化测试工具的特性和脚本风格往往需要花费很多时间。当自动化测试工具更新换代频繁时，你就丧失了一部分刚刚学习到的自动化测试经验。

5. 自动化测试工具对软件测试本身没有起到帮助作用

在很多软件项目中发生了这样的情况：自动化测试工程师认为实现产品的自动化测试比测试本身更有趣，他们不参与到软件测试的具体活动中。由于测试的自动化与测试的人为割裂，导致很多自动化测试工具对软件测试并没有太大的帮助。

当我们有了针对自动化测试实施风险的准备后，就可以开始考虑：需要在什么阶段开始启动自动化测试？自动化测试的人力投入方式如何？如何执行测试脚本才更高效？

首先，在何时启动自动化测试，每个公司的情况都不同。有的公司是在测试用例都手工执行过并且测试用例不再修改时，再开发相应的自动化测试脚本；而有的公司则是在开发测试用例的同时，就进行脚本的开发。如果团队中测试用例的设计者是一个有着丰富测试用例设计经验的工程师，他所开发的测试用例是高效的，未来改动较少，则可以考虑在开发测试用例的同时，同步开发自动化测试脚本。如果团队中测试用例的设计者是一个测试用例设计经验不丰富或是设计的测试用例质量不高效的人，其开发的测试用例需要在后期经常进行许多的改动，则还是考虑等到测试用例本身稳定后，再开始脚本开发。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

其次，自动化测试人力投入方式的选择也是有讲究的。据笔者了解，大部分公司是由专人进行自动化测试脚本开发的，少部分大公司则是全民开发自动化测试脚本。这两种方式都各有利弊：专人进行脚本开发，优点是开发脚本的专业技能可以不断地得到强化，开发效率大大提高；缺点是由于对开发模块的测试用例了解并不深入，有可能开发出的自动化测试脚本只是“翻译”测试用例，发现 bug 的概率较小。而有的大公司，由于员工的整体素质较高，通常都具备一定的开发能力，则由每个模块的手工测试者自行开发自动化测试脚本。虽然，手工测试者脚本开发的熟练程度没有专门的脚本开发者熟练，但是由于手工测试者是最了解测试用例真谛的人，因此他开发出的测试脚本就不仅仅是“翻译”，而可能是对测试用例的“升华”，其测试脚本发现 bug 的概率会更大。

最后，如何执行测试脚本才更高效呢？

(1) N 个测试环境同步并行执行测试脚本，可以将自动化测试脚本执行的总时间成本降低为 $1/N$ 。

(2) 由专门的自动化测试执行工程师来执行批量的自动化测试脚本。自动化测试脚本运行失败的前 3 大因素大致为：

- 测试环境问题；
- 脚本错误；
- 被测目标出现 bug。

由于专门的自动化测试执行工程师对大量失败的脚本分析经验的积累，通常可以非常高效地定位脚本失败的原因，提高自动化测试脚本执行的效率。

(3) 独立的自动化测试环境供脚本执行团队使用。如前所述，测试环境问题是测试脚本失败的原因。而测试环境影响测试脚本执行的两大杀手：一个是测试环境被前一个失败脚本破坏而未还原；另一个则是测试环境被其他项目的同事给破坏了。对于第一种情况，我们可以在测试脚本的代码结构中加入足够的系统恢复代码来解决；对于第二种情况，则只有依赖于公司领导的政策支持，是否愿意腾出足够的测试环境给自动化测试执行小组专用。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

(4) 在测试脚本中加入丰富的脚本失败的定位信息。自动化测试脚本一旦失败，我们就只有依靠脚本自身打印的信息进行定位了，定位问题的速度快慢除了依赖脚本执行人员自身的经验外，更依赖脚本中是否有着丰富的脚本打印信息。

(5) 使用自动化测试基线软件版本。当出现大批量测试脚本失败的情况时，可以在排除了测试环境问题后，直接把这些失败的测试脚本在基线软件版本中运行。如果在基线版本中运行全通过了，则证明脚本失败原因是产品新 bug 引起的，而不用逐个地去阅读这些失败测试脚本的源代码来分析脚本自身原因。

当决定了自动化测试的开始时间、资源投入模式和执行方式后，就可以对自动化测试的目标进行梳理了。针对测试目标的不同，来选择是否进行自动化测试：

- 基本功能回归测试

如果准备在每个版本都运行同样的测试用例，则这个环节值得你在自动化测试上进行资源投入。

- 配置测试

如果希望软件支持多种不同的平台，并支持在所有平台上进行测试，也建议采用自动化测试。

- 测试环境建立

对于大量不同的测试用例，可能需要相同的测试环境搭建过程，可以用自动化测试来实现测试环境的搭建。

- 非 GUI 测试

实现命令行和 API 的测试自动化比实现 GUI 自动化测试容易得多。

Case 案例

我们用以太交换机产品的自动化测试实施策略作为案例，来讲解如何进行自动化测试实施策略的制定。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

1. 自动化测试人力资源模式

鉴于我们已有的人力状况，决定实行 $1+n$ 方式。首先选出一名测试工程师作为自动化测试的总技术负责人，他将不再参与任何手工测试的工作任务，主要职责是开发一个标准的自动化测试脚本模板，以及自动化测试平台和测试脚本库函数的开发及维护，同时还担任所有自动化测试的技术支持工作。其他所有功能测试人员在经过简单培训后，由各自独立开发调试所测试模块的自动化测试脚本。

2. 自动化测试工具选择标准

(1) 首先寻找一个基于 Linux 的开源自动化测试平台，便于利用众多基于 Linux 开源的自动化测试工具的集成和后续维护。如果没有找到开源的自动化测试平台，可考虑申请投入 1~2 人用 3 个月时间开发一个简易可用的自动化测试平台，日后再不断完善和改进。

(2) 优先直接利用 Linux 开源测试工具。

(3) 任何自动化测试工具的选择型，必须经过整个测试部门所有技术骨干的集体决策。

3. 自动化测试脚本的开发语言

由于我们的产品主要是以基于命令行的操作为主，为了提高自动化测试效率，将采用 Tcl 语言作为脚本开发语言。

4. 自动化测试脚本开发启动时间和工程师投入时间

由于我们公司进行测试用例设计的手工测试工程师都是具有 1 年以上实战测试经验的熟手，因此设计的测试用例质量较高，可以在完成测试用例评审，以及两轮手工测试后，安排 40% 的时间给手工测试工程师开始编写自己所需要的自动化测试脚本。

5. 自动化测试脚本的执行方式

(1) 由专人执行，即自动化测试脚本的总体负责人来执行。因为他不但可以利用脚本执行时间并行从事一些测试脚本优化的工作，同时也可以通过在执行过

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

程中发现的脚本问题，来不断优化他所开发和维护的自动化测试脚本框架模板，以及总结出不同人在测试脚本编写过程中易犯的错误，形成一个所有自动化测试脚本错误经验的洼地，高效地把各类脚本编写错误统一起来，并定期为全体测试工程师进行脚本开发错误经验培训。

(2) 采用并行方式执行测试脚本。鉴于我们已有的测试设备资源状况只能搭建3个并行的测试环境，因此，需要由自动化测试脚本的总体技术负责人进行合理的测试资源分配。

(3) 自动化测试脚本的分组规则及执行优先顺序。由每个测试项目的测试项目经理与自动化测试技术总负责人共同制定执行优先顺序。

6. 自动化测试应用的预期目标

(1) 应用于所有的回归测试。

(2) 功能测试部分：争取第一年达到30%的自动化率，第二年达到60%的自动化率。

(3) 性能测试部分：争取第一年达到40%的自动化率，第二年达到70%的自动化率。

(4) 系统测试部分：争取第一年达到10%的自动化率，第二年达到30%的自动化率。

7. 自动化测试对测试用例的影响

(1) 所有新开发的测试用例，在开发和评审过程中都必须考虑测试方法的可自动化。

(2) 在正式开发自动化测试脚本前，由各手工测试工程师自行选出已有测试用例中急需转为自动化测试的部分，和可以容易转为自动化测试的部分。对于急需转为自动化测试，而又不易技术实现自动化的部分，可以集体开会讨论如何修改和优化测试用例，以保证实现自动化测试。

(3) 在项目间歇期，让每名手工测试工程师必须选出一定数量的测试用例进行优化，通过改进使尽可能多的测试用例可转成自动化。该目标由各测试组长进行监督和跟踪。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com



经验提示

“罗马城不是一天建成的。”自动化测试的实施也需要一个积累经验、循序渐进的过程，不要期望在短期内实现所有测试的自动化。成功的自动化测试需要制定相应的自动化测试计划，在没有进行计划的条件下，实施测试自动化只会带来混乱。最开始我们可以从整个测试计划的一小部分开始实施自动化测试，然后再依据资源情况逐步添加自动化测试集合。同时自动化测试脚本开发的策略应该包括：可以在不同测试中应用的测试函数库和采用将代码与数据分离的方式进行脚本框架的开发。例如，最新的第三代自动化测试技术就是把测试数据写入到简单表格中，用一个解析器来解释表格中的数据，并执行测试。这种方法被称为**表驱动或数据驱动**（data-driven），该测试脚本架构最大的好处是，它允许把测试内容写在具有一定格式的表格中，这样方便数据设计和数据检视。

好的自动化测试策略是自动化测试实施是否成功的第一步。只有充分考虑到自身实施自动化测试的风险、资源和目标后，才能制定出适合自己的自动化测试策略，并最终对后面的自动化测试实施过程产生帮助。

3.4 自动化测试实施过程

自动化测试实施的过程大致如下：

第一步：制定自动化测试方案。

第二步：开发自动化测试脚本。

第三步：执行自动化测试脚本。

3.4.1 制定自动化测试方案

制定自动化测试方案，至少需要包括如下内容：

- 自动化测试方案的目标和计划。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

- 自动化测试工具的选择。
- 测试报告的输出。

1. 自动化测试方案的目标和计划

- 项目目标：自动化测试能否帮助提高项目进度，降低项目延迟风险。
- 自动化测试覆盖率：能覆盖软件的哪些关键特性和功能。
- 实现复杂度：自动化测试是否容易实现，包括数据和其他环境的影响。
- 自动化测试计划时间表：项目时间计划表、脚本开发计划表。
- 自动化测试开发需要的资源：需要投入的人力资源、硬件资源和数据资源。
- 自动化测试的执行计划：是否由专门的自动化测试执行小组来执行，以及计划投入的人数。

2. 自动化测试工具的选择

为了达到事半功倍的自动化测试效率，我们应该根据各个自动化测试方案中测试对象的不同，来选择合适的自动化测试工具。目前有很多厂家提供自动化测试产品，每个解决方案都有自身的优势和劣势、独特的功能和市场应用环境。选择自动化测试工具的原则如下：

(1) 该工具的学习使用周期要短，如果掌握工具的使用方法需要耗费较多的时间和精力，则不如放弃这种工具。

(2) 该工具的开发脚本方式要简单，开发周期短。

(3) 最好支持标准的自动化测试脚本，如 Tcl/Perl/VBScript，既便于我们利用已有的资源进行学习，也能大大简化开发脚本的难度。

(4) 该工具提供相关 API，利于我们灵活地将其应用于自动化测试脚本开发中。

(5) 该工具能有一定年限的后续支持，毕竟任何工具难免都会有 bug，如果没有相关的开发维护团队，则这些无法修复的 bug 将会大大影响自动化测试的效率。这里需要强调的是，开源软件不代表没有后续开发维护团队，商业软件也不代表

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

未来 2~3 年还能有后续开发维护团队。

因此，对于自动化测试方案中第三方技术依赖性最大的自动化测试工具的选择一定要谨慎。毕竟，很多时候如果更换了自动化测试工具，基本上很多测试脚本都将无法使用，前期的自动化测试投资也将付之东流了。

3. 测试报告的输出

如果没有一份有效的自动化测试报告，即使进行了非常全面的自动化测试，人们也会因缺乏对完成的自动化测试的全面了解，而看不到自动化测试的贡献。在自动化测试的测试报告中，除了需要有明确的统计数据，如测试用例个数、测试用例通过率、测试用例失败率外，还需要提供失败测试脚本的跟踪信息，以及测试脚本失败的原因分析。对脚本失败原因的分析文字要具有良好的可读性，对问题有很好的描述与定位，可供自动化测试人员、开发人员等多方人员阅读，让大家对测试结果有很好的理解和定位。一份良好的自动化测试报告可以帮助我们跟踪项目进度，把握功能点的测试完成情况，并有利于 bug 定位，更是自动化测试工作成果的真实反馈。

3.4.2 自动化测试脚本开发

自动化测试脚本开发涉及如下领域：

- 自动化测试框架；
- 自动化测试脚本的文件分类；
- 自动化测试脚本的版本管理；
- 测试脚本代码的结构。

1. 自动化测试框架

自动化测试框架不仅指应该有一定的脚本函数库，还需要有一个能批量运行各种自动化测试脚本的图形化平台，通过这个 GUI 平台进行测试脚本的提交、暂停、停止等一系列操作。这个自动化测试框架将大大提高自动化测试脚本的执行效率，

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

也有利于我们通过编排测试脚本的执行次序，来制定各种灵活的自动化测试策略。

2. 自动化测试脚本的文件分类

我们在开发自动化测试脚本时，最好包括如下文件：脚本代码文件、脚本环境配置文件、脚本输入参数或测试驱动数据文件、脚本运行 log 文件。将脚本环境配置文件和脚本测试驱动数据文件，与脚本代码文件分离是为了实现自动化测试代码的松耦合，这样非常有利于代码的维护，也利于脚本执行人员分析定位问题。当测试环境或测试数据发生改变时，只需修改对应文件的相应数据项，而不用去管代码的实现，使得脚本具有非常强的灵活性和适应能力。脚本运行 log 文件除了包括运行步骤的记录外，还应该把各种脚本的报错信息加入其中。

3. 自动化测试脚本的版本管理

进行任何软件开发都需要进行版本管理，同样自动化测试脚本也需要版本管理。

这里主要与大家分享一些与自动化测试有关的版本管理特点。首先，每一个版本的自动化测试脚本都必须与一个测试用例的版本一一对应。其次，为了统一管理，只有脚本的开发工程师才能按需求修改脚本，并提交新脚本与变更需求，且只有经评审通过后，才能一同 check in 相应的脚本库中。最后，自动化测试脚本不用考虑向前兼容，但需要考虑向后兼容。因为，本公司的产品虽然在不断向前改变，但是依然有一些老产品也需要适当的维护，所以保持对老产品的自动化测试兼容就显得非常重要。

4. 自动化测试脚本代码的结构

主要包括两大要素：

(1) 需要在脚本开始执行部分或脚本结束部分进行 clean 环境操作。也就是把环境恢复成一个“纯”的、“干净”的测试环境，保证自动化测试脚本不会受上一个测试脚本的影响，也不会影响下一个测试脚本的执行。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

(2) 测试用例的代码应具有良好的健壮性，理想的测试用例代码不会经常报错。只有当被测模块真正发生了错误，或是测试环境被破坏时，才发出正确的告警或报错信息。不过，实际上在自动化测试实施过程中，测试代码的健壮性却很难保证。既有自动化测试代码编写人员本身编程水平的因素，也有由于功能需求和设计的变化导致测试用例本身也发生改变的情况。如果测试脚本更新不及时，就会导致被测模块的失败。因此，一旦测试用例本身发生了变更，则必须及时通知脚本开发人员更新自动化测试代码，以减少这种失误的产生。

3.4.3 自动化测试脚本的执行

在完成了自动化测试方案的设计与自动化测试脚本的开发后，自动化测试脚本的执行就是自动化测试效率的最后瓶颈。在上一节中，我们已经谈到了自动化测试脚本执行策略的选择和计划。本节的重点是关注自动化测试脚本执行过程中的一些重要因素。

建议在一个适合产品特点的自动化测试平台上进行自动化测试脚本的执行。通过自动化测试平台可以很方便地知道目前哪些测试脚本通过了，哪些没有通过，并且可以比较容易地将测试脚本按功能或测试环境进行分类，便于测试脚本执行的管理，大大提高了自动化测试脚本的执行效率。

建议由专人进行自动化测试脚本的执行，而不是由开发脚本的工程师来执行。毕竟分析和调试脚本失败的原因是一个非常需要经验和技巧的工作，唯有熟能生巧才能高效。

建议有一个专门的自动化测试环境，可大大减少自动化测试脚本执行过程中因环境问题而出错的概率。毕竟，环境问题是自动化测试执行过程中失败最多的因素，是排名第一的脚本失败之母。

建议有一个自动化测试基线软件版本。所谓“自动化测试基线软件版本”，就是指我们选择一个可以让所有自动化测试脚本都能全部正确、一次性通过的软件版本，将其保存为“基线软件版本”。当日后执行自动化测试脚本时，如果出现了大量自动化测试脚本运行不通过的情况，就可以先在“基线软件版本”上运

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

行该失败的自动化测试脚本。如果能够通过，则可以快速地排除脚本代码的问题，将失败的原因快速定位到产品本身的 bug 上。使用“基线软件版本”的方法大大提高了自动化测试时分析定位产品 bug 的效率。

建议最好将测试数据与脚本进行分离。这样每换一批测试数据，就相当于有了一批新的测试用例。经常改变测试数据，能帮助我们找到更多的 bug。

最后在自动化测试执行的用人标准上，**建议**不要用不尊重测试的自动化测试执行工程师，也不用不了解测试的自动化测试执行工程师。自动化测试执行工程师不能选择测试团队中所谓能力最弱的人来做，应该选择有很强的分析问题、定位问题能力，而发散思维和创造性相对较弱，有一定经验的测试工程师来执行，这样才不会让自动化测试执行成为整个自动化测试实施过程中的最后一个效率瓶颈。

3.5 自动化测试开发脚本

3.5.1 什么是测试脚本语言

“测试脚本语言”一般是指在测试过程中使用到的脚本语言。在测试过程中使用的脚本语言主要有 2 大类：一类是测试工具本身使用的脚本语言；另一类是通用的脚本语言。对前者来说，使用何种脚本语言主要取决于工具本身，例如，Robot 工具使用的脚本语言类似于 VBScript 和 JavaScript，LoadRunner 使用的脚本语言是类 C 和类 JavaScript 的。对于后者而言，Tcl、Perl、Python、Ruby、VBScript 都是常用的测试脚本语言。

测试工具使用的脚本语言有很多种，而且也不具有典型代表性，在此我们就不进行详细描述了。如果读者遇到了特殊的测试工具脚本，也能很容易通过测试工具自带的帮助文档掌握这些特殊测试脚本的使用。本文中我们重点关注最广泛使用的 3 种脚本语言，即 Tcl、Perl、Python。无论使用哪种语言，只要真正熟悉它，就可以用它完成所有你想要完成的工作。不过，如果在最开始选择脚本语言时能多花些工夫，对后续的工作会产生更大的价值。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

首先，要确认脚本语言是否可以在你的工作平台上使用。

其次，要确认脚本语言是否能满足你的测试目标的特性场景。例如：对于基于命令行的产品，自动化测试脚本语言通常会用 Tcl、Perl，效率会比 VBScript 更高。

最后，由于脚本语言都会提供很多扩展库来扩充自己，例如 Perl 的 CPAN 扩展等。如果某种脚本语言提供了一个非常适合你的测试要求的扩展，那么不妨首先考虑使用这种脚本语言。例如：Tcl 的 expect 提供很强大的字符处理能力，能大大提高测试脚本的字符判断处理能力。Perl 作为历史最悠久的脚本语言之一，拥有最大的用户群体、最完善和最广泛的各类开发库支持。Python 作为后起之秀，目前在国外发展非常迅猛，各种学习资料也比较充足。

由于脚本语言本质上也是一种语言，学习它们的方法与学习其他的编程语言并没有本质的区别。所以，在选定好脚本语言后，多实践永远是最好地发挥脚本语言功力的方法。在实际工作中学习，在实际工作中应用，遇到不懂的再去找文档补充学习。

下面，笔者将与大家分享脚本语言的过去、现在和未来的故事及感受。作为应用最广泛的脚本语言 Perl 和 Tcl 等与 C 或 Java 代表的系统程序设计语言有完全不同的编程形式，脚本语言使用无类型方法来实现比系统程序设计语言更快开发、更易使用的效果。计算机速度的增长和混合应用，使得执行速度不再是选择开发语言的最重要因素。执行速度稍慢，但更易快速开发的脚本语言将在今后的应用中越来越重要。在国外过去的 15 年里，人们编写计算机程序的方法发生了根本的转变，即越来越多的人开始从 C 或 C++ 等系统程序设计语言向 Perl 或 Tcl 等脚本语言过渡。最为典型的大型软件就是 Google 的各种网络软件，如 Google App、Google office 套件等都是由 Python 这类脚本语言来实现的。

与系统程序设计语言相比，不同的脚本语言为不同的工作而设计，这导致了语言间的根本不同。系统程序设计语言起源于像内存字等最初期的计算机元素，它为建立数据结构和算法而创建，追求的是速度和对计算机底层的控制；而脚本语言是为快速、高效开发而设计的。系统程序设计语言使用强类型定义来帮助处理复杂事务；而脚本语言使用无类型定义来简化组件间的联系，并提供快速应用开发。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

在过去脚本语言和系统程序设计语言互为补充，在 20 世纪 80 年代的 UNIX 机器上，C 被用于系统编程，B shell、C shell 等 shell 编程被用于应用开发。在现在已基本成型的网络时代中，Java 被用于系统编程，而像 JavaScript、Perl 和 Tcl 等语言被用于互联网应用程序的开发。

在过去脚本和系统编程是共生的，共同使用，它们能产生格外强大的编程环境：系统程序设计语言用于生成速度快、效率高的组件，然后用脚本语言把它们连接起来。在 UNIX 下经常使用 shell 来调用使用 C 编写的程序。这些语言在组件框架中有着典型的应用：组件由系统程序设计语言创建，并由脚本语言组合在一起。后来随着速度更快的机器的出现，更好的脚本语言、图形用户界面和组件构造重要性的不断提高，以及因特网的发展等，脚本语言执行速度慢的弱点不再成为关键因素，反而易用、快速开发的特点成为了最有价值的特点。使用脚本语言的人开始越来越多，而且据国外大师预测在今后的 10 年中，这种趋势将继续。

像 Perl、Python、Tcl 和 UNIX shells 等脚本语言与系统程序设计语言有完全不同的编程方式。脚本语言常用于扩展组件特性，但它们很少用于复杂的算法和数据结构。因此为了简化连接组件的工作，脚本语言被设计为无类型的。这种无类型对于大多数脚本语言而言，本质上是绝大多数的数据结构都被它看成字符型来处理，所有的内容无论是看起来还是使用起来都是完全一样的，因此它们可以互换。例如，在 Tcl 中一个变量可以一会儿处理字符串，一会儿又处理整型，代码和数据也可互换。

脚本语言和系统程序设计语言的另一个重要区别是脚本语言是被解释的；而系统程序设计语言是被编译的。被解释的语言由于没有编译时间而提供快速的转换，通过允许用户运行时编写应用程序，解释器使应用程序更加灵活。例如，一个基于 Perl 的网页浏览器可以通过把网页中的 HTML 转换为使用一些常规表达替代物的 Perl 脚本，从而从语法上分析网页，然后执行脚本。

虽然脚本语言不如系统程序设计语言效率高，但幸运的是，脚本语言的性能问题在如今这个硬件性能异常强大的时代已不是什么大问题了，我们在使用脚本语言时，基本感受不到执行速度稍慢带来的影响。相反在某些方面，脚本语言比

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

系统程序设计语言更高级，例如，脚本语言平均一个指令可以比系统程序语言做更多的工作。某些脚本语言指令甚至执行成百上千条机器指令，而某些系统程序设计语言指令却只能执行大约 5 条机器指令。

但是脚本语言也不是完美的，它并不是系统程序设计语言的替代品。脚本语言有自己的优势领域，系统程序设计语言也有自己的优势领域。在开发与硬件打交道的嵌入式程序时，系统程序设计语言有非常好的灵活性和速度优势；而在开发自动化测试脚本时，脚本语言几乎占据了垄断性的地位。脚本语言在测试领域充分展现它的最大优点：简单、够用就可以了。

不过，我们在考虑是否使用脚本语言或系统程序设计语言处理一项特殊任务时，还是可以参考业内行家给出的一些建议：

- 程序的主要工作是否是把已经存在的组件联系起来；
- 程序是否要操纵不同类型的事物；
- 程序是否包含图形用户界面；
- 程序是否做大量字符串操作；
- 程序函数是否能快速解决问题；
- 程序是否需要可扩展。

如果这些问题的答案都是：是，则表明这个程序使用脚本语言会更好。

另外，如果对下面的问题回答：是，就表明系统程序设计语言更适合这个应用程序。

- 应用程序是否执行复杂的算法或数据结构；
- 应用程序是否操纵大量数据集（像图像中的所有像素），因而执行速度很重要。

通过上面的介绍，读者可以了解到脚本语言并不是近 10 年的新鲜事物，而是已经存在了很长时间。只是由于最近几年的一些新因素促使了它被人们越来越重视，特别是因特网的发展使得脚本语言的优势得到了极大的发挥。例如：Perl 因为编写 CGI 脚本而流行，JavaScript 因为编写网页而流行。

脚本语言近年来高速发展的另一个原因是脚本技术自身升级和硬件的升级。现代脚本语言像 Tcl 和 Perl 已经进行了几个版本的升级，可支持的功能库越来越多，对用户而言功能越来越强大，几乎可以实现除了与硬件打交道的工作外的绝大多数任务。今天的机器比 1980 年的快 100~500 倍，并且仍在继续以每 18 个月翻一番的速度增长，脚本语言的执行速度已不再是影响其应用的一个瓶颈了，它是通过牺牲执行速度来提供更高的编程创作力的。

3.5.2 Tcl 的故事

上一节概括性地介绍了脚本语言的前生来世，相信读者在了解到脚本语言的价值后，现在可能很想了解具体的一个强大的通用脚本语言的故事。现在笔者就以过去在 UNIX 平台上流行最广的脚本语言 Tcl 的故事来作为开场白。2003 年笔者第一次使用 Tcl 语言时，在中国市面上只有一本关于 Tcl 详细介绍的图书，在笔者工作的城市中听说过 Tcl 的人也不到百人。

诞生于 20 世纪 80 年代的 Tcl，已经有 20 多年的历史了。Tcl 是由创造者 John Ousterhout 在 1988 年初创建的一种优秀的解释型语言，最开始的 Tcl 语言解释器提供了一系列相对通用的工具，如变量、控制结构、过程。每个使用这种语言的程序都可以扩展的形式把自己的功能加到语言中，达到用语言来控制程序的目的。Tcl 的全称“工具命令语言”正说明了这一点。John 在设计 Tcl 语言时，主要是从如下 3 个方面进行考虑的。

首先，这种语言必须是可扩展的，它应该能够让每个应用程序都可以很方便地把自己的功能加到语言的基本功能中，并且程序的特定功能使用起来必须非常自然，就好像一开始就是为这种语言设计的一样。

其次，这种语言必须非常简单、通用，能很方便地与许多不同的应用程序配合工作，而不会限制应用程序的功能。

最后，由于大部分出色的功能都来自于应用程序，所以这种语言的首要目的就是必须要有很好的整合能力。

自从 John 在伯克利大学上作了一个关于 Tcl 的演讲后不久，在 1989 年，已有

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

一部分人开始使用 Tcl 了。到了 1990 年，由于互联网的出现，许多 Tcl 的最初用户通过网络下载拿到了第一个免费的拷贝。

在这些最早的 Tcl 用户中有一个来自美国国家标准和技术研究院的用户对 Tcl 非常狂热，于是在 Tcl 世界中最有名的程序 Expect 通过这个狂人来到了 Tcl 的世界，从此以后方便了 Tcl 的用户进行文本交互程序的开发。正是 Expect 强大的正则表达式处理能力，又间接地让 Tcl 在更大范围内流行开来。

从 1990 年初开始 Tcl 的使用者数量每年都在迅速增加，原因主要有两个：

(1) Tcl 提供了最简单的在 UNIX 下开发图形用户界面的方法。Tk 的 GUI 工具既简单又强大，所编写的应用程序可以完全由 Tcl 编写而不用写 C 代码；换成基于 C 的工具来开发，比如 Motif 工具则要复杂得多，而且功能也比较少。人们很快就发现用 Tcl 来开发 GUI 比起用 Motif 可以少投入 5~10 倍的精力。

(2) Tcl 的可嵌入特性。Tcl 一推出就被广泛地用于股票交易、科学观测、生产自动化等领域。

到了 20 世纪 90 年代初，随着 Tcl 使用者的增加，人们在网上建立了一个关于这种语言的社区，目的是让大家发布功能强大的扩展和为新的用户提供帮助。Mark Diekhans 和 Karl Lehenbauer 是早期开创者中的两位，同时许多早期的 Tcl 开创者还编写了大量高质量的可扩展使用功能免费给 Tcl 的用户使用，比如 Michael Mcleannan 编写的 [incr Tcl]，为 Tcl 提供了面向对象的功能；George Howlett 编写的 BLT，为 Tk 加进了许多重要的构件；Tom Poindexter 编写的 Sybtcl 和 Oratcl，提供了对主流数据库的访问；Brian Smith 和 Lary Rowe 编写的 TcIDP，提供了 Socket 接口及一些有用的编程功能。为此，Tcl 的后来者应该感谢这些 Tcl 前辈们在 Tcl 社区中的无私贡献，才使得我们现在能获得如此高效的 Tcl。

到了 1993 年，随着 Tcl 用户越来越多，Larry Rowe 主动站出来，在伯克利组织了第一届 Tcl 专题讨论会，大约有 60 人参加，这次大会开得非常成功。从 1995 年开始，USENIX 协会开始对这个讨论会进行资助。因此 Tcl 大会就变成了一个有几百人参加，发表大量论文、教程，开展众多活动的全方位的大会。同时 Tcl 大

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

会也是世界上最大的参加者上身着装统一的大会之一，每年要分发 5~10 种不同设计图案的短袖衫来区别不同的扩展和活动。

1994 年 John 决定离开学术界，接受了 Sun 公司的邀请。在 Sun 公司的支持下，在 Sun 实验室创立一个小组，专门负责将 Tcl 开发成因特网上通用的脚本语言。

在以后的 3 年里，Sun 的 Tcl 小组成员发展到了 12 人。在 Sun 公司提供的许多帮助下，Tcl/Tk 的性能得到了大规模的提升。Scott Stanton 和 Ray Johnson 将 Tcl/Tk 输出到了 Windows 和 Macintosh 上，于是 Tcl 就成了一个跨平台的开发环境，目前有超过 2/3 下载 Tcl 的用户是用于 Windows 开发；Jacob Levy 和 Scott Stanton 仔细测试了输入/输出系统并加进了对 Socket 的支持，于是 Tcl 可以被广泛地应用于网络应用程序；Brian Lewis 为 Tcl 脚本创建了一个二进制编译器，这使它的运行速度提升了 10 倍；Jacob Levy 开发了 Safe-Tcl，这是一个强大的安全模块，使非信任的脚本能更安全地执行；Jacob Levy 和 Laurent Demailly 创建了一个 Tcl 插件，使 Tcl 脚本可以在 Web 浏览器中运行。动态加载、名称空间、时间和日期支持、二进制 I/O、额外的文件操作命令，这些如今非常有用的 Tcl 功能也都是在这个时间的 Sun 实验室中诞生和完善的。并且 Sun 公司一直信守诺言，在源代码论坛中继续公开 Tcl/Tk 的核心代码。

到 1997 年已经有成千上万的 Tcl 开发者开始使用新版的 Tcl 语言。Sun 公司的 FTP 站点的 Tcl 下载率由 1995 年底的 2000 人/星期上升到 1998 年初的超过 10000 人/星期。

到了 1997 年底，John 离开 Sun 创建了一家 Tcl 的商业公司，Sun 公司 Tcl 小组约一半的成员都跳槽加入了他的公司。TclPro 作为一套 Tcl 的开发工具于 1998 年 9 月完成，并推向市场，帮助 Tcl 开发人员提高开发效率。

1998 年，Tcl 公司为 Tcl 8.0 做了一些改进，修复了一些错误并加进了一些小的功能，比如可以更好地支持[incr Tcl]扩展。

1999 年 4 月，Tcl 发布了第一个最重要的开放源码的版本，即 Tcl/Tk 8.1，这个版本加进了对 Unicode 的支持（国际化）、线性安全机制（用于多线程服务程序）和

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

一个由 Henry Spencer 开发的全新的规则表达式包，还有一些类似对 Unicode 提供支持这样优秀的新功能。Tcl 成为第一个满足企业中关键程序各方面要求的脚本语言。

1998 年春，Tcl 获得了两个重要的奖项：一个是 ACM 软件系统大奖，奖给每年的“具有长远影响的软件系统”，以往这个奖项的获得者包括许多重要的系统如 TCP/IP 协议、第一个线性表、第一个关系数据库、因特网、UNIX 系统、PostScript、Smalltalk；另一个是 USENIX 软件工具用户组织（STUG）大奖，这个奖项奖给年度优秀的软件工具。

Ok! Tcl 的故事讲完了，相信读者通过了解 Tcl 语言的背景故事，能更深刻地了解到作为曾经在 UNIX 平台上广泛应用的第一脚本语言的价值和特点。

3.5.3 Perl 的故事

Perl 被称为“骆驼”语言。发明 Perl 的 Larry Wall 就说过：“Perl 可能不好看或者不好闻，但是它能完成任务”。Perl 的官方网站是 www.perl.org。

- Perl 是文本处理语言，它最初就是为了这个目的设计的。
- Perl 是网络编程语言，它内置了 socket/client-server 协议。
- Perl 是系统管理语言，全球网络中很多服务器上就运行着它的脚本，很多时候是单行的脚本。
- Perl 是 Web 脚本编写语言，至今它仍然是最流行的 Web 脚本编写语言之一。

同时还可以：

用 Perl（调用 TK）编写 GUI 程序。

- 在 C/C++ 中进行 Perl 编程，在 Perl 中进行 C/C++ 编程。
- 用 Perl 过滤邮件。
- 用 Perl 数据仓库编程。
- 在 Web 上进行搜索。

.....

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

Perl 是一种能完成任务的语言。从一开始, Perl 就设计成可以把简单工作简单化, 同时又不失去处理困难问题能力的语言。它可以很容易操作数字、文本、文件和目录、计算机和网络, 特别是程序语言。Perl 吸取了 C、sed、awk、shell scripting, 以及很多其他程序语言的特性。

Perl 的诞生有着很大的偶然性, 1986 年, Perl 的创造者 Larry 是一个系统程序员, 在做一个多层安全的广域网项目时, 他负责这样一个系统: 这个系统由西海岸的 3 台 VAX 和 3 台 Sun 机器, 通过一条加密了的 1200 波特的串行线路和东海岸类似配置的系统连接组成的, 因为 Larry 的主要工作是支持(他不是该项目的程序员, 只是系统专家), 所以他就有机会来开发和提高所有有用的工具, 比如 `rm` 和 `patch` 和 `warp`。一天, Larry 刚刚把 `rm` 撕成碎片, 把它一片一片地放在他的目录里, 大管理员就跑进来说, “Larry, 我们需要一个管理配置, 用它控制所有 6 台 VAX 和 6 台 Sun。我们想在一个月内就要它, 你做一个吧!” 所以, 从不逃避工作的 Larry, 开始问自己做一个关于两个海岸的 CM 系统最好的方法是什么, 它必须不用自己从头开始写, 并且还可以查阅两个海岸的问题报告及核准和控制。他想到的答案只有一个词: B-news。(注: 也就是 Usenet 传输软件的第 2 种实现。) Larry 着手在这些机器上安装了新闻软件并且增加了两条控制命令: 一条 “append” 命令用于向现有的文章追加内容; 另一条 “synchronize” 命令保持两个海岸的文章数目相同。CM 系统可以用 RCS (版本控制系统) 做, 而核准和控制可以用新闻和 `rm` 来做。新闻是在核心机器中一个独立的文件里维护的, 里面有许多文件间的交叉引用。Larry 的第一个反应是 “用 awk”, 但糟糕的是, 那个时候的 awk 无法做到以文件里的信息为基础打开和关闭多个文件。Larry 不想编写一个特殊目的的工具, 结果就产生了一种新的语言——Perl。

Perl 从一开始追求的就是简单, 解决一个一般的问题用几行代码就完成了, 一个稍复杂一点的问题代码也不会超过一屏。事实上, 大多数人用 Perl 写的程序大多都没超过 100 行。Perl 最初是当做一种 UNIX 的教学语言设计的, 但是它早就移植到大多数其他操作系统里了。因为 Perl 几乎可以在任何地方运行, 所以 Perl 可以说是当今最具有移植性的编程环境。要想写可移植的 C/C++ 程序, 就得在程序里加上一大堆 `#ifdef` 标签来区分不同的系统; 要想写可移植的 Java 程序, 就必

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

须理解每种新的 Java 实现的特质。令人高兴的是 Perl 避免了所有这些问题，还保留了这些语言中的许多优点，同时还有一些自己的特色。Perl 的特色来自许多方面：它的特性集的工具、Perl 社区的创造性，以及开源运动的大环境。不过，这些特性都是混合的东西；Perl 的身世复杂，它总是把事物看成优点的不同方面，而不是弱点。如果你觉得自己陷入一团乱麻之中，非常渴望自由，那么请使用 Perl。

Perl 是跨文化的。Perl 的爆炸性增长在很大程度上是因为那些前 UNIX 系统程序员的渴望，他们希望从自己的“老家”带尽可能多的东西。对于他们而言，Perl 是可移植的 UNIX 文化蒸馏器，是“此路不通”的沙漠中的绿洲。从另外一个角度来看，Perl 还可以从另外一个方向运转：在 Windows 上工作的 Web 设计者通常会非常开心地发现自己的 Perl 程序可以不加修改地在 UNIX 服务器上跑。

尽管 Perl 在系统程序员和 Web 设计师中间非常流行，这只是因为他们是最早发现 Perl 的，Perl 可以用于更广泛的用途。从 Perl 最早的文本处理语言开始，它已经发展成为一种非常复杂的、通用的编程语言，以及完整的开发环境，包括调试器、调节器、交叉引用、编译器、库、语法提示编辑器，以及所有其他“真正”的编程语言所具有的大多数特性，使用这些特性可以让我们处理很难的问题，虽然很多其他语言也可以做到这一点。Perl 之所以成为 Perl 是因为它从来不会因为保持事情简单化而丢失其他方面的特性。因为 Perl 既强大又好用，所以它被广泛地用于从宇航工程到分子生物学、从数学到语言学、从图形处理到文档处理、从数据库操作到网络管理。很多人用 Perl 进行快速处理那些很难分析或转换的大批量数据，不管是处理 DNA 序列还是网页都没有问题。实际上，在 Perl 社区有一个笑话就是：下次股市大崩盘就很有可能是由哪个家伙写的脚本里头有臭虫造成的。

Perl 是自由软件这一点无疑对它是有帮助的，但这一点并不足以解释 Perl 现象，因为许多自由软件包没能繁荣起来。Perl 不仅自由，而且好玩，人们觉得自己在 Perl 里可以有创造力，因为他们有表达的自由：他们可以选择是为计算机速度优化还是为程序员的速度优化，是冗长还是简洁，是选择可读性还是可维护性，或者选择复用性、移植性、接受性和传授性等。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

Perl 可以给予你所有这些自由，因为它是一门有着分裂人格的语言。Perl 同时是很简单并且很富有的语言，Perl 从其他地方拿来好主意，然后把它们安装到易用的框架中。对于只是喜欢它的人来说，Perl 是实用抽取和报表语言（Practical Extractoin and Report Language）；对于热爱它的人而言，它是变态电子垃圾制造者（Pathologically Electric Rubbish Lister）；在少数人眼里，Perl 是毫无意义的重复练习。不过世界需要一点点冗余，精简主义者总是想把事物分隔开，而我们则总是企图把它们合并到一起。

Perl 之所以称作简单的语言是有很多原因的。比如，你用不着知道什么特殊的指令就可以编译 Perl 程序，只要把它当做批处理或者 shell 脚本执行就可以了。Perl 的类型和结构很容易使用和理解。Perl 对数据没有任何限制，字符串和数组可以要多长就多长（只要有足够的内存），而且它们都会自动增长。Perl 不会强迫你学习新的语法和语义，Perl 从许多你已经熟悉的其他语言里（比如 C、awk、Basic 和 Python、英文、希腊语等）借来语法。实际上，任何程序员都可以从书写良好的 Perl 代码段中读懂它的含义。

最重要的是，你不用先学习所有 Perl 的东西就可以开始写有用的程序。你可以写很小的 Perl 程序；你也可以像小孩那样写 Perl 程序，我们保证不会笑话你。Perl 里的许多观点都是从自然语言中借来的，其中一条最好的观点就是只要能把自己的意思表述清楚，那么你就可以使用这些语言的一个子集。Perl 文化可以接受任何熟练程度的成员，我们不会在你背后放个语言警察。如果你的老板不炒你，而且你的 Perl 脚本也能完成工作，那么它就是“正确”的。由于 Perl 的继承性，就算它只是用做数据归纳语言时也有丰富的特性，Perl 一开始就设计成可以浏览文件、扫描大量文本并且生成动态数据，以及打印出具有良好格式化的数据报表。不过，随后 Perl 就开始风行，于是它就成为了可以操作文件系统、进程管理、数据库管理、进行 C/S 编程和安全编程、Web 信息管理，甚至可以进行面向对象和面向功能的编程语言。而且这些功能并非只是在 Perl 这边，每种新功能都和其他东西交流得很好。Perl 并不仅仅只能黏合它自己的特性，Perl 是设计成可以用模块扩展的语言。你可以用 Perl 快速设计、编写、调试和部署 Perl 应用，并且还可以在需要的时候很方便地扩展这些应用；你可以在其他语言里嵌入 Perl，而且也可以

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

在 Perl 里嵌入其他语言；通过模块输入机制，你可以把这些外部的扩展当做内置于 Perl 的特性，那些面向对象的外部库在 Perl 内部仍然保持面向对象的特征。

不过，偏执一点儿说，Perl 帮你的大部分内容和 Perl 本身没有什么关系，而是和使用 Perl 的人有关。坦率地说，Perl 社区的人可以说是地球上最热心的人了。如果 Perl 运动里面有那么一点点宗教色彩的话，那么这就是它的核心了。Perl 之所以强大，是因为有 CPAN (Comprehensive Perl Archive Network, 全面的 Perl 存档网络)，CPAN 上面有无数的开源模块，从科学计算到桌面应用再到网络等各个方面都有大量的模块，并且现在世界上也还有无数的人在向上面添加模块，如果你想用 Perl 实现某功能，不用自己做，在 CPAN 上面搜一搜，多半都会得到已有的结果。CPAN 是查找任何与 Perl 有关的东西的中心仓库，它包含从整个 Perl 社区收集来的智慧：成百上千的 Perl 模块和脚本，相当于好几本书的文档，以及整个 Perl 发布。如果有东西是用 Perl 写的，这个东西很有用而且是自由的，那么它很有可能就在 CPAN 上。CPAN 在全世界都有镜像，你可以在位于 <http://www.perl.com/CPAN> 的 CPAN 路牌上找到离你最近的镜像。那块路牌会记住你选择的是哪个镜像，并且你以后再访问 <http://www.perl.com/CPAN/> (注意最后的斜杠) 时就会自动重新定向到那个镜像。另外，你也可以从 www.cpan.org 开始，这个站点的界面不同，但是数据是一样的。

Perl 的未来会怎么样？Perl 会不会成为过去？Perl 一直在发展，本书编写期间 Perl 5 已成为了主流，Perl 6 也在规划当中。

3.5.4 Python 的故事

全球最大的 Python 应用就是 Google，非常多的 Google 互联网软件都是用 Python 开发的，例如：Google App、Google Office 套件等。

Python 是一种面向对象的解释性的计算机程序设计语言，也是一种功能强大而完善的通用型语言，已经具有十多年的发展历史，成熟且稳定。Python 具有脚本语言中最丰富和强大的类库，足以支持绝大多数日常应用。这种语言具有非常简捷而清晰的语法特点，适合完成各种高层任务，几乎可以在所有的操作系统中

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

运行。目前，基于这种语言的相关技术正在飞速发展，用户数量急剧扩大，相关的资源非常多。

1. Python 在编程语言中的定位

虽然 Python 可能被粗略地分类为脚本语言 (Scripting Language)，但实际上一些大规模软件开发计划，例如 Zope、Mnet 及 BitTorrent 都使用它，Google 也广泛地使用它。Python 的支持者较喜欢称它为一种高级动态编程语言，原因是脚本语言泛指仅用作简单编程任务如 Shell Scripts，而 Python 不能与 JavaScript 等只能处理简单任务的编程语言相提并论。

2. Python 的特色

可扩充性可以说是 Python 作为一种编程语言的特色。新的内置模块 (Module) 可以用 C 或 C++ 写成，而我们也可以为现成的模块加上 Python 接口，Python 可以使用户避免被过分的语法羁绊而将精力主要集中到所要实现的程序任务上。Python 也被称为一种清晰的语言，因为它的作者在设计它的时候，总的指导思想是：对于一个特定的问题，只要有一种最好的方法来解决就好了。Python 语言是一种清晰的语言的另一个意思是，它的作者有意地设计限制性很强的语法，使得不好的编程习惯（例如，if 语句的下一行不向右缩进）都不能通过编译，这样有意地强制程序员养成良好的编程习惯。

虽然 Python 是一个非常成功的语言，但是也有必要明白它的局限性。

(1) 运行效率低下

到目前为止，Python 可以说是所有主流脚本语言中速度最慢的，这与其脚本引擎的设计思路有关。如果你的应用对于速度有着较高的要求，就要考虑 Python 是否能满足需要。不过这一点可以通过使用 C 编写关键模块，然后由 Python 调用的方式来加以解决。

(2) 多线程支持欠佳

Python 支持多线程，但是其运行效率也不高。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

(3) 独特的语法

这也许不应该被称为局限，但是它用缩进来区分语句关系的方式还是给很多初学者带来了困惑。即便是很有经验的 Python 程序员，也可能陷入陷阱当中。最常见的情况是 tab 和空格的混用会导致错误，而这是用肉眼无法分别的。

(4) 无类型

作为一种动态语言，随时随地创建和使用变量是 Python 给我们带来的巨大的便利。但是它也会使得程序不严谨，某些错误只有在运行中才可能出现。所以，使用 Python 编程的时候，要对类型做到心中有数。这也使得 Python 的 IDE 工具无法提供便利的自动完成等功能。

目前 Python 在编程领域的占有率一直处于稳步上升之中，根据最新的数据，在世界上 Python 排名第 7。前 6 名分别是 Java、C、VB、C++、PHP 和 Perl。作为一个很年轻的语言，Python 的位置已经相当令人振奋了。随着微软将 Python 纳入 .NET 平台，相信 Python 将来的发展会更加强劲，Python 很可能会成为 .NET 平台快速开发的主流语言。读者若想了解这方面的情况，请参考 Iron Python 的相关信息。著名的搜索引擎 Google 也大量使用 Python。更加令人吃惊的是，在 NOKIA 智能手机所采用的 Symbian 操作系统上，Python 成为继 C++、Java 之后的第 3 个编程语言，由此可见 Python 的影响力之巨大。

3.6 自动化测试工具介绍

介绍优秀测试工具的目的，不是帮厂商推销产品，而是希望大家从这些优秀测试工具中去领会最新的测试思想和测试理念。

无论是直接引进这些测试工具，还是在其测试理念的基础上自行开发，对优秀测试工具的深入了解，是一种非常有价值的学习先进测试方法和测试思想的重要方式，尤其对我们测试人员更是如此。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

3.6.1 WinRunner

WinRunner——强大的企业级自动化测试工具。

Mercury Interactive 公司的 WinRunner 是一种企业级的功能测试工具，用于检测应用程序是否能够达到预期的功能及正常运行。通过自动录制、检测和回放用户的应用操作，WinRunner 能够有效地帮助测试人员对复杂的企业级应用的不同发布版进行测试，提高测试人员的工作效率和质量，确保跨平台的、复杂的企业级应用无故障发布及长期稳定运行。

企业级应用可能包括 Web 应用系统、ERP 系统、CRM 系统等，这些系统在发布之前、升级之后都要经过测试，确保所有功能都能正常运行，没有任何错误。如何有效地测试不断升级更新且不同环境的应用系统，是每个公司都会面临的问题。如果时间或资源有限，这个问题会更加棘手。人工测试的工作量太大，还需要额外的时间来培训新的测试人员等。为了确保那些复杂的企业级应用在不同环境下都能正常、可靠地运行，需要一个能简单操作的测试工具来自动完成应用程序的功能性测试。

用 WinRunner 创建一个测试，只需单击鼠标和按键，完成一个标准的业务操作流程，WinRunner 会自动记录你的操作并生成所需要的脚本代码。这样，即使计算机技术知识有限的业务用户也能轻松创建完整的测试。你还可以直接修改测试脚本以满足各种复杂测试的需求。WinRunner 提供这两种创建测试方式，以满足测试团队中业务用户和专业技术人员的不同需求。

WinRunner 的使用特点如下：

(1) 插入检查点

在记录一个测试的过程中，可以插入检查点，检查在某个时刻/状态下，应用程序是否运行正常。在插入检查点后，WinRunner 会收集一套数据指标，在测试运行时对其一一验证。WinRunner 提供几种不同类型的检查点，包括文本、GUI、位图和数据库。例如，用一个位图检查点，你可以检查公司的图标是否出现在指定位置。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

(2) 检验数据

除了创建并运行测试，WinRunner 还能验证数据库的数值，从而确保业务交易的准确性。例如，在创建测试时，可以设定哪些数据库表和记录需要检测；在测试运行时，测试程序就会自动核对数据库内的实际数值和预期的数值。WinRunner 自动显示检测结果，在有更新/删除/插入的记录上突出显示以引起注意。

(3) 增强测试

为了全面彻底地测试一个应用程序，需要使用不同类型的数据来测试。WinRunner 的数据驱动向导 (Data Driver Wizard) 可以让你简单地单击几下鼠标，就可以把一个业务流程测试转换为数据驱动测试，从而反映多个用户各自独特且真实的行为。

以一个订单输入的流程为例，你可能希望把订单号或客户名称作为可变栏，用多套数据进行测试。使用数据驱动向导，你可以选择订单号或客户名称用数据表格文件中的哪个栏目的数据替换；你可以把订单号或客户名称输入数据表格文件，或从其他表格和数据库中导入。数据驱动测试不仅节省了时间和资源，而且提高了应用的测试覆盖率。

WinRunner 还可以通过 Function Generator 增加测试的功能。使用 Function Generator，可以从目录列表中选择一个功能增加到你的测试中以提高测试能力。例如，你可以选择“calendar”，然后从日历功能的下属目录中选择，如 Calendar_select_date()，然后就可以直观地输入参数，把这个功能插入到测试中了。

针对相当数量的企业应用中非标准对象，WinRunner 提供了 Virtual Object Wizard 来识别以前未知的对象。使用 Virtual Object Wizard，可以选择未知对象的类型，设定标识和命名。在录制使用该对象的测试时，WinRunner 会自动对应它的名字，从而提高测试脚本的可读性和测试质量。

(4) 运行测试

创建好测试脚本，并插入检查点和必要的添加功能后，就可以开始运行测试

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

了。运行测试时，WinRunner 会自动操作应用程序，就像一个真实的用户根据业务流程执行着每一步的操作。在测试运行过程中，如有网络消息窗口出现或其他意外事件出现，WinRunner 也会根据预先的设定排除这些干扰。

(5) 分析结果

测试运行结束后，你需要分析测试结果。WinRunner 通过交互式的报告工具来提供详尽的、易读的报告，报告中会列出测试中发现的错误内容、位置、检查点和其他重要事件，帮助你对测试结果进行分析。这些测试结果还可以通过 Mercury Interactive 的测试管理工具 TestDirector 来查阅。

(6) 维护测试

随着时间的推移，开发人员会对应用程序做进一步的修改，并需要增加另外的测试。使用 WinRunner，你不必对程序的每一次改动都重新创建测试。WinRunner 可以创建在整个应用程序生命周期内都可以重复使用的测试，从而大大地节省了时间和资源，充分利用你的测试投资。每次记录测试时，WinRunner 会自动创建一个 GUI Map 文件以保存应用对象，这些对象分层次组织，既可以总览所有的对象，也可以查询某个对象的详细信息。一般而言，对应用程序的任何改动都会影响到成百上千个测试。通过修改一个 GUI Map 文件而非无数个测试，WinRunner 可以方便地实现测试重用。

3.6.2 Fanfare

Fanfare 公司在 2004 年成立于美国硅谷，致力于为高科技设备制造商和服务供应商提供设备自动化测试的软件解决方案，以简化和加速设备测试。通过使用 Fanfare 公司提供的技术，质量保证团队和开发人员可以构建并运行自动测试，评判“通过-失败”（Pass-Fail）标准并自动生成可共享的测试文档。Fanfare 提供的测试自动化解决方案能够缩短质量检测周期，加速产品上市，提高产品质量。

Fanfare 公司产品应用于通信、航空、国防、消费类电子产品、汽车、工业控制及医疗器械等行业，其客户包括网络设备、嵌入式系统和其他高科技设备的制

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

造商和运营商，公司的核心产品包括 iTestTM Personal、iTestTM Team 及专业的咨询和服务。

iTestTM Team 产品是一个面向开发人员、测试人员和自动化测试团队的集成测试环境 (ITE)，它提供了一种统一的方法来实现测试用例的快速开发、实施和维护。使用 iTestTM Team 可以通过不同的协议来测试复杂的设备和系统，这些协议包括：Command Line Interfaces (CLI)、Web Interfaces、SNMP、Command Shells、Tcl 等。iTestTM Team 使 QA 团队的所有成员都能够参与其中，iTestTM Team 采用创新的捕获机制，实现快速的测试用例构建、快捷的用例回放、自动生成文档并对用例进行扩展，进而完成复杂的测试工作。



1. 实现快速的测试用例构建

从简单的单设备测试到复杂的多设备测试，包括功能测试、负载测试、压力测试及特定场景测试，测试专家或新手都可以用 iTestTM Team 来定义和执行测试用例。iTestTM Team 允许用户通过 ITE 或者捕捉回放来快速设计测试用例。iTestTM Team 的捕获机制能够捕获每一个命令、动作及响应，并且将它们保存为测试用例，并可以在任何时候再次运行该用例。此工具的 Test Audit 功能可以对任何已运行的测试进行分析，并自动生成相应的报告，指出该测试是否通过。

iTestTM Team 还能通过非脚本命令方式逐行构建测试用例，对已有的用例进行修改并保存为新的用例。这种方法不需要测试人员具备编写脚本的技能，便能快速构建类似的新用例，确保高效的自动化测试。对于现成的测试用例，iTestTM Team 的分析功能和逻辑功能可对该用例进行扩展，以进行更深入、更详尽的测试。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

iTestTM Team 还可以给测试用例添加各种特性，创建更为复杂的测试用例，其中包括通过各种接口实现对多设备和多测试仪器的访问。

iTestTM Team 中测试用例的每一个步骤都清晰可见，容易理解。测试人员可以通过给每一个步骤添加各种规则，定制复杂的 pass-fail 标准，并通过支持 Procedures、Session Profiles 等高级特性来进一步加速测试过程。对于原有的某个 Procedures，在其他的测试用例中可以通过简单地拖曳该 Procedures 的图标实现对它的使用。类似的，对于已创建的 Session Profiles，也可以将该 Session Profiles 通过简单的拖曳加入到新的测试用例或者已有的测试用例中，这样就可以简单地实现对设备的访问，在原有基础上快速创建新的更为强大的测试。

iTestTM Team 可以基于每个测试用例快速生成全面的报告。每执行一个测试用例，iTestTM Team 就自动创建一个测试报告，其中包括捕获到的所有设置信息、每一个命令、动作，以及按照逻辑和先后顺序对应的响应，这些信息使用的都是设备自身语言。通过测试报告可以确认所运行的用例是通过还是失败。如果该测试发现了缺陷，可以将该报告作为缺陷追踪系统的一部分进行提交，任何一个使用 iTestTM Team 或者 iTestTM Personal 的测试人员都能够快速地复现该缺陷，并且快速进行修正。

2. 简化数据分析、实现自动 pass-fail 判定

无须任何编写脚本和程序的经验，测试人员能够对测试添加复杂的分析和逻辑功能，这大大提高了测试用例的效能。由于可以添加复杂的功能，以及实现文档共享，因此可以立即对 iTestTM Team 开发的测试用例进行回归测试，同时还可以对该测试用例增加执行条件或者进行迭代，自动进行多约束条件的 pass-fail 判定。

iTestTM Team 可以分别对测试用例、设备响应及测试环境进行提取，这样就可以对测试用例的每个要素进行独立管理。对于更新后的 Response Map 或者 Session Profiles 中的静态条目，所有引用了这些文件来获取信息的测试用例都将动态更新，这样就不用花费大量的时间去处理成百上千行的代码，或者更新每一个测试用例了，还能对其他被测设备（DUT）运行同样的测试，或者在设备及软件更改后继续进行测试。

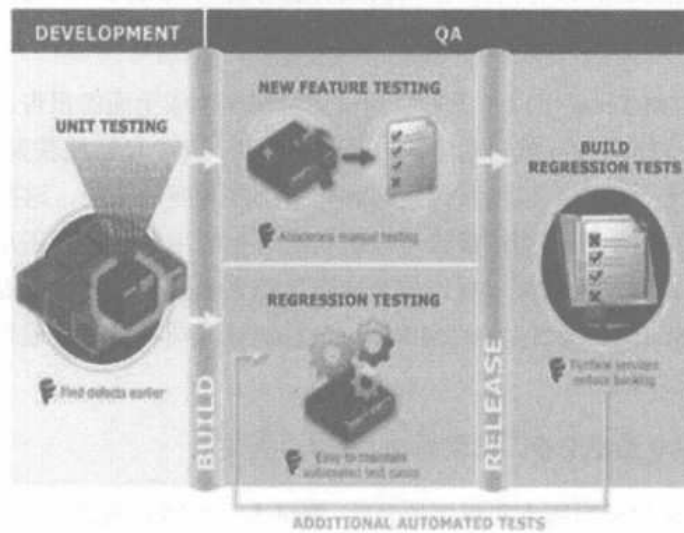
软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

iTestTM Team 具备配置和管理 DUT、网络流量发生器，以及测试环境中其他组成部分的能力。因此 iTestTM Team 是一个完善的、适用于多厂商产品环境的自动化测试解决方案。

3. 实现资源共享

iTestTM Team 提供 CLI 接口、Tcl API 及 Tcl 函数的输出接口，可最大限度地利用现有的测试资源，并且将 iTestTM Team 集成到回归测试过程中。另外，测试人员还可以使用团队中现有的缺陷追踪系统。



iTestTM Team 不仅能替代现有传统的手工测试过程，同时在实现自动化的过程中测试工作不会被中断。另外，iTestTM Team 也使得 QA 组织可以采用类似产品装配线的方式来实现测试自动化。在该过程中，开发人员开发一个简单的验证用例，功能测试人员添加自动的 pass-fail 判定标准，自动化团队通过使用提取技术来对整个测试进行扩充，直到将其构建为一个全面的能自动回归的测试。

Fanfare 产品所支持的测试领域如下：

iTestTM Team 可以使用多种协议来进行单个 DUT 及复杂系统的测试。另外，iTestTM Team 可以在以下复杂的测试环境中工作。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

- Data、Voice
- Wireless
- Video
- Telephony
- Storage

iTestTM Team 支持以下接口：

- CLI——通过 Telnet、SSH1、SSH2、TL1 发送命令。
- SNMP——获取全面的 SNMP 交互，支持 v1、v2 或者 v3。
- HTTP——在服务器或者设备上直接获取或者张贴 Web 页面。
- Web——在 Web 浏览器中仿真用户行为。
- Windows CMD——调用 PC 上的任何可执行程序。
- Mail——测试电子邮件发送过程及结果。
- Telnet/Localhost——通过 tunnels 和 VPN 捕获 loopback traffic。
- TrafficGen——用 IXIA IxExplorer、Agilent 或者 Spirent 生成并分析 traffic。
- Tcl Shell——使用集成的 Tcl Shell 来访问基于 Tcl 的工具。

下面是 Fanfare 已有的用户名单。



软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

第 4 章 性能测试与 Troubleshooting



本章将向读者展示什么是性能测试，以及性能测试与压力测试之间的关系。性能测试与压力测试是测试工作中对产品系统内部整体了解要求最高的测试阶段，需要测试人员能对产品系统有更全面和深入的认识。Troubleshooting 一节与性能测试和压力测试紧密相关，因为很多所谓不易重现的问题，大多是在性能测试和压力测试阶段发现的。分析定位问题的能力是成长为一个测试高手必备的能力，本章将与读者一起分享在分析定位方法上的经验，希望能帮助读者提高重现 bug 的能力，提高分析定位问题的效率。

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

4.1 性能测试

4.1.1 什么是性能测试

性能测试是所有测试中最常用的术语之一。性能测试关注的是系统的整体。它和通常所说的强度、压力/负载测试有着密切的关系，压力测试应该也属于性能测试的一部分。



举例说明

针对一个网站进行测试，模拟 10~50 个用户就是在进行常规性能测试，用户增加到 1000 乃至上万就变成了压力/负载测试。

有时性能测试值是正常使用的时间内系统完成一个任务所需要的时间，多任务同时使用时响应时间。以 J2EE 技术实现的系统在性能方面的要求为例，一般原则是 3 秒以下接受，3~5 秒可以接受，5 秒以上就影响易用性了。如果在测试过程中发现性能问题，修复起来是非常困难的，因为这常常意味着程序的算法不好、结构不好，或者设计有问题。因此在产品开发的开始阶段，就必须考虑到软件的性能问题

各家各门派对性能测试的定义有所不同，究其原因是因为软件产品所覆盖的应用范围非常广泛，所以人们从自己产品的领域出发提出了适用于自己的性能测试定义。

那么，是否有一种共性的性能测试定义呢？在 Wiki 上就试图对性能测试的共性给出了一种标准的解释：

In software engineering, performance testing is testing that is performed, from one perspective, to determine how fast some aspect of a system performs under a particular workload.

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

It can also serve to validate and verify other quality attributes of the system, such as scalability, reliability and resource usage. Performance testing is a subset of Performance engineering, an emerging computer science practice which strives to build performance into the design and architecture of a system, prior to the onset of actual coding effort.

译：在软件工程里，性能测试是指从一个角度出发用于测定系统的一些地方在特定负载下运行的速度。它也可以用于验证和校验系统的其他质量特性，例如可伸缩性、稳定性和资源使用率。性能测试是性能工程的一个子集。性能工程是一门新兴的计算机科学实践，它致力于在系统的设计和架构中构建性能。

随着计算机技术及软件产业的不断发展，性能测试的范围已经从最初简单的范畴衍生到一个非常广的领域，业界对其的称谓也是五花八门，例如压力（强度）测试、负载测试、并发测试等。由于在测试手段和技术上的相似性，它们更多地被看成是广义性能测试的几个子集，本节中所提及的“性能测试”一词就包含着广义性能测试的这几个子集。这些子性能测试的不同点如下：

- 性能测试：在一个特定的基准下（相对较轻的负载），测试系统的性能表现。
- 性能调优：通过性能测试帮助产品开发找到产品的性能瓶颈点，并通过一定方式解决瓶颈点，提高产品实际性能指标值。
- 负载测试：测试系统在多种负载组合下（相对较重的负载）的性能表现。
- 压力（强度）测试：测试系统在高或很高强度负载情况下的性能表现。

总的来说，在国内通常把性能测试的定义概括得更大一些，包含了国外常说的 Performance 测试和 Stress 测试。笔者以个人的经验来看，Performance 测试主要关注产品整体的某些度量指标值，而 Stress 测试则确保产品能在合理压力状况下长时间稳定运行。

一个性能测试方案的必要要素如下：

- 确定性能指标；

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

- 设计测试场景；
- 确定需要测试收集的数据；
- 确定测试方法。

根据微软公司的研究显示，性能测试的过程应该分为 6 个阶段，分别是：发现、探究、提案、执行、复查、收尾。

1. Discover the problem: 发现问题

这个步骤最重要的就是发现 (Discover) 问题、详述 (Discribe) 问题，并且正确而详细地记录 (Document) 下来。在进入下一步骤前，测试人员应该问问自己以下这些问题：

- 对问题是否已经有简明的描述；
- 用户的基线与期待在哪里。

2. Explore the conditions: 探究原因，为问题提供明确的定义与定位

这个步骤的主要任务：是广泛搜集相关数据，尽量了解系统的每一个方面，避免深入分析时，漏了某个关键的现象而误入歧途；重点是探索 (Explore)，寻找证据 (Evidence)，建立 (Establish) 整个问题的来龙去脉的假设。

有的时候在这个阶段就可以发现重大问题，一眼就看出关键点，例如硬件毁损，某个硬盘区块或内存块不稳，或者某个其他程序吃掉所有的内存，让 SQL Server 无内存可用，或者拖垮 CPU 等。

3. Track down possible approaches: 提供可能的解决方案

这个步骤的主要任务是：深入分析数据间的关联性，并对整个问题的前因后果提出假设，最后拟定出相应的策略 (计划)。如果前一个步骤做得不够翔实，这个步骤可能就会误判，导致努力了半天，但就是找不到瓶颈点。

这个步骤最重要的动作是：拟定计划。只有一个好的计划，才能知道方向与步骤。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

4. Execute the most likely approach: 执行最有可能的解决方案

这是 DETECT 方法中最简单的一步，因为只要执行上一步中所拟定的计划就行了。但是在执行计划时，仍然要注意系统的反应（随时都可能会放弃当前的计划，因为新的证据可能证明先前的判断错误，因而要修正计划，甚至退回到上一步以重新拟定计划。这时切勿急躁，因为整个性能测试的过程就是在考验团队的细心与耐力、知识的广度与深度），同时还要小心观察会不会有新的问题出现并严重影响当前系统的执行，不要变成原来系统迟缓，而你的测试却成为压垮骆驼的最后一根稻草。

5. Check for success: 确认解决方案成功与否

这一步骤的主要任务是：重新收集数据，以验证该计划的成功与否。如果证实假设是对的，则继续搜集相关数据，以确立接下来的步骤；如果到这一步发现执行的结果不如预期，证明我们的假设错误。这会让人非常气馁，进而放弃这个性能测试的方法，因为无法忍受整个时间的流失。其实，定错性能的目标是常有的事，这个方法论就是要你在错误当前，停下来好好思考，重新理出头绪，最重要的是要清楚地知道错在哪里，如此新的步骤就能更逼近目标。有系统的犯错，是整个计划的一部分，踩着错误走向成功是性能测试的常态。

6. Tie up loose ends: 完成收尾工作

重复前 5 个步骤直到达到目标。

当我们完成任务后，依然要注意以下的问题。

- 解决的方式是否有边际效应，造成其他的问题。例如：为了某类的查询工作建立了大量的索引，事后原本正常的新建、修改、删除都出现了性能问题。
- 是否真正根除了问题，还是仅表象地头痛医头，脚痛医脚。建立问题的假设时，很容易将问题特殊化，仅局部地解决该现象。例如：加了某个索引或稍稍改变查询语法，舒缓了当前的瓶颈，但当用户数量稍稍增加，或者

采用了不同的查询方式时，就老问题复发了。

- 是否要建立持续跟踪的计划。当你无法确定是否已经根除问题时，那可能就要拟定持续跟踪的计划了。决定是否要持续观察某些计数器，跟踪某些现象是否还会发生，若发生了要如何解决等。如此不但可以让用户安心，更可以让自己知道之前的行为到底有多少效益，下次的性能测试才有更完整的解决方式。

我们应该在什么时候开始性能测试呢？

以作者的经验来看，我们应该在施行了产品的功能测试确保产品主要模块联机联调通过后，才开始性能测试，这时的性能测试结果才有意义和价值。性能测试开展的时机不能操之过急。

什么时候性能测试可以结束呢？

只要产品的生命周期没有结束，性能测试就应该不断地进行下去。不是只有找到了 bug，才是对产品质量保障作出了贡献。通过性能测试的各种测试方法，不断促使产品的整体性能得到提升，也是提高产品质量的重要贡献。

性能测试用例的设计是所有测试用例设计中最复杂、对测试人员产品内部知识要求最高的工作。同时性能测试也是考验一个人能不断地坚持意志品质的工作，它需要我们不断精益求精地去追求性能的完美。

4.1.2 性能调优

性能调优的目的是为了突破系统的性能瓶颈。所谓瓶颈就是整个系统原本可以流畅地执行，但系统中若有一个点无法处理该需求量，导致整个系统执行效率被卡住，该点就是瓶颈。瓶颈 = 需求达到的处理量 > 实际处理量 (Throughput)。

针对整个系统而言，要找出整体性能卡在哪一点上，不是一件容易的事，因为系统的瓶颈点可能相当多，所以性能调优的重点是研究整体系统处理性能究竟卡在哪一点上。同时对于一个开发比较成熟的业务系统而言，功能虽然相对已经完善，但仍需要通过对系统进行配置修改或者产品框架调整来优化系统。在优化

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

系统过程中，最有效的手段就是对系统做性能测试。通过性能测试的结果不断进行系统优化，最终达到系统在大业务量情况下稳定运行的目的。

性能调优的第一步是找出系统拥有哪些瓶颈点；第二步是针对这些瓶颈点再拟定计划，逐项克服。然后重复前面的两步，不断寻找出未发现的瓶颈点，先处理执行成本较低但性能影响较大的部分。因此性能调优不是一次性的活动，而是一个不断改进、不断促进性能完善的过程。

在性能调优的第一步，我们应该如何开展工作才能更科学、高效呢？

首先，我们应该先确认产品未来的应用场景，收集确定产品应用场景对性能规格的要求，以及应用场景的实际配置状况。

然后，我们依据产品实际应用场景的配置来作为我们性能调优测试的基本配置参数。

最后，以实际应用场景的性能规格作为性能调优测试的目标，如果在性能调优测试中，数据无法达到应用场景的性能规格，则可以认为产品存在性能的 bug，需要进行性能调优来解决 bug。

只有结合实际应用场景的配置和业务要求，性能调优测试才能有目标性，有基线标准。

在性能调优的第二步，我们应该如何提高配合开发进行性能调优的效率呢？

首先，提供我们详细的性能调优测试的配置参数，以及所采用的测试方法的理由。

然后，要能确保性能调优测试结果是可重复的，排除偶然因素对性能测试结果的影响。

最后，在配合开发定位的过程中，不断丰富学习自有产品的内部实现原理。

那么什么样的测试人员能担当一个好的性能调优测试工程师呢？

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

由于性能调优测试不再只关注产品的一个软件模块，而是整个产品绝大多数产品模块联动的效果。因此，担当性能调优测试的工程师应该首先是一个对整个产品架构都有深入了解的人，才能设计出性能调优的测试用例。同时，性能调优的测试工程师也必须了解大多数产品的实际应用场景特点，避免性能调优测试用例设计的闭门造车。由于性能调优测试方法和性能调优测试结果有时具有一定的争议，开发者有时会怀疑性能调优的测试方法。所以，性能调优测试工程师也应该具备良好的沟通能力，把复杂的性能调优测试原理和场景向其他人清晰地描述清楚。

提高性能调优执行效率的建议：

因为性能调优测试会重复进行多次，所以将性能调优测试的过程转化成自动化测试，是提高性能调优测试工作效率的最好方法。利用自动化测试的标准化特点，也排除了人为影响性能调优测试结果的干扰因素。

4.1.3 压力测试

性能测试与压力测试的区别如下：

- 性能测试是母集，压力测试是子集。
- 狭义性能测试主要关注精确值，关注基线标准值；压力测试主要关注系统是否能够在高负载状况下不出现异常，更多还是设备的真实场景的可用性，甚至是恶劣环境下的可用性。
- 性能测试类似于我们读书学一门课程、掌握一项新技术，通过考试拿到的分数是性能测试值。而在实际工作中，如何应用该技术，甚至在各种环境压力情况下如何很好地应用该项技术，则是压力测试水平。

关于压力测试的准确定位，现在也是有很多种说法，这里笔者摘取了来自中国专业测试网站“51Testing 软件测试网”的一些说法和定义：

压力测试是指模拟巨大的工作负荷以查看应用程序在峰值使用情况下如何执行操作。对每个单独的组件进行压力测试后，应对带有其所有组件和支持服务的

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

整个应用程序进行压力测试。集中测试从最基础的功能测试开始，需要知道编码路径和用户方案，了解用户试图做什么，以及确定用户运用应用程序的所有方式。测试脚本应根据预期的用法运行应用程序。例如，如果应用程序显示 Web 页，而且 99% 的客户只是搜索该站点，只有 1% 的客户将真正购买，这使得提供对搜索和其他浏览功能进行压力测试的测试脚本才有意义。当然，也应对购物车进行测试，但是预期的使用暗示搜索测试应在测试中占很大比重。

压力测试是用来评估在超越最大负载的情况下系统将如何运行。压力测试的目标就是发现在高负载的条件下应用程序的缺陷 (bug)，包括：synchronization issues (同步问题)、race conditions (竞争条件)、memory leaks (内存泄漏)。压力测试能让你识别程序的弱点和在极限负载下程序将如何运行。

压力测试主要是为了发现在一定 (任意) 条件下软件系统的性能变化情况。通过改变应用程序的输入来对应用程序施加越来越大的负载 (并发、循环操作、多用户)，并测量在不同的输入时性能的改变，也就是通常所说的压力测试，考察当前软、硬件环境下系统所能承受的最大负荷并帮助找出系统瓶颈所在。其实这种测试也可以称为负载测试，但是负载测试通常描述一种特定类型的压力测试——增加用户数量以对应用程序进行压力测试。

压力测试是对系统不断施加压力的测试，是通过确定一个系统的瓶颈或者不能接收的性能点，来获得系统能提供的最大服务级别的测试。例如，测试一个 Web 站点在大量的负荷下，何时系统的响应会退化或失败。

对于以上众说纷纭的压力测试定义，笔者认为定义都没有错，但是又都不全。这些定义更多只是压力测试的一部分——静态压力测试。也就是说，被测试设备或被测软件在压力流量下进行着长时间的静态测试，没有在有压力背景流量下进行一些常规操作的验证。

那什么是非静态压力测试呢？这里以网络设备的压力测试为例：

第一阶段：长时间进行持续大压力流量测试。

第二阶段：压力流量大小进行动态变化。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

第三阶段：模拟产品的用户在真实状况下的行为场景——在足够压力背景流量下，对设备进行的各种可能操作。例如，在足够压力背景流量下：

- 对设备进行基本配置操作——设备能正常配置；
- 对设备进行各项状态统计信息的提取——都能实时收集到；
- 对设备进行升级操作——能正常完成升级；
- 对设备重启操作——设备能正常启动；
- 对设备进行各种备份操作——能顺利完成备份操作。

其中第三阶段的测试方式就是一种动态压力测试，即压力测试的过程不是让设备静静地待在一个地方，而是在有压力背景的条件下，进行着各种常规的操作而产品无异常。动态压力测试方法更多的是一种模拟用户真实应用的测试方法。在产品的实际应用中虽然可能有 95% 的时间内产品是处于静态压力状态，但是却有着 5% 左右的时间，产品会在有压力的情况下进行着一些状态的改变，为了保证在这种场景下产品不发生异常，需要在压力测试中模拟用户的操作进行动态压力测试。在任何公司的产品上进行动态压力测试肯定会发现不少严重的 bug，在动态压力测试中投入的成本是值得的。

因此，在进行压力测试时，需要充分考虑在实施静态压力测试的同时，还应该充分考虑动态压力测试的方案。动静结合才能使得压力测试阶段的价值更大，更能贴近和满足用户的真实需求。

4.2 Troubleshooting

什么是 Troubleshooting? Trouble 的含义是困难，Shooting 的含义是瞄准、射击。直译：困难瞄准。专业说法是分析定位问题。

对于从事 IT 技术的各种工种，都需要有足够的分析定位问题的能力，测试人员、开发人员和技术支持人员都需要。Troubleshooting 是一种能力，它对 IT 技术人员的意义，类似于艺术家的创作灵感，销售人员对客户需求的理解能力。分析定位问题能力是每一个 IT 技术人员必备的能力，尤其对软件测试人员要求会更高。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

为什么对软件测试人员要求会更高呢？

当软件测试人员发现了一个 bug 后，首先就应该确认是否可以重现该 bug，而重现 bug 的前提就需要测试人员能分析自己的测试步骤是如何触发 bug 的，并定位到重现 bug 的最小操作集。

其次，软件测试人员会经常进行各种测试环境的搭建，并使用很多测试工具来辅助搭建测试环境。而在搭建测试环境时，常常会遇到各种各样的环境故障，排除这些环境故障就必须依赖测试人员的分析定位能力。

最后，软件测试人员要得到开发人员的尊重和信任，就必须能与开发人员一起分析定位各种 bug，在讨论分析过程中，测试人员不但展现了自己的能力，而且还担任着一个系统级 bug 的定位总负责人的责任。例如：在压力测试过程中，某测试人员发现了一个协议状态异常的 bug，因此他请来了协议开发人员，进行共同定位发现协议异常引发原因来源于驱动。于是测试人员又请来了驱动开发人员，与驱动开发人员一起讨论分析定位后，发现问题来自硬件。接着测试人员又请来硬件开发人员，并将前期所进行的所有分析定位过程和分析定位的结果向硬件开发人员描述，最后帮助硬件开发人员定位到了具体的硬件问题。

所以，对于软件测试人员而言，掌握和具备较强的分析定位问题的能力，无论对于解决测试环境搭建的问题，找到重现 bug 的方法，还是协助各类开发人员定位到 bug 产生的原因都是非常重要的必备能力。

那么对于一些不易重现的问题应该如何进行问题的分析定位呢？下面笔者和大家分享一点个人的经验。

- (1) 首先确保被测试目标的软件版本是发现问题时的版本——版本不一致就没有分析定位问题的前提了。
- (2) 如果被测试设备是硬件设备，要确保重现问题时是同一台硬件设备、同一张硬件板卡——因为有可能问题就是由某种硬件板卡的个性故障引起的。
- (3) 搭建测试环境的辅助软件和辅助设备要与发现问题时是同一台设备及同

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

一软件版本——因为问题有可能是由于辅助软件和辅助设备的故障所引起的，或者由于辅助软件和辅助设备的某个异常才触发了被测试设备的问题。

(4) 所有测试环境的硬件条件都确保一致后，就需要确保测试目标的功能配置及参数与发现问题时的数据是一致的，特别是输入值应该与出现问题时一样。

(5) 完成前面 4 步准备工作后，至少要让一名开发人员陪同你一起开始重现问题，因为他会从实现内部原理的角度给你提供很多提高重现工作效率的意见和建议。同时，他还可以通过在测试过程中不断查看各种重要的系统数据来找到一些可帮助快速突破的蛛丝马迹。

(6) 如果在第(5)步中没有找到可以定位问题的原因，就需要再回忆第一次遇到问题时的所有场景，查看自己有无遗漏信息，并继续坚持去尝试重现。因为确实有些问题是小概率发生的，是多种瞬间状态临界值的交集。即使定位到了触发条件和原因，在操作上也很难每次都掌握到那个临界状态。只有不断尝试和坚持才能把问题重现出来，帮助开发人员修改该问题。



案例：一个 ATM 性能 bug 定位的故事

当时我们搭建了一个由数十台辅助测试的发包路由器和两台专业性能测试仪器 Smartbits 构造压力流量的测试环境，确保设备 ATM 卡所能支持的所有通道数和最大带宽都达到了上限。在压力测试早期，48 小时过去了，ATM 卡依然正常地工作着，没有任何异常。但在第 3 天，我们忽然发现 ATM 卡重启了，ATM 卡的协议开发人员和驱动人员都立刻来到了现场进行信息收集，在忙碌了三四个小时的信息收集和分析后，大家都没有在现场找到问题的原因。于是开发人员又回去分析讨论定位重现方案，最后他们在代码中新添加了一些辅助定位的打印信息，希望能在问题再现时，打印出更详细的信息以供定位。

于是我们开始了第一次的问题重现，压力测试继续跑起来，可是让人失望的是居然在 72 小时内都没有重现该问题。不过，我们没有把压力测试停下来，而是继续下去。在第 5 天，问题终于重现了，这次来现场收集信息的人更多了，当然问题发生时各模块统计打印的信息也更多、更全。后来经过开发人员 2 天的会诊，

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

最后定位到了原来是驱动层的数据转发的发送指针跑到接收指针的前面去了，发送指针的调度速度比接收指针的调度速度快了。

之后我们开始了对该问题解决结果的验证，可奇怪的事又发生了，在测试的第 2 天设备又重启了。很多开发人员又跑来定位，大家都觉得不可思议。难道分析判断错了吗？大家带着极大的疑问，仔细地看完了所有的打印数据，发现这次的出错数据与上次的问题又不一样了。难道又有新问题发生了？开发人员又在代码中加入了更多的打印信息。

第三次问题重现又开始了，这次还好在测试的第 3 天重现了第二个问题。开发人员看了自己第二次加入的打印信息后，感到非常迷茫，“不可能，不可能啊”是挂在他们口中的惊讶。原来他们从打印的信息中发现居然转发芯片不工作了，导致上层驱动的指针又飞了，才导致设备重启的。可转发芯片怎么不工作了呢？没办法，开发人员只好把所有现场收集到的数据拿回去分析研究。几天后，他们得出一个自己都觉得很奇怪的结论：芯片在停止工作前，收到了错误的 ATM 协议报文。可是错误的 ATM 协议报文是从哪里来的呢？由报文的源地址来源判断，居然是我们的测试用路由器产生的。原来，由于中低端路由器本身 CPU 频率不高，在长时间进行数据包构造后，在某一时刻把一个没有构造完全的残缺数据包发送出来了，导致 ATM 卡收到了一个自己从未预料到的报文结构，ATM 芯片无法处理而停止转动。后经由芯片厂商确认，证实了确实是芯片的问题。我方的开发人员只好通过软件来补救芯片解决该问题。

第四次问题重现开始了，这次的重现既要验证发送指针跑到接收指针前面去的问题，又要同时验证补救芯片代码的实现。幸运的是，这次压力测试在经历了连续 10 天的测试后，ATM 卡依然安然无恙。虽然中间观察到了几次瞬间复位的状态，但毕竟没有出现整个 ATM 卡重启或瘫痪的情况。现在看来前期发现的两个问题都应该被解决了。

这个案例是笔者测试职业生涯早期印象特别深刻的，印象深刻的原因就在于问题出现现象的诡异性，问题出现的场景及触发条件让人很难事先预料到。这场战斗持续了近 1 个月的时间，让开发人员多个夜晚加班分析，那段时间在笔者的办公桌边总是有几个人陪着一起盯着屏幕上不断打印出的各种信息，心里期盼着早点看到自己想看到的信息，可是偏偏信息就是不出现，而且出现的还不是自己

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

想看到的信息，却是一个新麻烦的开始。在这个案例中的两个 bug，都不是那种只要找到逻辑就能马上复现的 bug。而是那种需要在某种临界瞬时状态下才会出现的问题，也就是大家最爱说的不易重现的问题。也正是这场 bug 定位战斗，让笔者坚信了一个真理：“没有重现不了的 bug”。所以，以后只要有新员工对笔者说“这个 bug 不能重现”，笔者就会马上纠正他，应该是“这个 bug 不易重现，世上没有不能重现的 bug”。

在以后的测试工作中，渐渐的在一次次找到“不易重现的 bug”的必现原因和最小必现操作步骤后，笔者发现自己越来越自信了，越来越坚信自己的结论“没有重现不了的 bug”，并且喜欢上了分析定位 bug 的过程，以及成功后给自己带来的成功感和业务水平上的提高。而且有时参与一个难重现和分析 bug 的过程，可以让测试人员把整个产品的内部实现流程和主模块实现原理全部都深入地学习一遍，其理解和感受之深刻，是看任何静态资料都无法达到的效果。如果说在测试工作中，笔者最愿意让领导分配的工作是什么？那就是非常高兴地听到领导说：“董杰，那边又出现了一个难分析定位的 bug，帮他们重现和定位一下吧。”一场战斗又打响了！

最后总结 Troubleshooting 的核心是：

- 头脑要清晰，思维要收敛，而非发散。
- 与开发人员保持良好、准确的沟通。
- 细心、耐心。
- 不轻言放弃，即使开发人员也快失去了信心，你也不要放弃。
- 自信，有些低水平的开发人员分析定位问题的能力比测试人员要弱。

4.3 优秀性能测试工具推荐

4.3.1 LoadRunner

Mercury LoadRunner 是一种预测系统行为和性能负载测试工具。通过模拟上千万用户实施并发负载及实时性能监测的方式来确认和查找问题，LoadRunner

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

能够对整个企业架构进行测试。通过使用 LoadRunner，企业能最大限度地缩短测试时间、优化性能和缩短应用系统的发布周期。

目前企业的网络应用环境都必须支持大量用户，网络体系架构中包含各类应用环境且由不同供应商提供软件和硬件产品。难以预知的用户负载和越来越复杂的应用环境使公司时时担心会发生用户响应速度过慢、系统崩溃等问题，这些都不可避免地会导致公司收益的损失。Mercury Interactive 的 LoadRunner 能让企业保护自己的收入来源，无须购置额外硬件而最大限度地利用现有的 IT 资源，并确保终端用户在应用系统的各个环节中对其测试应用的质量、可靠性和可扩展性都有良好的评价。

LoadRunner 是一种适用于各种体系架构的自动负载测试工具，它能预测系统行为并优化系统性能。LoadRunner 的测试对象是整个企业的系统，它通过模拟实际用户的操作行为和实行实时性能监测，来帮助测试人员更快地查找和发现问题。此外，LoadRunner 能支持广泛的协议和技术，为特殊环境提供特殊的解决方案。

1. 轻松创建虚拟用户

使用 LoadRunner 的 Virtual User Generator，能够很简便地创立系统负载。该引擎能够生成虚拟用户，以虚拟用户的方式模拟真实用户的业务操作行为。它先记录下业务流程（如下订单或机票预定），然后将其转化为测试脚本。利用虚拟用户，可以在 Windows、UNIX 或 Linux 机器上同时产生成千上万个用户访问，所以 LoadRunner 能极大地减少负载测试所需要的硬件和人力资源。另外，LoadRunner 的 TurboLoad 专利技术能提供很高的适应性，TurboLoad 可以产生每天几十万名在线用户和数以百万计的点击数的负载。

用 Virtual User Generator 建立测试脚本后，可以对其进行参数化操作，这一操作能让你利用几套不同的实际发生数据来测试应用程序，从而反映出本系统的负载能力。以一个订单输入过程为例，参数化操作可将记录中的固定数据如订单号和客户名称，由可变值来代替。在这些变量内随意输入可能的订单号和客户名，来匹配多个实际用户的操作行为。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

LoadRunner 通过它的 Data Wizard 来自动实现其测试数据的参数化。Data Wizard 直接连接数据库服务器，从中可以获取所需要的数据（如订单号和用户名）并直接将其输入到测试脚本。这样避免了用人工来处理数据，Data Wizard 为你节省了大量的时间。

2. 创建真实的负载

建立虚拟用户后，需要设定负载方案、业务流程组合和虚拟用户数量。用 LoadRunner 的 Controller，能够很快地组织多用户的测试方案。Controller 的 Rendezvous 功能提供一个互动的环境，在其中既能建立持续且循环的负载，又能管理和驱动负载测试方案。而且，可以利用它的日程计划服务来定义用户在什么时候访问系统以产生负载，这样就能将测试过程自动化。同样还可以用 Controller 来限定负载方案，在这个方案中所有的用户同时执行一个动作——如登录到一个库存应用程序来模拟峰值负载的情况。另外，还能监测系统架构中各个组件的性能，包括服务器、数据库、网络设备等，来帮助客户决定系统的配置。

LoadRunner 还通过它的 AutoLoad 技术，为你提供更多的测试灵活性。使用 AutoLoad，可以根据目前的用户人数事先设定测试目标，优化测试流程。例如，目标可以是确定应用系统承受的每秒点击数或每秒的交易量。

3. 定位性能问题

LoadRunner 内含集成的实时监测器，在负载测试过程的任何时候，都可以观察到应用系统的运行性能。这些性能监测器为你实时显示交易性能数据（如响应时间）和其他系统组件，包括 Application Server, Web Server、网络设备和数据库等的实时性能。这样，就可以在测试过程中从客户和服务器的双方面评估这些系统组件的运行性能，从而更快地发现问题。利用 LoadRunner 的 ContentCheck TM，可以判断负载下的应用程序功能正常与否。ContentCheck 在虚拟用户运行时，检测应用程序的网络数据包内容，从中确定是否有错误内容传送出去，它的实时浏览器帮助你从终端用户角度观察程序性能状况。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

4. 分析结果

一旦测试完毕后，LoadRunner 收集汇总所有的测试数据，并为你提供高级的分析和报告工具，以便迅速查找到性能问题并追溯缘由。使用 LoadRunner 的 Web 交易细节监测器，可以了解到将所有的图像、框架和文本下载到每个网页上所需要的时间。例如，这个交易细节分析机制能够分析是否因为一个大尺寸的图形文件或第三方的数据组件造成应用系统运行速度减慢。另外，Web 交易细节监测器分解用于客户端、网络和服务端上端到端的反应时间，以便于确认问题，定位查找真正出错的组件。例如，可以将网络延时进行分解，以判断 DNS 解析时间、连接服务器或 SSL 认证所花费的时间。通过使用 LoadRunner 的分析工具，能很快地查找到出错的位置和原因并作出相应的调整。

5. 最大化投资回报

Mercury Interactive 的所有产品和服务都是集成设计的，能完全相容地一起运作。由于它们具有相同的核心技术，来自于 LoadRunner 和 ActiveTest TM 的测试脚本，在 Mercury Interactive 的负载测试服务项目中可以被重复用于性能监测。借助 Mercury Interactive 的监测功能——Topaz TM 和 ActiveWatch TM，测试脚本可以重复使用从而平衡投资收益。更重要的是，能为测试的前期部署和生产系统的监测提供一个完整的应用性能管理解决方案。

4.3.2 SmartBits

SmartBits 是数据通信业界广泛认同的对于网络设备及网络性能测试的标准测量仪表，在北美、欧洲和亚洲被广泛地用于政府、大学科研机构、各大网络通信公司、运营商等相关组织中。在中国，SmartBits 用户遍及全国，成为网络设备从业人员不可缺少的工具之一，在同类产品中具有绝对优势。使用 SmartBits 可以帮助用户测试设备及网络性能，如吞吐量、延迟、丢包等，更可以在一个端口中模拟上千万个网络的数量，还可以对其各自的性能进行分析，测试出不同的 QoS 下不同流量的表现。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

SmartBits 提供丰富的接口，包括 10M USB 接口、10M/100M/1000M 以太网接口、E1/25M/Oc3/Oc12ATM 接口、Oc3/Oc12/Oc48/Oc192POS 接口和 1G/2G Fiber Channel 接口，并支持不同接口类型之间的互通测试，支持多用户操作，不同的操作人员可以使用同一机箱上的不同板卡，从而提高了测试设备的使用率。SmartBits 还可以通过机箱之间的扩展达到超过 1000 端口的测试密度，而且支持远程 GPS/CDMA 的时钟同步系统进行精准的异地延迟等高要求的测量，整个系统有良好的扩展性。

SmartBits 提供了丰富的测试软件，用户可以通过安装了测试软件的 PC (Windows 9X/NT/2000/XP) 控制 SmartBits 设备，这种控制既可以是本地的，也可以来自远程。同时 SmartBits 提供灵活的编程接口，支持 Tcl 和 C 语言等。用户既可以选择使用功能灵活的软件自定义测试方法与网络拓扑，实现任意功能或性能测试，也可以使用基于标准的定制软件进行基准测试。

SmartBits 的测试解决方案非常多，与路由设备相关的测试方案包括：利用 SmartBits 的 SmartApplications 测试软件进行 RFC 2544 自动化测试；利用 SmartFlow 进行基于多流的吞吐量、平均延迟、延迟分布、定长时间延迟测试及帧丢失测试；使用 SmartMulticast IP 进行组播性能测试。考虑到测试人员的实际需求，思博伦通信公司开发了基于 Tcl/Tk 语言的测试脚本开发工具，用户可以方便地编写定制的测试程序，这样可以大大简化测试的过程和缩短测试的时间，同时也使具有特殊要求的一些测试需求得到满足。

4.3.3 IXIA

美国 IXIA 公司（纳斯达克上市公司，代号 XXIA）是高性能 IP 网络测试解决方案的全球一流提供商，100%专注于开发 IP 测试解决方案。IXIA 的统一化测试平台配合相应的测试工具可以实现完整的 2~7 层测试方案，能够实现的测试主要有：

- 2~3 层转发设备的流量分析与测试；
- 路由器、交换机、MPLS 测试；
- Edge Router/Switch/DSLAM 组播测试；

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

- 电信级以太网（Carrier Ethernet）测试；
- 宽带接入设备（BRAS、DSLAM）测试；
- PON 系统性能和业务验证测试；
- 信息安全产品（防火墙、IDS/IPS、垃圾邮件网关和 NAC 认证设备等）测试；
- IPSec 网关性能测试；
- SSLVPN 测试；
- IPTV 设备和网络测试；
- 语音/IMS 性能测试；
- 新业务验证及网络端到端测试；
- 各种应用服务器性能测试；
- DPI（深度数据包检测）设备测试；
- 家庭网关/E-home 终端测试；
- WLAN 产品测试；
- Wi-Fi 手持终端性能测试；
- 协议一致性测试（Conformance Test）；
- 自动化测试。

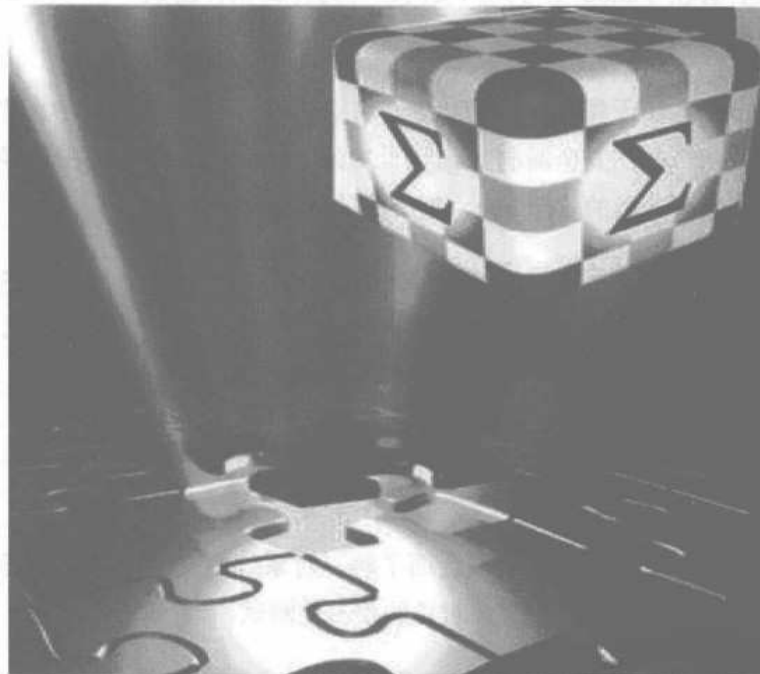
IXIA 产品得到用户广泛的认可，主要得益于产品所具有的特点和优势：

- 真正的 2~7 层测试平台：测试模块的每个端口在产生 2~3 层流量（包括协议和普通流量）的同时，还可以产生具有状态的（Data、Voice、Video）4~7 层应用层的流量。这是 IXIA 平台的最大特点。
- 基于端口的多用户操作：每个端口都有独立的 CPU 和内存，性能指标高并且可以大大提高使用率。这是 IXIA 产品在技术上的最大优势。
- 开放式结构和自动化操作：每个测试端口都运行有标准的 Linux 协议栈，可以根据客户的需要方便地集成新技术，并提供所见即所得的自动化测试脚本，这样可以使 IXIA 测试平台的使用率最大化。
- 高端口密度并行稳定测试：IXIA 产品已经在用户实验室得到验证，同时支持 1500 个 GE 端口的测试；一个机框最多 40 个用户同时执行不同的操作可以长时间稳定运行。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

第5章 安全测试技术



为什么说黑客是高级的软件测试人员，本章将为读者奉献一个在互联网上广为流传的一个中国黑客高手的故事，来体会测试技术与黑客技术本质上的相似性。同时，本章还会告诉读者产品安全测试应该包含哪些内容，以及如何开展安全性测试。并推荐在一些领域较好的安全性测试工具，以供大家研究了解世界最新安全性测试技术的趋势。

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

5.1 终极测试人——黑客

5.1.1 黑客本质

黑客本质是世界上最厉害的软件测试高手，因为他能找到经过若干测试人员测试后未能找到的产品 bug。

作者曾在 Yahoo 上看到网友问：“为什么软件测试工程师需要懂一些黑客的技术？”

一方面，由于黑客人群的壮大，现在的软件产品会受到许多黑客的攻击，他们试图找到产品的安全漏洞 bug，因此如果软件测试工程师能在黑客前先用他们的攻击方法找到产品的安全 bug，则产品的质量不是更高吗？

另一方面，黑客的攻击技术也是软件测试技术的一个领域。学习黑客的技术和思想，也就是多学习一种软件测试的技术和思维，对于提升我们自己的软件测试水平也是百利无一害。

黑客：基于对系统的猜测，依据经验进行排雷式探索性攻击，具有很大的盲目性。

测试人员：基于对系统的全面了解，对系统的安全性进行验证确认，有的放矢地对可能的隐患进行更有针对性的攻击性测试。

因此仅针对被测试系统来说，测试人员的测试比黑客更全面，也更有针对性，同时更有效率。

5.1.2 知名的测试黑客的故事

下面是中国一个知名的优秀软件测试黑客的故事。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

发现了一个黑客，在中国！

“我们发现了一个黑客，在中国。”

2003年1月的一个早上，张亚勤来到办公室，打开电子邮箱，这一行字立即弹出来。

邮件来自微软公司总部的安全小组。该小组的职责之一是监视因特网浏览器在全球的运行情况，专门寻找“臭虫”，然后弥补，同时还要监视网络上神出鬼没的“黑客”行踪。

对于微软公司来说，网上“黑客”的性质是不同的。有些“黑客”的确对微软抱着敌意，专门寻找微软软件产品中的“臭虫”，利用软件本身存在的漏洞，去攻击那些软件使用者。这种攻击带有极大的破坏性，还让微软公司难堪。另外一些“黑客”则纯粹属于“技术狂”，他们专找大公司产品的毛病。

“可以肯定他是一个中国的学生。”他们告诉张亚勤，“在湘潭大学读书，名叫刘蝶雨。”

湖南·湘潭大学·刘蝶雨

刘蝶雨坐在他的拥挤不堪的房间里，盯着电脑屏幕，目不转睛，已经好几个小时一动不动。上面是微软公司最得意的产品之一——“因特网浏览器”。他知道他想找的东西就在那里，他已经清晰地感觉到它的存在，可它为什么还不出来呢？……

他一边想，一边用手指敲击键盘。就在那一瞬间，“因特网浏览器”出了麻烦。

他咧开一嘴大板牙，嘿嘿笑了：“我破坏了它的规则，是不是？”

他身边有不少恨微软的人，看了他做的事情，不免大快人心。开心完了，又在猜测微软那边的动向。“真奇怪！”有个同学对他说，“微软还没有对你采取行动啊？”

正说笑着，电话铃声响了。他拿起来，就听那边有个声音说：“我是微软的。”

刘蝶雨当即目瞪口呆。

“也许你愿意和我们合作。”

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

电话这边真是微软的，叫林斌，是微软亚洲研究院新技术开发部的经理。他接到张亚勤转发的邮件，按图索骥，很容易地找到了刘蝶雨。

“我们找你，没有什么特别的。”林斌感觉到对方的紧张，希望缓解气氛。

对方“啊”了一声，还是不说话。

林斌似乎想起了什么，赶忙自我介绍。说自己在北京，在微软亚洲研究院负责一个工程师小组。

“啊，啊。”

“我只是微软的一个技术人员。”

“啊，啊。”

“我们的院长是张亚勤。他是世界一流的计算机科学家。”

“啊，啊。”

“我曾经参加了视窗 2000 的制作。”

“那很酷。”对方终于说话了。

“我们对你的工作非常感兴趣。”

“啊。”刘蝶雨又不说话了。

“我们很想请你到北京来看看。也许你愿意和我们合作。”

“啊……”

“你寒假如果没有事的话就来吧。坐飞机坐火车都行，我们负责你的旅行费。”

“我很愿意。”刘蝶雨说，“也许吧。”

“他们是微软的吗？”

刘蝶雨放下电话，心里还在咚咚跳。他的第一个念头是：“这年头骗子太多啦。他们是微软的吗？”接着又一个念头：“就算他们是真的，到底为什么找我呢？该不是要把我关起来吧？”

刘蝶雨这样想着，第一次感觉到“黑客帝国”的惊险离奇。

实际上他不喜欢别人叫他“黑客”。“我没见过黑客。”有一次他这样说，“有的人把人家的网站都弄垮，叫黑客；有的人技术很好，也叫黑客。黑客的意思太多了，如果你认为黑客也是好人，那我就是黑客。”

说老实话，他完全是在不经意中走上这条路的，只是带着几分好奇、几分兴趣，还有几分想要证明自己能力的欲望。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

事情是从2002年6月30日开始的。那一天，他偶然看到一篇文章，是一个外国人写的，举了不少例子说明怎样发现一个程序中的错误。很多在刘蝶雨看来非常困难的事情，顿时变得简单了，就像文章作者说的，“并不需要你有多么高深的知识和多高明的技巧，你只要从特殊的角度去看它。”

“我也一定能做到。”刘蝶雨这样想。

他的思路屡试不爽，他在微软的“浏览器”上到处找，不停地找到“臭虫”，但是微软也在找他。现在，微软终于找到他了。

北京·微软亚洲研究院

林斌放下电话，就去找研究院人力资源部的经理王谨，请她安排刘蝶雨在寒假期间到微软亚洲研究院来实习。

“这些都可以是假的，我只认电子邮件。”王谨觉得自己似乎不能说服刘蝶雨，只好转回头来找林斌：“这个孩子怎么这么有意思啊？他问你是不是微软的？”

林斌的电话再次来到湘潭：“你相信电子邮件，那很好，我们的电子邮件地址有‘微软’字样，这不会有假吧？”

春节过后，刘蝶雨终于来到希格玛大厦。

林斌致电微软总部，说他们要找的“黑客”现在就在他身边。

“太棒了。”对方兴奋地说，“我们在一起开个会吧。”

“你们想要我做什么？”

林斌对他说：“你想做什么，就做什么。你如果想继续在‘浏览器’上找‘臭虫’，我们总部的产品组可以和你合作。你找到漏洞，他们马上弥补，打包，然后再给你。你能得到最新的程序，还没有公布的。”

两个人正在说着，刘蝶雨的手机响了。那是他父亲从湘潭打来的。

“没问题，我很安全。”刘蝶雨对着听筒说，又抬起头看着林斌：“我觉得你们微软和外面说的不大一样。”

刘蝶雨开始工作了。连续7天，他在“因特网浏览器”上找到7个“臭虫”，一天一个，而且都是很难找到的。他自己很兴奋，微软总部的人更兴奋，惊叹“这个人怎么这么厉害”。

“他的工作简直太好了。”林斌说。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

第八天，林斌给了刘蝶雨一本书，是专门讲怎样编写安全代码的。作者是微软公司产品部门的一个经理，在书中列举了程序员常犯的错误，极为精密周到，所以这本书成为微软程序员的必读书。

林斌说：“读读这本书，你能更好地发现程序员的弱点在哪里。”

刘蝶雨大喜，拿回去看了第一章，脑子里马上有了新主意。他试图以其人之道还治其人之身，直接深入到书的作者领导的那个小组中去。

林斌听了他的想法，觉得不会有结果：“书是他自己写的，他不会在他领导的小组里犯错误。”

但是刘蝶雨更相信自己的直觉。

接下来的事情，让所有人都感到意外。刘蝶雨在“浏览器”的地址栏里发起了他的“战役”。当他使用一种方式来表达一个字母的时候，“浏览器”没有任何问题。他继续扩张到第二种方式，仍然没有问题。他一口气做下去，不断扩张出新的表达方式，字符串延伸到第81次，系统的毛病显示出来，他笑了，再接再厉，扩张出至少200种变化，字符串形成前所未有的长度。结果发现，从第81种变化开始，一直到第100种，系统都会出现问题，最后他莫名其妙地进入一个银行的网页。

“真是太绝了。”林斌说，“别人的测试，只变化十几次、二十次，已经不得了。实际上他的变化是从0到200多次，就是在中间一个短暂的阶段，他发现了问题。问题报告总部，那边的人佩服得一塌糊涂。”

“这种测试，我们根本想象不到。”微软总部的安全小组回电说。

相信读者看完这个故事，会直观地认识到原来软件测试活动本身与黑客的活动本身居然有着如此的异曲同工之处啊。

5.1.3 安全测试应该包含什么

不过真正的安全测试的范围要比黑客的行为广，黑客攻击只是安全性测试的一个子集。那么安全测试应该包含哪些内容呢？目前安全性测试分为主机端测试和网络电信设备类测试。

- 主机端的安全性测试主要针对数据库、网页、网站服务器、Windows 操作

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

系统等的测试。

- 网络电信设备类的安全性测试主要进行拒绝访问服务攻击、bypass 攻击、协议畸形报文攻击等。

这些攻击存在的价值来源于软件设计上对异常场景和负面状况没有做到充分的“防备”。通过这些攻击可以使得产品设计中遗漏的异常处理缺陷提前暴露，帮助大大提高产品的健壮性。

以美国某公司安全性测试方案为例：

第一步，收集已知的各种攻击手段来测试产品的安全性。

第二步，根据进一步的安全性测试需求，开发相关的安全性攻击工具继续开展安全性测试。

第三步，由经验丰富的测试工程师根据自己产品实现的特点，提前预演一些黑客攻击手法。

软件测试厂商核心安全技术公司 CST (Core Security Technologies Inc) 透露，运行在苹果电脑 Mac OS X 操作系统上的 iCal 日历应用程序存在 3 个致命的安全漏洞，给 Mac 用户造成了严重的后果。CST 的测试报告指出，这 3 个安全漏洞中最严重的一个是由资源库软件瑕疵引起的内存破坏造成的，潜在的攻击者（黑客）能够利用这个安全漏洞特别制作恶意的日历文件；该恶意日历文件进而利用其他两个安全漏洞造成 iCal 程序崩溃。同时，攻击者（黑客）可以利用这些漏洞轻松入侵到 Mac 的授权用户，并对这些计算机进行控制。

对此，公司高级测试工程师托马斯表示，该程序出现安全漏洞的主要原因是测试不足导致的。

一些常见的主机端安全性问题如下：

- SQL Injection (SQL 注入)

SQL 注入是最常见的攻击方式，它的主要原理是：攻击者通过改变 Web 页的

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

参数（如 GET/POST 数据或 URLs）直接将 SQL 片段提交到服务器，并在服务器端执行的过程。

- Cross-Site scripting (XSS) (跨站脚本攻击)

XSS 定义：由于 Web 程序没有对用户提交的 HTML 内容进行适当的转译，这样攻击者就可能在你的 Web 页中插入一些 HTML 语句，这些语句通常以 <script> 标记的形式出现，攻击者通常使用 XSS 攻击来窃取 Cookies 和 Session 信息，或者欺骗用户将隐私信息暴露给错误对象（又称为钓鱼）。

- Cross-Site Request Forgery (跨站请求伪造)

CSRF：在 Web 页或在给用户发送的邮件中插入恶意代码（通常是链接或脚本），比如发送一个带有银行取款链接的图片或脚本（通常是 HTML 或 JavaScript），当用户访问这个图片时，页面加载图片过程会隐密地链接到一个远程页面，这个页面会自动向目标站点发起请求，如果这个目标站点仍保留这个用户的 Cookie 信息，并且这个 Cookie 未过期，那么攻击者就可以在用户不知情的情况下以用户的身份登录银行或执行取款操作。CSRF 的特性就是利用网站对用户标识的信任，欺骗用户的浏览器发送 HTTP 请求给目标站点。

- E-mail Header Injection (邮件标题注入)

邮件标题注入与 SQL 注入的原理类似，它的原理是：通过在 E-mail 的 Subject 中输入一些特殊语句如“\n”，攻击者可以利用这个缺陷通过邮件服务器发送垃圾邮件。

- Directory Traversal (目录遍历)

目录遍历是另一种注入类型的攻击，攻击者欺骗文件系统读或写服务器不允许操作的文件。

除了主机端的安全性测试外，我们为什么还需要对网络电信类设备进行安全性、健壮性测试呢？

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

网络高速发展，新技术、新应用不断涌现，不断给我们带来惊喜，同时也在越来越深入地改变着我们的生活，网络已经渗透到了生活的每一个角落。正因为如此，网络的安全性也越来越受到人们的重视。在网络安全领域，病毒、黑客入侵、DoS 攻击等概念已经深入人心，但是对于另外一些威胁网络安全的因素，人们还未给予足够的重视，这就是网络设备的安全漏洞。Windows 操作系统的安全漏洞大家比较熟悉了，但网络设备（比如路由器、以太网交换机等）、网络安全设备（防火墙、UTM 等）以及服务器（Web 服务器、应用服务器、固网和移动网络的核心网设备也可以归于这一类）等也经常存在安全漏洞，这些安全漏洞可能会导致网络设备性能下降、拒绝服务甚至瘫痪。

我们知道，现有的网络通信协议非常多（IETF 发布的 RFC 目前已经有 4000 多个），而且有很多协议都相当复杂，目前的网络设备往往支持十几种，甚至是几十种网络协议，出于协议本身，以及各厂商对协议理解程度等原因，网络设备在协议的处理上，难免会存在诸多没有考虑到的情况，这种没有考虑到的情况就是漏洞 bug，这些漏洞如果被人利用并实施攻击的话，可能会造成网络设备的异常甚至瘫痪。如果受到攻击的设备处于网络的关键位置，造成的损失将无法估量。

常规的网络电信设备安全性测试有如下两种方式：

(1) 针对某些特定攻击的测试

如 DDoS 攻击、ARP 类攻击的测试，这类测试往往是通过模拟一些异常的数据流量来对设备进行测试，以验证设备对这类攻击的抵抗性。

(2) 针对设备安全漏洞的测试

这类测试一般是通过工具对已公开的漏洞进行扫描；或者通过安全专家，利用简单的工具来构造一些攻击，以检测设备对这些攻击的抵抗性。

第 2 种方式实质上是通过对协议报文进行变异，将这些异常的报文发送给被测设备，以检测设备在收到这些异常报文时的应对能力。

最后为了让读者直观地意识到网络设备漏洞问题的普遍性，作者特意为大家收集了网络设备公司即美国思科公司已公开的产品安全漏洞。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

VU#744249 01/08/2007 Cisco Secure Access Control Server vulnerable to a stack-based buffer overflow via a specially crafted "HTTP GET" request
 VU#778648 11/01/2006 Cisco Security Agent Management Center vulnerable to authentication bypass
 VU#123140 09/20/2006 Cisco products contain hard-coded SNMP values
 VU#821420 09/13/2006 Cisco IOS fails to properly handle summary packets in the VLAN Trunking Protocol
 VU#542108 09/13/2006 Cisco IOS contains buffer overflow in VTP VLAN name handling
 VU#175148 09/13/2006 Cisco IOS fails to properly verify the VTP configuration revision number
 VU#642076 09/20/2006 Cisco Intrusion Prevention System administration interface fails to properly handle Secure Socket Layer packets
 VU#658884 09/20/2006 Cisco IPS fails to properly check fragmented IP packets
 VU#236045 09/07/2005 Cisco IOS Firewall Authentication Proxy vulnerable to buffer overflow via specially crafted user authentication credentials
 VU#472582 01/26/2005 Cisco IOS IPv6 denial-of-service vulnerability
 VU#697049 04/23/2003 Cisco Secure ACS for Windows CSAdmin vulnerable to buffer overflow via login requests
 VU#463944 12/20/2000 Cisco Catalyst reboots in response to an SSH "protocol mismatch" error
 VU#486224 07/21/2004 Multiple Cisco ONS control cards fail to properly handle malformed UDP packets
 VU#721092 01/21/2004 Cisco IBM Director agent does not properly handle arbitrary TCP packets to port 14247/tcp
 VU#362483 11/28/2001 Cisco IOS Firewall Feature Set fails to check IP protocol type thereby allowing packets to bypass dynamic access control lists
 VU#352462 12/10/2003 Cisco ACNS contains buffer overflow vulnerability in the authentication module when supplied an overly long password
 VU#310387 02/27/2002 Cisco IOS discloses fragments of previous packets when Express Forwarding is enabled
 VU#760432 07/21/2004 Cisco Transaction Language 1 (TL1) interface fails to properly validate accounts with blank passwords
 VU#904310 04/08/2004 Cisco IPsec VPNM vulnerable to DoS via malformed IKE packet
 VU#968187 01/09/2002 Cisco SN 5420 Storage Router vulnerable to DoS via HTTP request containing long headers

 VU#495275 03/27/2002 Cisco CallManager contains memory leak
 VU#800384 07/21/2004 Multiple Cisco ONS control cards fail to properly handle malformed TCP packets
 VU#579324 07/31/2003 Cisco IOS HTTP Server vulnerable to buffer overflow when processing overly large malformed HTTP GET request
 VU#277048 07/21/2004 Multiple Cisco ONS control cards fail to properly handle invalid TCP responses
 VU#443257 04/24/2003 Cisco Catalyst switches allow access to "enable mode" without password
 VU#602734 01/21/2004 Cisco default install of IBM Director agent fails to authenticate users for remote administration
 VU#969344 07/21/2004 Multiple Cisco ONS control cards fail to properly handle malformed IP packets
 VU#812515 06/27/2001 Cisco IOS HTTP server authentication vulnerability allows remote attackers to execute arbitrary commands
 VU#162451 04/20/2004 Cisco IOS fails to properly process solicited SNMP operations
 VU#548968 07/21/2004 Multiple Cisco ONS control cards fail to properly handle malformed SNMP packets
 VU#810062 02/03/2004 Cisco 6000/6500/7600 series systems fail to properly process layer 2 frames
 VU#411332 07/16/2003 Cisco IOS Interface Blocked by IPv4 Packet
 VU#855195 01/09/2002 Cisco SN 5420 Storage Router vulnerable to DoS via fragmented packet sent over Gigabit interface
 VU#26825 07/11/2000 Cisco Secure PIX Firewall TCP Reset Vulnerability
 VU#106392 05/10/2001 Cisco IOS vulnerable to DoS via unrecognized transitive attribute in BGP UPDATE
 VU#918920 07/21/2004 Multiple Cisco ONS control cards fail to properly handle malformed ICMP packets
 VU#245190 06/09/2004 Cisco CatOS TCP ACK handling vulnerability
 VU#174248 04/04/2001 Cisco Content Services Switch (CSS) permits non-privileged user to enter debug mode
 VU#127545 04/25/2007 Cisco NetFlow Collection Engine contains known default passwords
 VU#544484 06/28/2006 Cisco Access Point Web Browser Interface contains a vulnerability
 VU#853540 11/22/2005 Cisco PIX fails to verify TCP checksum
 VU#562945 11/02/2005 Cisco IOS heap integrity checks are insufficient
 VU#959203 12/27/2002 Cisco IOS OSPF neighbor IO buffer overflow

软件测试精要

每月及時觀看電子月刊書籍
 就上溜客安全網 www.176ku.com

5.2 优秀安全性测试工具推荐

5.2.1 针对 Web 安全性的工具

Wapiti 是一个开源的安全测试工具,用于 Web 应用程序漏洞扫描和安全检测,可到 <http://sourceforge.net/projects/wapiti/> 下载。

Wapiti 是用 Python 编写的脚本,使用它需要 Python 的支持,也就是说,要先安装好 Python。

Wapiti 执行的是“黑盒”方式的扫描,也就是说,直接对网页进行扫描,而不需要扫描 Web 应用程序的源代码。Wapiti 通过扫描网页的脚本和表单,查找可以注入数据的地方。Wapiti 能检测以下漏洞:

- 文件处理错误。
- 数据库注入(包括 PHP/JSP/ASP SQL 注入和 XPath 注入)。
- XSS(跨站脚本)注入。
- LDAP 注入。
- 命令执行检测(例如,eval()、system()、passtru()等)。
- CRLF 注入。

之所以把 Wapiti 称为轻量级,是因为它的安全检测过程不需要依赖漏洞数据库,因此执行的速度会更快些。

5.2.2 针对设备的安全性工具

Mu-4000 是美国 Mu Dynamics 公司(原 Mu Security 公司)研制的一款网络安全测试分析设备,该设备可对网络设备提供基于协议的安全分析测试。作为一款基于协议的安全测试系统,是业界第一款系统化的和可重现识别任何基于 IP 设备的各种已知和未知的安全漏洞的产品,其测试方法不需要知道被测设备的任何源代码。Mu-4000 的测试方法从攻击方式看十分类似于黑客入侵,其分析报告基于强大的审计监督功能能够给网络设备测试人员带来极大的方便。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

Mu-4000 将被测设备置于成百上千的协议 Fuzzing 攻击手段下,同时监测设备的反应,并以可再现和可执行的方式提供故障的定位方法。这种安全分析的生命周期方法使得产品自身协议的安全性不断得以加固,可适用于产品研发和部署的各个阶段。

Mu Security 安全分析平台见图 5-1。



图 5-1 Mu Security 安全分析平台

1. 产品特点

系统化的安全分析在业界一直是个难题,脆弱性攻击测试中不可预期输入的范围和数量远比预期输入的数目多得多,然而安全分析不仅仅是设计攻击,对平台和协议都有要求,系统必须允许一种自动的、方法论的、系统的和可重复的方式来执行安全分析系统必须支持使用的不同模式。美国 Mu Security 公司正是基于

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

自己独特的“蜘蛛协议”（Spider Protocoling）技术，为业界提供了一种全新而领先的安全分析方法。

Mu Security 协议安全测试平台 Mu-4000 目前广泛支持 9 大系列的协议簇：核心网、多媒体、路由、企业网、城域以太网、SCADA、网络管理、组播、存储。

下面是支持的部分协议（包含但不限于）：

IPv4、UDP4、TCP4、ICMP4、ARP、FTP、TFTP、DHCP、HTTP、SNMP、IPv6、TCP6、UDP6、ICMP6、SSH、ISAKMP、SSL、RIP、RIPng、BGP、OSPF、PIM、IGMP、SMTP、POP3、IMAP、PPPoE、IEEE 802.1X、IEEE 802.1Q、VRRP、RADIUS、LDAP、TACACS+、SIP、MGCP、RTSP、SSDP、CIFS、NFS、DNP3

同时，除了支持不断更新的各个厂家漏洞安全公告（Published Vulnerabilities）和自己独特的变异攻击分析外，Mu-4000 还是一个很灵活可扩展的安全分析平台，它允许整合加入您自己的攻击分析工具和脚本，确保即时作出对新安全攻击的精确响应。

2. 应用场景

Mu-4000 可以对基于 IP 进行通信的多种网络设备进行安全性测试分析。Mu-4000 可以生成异常流量发往被测设备，以检测设备的安全性、健壮性。其原理图如图 5-2 所示。

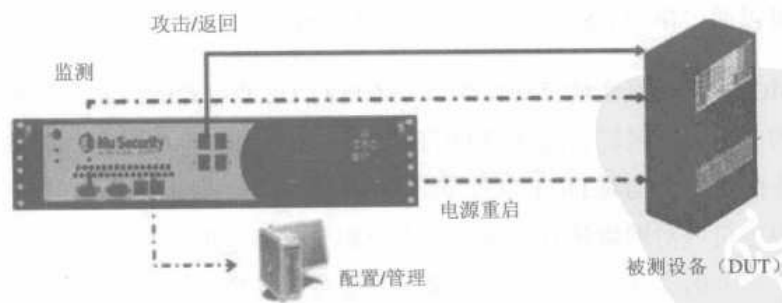


图 5-2 检测原理图

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

Mu-4000 产生的异常流量包括如下几类：

(1) 异常的协议报文

针对各种通信协议（目前 Mu-4000 支持 60 余种协议），Mu-4000 能够产生数以百万计的异常协议报文，这些异常报文从状态、结构、语义等角度对正常的协议报文进行“变异”，覆盖了各种可能性，能够全面检测 DUT（被测设备）对该协议处理上的安全性、健壮性。

(2) DoS（Denial of Service，拒绝服务）攻击流量

Mu-4000 能够产生多种 DoS 攻击流量（目前支持 40 余种类型的 DoS 攻击），用户可以配置攻击类型、流量模型（包括强度、持续时间、流量变化方式）、攻击报文的负载等，通过灵活的配置，可以模拟出各种 DoS 攻击，以检测设备对 DoS 攻击的抵抗能力。

(3) PVA（Published Vulnerability Attack，已公开漏洞攻击）流量

Mu-4000 还能够根据目前已公开的安全漏洞，产生相应的流量，检测设备是否已经修补了已公开的漏洞，以及对已公开漏洞攻击的防御能力如何。

Mu-4000 通过 DUT 对正常协议报文的响应情况来判定 DUT 的问题，与此同时，还可以通过监测设备的内部状态来更加准确地定位问题。对 DUT 内部状态的监测可以通过串口、SSH、Telnet 等方式实现。

Mu-4000 可以实现自动化测试，通过对 DUT 电源的控制，在碰到 DUT 不可恢复的问题时，可以自动重启 DUT，使测试继续运行（也可以通过事件触发，使用命令的方式自动对 DUT 进行重启）。Mu-4000 一般用来对单一网元设备进行测试，但也可以对网络进行测试，以衡量整个网络（服务）的安全性、健壮性。对于网络服务的测试，是在网络环境下对服务器（协议处理实体）进行的测试，其原理图如图 5-3 所示。

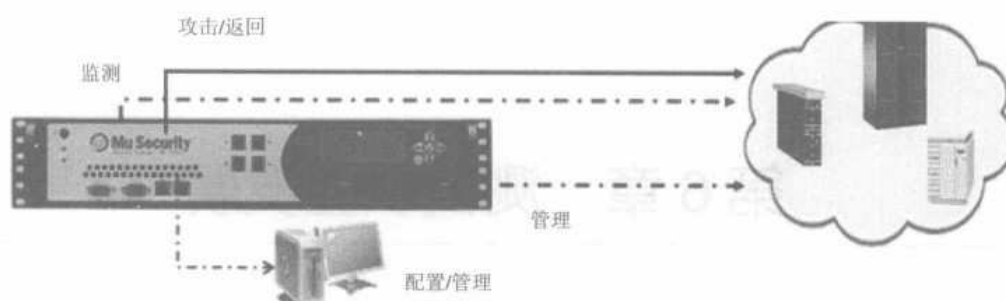


图 5-3 测试原理图

在对网络进行测试的时候，需要预先制定相应的保护措施，以避免测试引起的网络中断带来的损失。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

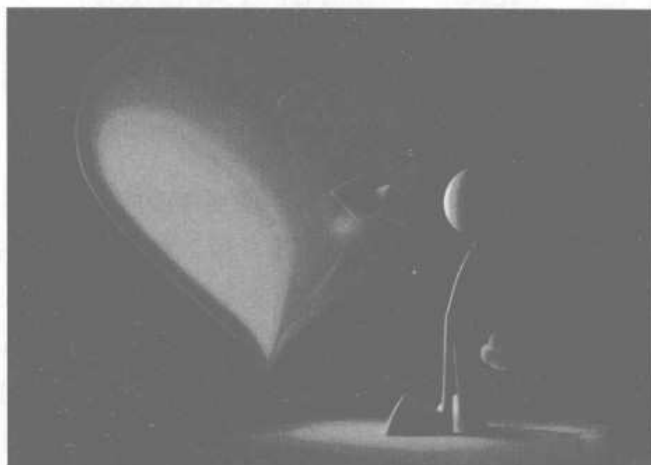
第6章 测试职业发展



当读者阅读完前面几章关于测试技术的内容后，相信不少从事测试的朋友能对软件测试有一个新的认识和理解，让自己重新对软件测试树立起新的兴趣和信心。但是测试朋友们对于更多非测试技术外的困惑，我们应该如何来解决。本章将重点与测试朋友们在非测试技术领域进行一些经验的分享，希望能帮助大家解除心中的一些疑问。本章将从一个测试人的思考角度出发，将如何规划测试从业者的职业发展和职业选择作为非技术疑惑的解惑开始。

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

6.1 选择——听你的心



本章不是和大家讨论选择测试作为自己职业的理由，笔者也不会给大家讲解从事软件测试有多么的美好，前途多么的光明，而是希望和大家在更宽泛的层面讨论我们选择职业的依据，重新审视自己是否应该选择测试作为自己人生路上的一个职业。如果你已从事了软件测试这一职业，但是觉得自己想离开这个职业，去看另一片新天地，笔者将会把自己身边测试同事转行的各种案例与大家分享，希望大家在进行测试转型的选择时，能少一些迷茫与恐惧，找到自己新的起点。

首先，问问我们自己职业选择的依据是什么？

你是如何进入测试这个行业的呢？得到最多的答案是：刚毕业时，没有其他岗位的选择，只有做测试；刚毕业时，为了进一家大公司，暂时委屈选择了做测试；受到媒体的鼓吹，进入了测试行业；不喜欢写代码，做测试简单些；被招聘官忽悠了，说干测试大有前途；进入公司后，服从公司的岗位调整等。

总的来说，大多数人在选择自己的第一份职业，甚至第二份职业时都是稀里

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

糊涂地选择的。在笔者工作四五年的时候，经常反思自己，我要选择的职业是什么？我自己想要的是什么？我下一步应该从事什么职业？一位美国 40 多岁的 IT 老前辈给了笔者一个建议：“听你的心，不要动脑去想。”这不正好与我们大家平时选择职业时，要求理性分析对比地选择方法相反吗？后来，笔者慢慢地明白了为什么要“听你的心”。因为，你的心才是你自己的核心价值观，只有当你的核心价值观得到了满足，你才能体会到幸福，体会到快乐。

这里的核心价值观，不是传统意义上的要事业成功，工资要达到几万，要当上经理，要发财；而是你不可动摇的快乐来源。拿笔者举例：在经历了 5 年的迷茫和面临多次工作选择后，终于明白了：我最核心的价值观和选择底线是什么了。我不是一个能长期过流浪生活的人，在成都的生活是我最快乐的；我不是一个喜欢冒险的人，因此我还是应该选择一个比较稳定的工作组织；我是一个喜欢 IT 技术的人，因此不要让我的工作内容离开 IT 技术；我是一个希望工作可以让生活快乐的人，所以，当面对新工作选择的时候，只要会影响我的最核心价值观，破坏我的快乐底线，那么我宁愿选择放弃。事实上，在不知不觉中，过去几年的选择都在不经意间为这些价值观服务。

希望大家能从自己快乐的角度整理出有自己特色的核心价值观。当然，如果目前还在为生存、生活所迫从事让自己不快乐的工作，也可以先潜伏其中，待机会成熟时再去追求“自己的心”。

以往笔者在选择职业和工作时除了听自己的心，即自己的核心价值观这一本质因素外，还会上网去看看相关的信息或向一些过来人请教工作的选择。现将笔者曾经参考过的一些观点与大家分享。

1. 选择在什么地方的什么组织做一份什么样的工作

在什么地方？

我们在选择职业时，常常忽视了工作地点的选择。很多人会为了一份看似很美好的工作到一个没有亲人、没有多少朋友、很陌生的地方工作。当一个人孤独

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

地在异地工作生活时，很可能需要更大的坚强来支撑熬过困苦，需要独自一人来分享快乐。笔者一直认为，没有生活的快乐和精神支撑，要想长期沉下心来专注事业，那需要非常大的坚强和毅力。因此，笔者建议大家寻求发展时，最好还是能在自己的家乡、自己的同学多的城市发展。这样，在精神上不会特别的孤单，生活也会愉悦很多。

什么组织？

本章后面的内容会专门谈到如何进行外企和民企的选择；大公司和小公司的选择；外包公司和自主产品公司的选择。

什么样的工作？

工作的内容，以及从事这份工作后对自己的生活、家庭、性格的影响，是否会破坏你的快乐来源。有的工作内容会让你常年过上流离漂泊的生活；有的工作内容会很艰难，但却可以让你从艰难中学到更多；有的工作内容会从此改变你的生活习惯，从而改变你的性格。无论什么样的工作内容，不能给你带来快乐的工作，是不会让你的这份职业长久下去的。

2. 是否自己工作起来整天提心吊胆的

这句话是一次笔者出现人生选择时，一位好友给的建议“是否自己工作起来整天提心吊胆的。”出现这种状态通常有两种可能：所工作的企业经营风险极大，说不定哪一天就破产，失业了；所从事的岗位，具有非常强的不稳定性，或担心未来是否还能有劳动力竞争力。对于大多数平凡人，如果你在选择工作时有这种感受，建议不要选择这样的工作。对于少部分想创业、想靠自己的一份力量得到超常规回报的朋友如能承受这样的风险和失败，也许真能心想事成。

曾经有一次，笔者与高中时代的一位好朋友聊天。在他材料学博士毕业前夕为他参谋选择就业方向，是出国还是留校？是留在母校还是去某某大学？通过深入分析内心的感受得知：选择出国的最大原因是在国外工作，可以马上实现年收入30万元人民币这一物质目标，其实是为了在同学中有面子，为了在家乡的亲人

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

中有面子。选择留校，虽然开始工资只有 3000 元，但是可以在自己熟悉的工作环境和生活了 9 年的城市中继续过着自己心灵满意的生活和工作状态。如果选择去某某大学，仅仅因为他是四川人，可以离父母更近一些而已，但却会失去自己所熟悉的城市的生活和如鱼得水的科研环境。

最后与大家共享一段关于工作、学习、兴趣三者关系的一些观点。

“工作就是一种愉快的带薪学习。只有我们认真地思索这句话才能够真正地理解工作和学习的关系，工作和自我职业发展的关系，工作和知识技能、工作与经验积累之间的关系。工作的过程就是学习的过程和自我知识积累的过程，只有提升自己的专业知识技能并真正转化为经验，自己的个人价值才能够被认可。在工作中不思索、不学习，那么本来开始从事的是一种脑力劳动，到后面全部转化为一种重复性的体力劳动，工作后期完全无任何经验积累，反而知识在不断地退步。

不重视自己的工作，自己都看不起自己的工作，将失去工作的兴趣，失去在工作中学习和持续改进的兴趣。为了工作而工作，自己被企业所束缚，无发展的兴趣和源动力。或者眼高手低，好高骛远，当前工作没有做好或者基础没有打好就期待更高层的工作，这样最终将跌得很惨。不要把自己的关注点仅仅放在工作所涉及的内容上，工作内容往往只是一个完善的知识体系的一部分。应该多关注工作所涉及的业务领域和技术领域，多关注领域发展趋势和前沿论点，多注意平时的资料收集和积累，多参加各种培训和实践活动巩固知识。周期性地整理自己的知识体系，使知识和流程都能够系统化和结构化，多写工作总结和学习心得，多思考可以改进的过程，多尝试和实践新方法。专业基础扎实，知识面宽广，两者都显得很重要。”

最后希望通过这些结合了作者自己的经验感言与其他网友的经验感言，能对本书的读者起到一定的抛砖引玉的作用，帮助大家减少一些工作机会选择时的恐惧与犹豫，少走弯路。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

6.2 职业选择的象限



按照《穷爸爸，富爸爸》书中的说法：职业选择通常有 4 个象限，即雇员、自由职业者，做生意/创业者和投资者。作为软件测试行业的从业者，很显然大多数都属于雇员这一象限，可能不到 1% 的测试人能做到自己创业，开创一些第三方测试机构，或者一些测试外包公司、测试工具销售公司、测试咨询服务公司。

也许是经历的缘故，笔者个人不太赞同通过测试来创业或成为自由职业者。因为通过笔者的工作经验体会出，以测试作为创业和成为自由职业者的技能将非常难获得足够的商业机会，并取得成功。毕竟测试是一个服务性行业，它需要和开发密切联系在一起，两者是需要密切沟通配合的，才能做好各自的工作。离开了开发，测试本身就失去了所服务的对象和价值。

那么对于 99% 的测试人是否永远都只能处在雇员这个象限？答案是“测试人可以同时处在雇员和投资者两个象限”。笔者对“投资者”的理解不仅仅限于手拿资金投资的人，或是进行风险投资的 VC 们。“投资者”的含义还应该包含如下两种情况：

- 第一种情况：将我们自己的职业生涯长期地投资到一些能取得高速成长的企业，并能与企业一起分享成长所带来的高额回报。
- 第二种情况：尽早进行一定的个人资金理财。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

 举例

第一种情况的案例是非常普遍的：一位测试工程师通过加盟一家企业，并长期坚持服务于这家企业，最后靠企业的股票或期权获得较多财富回报的案例在如今的中国是非常多的。这样的方式对于广大的普通测试工程师而言，基本上算是最好的发财致富成功的好方法了。我们唯一需要做的是，擦亮眼睛，寻找这样的潜力股。

虽然，第一种情况的回报会非常大，但同样风险也比较大。有很多测试工程师靠公司成功后股票发财，可也有更多的测试工程师由于公司创业的不成功，在付出了几年的机会成本和时间成本后，手中的股票或期权却成为了一张废纸。那么对于一部分不愿意去冒这些风险的测试工程师，又该如何选择自己的成功象限呢？第二种情况似乎是最好的选择，尽早进行一定的个人资金理财。通过把低风险的测试职业和高风险的投资相结合，两者取其平衡。理财投资让那些喜欢待在大公司的测试工程师或是厌恶风险的测试工程师也有了一定致富的可能，而测试工作所带来的稳定收入又可以抵御一定的投资风险，让你有了投资失败后东山再起的本钱。

笔者曾在一家外企工作过，在这家企业服务期间，接触到一些来自美国硅谷的测试同行。他们大多做了十多年，甚至二十年的工程师，有的人有好几百万美元的家产，有的人甚至拥有了上千万美元的家产。他们的这些主要收入来源就是靠工程师的工资和投资股票的收益，目前过着既健康、有规律，又比较富裕的生活。其中一位测试工程师的故事更加传奇：他在1990年加入思科后，依靠思科的期权拥有了第一桶金。在工作之余做了天使投资，投了另一家高科技企业，7年后这家高科技企业帮他赚了100倍以上的投资回报。现在这位老工程师已专职在家从事天使投资的工作。

最后再给大家讲一个老故事：古代有条河，河两边有两座山，每座山有一个寺庙。一个寺庙住着一个年轻的和尚，另一个寺庙住着一个老和尚。两个和尚每天都会下山打水，并互相打招呼，关心彼此。忽然，连续几天老和尚没来打水，年轻的和尚担心老和尚是否病了，就到老和尚所在的山上来慰问他。可当他来到

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

老和尚的寺庙时，吃惊地发现老和尚居然在打太极拳，看似非常精神的状态。年轻的和尚满脸惊讶地问老和尚：“老师傅，好几天你都没打水了，寺庙里不缺水吗？”老和尚微笑着给年轻和尚指了指身边的一口井说：“有了它，我再也不需要下山打水了。”年轻和尚充满疑惑地问老和尚“既然你有井，为什么这么多年来还坚持每天下山打水呢？”老和尚说：“我这山上原本没有井。这10年来，我每天打水回来后，都坚持在这里挖1小时的井，功夫不负有心人，我终于挖出了一口好井。现在好啊，我再也不用下山打水了。”

做测试的朋友们看了这个故事后，是否觉得我们可以打好测试这桶水的时候，也应该开始准备自己的井。无论是加入创业公司，还是个人理财，都是为将来实现自己的幸福生活而准备。所以，很希望广大的测试工程师们能多利用一些个人休息时间和精力，学习如何选择一个好企业，去积极关注IT行业的发展趋势，让自己能有识别潜力股企业的能力，或是尽早开始个人的资金理财之路。

6.3 测试职业种类

6.3.1 普通测试工程师

普通测试工程师主要指：大多数进行黑盒功能测试的工程师和自动化测试脚本开发的工程师。

其主要职责为：

- 基于需求文档，从用户体验的角度来编写测试用例。
- 基于测试用例，进行测试执行工作。
- 基于测试用例，进行自动化测试脚本开发。
- 对测试执行中发现的 bug，进行认真分析和详细、准确描述。
- 进行手工回归测试的执行和自动化回归测试的执行。
- 为市场提供技术支持。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

基本任职要求：针对没有经验，但期望进入测试行业的人员。

- 思维严谨和发散——才能设计高覆盖率的测试用例。
- 有一定的沟通表达能力——能与开发人员进行准确的 bug 状况讨论和分析。
- 有一定的编程能力——主要针对自动化测试脚本开发。
- 很强的学习能力和适应能力——毕竟不是每一个测试的新模块都是以前学习过的，因此对任何新的测试需求能够快速的学习和深入了解的能力就显得非常重要。
- 热爱测试工作——只有热爱本职工作，才能做到精益求精。
- 具备相关计算机技术或业务相关知识——这些知识是测试人员开始测试和完成测试的一切知识基础。
- 团队协作精神——测试工作是没有孤胆英雄的，与开发人员合作、与其他测试人员合作都是最基本的要求。

针对已有一定测试经验的人员，除了基本任职要求的条件外，进阶任职还需具备如下要求：

- 足够丰富的业务知识。
- 良好的测试规范习惯。
- 熟悉测试流程。
- 掌握常见的测试工具。
- 较强的 bug 分析和定位能力。
- 能独立编写测试用例。

作一名普通的测试工程师是进入测试行业的第一步，对于软件测试的真正理解和最基本的经验都将从这个阶段开始。因此，对于准备进入软件测试行业的朋友，无论你未来的目标是做自动化测试脚本开发还是测试工具，建议还是先要有一定的手工测试经验，对软件测试有第一线的实战测试经验对于在软件测试领域的发展都将非常重要。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

6.3.2 性能测试工程师

性能测试工程师主要指：对产品进行性能指标测试和压力测试的工程师。通常性能测试工程师都是由有着一定手工测试经验，且对整个系统有整体了解的人来担任。

其主要职责为：

- 进行性能测试方案的设计和执行。
- 进行压力测试方案的设计和执行。
- 性能测试数据的准备。
- 对性能测试和压力测试发现的 bug 能进行准确的分析和定位。
- 性能测试环境的搭建。
- 能提出辅助性能测试工具的开发需求。
- 性能测试工具的选型。
- 对产品性能模块提供设计建议。
- 确保产品性能参数达到或超过行业标准。
- 编写详细、准确的性能测试报告。

其基本要求为：

- 掌握被测产品的性能参数指标和常见性能测试方法——性能测试的基础知识。
- 能设计和搭建高效性能测试环境——性能测试环境的搭建是性能测试的前提。
- 掌握相关性能测试工具的使用。
- 良好的团队沟通、协作能力——由于性能测试的设计和搭建的工作量非常大，因此，很多时候都是由多人组成性能测试团队。
- 至少有 1~2 年功能测试经验——如果对产品的应用模式没有实际测试经验，不懂如何定位 bug、分析 bug，则很难完成性能测试的设计和執行工作。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

- 丰富的行业业务经验。
- 了解常见用户应用模式。

作为功能测试工程师进阶的一个发展方向——性能测试工程师，不但要求在有实战测试经验的基础上能对产品的性能和产品实现构架有一定的认识，而且还要求性能测试工程师必须要有比较丰富的 bug 分析定位能力，才能协助开发人员重现 bug 和定位 bug，提高性能测试的效率。因为，在性能测试和压力测试中发现的 bug 通常是整个产品中破坏力最大，负面影响最大，也是最难定位的 bug。

所以，性能测试工程师对于企业的重要程度和人员素质要求层次都要比普通的功能测试工程师和自动化测试脚本工程师高。在做功能测试工程师的时候，通过大量的功能测试实战可以帮助你不断提升 bug 分析定位的能力。同时个人再积极地学习产品原理和实现架构，将帮助你不断加强内功，为将来成为一个优秀的性能测试工程师打下坚实的基础。

6.3.3 测试设计专家

测试设计专家主要指：进行测试策略制定和测试设计的测试工程师，也包括进行系统测试和解决方案测试的工程师，他们是测试团队中的技术骨干力量。

其主要职责为：

- 前期测试需求收集和分析。
- 测试方案设计，包括自动化测试方案设计等工作。
- 解决测试中的测试难题，提供测试领域中的解决方案。
- 基于项目计划和项目目标制定测试策略和测试计划。
- 利用自己丰富的测试经验指导和帮助测试工程师开发测试用例。
- 参与各种测试策略和测试设计的评审。
- 参与各种测试用例的评审。
- 进行性能测试的设计和分析。
- 去用户现场了解用户的产品使用模式。
- 按产品的实际使用模式来指导测试策略的设计。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

- 参与系统测试和解决方案测试的设计和 execution 工作。
- 参与产品规划和产品架构设计的评审会。
- 测试工具选取的评审。
- 测试工程师的培养。

其基本要求为：

- 丰富的功能测试实战经验。
- 丰富的性能测试经验。
- 丰富的测试业务相关行业经验。
- 丰富的测试设计经验。
- 有针对整个产品全局测试设计的经验，而不仅仅是只有几个模块的测试经验。
- 深厚的性能测试计划、设计、分析能力。
- 较强的沟通、协调和表达能力。
- 具备能与用户进行技术沟通交流的能力。

作为软件测试领域中测试经验最丰富、测试技术最全面的技术角色——测试设计专家，就如同软件开发组的软件开发架构师一样，是从全局的角度来把握公司整个产品的软件质量，并通过自己的测试经验、技术和领导能力来影响整个测试团队的测试设计水平的高低和测试效率的科学性。

测试设计专家是测试领域中众多职业发展方向上，对测试团队的绩效、对企业和组织的贡献最重要的一个岗位，也是对测试技术经验要求最全面的一个岗位。由于测试设计专家不仅要精通各种测试理念和测试方法，还需要精通公司的产品业务知识才能胜任这样的职务。其重要性不仅可直接影响着—个公司测试水平的层次高低，也直接影响着—个公司产品的质量高低及市场竞争力。

在一个重视产品质量的公司，测试部门不仅仅被理解定位成—个产品验收的末端部门。测试应该是和开发同步开始工作，从产品需求规划开始就进入参与，并在产品需求阶段、产品设计阶段、产品开发阶段、产品验收测试各阶段

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

都积极参与其中的一种工种。如果有某家公司对测试的定位是一个可参与产品研发全阶段的岗位，那么这样的公司就更需要测试设计专家，也更容易培养测试设计专家。

所以，测试并不是很多基层手工测试工程师所理解的简单重复的工作，同样测试领域也有可以逐步发展成为对产品研发大局有所掌握和了解的高级技术岗位。对于立志向测试设计专家领域发展的朋友，为了能实现自己的目标，需要擦亮自己的双眼来寻找可以给你提供这样机会的好平台。一些国内重视品牌、重视质量的龙头自主研发企业可以为你提供成为测试设计专家的机会，而一些所谓名气很大的外企相对在中国这样的机会就要少一些。

6.3.4 自动化测试专家

自动化测试专家不仅仅指只会进行自动化测试脚本开发的工程师，还应该具备自动化测试框架构建的经验 and 能力。

其主要职责应该包括：

- 建立自动化测试框架。
- 负责设计自动化脚本框架。
- 优化自动化测试流程。
- 自动化工具及测试相关新技术研究。

其基本要求为：

- 对 C/Java/C++/.NET 任一种编程语言，有 2 年以上编程经验。
- 精通各类测试脚本开发语言，如 Tcl、Perl、Python、VBScript 等。
- 有开发自动化测试工具，以及搭建自动化测试平台的实战经验。
- 建立过专业化的软件自动化测试工具平台，支撑产品线测试工作并提高软件测试效率。
- 组建过专业化的软件自动化测试团队。
- 非常强的技术背景，精通软件测试理论、方法和过程，并能够不断地学习、总结和提高。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

目前，在中国由于测试起步较晚，自动化测试起步得更晚。因此，市场上真正具备自动化测试专家所要求的技能和经验的人才非常少。如果要在公司内部成长为自动化测试专家，首先需要有一个资源足够丰富的平台才有这样的锻炼机会。其次，需要有一定的实力能担负这样的职责。最后，还需要通过多了解商用的自动化测试产品或开源的自动化测试产品来了解最新的自动化测试框架的发展趋势。通过到一些已有多年自动化测试经验的大公司工作，也是一种很好地了解世界先进自动化测试框架的机会，如华为、Motorola、Cisco、Juniper、Microsoft 这样的公司。这些大公司对测试很重视，在自动化测试上已投入很多，至少都在10年以上，站在巨人的肩膀上，即使无法参与自动化测试框架平台的开发，也至少可以通过使用这些先进的自动化测试框架来理解其架构和先进的理念。然后你再带着这些先进的自动化测试框架理念和思想，到一个准备启动构建自己的自动化测试框架的公司把你的才华施展出来。

最后，需要补充一点，在测试组织中，自动化测试只是一个辅助性的工具。如果你的目标是成为一个专注于自动化测试的专家，则可以全身心地投入到自动化测试的职业发展方向上；如果你的目标是成为测试专家，则自动化测试的相关知识和经验只是这条发展历程上一个不可或缺的技能，但不是决定性的因素。即使你在自动化测试领域的经验再丰富，如果在测试策略、测试方法领域经验不足的话，也只能成为一个优秀的自动化测试专家，不可能成为一个全面的测试专家。所以，自动化测试专家是一个不错的职业发展方向，但是并不是测试领域的技术顶峰。测试的职业目标如何取舍和发展，将取决于你自己的兴趣和志向。

6.3.5 测试工具开发专家

作为测试团队自己的“兵工厂”，将为测试团队提供相应的“测试武器”，帮助手工测试团队和自动化测试团队提高工作效率，实现一些很难只通过手工测试或现有工具就能完成的工作。

其主要职责应该包括：

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

- 根据测试工作开发需求，带领团队或独立开发出适用、稳定的测试工具。
- 为测试工具的使用提供相应的技术支持。
- 与测试团队一起很好地规划测试工具的开发计划。
- 密切关注最新的开源测试工具。
- 进行商用测试工具的选择决策。

其基本要求为：

- 有实战手工测试经验尤佳。
- 熟悉所测试行业的业务知识。
- 熟悉相关行业的测试工具情况，包括开源和商用。
- 遵守开发流程，依据产品线特点进行测试工具的需求分析、设计、编码和测试，以及维护、改进工作，并在测试部门应用和推广。
- 理解软件测试工作的本质。
- 数年丰富的 C、Java 或 .NET 开发经验。
- 自动化测试脚本开发经验。
- 具备软件系统设计及架构经验。

测试工具开发这一职业发展方向不仅仅只有那些常年从事软件开发或自动化测试脚本开发工作的人才能胜任，对于有足够软件开发能力的手工测试工程师，也可以向这个领域发展。因为手工测试工程师对测试工作的理解和业务知识的理解非常深刻，开发出的测试工具能更满足测试的本质需求。当然，由于手工测试工作的特殊性，手工测试工程师需要利用一些个人休息时间来学习一些软件开发技能，为未来某一天转向测试工具开发进行一定的开发经验和技能的积累。

对于一些希望未来能转到开发岗位的手工测试工程师，通过测试工具的开发来积累软件开发经验也是一个不错的准备。同样需要澄清的一个误区是：测试团队中真正的牛人或贡献最大的人，并不是测试工具开发高手，而是找到关键 bug 最多的人。如果你对代码开发非常感兴趣，不是很希望向测试策略和方法领域发展，但又比较难在当前的条件下转到开发部门时，测试工具开发这条路是一个很好的选择。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

6.3.6 测试管理者

通常测试团队的管理者主要分为两类：测试项目经理和测试部门经理。在研发规模较小的公司，一般情况下只有1~2个测试项目在同步进行，这时的测试项目经理只有一人，基本也就同时担负起了测试部门经理的职责。而在一些研发规模较大的公司，测试项目通常有多个，甚至10个以上，这时就会在多个测试项目经理之上有一位统管整个测试部门的测试部门经理。那么如何区分测试项目经理和测试部门经理的职责呢？

测试项目经理只对测试的项目负责，也就是大家常说的测试 Leader 或者测试组长。他将负责一个产品或者一组重要功能的测试计划，然后按照计划带领一个测试团队去完成该产品的测试任务。测试计划里除了要规划哪些功能要测、哪些功能不用测之外，还要详细解释各种测试方法在该产品测试中的应用方法，以及设置其优先级。其性质更多是偏技术型，略带一定人员管理职责。

测试部门经理主要负责制定部门的测试规范、测试流程，以及测试资源的调配。他只对整个测试部门负责，不具体到某一个项目中，但每个项目的状况都需要了解。如果有项目中的状况需要部门解决，就需要部门经理负责，例如，人员培养、资源调配、人员考核等。因此，其性质更偏重于管理型，技术内容相对偏少。

测试项目经理的主要职责有：

- 参与测试项目的需求分析和评审，关注项目需求的可测性并设定测试计划或方案。
- 参与或指导测试工程师编写测试用例，开发测试脚本及执行测试。
- 负责完成测试环境搭建和测试执行工作。
- 按照软件工程规范和项目管理流程，实施、管理和执行软件不同阶段的各种测试并提交测试报告。
- 指导下属编写测试文档，收集并分析测试中发现的问题，根据测试分析结果向研发部门提供产品技术性能改进方面的建议，并追踪落实。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

- 响应相关部门的技术支持需要，组织相关人员协调解决相应问题。
- 测试团队成员工作分配，绩效考评。
- 与其他开发项目组进行沟通，协调确保测试项目按时保质完成。

测试项目经理的基本要求为：

- 有丰富的被测目标的相关测试经验。
- 熟悉测试流程的制定和用例编写方法，能够指导测试工程师开展测试工作。
- 对测试技术有深入的理解，掌握 2 种以上常用的测试工具。
- 具有高效协调工作和组织管理能力、较强的分析和判断能力、沟通能力。品行端正，有良好的考核观；工作作风踏实、认真，有较强的责任感和敬业精神。

测试部门经理的主要职责有：

- 测试规范与测试计划的制定：根据实战经验制定高效率的测试规范，测试流程优化，并根据公司的产品特性，沟通测试工作的目标与提交的成果，组织下属制定软件测试计划。
- 按照测试计划，分配测试任务，协调测试人员，组织编写软件测试用例，并评审测试用例和脚本，确保测试的顺利开展。
- 组织软件产品的测试实施工作，指导和监督测试的进展和完成情况，及时沟通解决重大测试问题，确保测试目标的达成。
- 根据产品发展方向，及时追踪、收集软件测试新技术、新动态的资料，组织管理、更新测试用例库，与人力资源部门沟通部门人员的培训需求。通过技术培训、交流等方式，提高软件测试工程师的业务能力。

测试部门经理的基本要求包括：

- 具备缜密的逻辑思考能力，耐心、细致的观察能力。
- 工作认真、负责，学习能力强，有创意。
- 成熟稳重，主动积极与下属沟通，有一定的领袖气质。
- 非常熟悉各种测试方法和科学规范的测试流程。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

- 熟悉公司产品相关领域的业务知识。
- 具有软件测试行业4年及以上专职从业经验，熟练掌握各种测试工具。
- 拥有负责5人以上测试团队实际管理经验，具有良好的管理意识与能力，以及高效的执行能力，能够激励并带动团队高质量地达成工作目标。

无论是测试项目经理还是测试部门经理，对于测试人员的要求都已不只有技术方面的要求，在需要做好事的基础上还要求做好人。毕竟这些工作都有了一定的人员管理的职责，需要在工作中体现出领导力，而不再仅仅是以前做技术的思维——做好自己手里的事就OK了。领导者与普通基层员工相比，有更多的要求：领导者必须既要做好事也要做好人的工作；既要做好自己手头上的事，更要关心整个团队的绩效目标；要时刻关注团队成员的工作状态和个人状态，唯有团队成员的个人状态好了，才有可能保证工作状态良好。同时，当团队遇到困难和挫折时，领导者还需要自己主动站出来稳定住局势，并带领大家渡过难关。

笔者知道，很多工程师都立志走管理路线，觉得做了管理就是当官了，可以轻松拿大钱，可以安排他人干活而不是被安排干活。事实是真正的好领导，是一个服务者，而不是当官。团队中的每个基层员工都是领导者服务的客户对象，而不是领导者自己的附属物。领导者对内，要把每位基层员工“服务舒服了”，满足了他们的需求后，他们才可能团结一致，跟着你这个领导一起往前冲。领导者对外，遇到了新任务或新困难，肯定要成为最坚强者。既要尽量保护自己的团队成员，又要完成所接到的外部任务。

所以，要做一个合格的领导者，决不仅仅是大家所想象的“只要我是技术高手就能做好一个管理者”。很多公司的高层都这么认为：领导管理岗位不应该是给技术高手的唯一奖励，真正的好领导需要的是除技术外的更多综合素质，如成熟、有大局观、待人待物能力、积极主动沟通能力、正直公正、很强的协调能力。对于一些技术骨干，如果天性上不擅长做人的工作，何必又要强求做一个整脚的领导者呢，还不如专注在技术路线上，成为一个越来越强的技术高手。

虽然，目前很多招聘要求中，只要求测试工程师能有2~3年的经验就做测试Leader或经理。实际上，这些信息说明了目前非常缺少既有足够测试经验又

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

具备足够综合素质的测试领导者。对于性格不是完全内向，也乐意主动与人交流的测试工程师，可以试着向测试的管理领域发展。可是中国目前很多的测试管理者仅仅是依靠资历而成为了一个蹩脚的管理者，这也是不少在基层工作的测试工程师经常抱怨公司管理的一个很重要的因素。有个现象，基本上大多数人都发现了，就是基层员工所谓的对公司不满意，其实真相更多只是对他的直接主管不满意而已。除非你所在的企业可以依靠远高于行业平均待遇的收入标准来把人套在这里。否则简单粗暴的管理风格，缺乏服务意识的官僚主义管理风格，缺乏真心的人文关怀的管理风格将很难留下足够好的技术骨干。长久以往，你的团队不是优胜劣汰，而是劣胜优汰，成为一个缺乏创新和激情的团队。

最后，对于目前已是团队中技术骨干的朋友，需要对自己的个性特点有一个清楚正确的认识，不要仅仅为了心中那个“当官好”的目标，用自己的短板来发展，反而耽误了自己的技术发展前程，并给公司的老板制造了难题。也许此时在老板心中并不期望你来做一个管理者，因为他会担心在得到一个平庸的管理者的同时，会失去一个优秀的技术人才，甚至还担心由于平庸的领导能力会耽误团队中有潜力的新骨干的挖掘和成长。

中国在软件测试领域虽然缺乏大量优秀的领导者来改善中国测试行业的管理水准，但是更缺乏大量的有丰富经验的测试专家。物以稀为贵，当人人都力争去做测试管理者时，而你根据自己的特点，不用自己的短板强求去做管理工作，坚持自己的长板在技术领域不间断地积累，成为一个测试专家。那么你就更能成为一个市场上奇货可居的抢手人才。

6.4 突破测试的选择

把软件测试作为职业有着怎样的转行机会？答案是多样化的。因为相比软件开发人员，测试工程师对产品的认识更广泛。因此，很多有着丰富经验的测试工程师或测试管理者会得到一些测试以外的新工作机会，如项目经理、售后技术支持、售前技术支持等。笔者在一家公司从事了2年的测试工作后，的确也有机会得到售后技术支持和售前技术支持的新的职务机会。所以，测试人并没有被锁定

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

到做了测试，就只有干测试。甚至，测试人并没有固定在一个公司、一个职业发展方向或一个工作岗位上，我们可以大胆地改变职业发展方向并追求其他目标。事实上，也的确有不少的测试同行在工作一段时间后，转到了测试以外的行业继续发展。本节，笔者将把自己身边的测试同行转行后的一些故事与大家分享，让从事测试的朋友在试图改变自己的职业轨迹时，多一些有用的参考。



6.4.1 市场

通常测试人员口中的做市场更多的仅仅是指销售。从商科的角度来说，市场更多的是指 marketing，销售是 sales。在笔者身边确实有不少测试同行转去做销售，有的去卖测试工具或自动化测试产品，也有的在自己公司卖自己所测试的产品，还有的转行到别的行业，甚至离开 IT 行业去做销售。同时，有很少数的测试同行转去做 marketing，例如，产品规划、产品调研、产品营销策划或产品经理。因为笔者也没有太多关于 marketing&sales 的全面理解，就只能从这些转行的朋友的发展和特点来描述一下做 sales 或 marketing 的不同感受。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

1. sales

对于心理素质好，不怕各种打击和失败，做事非常执着的朋友，如果想去磨练自己的心理素质，或是对所谓 sales 工作充满无比完美向往，也可以选择去尝试转型做 sales。不过，让大家意外的一个现象是销售测试工具特别是自动化测试工具会非常难，很打击刚转型做销售的测试朋友。这个结论是一位在美国原厂从事过测试工具销售的朋友告诉笔者的。原因大概是因为目前中国还不重视测试，因此在测试工具的投入费用上还比较少。所以，建议想转型做 sales 的朋友，最好从自己公司入手，找一个成熟市场销售，开始时相对难度会小一些。

有一个朋友所在的公司规模太大了，基本不可能转岗，就找了个小公司去做销售。由于这个公司的资源太少，平台太差，结果是 2 个月后，还是重新回归到了测试行业。目前，他已坚定在测试行业继续发展了。的确也有一些因为听信了周围朋友的说法“你性格挺活跃的，应该去做销售”；或是误信“销售轻松，钱又多”这样的片面之言，而决定到销售行业去发展。肯定也有人在销售行业中创造出了比测试行业更好的一片天，但是，也有转销售的人最后还是回归到测试。究其原因：销售这件事，太难了。比做测试难很多，需要你付出更多的时间、精力和脑力。笔者有一位先做过 5 年开发测试，后转型做了 10 年销售的朋友曾说过：“销售不只是吃喝玩乐。销售比干技术更需要付出大量的脑力，需要面对和处理的困难复杂度比技术要难得多，而且个人生活上也会有很大的牺牲。”可惜人们往往只看到销售拿奖金的一刻，没有看到在拿到奖金前，销售人员需要做大量外部工作和内部工作，花费非常多的心力去解决一个个看似不可能解决的问题，才走到最后一步的成功。远比做技术时难得多，也更辛苦。他也曾给笔者建议：“如果你想折磨下自己的意志，或是让自己明白做技术真正的好处，可以去试试。也许试过后，你会发现做技术的好处也有不少，反而能沉下心来钻研。”

2. marketing

marketing 主要包括如下工作内容：品牌宣传、产品规划、产品调研、产品营销策划、产品经理等职务。总地来说，marketing 的工作压力比 sales 小，应该还算比较轻松的工作。特点是务虚较多，务实少，做时间长了，会让人感觉自己太空，

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

心里不踏实。而且绩效考核比较难，工作成果的认同感不一定强。笔者身边有几个朋友通过公司内部应聘转到了品牌宣传、产品规划、产品调研、产品经理。转到品牌宣传，基本上也就告别了技术的生涯。而一些技术实力较强的测试人员，如果综合素质好，沟通能力也不错，则有机会向产品调研、产品规划方面发展。

当然，对于少数技术、沟通、协调、市场意识都很强的人，则有机会成为产品经理，统领整个产品的规划、产品发布计划。好像看起来，测试转到 marketing 是一个不错的选择，压力和工作量没有 sales 大，而且也比纯测试有趣。不过凡事也有例外：笔者有两个朋友都是从公司的测试部门被抽调出来分别从事产品规划和产品调研工作的，结果在 3~6 个月的时间内，两位朋友分别离开了让很多测试人员很羡慕的职务。一位到 Cisco 做技术去了，而另一位去某美国外企做测试去了。和他们聊起原因，主要还是觉得 marketing 工作太虚了，心里不踏实。因此，看似轻松些的工作也有一些同志不一定能心安理得地去喜欢。

总的来说，测试工程师如果想向商业领域转行，marketing 较 sales 轻松，技术也没有完全丢掉。如果心理上能够适应新工作带来的新感受，在自己的公司转型到 marketing 也许是一个不错的机会，只是这样的机会相对销售会少得多，需要大家平时多积累在公司内的人脉，形成好名声，或许能把握住这些稍纵即逝的机会。

6.4.2 项目管理

此处所谈的项目管理并非仅限于软件测试项目管理，而是指整个产品的项目管理或整个软件的项目管理。一个项目经理是否优秀，主要是看他/她能在多大程度上提前识别并消除风险，是否能在项目相关各部分有足够的沟通、协调、组织能力。有的公司项目经理的权限更大，还能够进行资源申请、分配、任务划分及进度制定。风险对于一个项目就如同 bug 对于一个产品，影响项目正常进度的风险总是会存在的，而如何管理好风险，提前进行风险预判并做好相应的对策，就需要项目经理对软件研发流程的各个环节都有所了解，并有较好的风险意识。

作为整个项目的第一负责人，项目经理由于需要与各部门之间进行进度跟踪、资源协调、问题沟通，因此良好的沟通、协调技巧就成为了必备基本功。对于从

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

事过系统测试的测试工程师来说，要转型成为软件项目管理者，是比较对口和容易的。例如，微软的开发方式是“测试驱动”的，在测试过程中发现了墙角还有没涂到油漆的小块，开发则根据这个思想再补上那一块。如果能有一段从事过软件测试的经历恰好能让人更能从用户的角度来考虑问题，更能深入地了解程序开发过程中可能出现的问题，这都是成为一个优秀的项目管理者的必要条件。所以先做测试后转为软件项目管理，就应至少具有如下 4 个优势。

(1) 有“先知先觉”的思维习惯，能提前预判 bug 所在处，正如项目管理中的风险预判。

(2) 在进行 bug 分析定位时，锻炼了与研发内部各部门打交道的能力，常常是一个异常能牵动所有部门的人，如协议开发组、驱动开发组，甚至硬件组；并在论证是否为一个有效 bug 的过程中锻炼了较好的表达、沟通、组织能力。

(3) 为了配合售后人员解决售后问题，以及为售前提供解决方案，测试人员也具备了一定的与销售团队打交道的能力。

(4) 非常了解用户的应用模式，可以很好地驱动项目的执行。

笔者有一个做了 4 年测试经理的朋友，在有了足够多产品测试全流程经验和产品应用模式后，成功转型做了整个研发项目的项目经理，到目前为止，他做得还不错，而且自己感觉也还不错。

一个好的项目经理不仅仅是“沟通、协调”，如果做不好就会沦为整个项目组的秘书，而不是项目的总领导者。笔者曾见过有的公司拿一些刚毕业的人来直接做项目助理，实际上由于缺乏研发全流程，以及足够业务技术背景，而听不懂各业务部门的内容，基本上他们将一直停留在一个助理的角色，开会成为书记员，会后填写报告，很难成长为合格的项目经理。因为项目管理是一个靠业务经验吃饭的职务，工作年限较短的朋友切勿急于求成转成项目管理助理，否则会由于缺乏足够的业务知识，而一直从事秘书性质的工作，将来也就一直成为项目秘书。

所以，对于职场新人最好多花几年时间融入到实战的测试工作中，积累技术业务知识、研发全流程的经验和与人打交道的能力，厚积薄发才能做好真正的项

目管理经理。鉴于从事测试经理能给测试人员带来足够丰富的各部门协调、沟通的能力，并可以通过测试项目管理作为对项目管理经理的一个项目管理经验的积累。因此建议：如果未来期望向项目管理方向发展，可以考虑先从内部的测试 Leader 或测试经理做起，积累全面的项目管理经验后再考虑转型。

6.4.3 售前技术支持

售前技术支持在有的公司也叫产品经理。其主要职责包括：根据该产品区域规划，在区域市场提供产品推广和销售支持；负责该产品区域市场的分析、市场机会点的发掘；负责对品牌营销活动进行策划、组织实施，落实本区域市场的产品规划及产品品牌的树立和提升；负责对项目进行策划，对项目进行技术支持。对于大多数普通的测试工程师，只要为人成熟，性格不太内向，都可以向售前技术支持发展。

由于笔者所工作过的公司主要是复杂产品型企业，因此基本所有的销售项目都需要有售前技术支持这样一个角色。很多测试工程师于是把自己转行的目标放在了售前这个角色上，毕竟这个岗位对技术的要求更多的是产品应用上而不是代码级或非常深的内部设计上，而产品的应用模式又是测试工程师最熟悉和擅长的。因此，测试工程师转到这样一个岗位上，以前的技术知识和业务知识的积累将会得到很大的发挥。同时，借助这样一个技术和销售结合的岗位，可以通过该岗位的工作特点逐步了解销售的工作情况，如果觉得销售适合自己，从售前转销售是一个非常容易的转换。由于了解了销售的工作内容和工作状况，认为做销售不适合自己，则可以继续在售前的岗位上工作。不过，在笔者身边也发生过做了一年售前工作后又转回测试的案例，究其原因，也许只有自己做过售前后才知道吧。

当然，也有测试朋友在转售前工作几年后，由于以前积累了很多产品研发的经验，再结合对客户需求的深入了解，最后发展成为了产品规划经理和产品经理。当然，这样的案例在笔者身边只有一个，而且其本身在做测试时就是一个对产品内部构架原理非常熟悉的人。相对于前面提到的市场类、项目管理类的转岗，在笔者身边统计发现，从测试转到售前，并成功胜任新工作职责的测试朋友比较多。而且作者也认为从测试转到一个新岗位，若能充分利用以前做测试而积累的大量

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

产品应用的经验和知识，也将会大大减小转岗初期的压力和风险。

因此，如有测试朋友希望以后转到售前技术支持这一职务，请在平时的测试工作中，多从全局、系统的角度来理解产品的应用，并多多练习自己的沟通、演讲和表达能力，并有意识地修炼自己的性格和脾气，多多了解同行的相关产品和应用，多了解一些行业知识，以便增强转岗售前的成功率。

6.4.4 售后技术支持

售后技术支持主要进行：产品安装、故障定位排除，有时也进行一些针对客户的产品技术培训和售前技术交流。该岗位的特色是纯技术工作内容比例高，与客户沟通相应比例低。从笔者身边测试工程师的民意调查来看：大多数的测试工程师都不愿意转到售后技术支持的岗位上。究其原因，主要是有时在客户现场进行故障定位时技术压力大，出差频繁，做的苦活、累活多，感觉没有做售前轻松。笔者认识两位测试工程师在转做售后技术支持后，反而取得了自己从未想到的成就。当时他们为了响应公司的号召到市场一线去，原本计划做售前技术支持，可是销售部的领导担心他们暂时还缺乏与客户进行沟通的能力和经验，因此让他们先从售后技术支持做起。结果不到半年，两人就在公司的售后技术支持团队中靠优异的技术实力出名了。不到一年，一位成为了售后技术支持团队的高级技术工程师（基本相当于技术团队中前几名的角色），另一位成为了负责一个片区售后技术支持团队的经理。

为什么这两位朋友能在售后技术支持团队中这么快就取得不错的成绩呢？后来大家分析最根本的原因就是：因为他们是做测试出身，对产品的功能、原理、应用的理解和经验都远强于普通的售后技术支持工程师。再加上在从事测试工作时积累了大量分析问题和定位问题的工作经验和能力，使得他们在客户现场进行问题分析定位时的效率和准确率更强。因此，当他们到了售后技术团队后，他们原有的技术得到了游刃有余的发挥，个人价值也得到了很大体现，很快就成为了售后技术团队的技术明星和骨干力量。

最后，无论是他们个人还是公司都发现，原来测试转到售后技术支持团队会

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

大大帮助售后技术团队提高技术水平，对公司是非常好的一项举措。而且对测试人员而言，在测试期间的积累还能更好地发挥作用，让自己更容易取得成绩。因此，对于一些性格偏内向，不太喜欢与人沟通交流的朋友，在转行时，可以考虑充分利用好做测试积累的业务知识和产品知识，相比做一个普通的售前，还不如在一个纯技术领域做一个优秀的骨干售后工程师。

6.4.5 技术培训

从测试往技术培训方向发展也是一条不错的选择。只是这里提到的技术培训不仅仅指在一些培训学校进行测试技术培训，还包括在产品公司进行产品知识培训和业务技术培训，培训的对象包括公司内部人员、市场技术人员和客户。

到一些软件开发和测试培训学校进行测试技术培训，很类似于当老师。因为这些组织中培训的对象都是没有多少相关测试知识的年轻人，培训的内容也主要集中在测试工具、测试流程或一些测试基本思想上。培训的方式需要从零开始传授知识，既要细又要慢，和学校的方式类似。但是，并不是每一个想转到培训的测试工程师都适合这种学校老师性质的培训工作。有些测试工程师经过长期的测试工作后，对产品的应用和行业业务知识变得非常精通，但对测试的一些知识掌握较少，这类测试工程师也并不是很适合到培训学校进行测试知识的培训工作。

那么对于不适合当学校老师或不擅长测试理论培训的测试工程师，还有什么好的方法转到技术培训的方向吗？有，在产品公司进行产品知识的技术培训。笔者曾有一个女同事，学习能力和智力水平都非常强，对产品的复杂功能都能较容易地深入掌握，的确是一个技术女强人。可最后她却放弃了测试经理的选择，而去中兴通信做了一名产品技术培训师。3年后一次偶遇，她告诉笔者，自从做了产品技术培训讲师后，她很高兴，只要中兴不开除她，她将一直干下去。因为技术培训是她最喜欢的工作，也是她做得非常好的工作。在写这本书期间，有一天一位做了近5年测试的女同事说：“她要到华为做技术培训去了，虽然现在市面上有5年测试经验的人非常抢手也很吃香，但是她还是想体验一下新的工作和生活方式，所以选择了到华为去做技术培训。”这是笔者身边发生的两位从软件测试转到产品技术培训的真实案例。既然用人部门愿意录用软件测试人员来从事产品

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

技术培训的工作，那就证明软件测试人员可以向这条道上去发展。

综合这些从事产品技术培训朋友的体会，产品技术培训的主要培训对象都是有一定行业知识基础的用户和公司的市场技术人员，在进行技术培训时不用像教学生一样侧重于教学，其侧重的技能不是教书和测试理论，而是演讲能力和行业业务知识。通常都是系统一级的知识，而这些知识正好是测试人员平时工作时的知识深度层次。从事产品技术培训工作，没有研发的进度压力，也没有市场的销售压力和现场问题解决压力。相对来说，比较轻松，也不用太费心费力。缺点可能就是在培训部门工作，不如在销售部门容易做出大成就，得到大家的赞许。在就业市场上招聘对应产品技术培训的岗位机会也比较少，市场上对培训人才的招聘不如测试人才那么紧迫和稀缺。不过，总的来说，软件测试人员向产品技术培训方向发展，既不浪费以前做测试积累起来的行业业务知识，也比较适合不追求太大成就，但希望工作轻松一些的朋友。

6.4.6 转向开发

在软件测试行业里，也许有一些朋友是误打误撞进入这个行业的，还有一些朋友是为生存所迫而从事测试的。在他们心中一直有着强烈的开发情节，不愿意在测试领域投入和发展。或许现在这些有开发情节的朋友内心正在痛苦地挣扎，埋怨“当初为什么要进入软件测试行业？自己再也回不到开发领域了”。其实，这类朋友不用太过悲观，从测试转回到软件开发这条路上也有很多成功案例。软件开发和软件测试之间的互换对一部分人来说是可以相通的。在笔者身边就有 4 个成功案例，暂时取 A、B、C、D 4 个代号。

案例 A

A 大学时代是一个开发高手，毕业后由于公司资源调度的原因进入测试部门，很快就成为了测试部的测试牛人。在完成测试工作之余，他自己利用晚上的时间写了不少有用的测试小工具。由于他是测试部门中各类技术评估中的第一牛人，他的工资标准比同期进入开发组的人都要高。不过，3 年后他由于个人职业生涯选择缘故，离开了公司，到一家研究所从事软件开发去了。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

 案例 B

B 与 A 同期进入公司，也被调配到了测试部门。在测试部门从事了 2 年手工测试后，开始逐步负责公司自动化测试框架搭建的工作。后因开发部缺人，他在测试部门待了近 5 年后，转到了开发部，干得非常不错。

 案例 C

C 是一个非常聪明的人，其学习能力和掌握知识的能力在他们同期进来的本科生中是最厉害的，每天的培训考试都考第一。同样是属于开发招进公司后，分配到了测试部门。在测试部门工作期间，对他来说确实是一种痛苦，他内心的开发情节非常强，常常利用工作之余自己学习和练习一些嵌入式开发的知识。在测试部门很平凡地工作一年后，因开发部缺人而转到了开发部。从此在开发部如鱼得水，丝毫没有因为一年没有直接参与产品开发而跟不上开发进度。现在此君正在一家欧洲企业带领着几个人的开发团队。

 案例 D

D 兄的职业之路非常怪异。此兄刚毕业时，从事的是嵌入式开发。一年后，跳槽到了一家外企被转到了软件测试岗位。从事一年测试工作后，又被公司调为嵌入式开发。又是一年开发工作后，D 兄跳槽到了 NOKIA，又做测试工作。开发→测试→开发→测试。D 兄的工作适应能力确实非常强，基本打通了开发和测试的隔阂，在两者间可以自由穿梭往来。

最后，额外补充半个案例。之所以称为半个案例，是因为得到了转软件开发的机，但是主动放弃了这个机会。该测试工程师毕业后从事了近一年手工测试工作，一次偶然的机会被华为叫去面试，通过笔试和技术面试得到了华为的软件开发的工作机会，只是由于该测试工程师喜欢上了从事系统测试所带来的能快速、全面掌握系统知识的好处。最后，在这次转向软件开发的机会面前选择了放弃。

以上 4 个成功从测试部门转到开发部门的案例，具有一定的代表性。大家能从中发现一些规律，一名测试工程师要想成功转行到软件开发工程师并不是一件

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

难于上青天的事情。如果你能在测试工作之余，自己多花些时间在测试工具开发、自动化测试开发上，甚至自己学习并练习所在公司的软件开发技能，同样有非常多的机会转向软件开发。如果你在大学阶段打下了良好的软件开发基础并积累了足够的开发经验，工作后即使第一份工作从事的是软件测试工作，同样不影响你未来重新转回到软件开发。

6.4.7 做生意

也许这个建议不该出现在这本书中，因为做生意好像与做软件测试没有什么必然联系。事实也确实是这样，转行做生意与做软件测试无关。不过，在软件测试的从业人员中也的确有着这样非常小的一部分人最后转行做了IT行业以外的生意。笔者就曾有软件测试的同事转行在IT行业外做生意，其中有一位曾做到测试经理，在做了近5年测试时，选择了下海做生意，开始时做了一个餐饮行业的加盟店，后来又开始销售户外用品。从他下海到今天也快4年了，也不知他现在的道路走得怎么样，祝他一路走好。后来又有一位女测试同事，也在从事了4年左右软件测试后，下海开了服装店，不过这个店开了没1年，她又重新回到了软件测试行业，并且一做又快3年了。另外，从一个美国同事口中得知，原来在美国也有做软件测试的工程师转行去做小生意，其中一个华人软件测试工程师开了家卖PC的商店，一个印度软件测试工程师开了家小型运输公司。

虽然，离开软件测试行业，甚至完全离开IT行业去转行的人是非常小的一个群体。但是，笔者却认为他们是很有个性和想法的一群人。他们不是在软件测试行业做得不成功的人，也许他们觉得在软件测试行业做得足够好了，没有了新的挑战高峰，于是就想体验一些新的生活或新的挑战。当然也有可能他们厌倦了IT行业，所以，没有像绝大部分的软件测试工程师一样在IT行业内部进行新的选择。当然，并不是我们每个人都适合或喜欢做生意，正如那位开了服装店后又重新回到软件测试行业的同事一样，或许她更喜欢做软件测试工作的那种生活节奏或工作节奏，也可能是觉得自己不擅长做生意，于是放弃了继续做生意的想法。因此，如果你想尝试做生意的感觉，可以考虑先拿1年时间勇敢地去冲，即使最后发现自己不喜欢做生意，或是不适合做生意，再回来做技术，这时心态将会更沉稳一些。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

6.5 外企和民企的选择

外企和民企，选择哪一个呢？“外企是大材小用，民企是小材大用”。“以前在民企，工作技术含量高，可惜工资低。现在在外企，工作技术含量低，但是工资高”。这是一位在大外企工作的朋友说的一段特别深刻的话，正好总结了目前中国在 IT 行业内外企和民企鲜明的特点和不同。

大部分人之所以羡慕到外企工作，其好处主要在于：待遇不错，门面不错，国际大公司的名份能给人足够的虚荣。在专业和管理上，外企有着一些先进的流程、人性化的管理方式，以及大多数外企员工较高的个人修养和素质、国际化的工作氛围，这些正是目前国内民企所欠缺的，也是值得我们去取经学习的地方。正所谓有得必有失，任何选择都存在机会成本，选了 A 就不能选 B，而 B 就是选 A 所要付出的机会成本。外企在给予的同时也在剥夺，在培育你的同时也在裁剪你。在对外企心存向往之时，也要意识到进入外企是需要付出代价的，最大的代价莫过于：你的未来，取决于外企想要你成为怎样的你，而不取决于你想成为怎样的你。

到目前为止，把研发和测试的高端职务放在中国的大外企还比较少，再加上这些外企在国外已发展多年，早已有不少高端的专家在把持着技术含量高的技术岗位，而中国大陆的工程师更多能得到的是一些中低端的职务机会。究其本质还在于，目前在中国的外企研发中心对中国还是有一定的严格核心技术保护，到中国来建立研发中心，少量因素是看重中国人才的素质，大多数原因还是看重较低的工资成本和运营办公成本。因此，外企更愿意把一些高端设计的工作放在本国，而将金字塔下半部分的工作内容放到中国。同时外企中每个人的职能和目标都有着明确的限定，并要求只能在这个范围内去发挥个性和创造性，就像你可以在舞台上随意表演，但那个舞台却很小。于是，大家常说的“外企天花板现象”就这样诞生了，不光是管理层面，就是技术层面也同样有着这样的问题。

一位在 IT 外企工作了 10 年的畅销小说作者在一次讲座上曾说，“现在很多在国内经营的外国大公司都不愿意培养中国本土人才，它们把自己看成一架机器，中国员工就是螺丝钉。哪个螺丝钉脱扣了，换一个就是。进这些大公司，不光学习不

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

到什么东西，而且职位提升也很困难，许多人从外企干几年出来，要么是一颗闪闪发亮的螺丝钉，要么是一颗锈迹斑斑的螺丝钉，反正都不会变成飞上枝头的凤凰。虽然这些话说得有点绝对，但也不无道理。目前国企、民企的领头人中，在外企干过多年的有多少？那些自主创业成功的老板中，从外企出来的人又有几个？为什么外企到国企、民企的空降兵纷纷折戟？为什么外企出来创业的大多举步维艰？就是因为外企培养锻炼人的目的，是企业自身所用，是为了适应企业自身环境和发展，而不是为了让这个人未来创业，或者到其他的环境中去大展宏图。”

反观国内的一些好民企如华为，它就能为我们中国人提供一些很高端的技术岗位，虽然给的工资可能没外企高，加上工作的难度大，会让人感觉压力大，很累、很辛苦。但是对于期望在技术路线上成长为全面技术专家的朋友，到一个好的民企去工作，机会将比在外企多得多。例如，华为内部就有很多高端的开发或测试的职务，这些岗位是外企在中国基本不太可能会提供的。下面给大家举一个猎头给笔者的高端系统测试工程师的职务描述。

高级测试系统工程师

工作职责：

- 负责接入网产品的测试设计工作，包括产品的前期测试需求收集和分析、测试规格输出、测试方案设计，包括自动化测试方案设计等工作。
- 带领有经验的测试设计人员，探索和开发测试领域的工程方法、设计流程。
- 带领有经验的测试人员，解决产品测试中测试难题，提供测试领域中的解决方案。
- 和客户进行交流，了解和交流运营方面的需求，调研产品的运用场景。

该职位挑战：

- 测试是研发过程中的一项非常重要的工作，而测试设计是测试的前端工作，决定了测试质量。
- 如何在成本、进度、人力的压力情况下，寻找测试设计工作方法的突破，提高测试质量和测试效率。
- 需要尽快熟悉公司现有工作流程、工作制度和相关产品知识，需要尽快融入研发环境。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

该职位机会：

- 表现优异可以成为测试技术带头人，一段时间内可以成为某个测试领域的首席工程师。
- 可以带领一个测试设计团队从事有开拓性测试工作。
- 在技术路线发展上，最高可以成为公司的首席科学家，待遇等同公司级的副总裁。
- 年薪 25 ~ 80 万。

笔者相信不少读者看到这样的测试工作内容、发展空间及对应的待遇也有心动的感觉。在测试的技术发展路线上，以上的工作内容应该算是高端的了。在国内少数优秀的民企就可以提供这样的工作机会。

虽然部分优秀的民企提供了专业上可让你自由驰骋的舞台，但同时也有大量不够优秀的民企和由于老板自身的见识对测试不重视，给测试人员提供的舞台可能就相对较差。测试工程师在这样的平台既无法从事高技术含量的工作，也无法学习到成功外企规范的流程和高水平的管理水平。因此，笔者对外企和民企初步分为如下 3 档，作为大家就业选择的依据。

- 第一档高水平优秀民企

即能提供良好的技术平台和无限的技术上升空间，从长期发展来看待遇未必比外企差。

- 第二档外企

外企先进规范的研发流程和较好的管理水平，会让人受益匪浅。

- 第三档不优秀的民企

作为进入 IT 行业的第一个敲门砖吧。

最后再给准备毕业的大学生一些就业选择的建议。首先，任何选择都取决于每个人的权衡，要因人因地因时制宜，不可一概而论。其次，大多数人想走的路，不一定是对每个人来说都适当的路。有时正相反，另辟蹊径远比随大流、从众要

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

好得多。这里，再引用一位网友的话：“刚毕业的大学生，职业规划最好不要超过3年，因为在选择了一条路时，你可能就放弃了途中遇到的其他岔路的机会。”一路走到底固然是好事，但也有可能这条路是一条永远也走不完的路。在自己能力所能选择的范围内，听从自己的心。

6.6 大公司和小公司的选择

很多职场老兵都爱说这句话：“大公司做人，小公司做事。”这简单的10个字也许正好概括出了大公司和小公司的最本质区别。不过要强调的是此处对大公司、小公司的定义不仅仅是指人员的规模或销售额的大小，而是指这个公司的企业文化是一个大公司的企业文化，还是小公司的企业文化。有的公司或许人数有了5000人，但是依然有着小公司的那种创新精神，讲究高效率的办事风格，内部充满激情，在制度上没有冗余的啰唆条款，没有官僚的组织文化。而有的公司可能人数不到100人，却已丧失了创新、激情、高效率的办事风格，并出现官僚的部门文化，部门之间通过冗余复杂的流程制度来互相推托责任和风险，人人都在保守地保护自己的利益，而不敢承担责任和风险去积极进取。在小公司的文化中，每个人的绩效压力都会比较大，因为都要靠做好事才能达到自己的工作目标。而在大公司的文化中，因为需要花不少的精力与其他同事之间处好关系，因此如何做人就比做好事更重要了。

那么除了“大公司做人，小公司做事”这一文化上的本质区别外，在那些组织机构比较庞大的大公司工作与在组织机构相对较小的小公司工作还有哪些区别和不同呢？

大公司自然有大公司的好处，如果把给大公司打工也看作一个企业的话，服务于大公司就意味着你有稳定的现金流和各种福利保障，也就是说，生存安全指数比较高。至于未来超额回报的收益，则主要看你在职场中的业务能力和政治技能。在大公司中，你除了增长自己的业务能力外，最重要的是可以快速建立人脉关系并积累资源，这些东西的重要性会随着年龄增长越来越重要，远远超过你自身的能力增长。为大公司工作有许多便利的条件，明确严谨的计划和有条有理的

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

管理秩序使你工作稳定。有数以千计的雇员，能为你提供在不同环境中与不同的人共事的机会，有利于结交朋友，获得良师益友的帮助。按照公司提升政策，你或许有“安全”的升迁。而且还可保持长期以来在公司享受既得的利益，薪水和红包优厚，能提供良好的医疗、养老保险、带薪休假，以及继续受教育。

但是大公司并不完美，大公司里官僚作风严重，有许多条条框框，做事情“墨守陈规”。而你在公司中从事的仅是整个项目中的一小部分工作，比如设计螺栓。但作为一个小小的螺栓设计者你往往被束之高阁，要想冲破束缚是很困难的，你在单位的短暂半生中，如果只是年复一年地设计螺栓，必须把握每一个机会，才能使你成为这个大公司整体运转不可缺少的部件，对此你必须时刻警惕。在那些成百上千的雇员中，许多人也不情愿成为附属品，所以你要出头所受到的竞争就会更多。如果你只想做好本职工作并得到认可，或许在小公司里更容易。

那么小公司的特点是什么呢？小公司人员少，你有可能在一开始就被委以很大的责任，不仅是对你所设计的螺栓，而且还要对整个运行系统负责。小公司可以在广阔的领域内给你提供获得丰富经验的机会。在一个小公司里，你的一切都是有价值的，且是必需的，不会被视为附属品。你可能发现薪水稍低一些，但更具有竞争性。薪水加薪比率、职位提升，以及红利的分配没有固定的规律。当你干一段时间以后，你的雇主会给你不定期地加薪。但是公司没有或很少有较高的职位或部门，特别是在一些“家庭式”的私人公司，而你很难成为其“家庭”中的一员。小公司前途未卜，你也就较少安全感。当然，它也给你带来更多机遇和挑战。

我们应该如何选择呢？你必须掂量一下什么东西对你是最重要的，并以你的判断为基础作出选择，而不是别人的指手画脚。其实无论是哪条道路，毕竟人生有了不同的选择。很显然，如果你愿意承担较大的压力去多做有挑战的事，而不想在人际关系上花太多时间和精力，则可以考虑到小公司文化中去奋斗。但是如果你不喜欢长期在高绩效压力下工作，希望能在一种大家都轻松和谐相处的环境中工作，则大公司的文化更适合“养人”一些。一般而言，刚工作不久的年轻人多半会选择去充满激情的公司奋斗，而一些工作时间较长的人会由于失去了激情或生活的负担而选择在一个比较适合“养人”的公司工作。甚至，也有人先在适合“养人”的公司工作，后因内心强烈地渴望“成就”，于是又重新选择到有激

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

情的公司工作。无论选择哪种类型的公司，只要你在此工作时，能感觉到满足感，对你就是最好的选择。

6.7 外包公司和自主产品公司的选择

笔者曾在自主产品公司工作过，也在外包公司工作过。应该说两种类型的公司各有好处，也各有不足，适合不同人的需求。由于大家对外包批评较多，下面先谈谈笔者对外包的看法。

(1) 外包公司较缺乏归属感，有的公司待遇高，而有的待遇偏低。在美国，做外包是为自己而工作，有不少技术经验丰富的人很喜欢待在外包公司。因为在美国做同样的工作，在同一时间内，外包员工的工资比正式员工的要高。有时候，做 10 个月外包员工拿到的总收入就已相当于做 12 个月正式员工的总收入，故而在同等收入的情况下，你就多出了 2 个月的自由时间，有的美国人就用这 2 个月时间去全球旅游。美国人做外包还有一个特点，即只专注于一种技术，只做一种业务。例如，有的人 10 年来都只用 Perl 写自动化测试的脚本，有的人在 A 公司做 X86 的驱动，到 B 公司、C 公司还是只做 X86 驱动。这样他们因为常年专注于这一工具和这类业务，所以其劳动生产率比大部分人都高，同时自己的专业水平也越来越娴熟，经验不断地在一个领域积累，进而竞争优势更明显。这类只专注于某一个领域的外包员工在一段时间后大多都发展成为了这个领域的专家。

在中国真正优秀的外包公司通常给的待遇都比客户的正式员工高，如瞬联（先特别声明，笔者不是瞬联的员工也不是他的枪手，而是有同学在里面工作，大致了解这个公司的情况）。这家公司基本就是美国的外包模式：让你只专注于一个电信的研发领域，不会让你在不同的行业间换来换去，这样外包员工常常对某个具体的技术比一些正式员工的经验还要丰富、还要强。几年下来，在电信大行业中，这些外包员工就都有了一个非常专注的一技之长，做到“一招鲜”吃遍天。另外，有的外包公司也可以让你在一定聚焦基础上的广度扩展，如杰华科技。笔者的朋友在里面干了 2 年时间，虽然经历了 3 个不同的产品测试项目，但却都有一个共同点，就是都用 Tcl、Perl 和 Python 等脚本语言进行网络设备的自动化测

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

试脚本开发。在这 2 年时间内，他在网络产品的自动化测试技术方面长进不少，见识了不同厂家自动化测试的框架和实现方式。笔者认为他现在在中国任何一家网络产品公司从事自动化测试脚本开发，都是没有任何问题的。而在到杰华科技之前，他在一家做自主产品的公司从事的测试工作，只有机会从事 Tcl 脚本的开发，有时还被安排去做一些招标测试、售后技术支持等工作，简直成了一个杂家，根本无法达到在网络产品的自动化测试领域发展成为一个经验丰富的专家水平。

总的来说，好的外包公司既可以满足部分人成为某个专项技术专家的需求，又可以提供比正式员工还高的待遇。对于一个没有多少经验的人来说，先通过外包公司这个渠道来获得经验，也是一个不错的选择。

(2) 自主产品公司归属感肯定不是问题，待遇有高有低。最大的好处就是技术修炼上可以往更深的层次去发展，因为公司有着丰富的研发资源让你学习和交流，很容易成为某个产品的测试专家，前提是公司不会经常让你调换测试的产品。如果你的目标不是成为某项自动化测试技术的专家，产品公司或许能给你提供很多其他测试岗位尝试的机会。以网络电信设备的测试为例，公司可以给你提供从事系统测试、解决方案测试的选择，也有转岗到售前、售后技术专家的机会。如果你对一个产品的测试能保持较长的延续型，那么将比外包员工更了解产品的内部构造、实现原理，成为本公司产品专家的机会将会非常多。劣势：做同样的工作，可能收入没有外包的员工高。很难有机会去不同的公司体验不同的企业文化，自己的职业命运与本公司的命运联系得更紧密。

最后，总结一下外包和自主产品公司的特点。

- 如果你想成为专注于产品测试的专家，长期待在自主产品公司会更好。
- 如果你希望有强烈的归属感，待在自主产品公司较好。
- 如果你想从测试转换到别的岗位，自主产品公司机会更多。
- 如果你想要一种自由的工作状态，外包公司更适合。
- 如果你想专注于自动化测试领域成为专家，去外包公司可能会更好些。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

第7章 测试组织架构与测试管理



本章将对测试人员在测试团队中所处的价值和地位，通过与公司架构和军队组织架构进行类比，让测试人员直观地感觉到自己所处的位置和价值所在。同时，本章还会针对测试管理的现状和测试新人培训过程中容易疏忽的地方进行一定的经验分享。上一章的内容帮助测试人员看清未来的困惑，而本章将帮助测试人员看清当前的困惑。希望，测试人员通过这两章内容，不再对当前所担任的测试职业及其未来而迷茫。

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

本章的内容是将测试这一新兴行业的组织架构与公司架构、军队组织架构做一一对比，让所有在测试组织中工作的各类人才能够大致感知到自己在组织中所处的位置和重要性。

7.1 测试部架构与公司架构的对照

7.1.1 功能测试工程师——销售部

测试团队中冲到最前线的工程师肯定是功能测试工程师，他们就如同公司中的销售人员一样，执行着公司的策略，拿着公司后台开发出的工具，在软件上进行着冲杀。他们不断地努力找出更多的 bug，与各种预料到、未预料到的困难进行着斗争，最终的目标就是找出尽可能多的 bug，保证测试团队整体的绩效体现。其职责和作用，类似于公司的销售部——销售部执行着公司的测试，拿着公司提供的产品或一些销售资源，不断与各种销售困难做斗争，力争创造出尽可能多的销售额，让公司所有部门的劳动成果得到最大化的体现。

7.1.2 测试工具开发和自动化测试开发——研发部

虽然功能测试工程师的工作成果更多地需要依靠自己的努力、创造力、业务理解、测试经验来寻找产品的 bug，但是很多时候如果有一些锋利无比的利器可以帮助功能测试工程师，则他们的工作将事半功倍。测试工具开发和自动化测试开发就是这样一个能不断给功能测试工程师提供各种测试武器的助手，它的作用类似于公司的研发部，开发出各种很有市场威力的产品，提供给销售人员，帮助销售人员开拓更多的市场，能够在市场上更加所向无敌，开创出更多的销售业绩。

正如产品研发要结合市场需求一样，测试工具开发也需要和功能测试工程师们紧密联系，开发出的测试工具只有完全切合测试人员的需求，才能体现出测试工具开发的价值。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

7.1.3 测试设计工程师——公司的战略规划和市场部

公司的战略规划和市场部，是以全局的眼光来分析行业市场情况，并以知己知彼的战术思想来确定公司的发展方向和 market 策略，然后制定出合理的 market 策略和 market 计划交给强有力的执行部门来实现和完成。

而测试设计工程师的主要职责是以全局的眼光进行每一个项目的测试设计。为了帮助测试部门达到测试资源和测试时间的最佳利用效果，他们将在测试执行前进行测试策略的设计和规划，并将设计出的测试策略和测试用例交给一线的测试工程师来执行和实现。

因此，测试设计工程师的角色就是测试团队内部的战略规划部和市场部。

7.1.4 解决方案测试工程师和系统测试工程师——销售部中的高级销售人员

作为测试工程师中较高技术级别的解决方案测试工程师和系统测试工程师，由于比初级的功能测试工程师具有更全面、深入的测试经验和测试技能，因此他们主要是执行一些难度大而且重要的测试任务。其作用相当于销售团队中的高级销售人员，专门针对对公司重要而又难度较大的客户群开展销售工作。

7.1.5 测试领导者——公司的管理部门

作为公司的总经理或高级管理者需要管理和协调好公司的销售团队、市场团队、研发团队、财务、人力、行政等团队。测试部门的领导者正如测试团队的总经理，同样需要管理好和协调好测试的销售团队——功能测试工程师、系统测试和解决方案测试工程师；测试的市场团队——测试设计团队；测试的研发团队——测试工具开发团队和自动化测试开发团队。所以测试部门的管理者要像公司的总经理一样工作，需要思考各种团队的未来发展规划，平衡有限资源在不同团队中的投入，尽量不出现拖累测试团队整体发展的短板，确保其领导的测试团队是一个团结、高效的团队。

7.2 测试部架构与军队组织架构的对照



7.2.1 功能测试工程师——一线战斗的步兵

测试团队中数量最多的工程师肯定是功能测试工程师，正如军队中数量最多的是一线战斗的步兵。功能测试工程师团队是发现 bug 数量最多的测试团队，是整个测试组织所有团队如测试设计、测试工具开发等多部门劳动成果的转化体现。虽然，功能测试工程师入行的门槛要求比测试设计、测试开发要低，但却是成长为测试设计和更高层次测试专家的必由之路。甚至测试部门经理都必须要从功能测试工程师中通过实战，在一线战斗中积累业务经验和领导管理能力。因此，要想成为优秀的测试人才，要想从众多功能测试工程师中脱颖而出，必须像一个优秀的战士一样，既要有过硬的军事技能，还要有很强的综合素质，例如坚强、执着、团队精神、牺牲精神。

总地来看，功能测试工程师正如一线步兵一样虽然起点低，但是其中真正的优秀者却可以通过一线真枪实弹的磨炼，成长为优秀的领导者和高级的专业人才。

7.2.2 测试工具开发——军工厂给战士提供先进的武器

测试工具开发部门就犹如军工厂一样重要，很难想象一线的步兵能在没有武

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

器的情况下，徒手空拳就能取得好的成绩。同样，功能测试人员如果没有足够的测试工具，只靠人工进行有限的测试手段测试，几乎很难取得良好的测试成绩。因此，对于任何一个测试团队，都需要在测试工具上有所投入，如购买商用武器，或是有专人进行相关测试工具的开发。倘若一位功能测试人员发现所在测试组织没有计划为测试工具进行任何的投入，那么这个组织也不会是真正重视软件测试的组织。

7.2.3 测试设计工程师——军队参谋和各级指挥官

对于任何有一定经验积累的测试团队，肯定会从功能测试人员各自为战、自由发挥的阶段进阶为有计划、有策略的阶段。在每个测试项目启动前，测试设计工程师就如同军队中的参谋一样，会根据作战地图、作战情报、敌我态势优劣来制定本次战役的总体打法和指导方针，让一线的功能测试工程师用更少的时间找到更多的、重要的 bug。测试设计工程师将使得软件测试的过程组织是有章可循的，不再重复早期测试时测试人员如无头苍蝇一样，无方针、无计划地随机乱找 bug。

正因为从事测试策略设计的工程师，有着军中参谋一样的作用。所以，从事测试设计的工程师需要像军事参谋一样，在制定新的测试目标和测试计划前，要多了解测试目标内部和外部的情况，了解自身测试团队的人员资源情况、已有测试工具状况，按照知己知彼的方式来规划或建议如何开展本次测试项目的策略和方针，指导后续的测试执行工作，保证测试执行工作高效、科学。

7.2.4 解决方案测试工程师和系统测试工程师——军中的空军和海军

虽然解决方案测试工程师和系统测试工程师也是在测试一线利用手中的测试工具和一些测试方法直接寻找 bug，但是他们却在找 bug 的工作方式上与功能测试工程师有着显著的不同。通常解决方案测试工程师和系统测试工程师可使用的测试工具比功能测试工程师的工具更多、更先进。他们就类似于军队中有着特殊高级武器的空军和海军，当步兵出现了难以攻克的难关时，手中握有高级武器的特殊兵种就可以帮助其完成最后的任务。而实施解决方案测试和系统测试就是为了实现一些功能测试无法达到的测试目标，专门进行重点和难点的突破，提高整个

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

测试团队“最后一公里”的测试水平。

解决方案测试和系统测试对人员的要求相对要比功能测试高。首先，要求一定是要有足够测试经验的人；其次，要求具有能从全局的角度进行测试的能力；最后，要求能掌握和高效使用各种测试工具，并提出整体测试方案。空军和海军通常在选拔人才时的要求比步兵要高一些，要么需要具备较多的技术知识，要么就是有特殊的身体条件要求。因为这些特殊兵种都需要去完成步兵无法完成的一些特殊任务。所以，解决方案测试工程师和系统测试工程师在测试部门中的作用和要求就很类似空军和海军，人数不一定多，但是却是能对功能测试起到很好补充的一股力量。

7.2.5 自动化测试脚本开发工程师——军中的炮兵

自动化测试脚本开发工程师的劳动成果是大量的自动化测试脚本，这些自动化测试脚本能帮助功能测试工程师节省很多的时间和精力。在先运行完自动化测试脚本后，功能测试工程师就只需要对少量无法进行自动化测试的功能进行手工测试即可。因此，自动化测试脚本开发工程师就很类似于军中的炮兵，他们开发出的测试脚本就类似炮兵手中的炮弹。在战斗开始前，可先用大量的炮弹消灭一些阻力，这样后续的步兵再跟上，专门解决炮兵无法解决的一些问题。这种炮兵先行，步兵后动的方式，不但大大提高了完成任务的效率，而且节省了大量的时间和步兵资源。否则，全靠步兵一点一点地去解决困难，不但需要的人力资源非常多，而且完成的时间还会更长且效率不高。

“自动化测试+手工测试”的组合与“炮兵+步兵”的组合性质一样。自动化测试来批量地解决大量可以通过自动化测试实现的测试任务后，再交由手工测试工程师来完成自动化测试无法完成的任务。采用先自动化测试、后手工测试收拾战场的方式，不但提高了测试效率与测试准确率，而且也大大减少了测试时间和测试人员资源的需求。因此，进行自动化测试开发的工程师就类似于军中的炮兵部队，开发出的自动化测试脚本就是一颗颗威力巨大的炮弹，帮助手工测试工程师扫清前面的乱石。所以，现在很多公司的测试团队中已经组建和准备组建自动化测试队伍，以便提高整个测试团队每次测试时的效率。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

7.2.6 测试领导者——军中的军长和政委

在中国军队的管理体系中，通常有一个主抓业务的军长和一个主抓内部思想的政委。既然把一个测试部门比喻成一个具有完整组织结构的军队，那么这支测试军的领导者自然也就担负起了军中军长和政委的角色。首先，对内测试规范和测试流程的制定、各测试团队的发展规划、员工培训、员工思想沟通的工作就类似于军中对内主持内务的政委所担负的作用。其次，与各开发部门进行测试目标确认沟通、为每个测试项目进行调兵遣将、资源合理配置、最新测试技术发展跟踪，以及整个测试团队测试技术实力提升和对所有测试行为输出结果负责，其作用就更类似于军中主抓军事技能的主将。

没有上过前线的兵不是好兵，只有从一线实战中摸爬滚打出来的将军才是最了解队伍的将军，才是真正能打仗的将军。所以，要当好测试团队的领导者，最好是从第一线的功能测试工程师开始，从实战中成才、成功，才能当好这支测试队伍的军长和政委，这支队伍才会避免被一个纸上谈兵的领导所耽误，并成为一支真正的内部团结、对外业务能力突出的测试团队。

7.3 测试团队管理

笔者在测试这个特殊的团队中，多数时间是被领导和被管理，少数时间也从事过管理和领导工作。在工作中体会过民企的管理，也在美国工作过多年的几位测试总监和测试经理身边工作过几年。在管理与被管理中经历和了解了不少美国的优秀管理经验和中国式管理的经验，结合多方成功的和不成功的测试管理经验，以及许多测试同行的测试管理经验，特别是被管理经验，希望能与读者一起分享一些关于测试管理经验的观点。

目前，中国的现状是不少公司的测试领导者通常是由一些工作年限较短、人生阅历较少的人来担任。多数属于跑步上岗，而跑步上岗后又没有与管理相关的培训跟上，所以导致不少的测试管理者，都分不清楚做了测试领导工作后应该与以前做工程师有什么样的不同。很多测试领导者就这样在还未完全具备领导素质

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

的情况下，只因为早几个月熟悉项目，就被委任为带头人。结果，很多自己做人还不够成熟的人，在所谓的领导岗位上无意间伤害了组员的积极性，加速了团队成员的离职，让公司承担了各种招聘成本和项目风险。

那么什么才是领导者？简单地说就是：带领大家奔向成功，指导大家克服困难、取得进步的带头人。同时作为测试团队的领导者应该在言语上鼓励远多于批评，毕竟测试做久了，是容易让人感到气馁的工作（bug 越来越难找了）。这时更需要领导者，像关心自己家人一样关心自己的组员，只有组员成功了，才有团队的成功。

关于测试管理者与组员的团队关系，有这样一个观点：管理者与被管理者如同一对新婚夫妻，在年轻时因为自己的不成熟，因为自己缺乏足够的婚姻生活经验，很容易互相抱怨、埋怨而吵架。但如果双方能坚持互相理解、互相包容、互相扶持，总有一天，大家都能修炼成熟，吵架和埋怨就会越来越少，最后大家便能和谐地长期相处下去了。因此，当我们基层的测试人遇到很年轻不够成熟的管理者时，应该多一份理解、多一份包容，帮助他也是在帮我们自己。这里笔者谈一个亲身的经历：在笔者早期的一个测试项目中，笔者的 Leader 只早进入项目 1 年。有一次，在一个项目中由于销售压力，在时间、人力、资源都很紧张的情况下，在离最终目标日期还有四五天的时候，我们的 Leader 看着目前项目的进度，顶不住来自其他部门的压力了，变得脾气急躁，并直接表现在与我们的谈话和工作要求上，基本上小组的每个人都感受到了他脾气的变化。于是在午餐时，笔者特意坐到了 Leader 的对面，和他简单讲了一段话“越是困难，越是危险的时候，作为千军万马的带头人越要镇定，泰然自若。如果领军人表现出了恐慌、急躁、慌乱，肯定下面的部队也会恐慌、急躁、慌乱，更不利于大家一起齐心渡过难关”。下午，Leader 悄悄发了封 E-mail，表示很感谢中午给他的建议。很快他变得很镇定了，对组员们也不再表现出一副很心急的状态，并积极与每名组员进行沟通，以寻求好的解决方法。

给测试管理者的一些建议：

- 测试比开发更易困惑和消极，需要更多的鼓舞。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

- 测试比开发更易职业迷茫，需要更多的开导和沟通。
- 选择合适的人做测试。
- 要为测试人员争取应得的物质待遇。
- 测试工程师不受欣赏、不受重视，是测试经理的错。
- 测试部门不受重视是公司老板的错，一个对质量不够重视、不愿投入的企业，很难得到长期的大发展。
- 要敢于选拔培养人才，如果认为在自己的企业内不会出现英雄，那么就一定不会得到英雄。
- 当你的下属有更好的发展机会时，只要对公司有利、对下属个人有利，虽然有可能短期内自己负责的部门会受一定影响，还是希望能放下属一马，让他成功，也是让公司成功，并且自己也将多一个感恩的朋友。否则，为了自己部门的小利益，虽然留下了一个人在但心不在的员工，但是却有可能因为他内心的不满反而影响了整个测试团队团结的气氛，公司整体利益也可能受到一定影响。
- 在对人的管理上，要加强对下属的关心和爱护，多一些沟通和辅导，帮助下属提高技能和效率，培养工作中的自信心、成就感。
- 在平时的工作生活中，需要加强对下属的关心和爱护，包括针对个人的关怀。多一些沟通和辅导总是好事，总比高高在上，刻意去树立管理者威严，进而疏远了基层员工与管理者的关系要好。
- 在专业上，要经常帮助下属提高技能和效率，培养其工作中的自信心、成就感。应该多多给予赞许，不要吝啬给下属的表扬，甚至可以找理由来夸奖、肯定下属。

中国人总是喜欢批评而不好意思去表扬，我们这个传统习惯确实需要向美国人学习，在美国人的管理文化中“表扬”、“肯定”、“赞誉”的话语无处不在，而真正批评的话语则会非常谨慎、吝啬地使用。

笔者一直有一个观点：在一个项目中同事的关系永远只是暂时的，但大家脱离项目后，人与人之间的关系却可以是一辈子的，所以尽量避免在项目中出现影响人与人之间感情的事发生。测试团队的管理者除了需要熟悉测试的相关业务知

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

识，还应该做很多人的工作。既然是做人的工作，那么及时真诚地沟通和对下属的关怀将对鼓舞测试人员的士气起到非常大的影响。

美国人的管理观点：

多数测试人员都是执行经理，要以此为基础对他们进行管理，而不能像对工厂工人那样来对测试人员发布命令，更不能试图将测试人员的工作转为工厂化的流水作业，需要在工作中给测试人员发挥自己的创造力、主观能动性的机会。并要理解不同的测试人员有着不同的强项和兴趣，进行针对性管理。例如，有的测试人员喜欢写代码，就可让他多做一些自动化测试开发的工作或测试工具开发的工作，而不可强求他一直做一些手工测试的工作。而有的测试人员喜欢去思考，宁愿多创造一些新的测试方法，而不愿每天待在实验室中搭建环境，或只是执行他人的测试用例。管理者只有将测试人员的兴趣和特长与所分配的工作结合起来，才能让他们激发出工作热情，并乐于承担责任。

软件测试工作不像销售，可以简单地进行数据量化。有经验的测试人员都明白，测试人员努力工作的价值更多的是靠直接管理者的评价和反馈来体现。可是如何进行一个科学客观的绩效评估，对管理者则是头痛的难事。如果稍与测试人员的个人期望发生偏差，就很容易影响管理者与测试人员之间的关系，并打击为测试工作付出辛苦测试人员的积极性。借用美国的一些经验，在评价员工工作时可以进行如下工作。

- 阅读其错误报告，既检验其是否写得准确精炼，又可以判断是否找到重要的 bug 和较难找到的 bug。
- 阅读其自动化测试代码，可以了解其代码开发的水平。
- 阅读其编写的测试用例，是否是高质量的用例。
- 收集与其一起工作的开发人员、项目经理和其他相关人员的评价。
- 是否他的工作都按时完成了。
- 是否乐于帮助其他测试人员或开发者。
- 是否主动积极学习新的技能。

测试管理者可以针对以上几个方面，与测试者一起进行面对面的沟通，并双

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

方达成共识。考评的周期间隔最好在 1 个季度内，因为时间久了，有很多事实细节会因为双方记忆的不同，产生偏差。最终通过这些及时的双向沟通确认形成的绩效评价报告，将是测试人员心服口服的评估报告。这样的评估报告不会影响管理者与测试人员之间的关系，也不会影响测试人员的士气。

很多测试管理者在未升任管理职务前，都是对业务和测试技术非常熟悉的技术骨干。但由于作为管理者，需要花不少精力在沟通、协调、组织上，往往一段时间后，会渐渐跟不上业务的发展、测试技术的进步，并逐渐丧失了很多技术能力，而这也影响对测试人员技术工作的技术评估。因此，建议测试管理者可以适当安排一定时间参与到一个测试项目中，具体去了解公司最新的产品业务发展技术和测试技术。这样才能真正地明白，哪些工作难，哪些工作易，从而对测试人员技术工作给出客观、科学的绩效评估。

“士气”的重要性：

在任何团队活动中，“团队士气”都是非常重要的一股力量，甚至有时是决定性的力量。让测试人员感觉到自己的工作很重要，自己对公司是重要的资产，这样测试人员才会得到肯定、个人价值得到体现，进而充满士气。当整个团队失去士气时，责任肯定 100%落在测试管理者身上，因为测试管理者是士气最重要的保护者。

那么测试管理者应该如何来保护团队的士气呢？

- 不但自己要礼貌对待测试人员，尊重测试人员，而且要求其他部门的人也要礼貌对待测试人员，尊重测试人员。
- 让测试人员感觉到你在时刻关注着他们的工作，不要让他们觉得自己的工作无人关注。
- 经常赞誉做得好的工作，常表扬热心助人和诚实努力的员工。
- 如果测试人员任务完成得不顺利，就要及时派人进行指导和帮助。
- 多给测试人员提供培训机会，让他们感觉到公司很看重测试人员的培养和专业发展。
- 测试管理者要公平对待每个测试人员，并要求其他部门也公平对待他们。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

- 不要对测试人员大喊大叫，不要利用自己的权力强制要求测试人员接受自己的观点。
- 避免公开批评测试人员，但可在私下指出其错误和问题。
- 在工作量分配上，不要让测试人员被过多超额使用，这样会让测试人员太过劳累。
- 不要随意让测试人员加班，长期加班会使人体力透支，对公司不满，并导致工作低效。如果测试人员加班，管理者也应该陪着他们一起加班，让测试人员感觉管理者与他们一起在奋斗。
- 管理者应该主动为测试人员提供精神支持，并解决各种不公正的待遇问题。
- 管理者应主动去物质奖励测试人员，而不要永远只有精神奖励，或是对测试人员做出的辛苦贡献视而不见，无任何奖励。

如果一个测试管理者能够全部或大部分做到如上这些要求，相信绝大多数的测试人员都会认同管理者的工作，自然整个测试团队的“士气”也一定是高涨的。

最后引用来自互联网的一个建议：“因为测试经理和测试工程师是有本质区别的，也许一个测试经理在某些专业技术方面要比某个测试工程师差，但是测试经理关注的是更高层次的问题。比如，利用现有的资源，制定出符合公司发展需要的测试部门的发展规划；要考虑如何将测试技术应用在项目中而不会影响项目的进度；要考虑如何分配人手，如何利用有限的资源完成更多有效的工作；如何协调不同部门间，以及团队内成员间的关系和配合；如何完成剩下的工作；如何向公司高层解释现有的条件下所能完成的工作将帮助软件达到什么样的质量……从实践中慢慢总结，然后升华成理论，再实践。

前面讲了这么多的大道理，下面笔者很乐意把一次亲身经历和实施的一个测试项目的 Leader 经验贡献出来，以供大家参考。

案例

以前所带领的测试小组成员组成非常有特点，团队成员一共有 4 个：

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

小侯：学习能力超强，逻辑能力非常强，创新不是很多，心里不热爱测试。

小沙：工作态度最好，最积极、最勤奋，但创新和逻辑稍弱。

小朱：创新超强，只是有时看起来做得有些慢。

小马：热爱测试，创新能力和逻辑能力居中，不太爱集体活动。

每周一我会将上级分配下来的任务包分给每一个人，然后基本上至少每隔 2 天就会主动去了解他们执行的情况，如有困难希望他们早点提出来，但是我不会每天偷偷地站在他们后面偷看他们的工作。因为我觉得偷偷监视他人工作是对他们工作的不尊重，应该给予他们足够的自由，即使他在工作时间会开小差，只要能在计划前完成任务就可以。

中午吃饭时，尽可能和小组成员们一起吃饭，一起散步。多聊一些工作以外的话题，如他们的生活、租房等话题。

在工作上，当小组整体受到不公平评估时，就会努力去改变。曾有一次，测试部门准备试行以 bug 数来考评业绩的方法。由于每个组所承担的模块不同，有的组的模块比较新且功能多，复杂度高，很容易一天发现几个 P1 级别的 bug。而我们组所承担的模块，都是一些已经过 2~3 年锤打的老模块，而且功能点少，复杂度也不高，因此找 bug 的难度非常大，更不用说找 P1 级别的 bug。所以，当这项政策刚一出台时，我们全组组员都感觉对我们很不公平。因此，我当时就和我的主管进行了沟通，表达了我们的看法和观点。同时，回到组内还是动员大家先不管公司如何考评，我们先一起想办法，再梳理梳理我们手中的软件模块和每个最小的功能点，看看有无一些以前可能被疏忽的小地方，并从新的角度来找 bug。最后，我们讨论发现设备的 WebUI 部分平时大家不是很重视，找的 bug 较少。所以，我们又各自分头开始对各自负责模块的 WebUI 部分进行狂轰滥炸，甚至用出了一些非法手段来找 bug。新的 bug 被发现了，但是客观分析这些 bug 中的有些测试方法是不合理的，是为了找 bug 而找 bug，有些没有实际意义的 bug 只会是浪费开发人员宝贵的时间。而这种“为找 bug 而找 bug”的现象基本上出现在了每一个测试小组。也许是测试的领导们看到了，只以 bug 数论英雄的方式所带来的弊端，一周后测试部门中止了这种简单的绩效考评方式。我们团队又恢复到了往常的工作轨道上，在一块块金矿被挖走很多的地方继续寻找真正有价值的金矿。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

每周五 17:30 至 18:00, 我们小组有一个内部活动。在我们 5 个人的工作区随便找个地方, 大家围坐在一起, 既进行一周的工作总结也进行一些思想碰撞。工作总结, 主要是让每一个人都说说本周自己找了哪些自己认为比较经典的 bug, 可以是找的过程比较经典, 也可以是这个 bug 的重要性比较大。这样, 一方面大家可以从互相学习找 bug 的思路和方法, 另一方面可以给这些刚从学校毕业不久的年轻人进行技术表达能力和公众场所敢发言能力的锻炼。其间我遇到了这样一个难题, 由于小侯不喜欢做测试, 小马性格比较内向, 他们俩发言不是很积极。这时, 我会停在这里, 大家一起等他们发言, 哪怕即使只说几句不着边的话。只有每个人都发言了, 每个人才能融入这个团队。也不知现在小马和小侯能否理解当年我给他们提供的这样一个锻炼表达能力的机会。

小侯一直很想去开发, 我也私下悄悄给我的主管提出给他转开发的机会, 只是一直没有机会。只能在测试工作分配中, 尽量指派一些学习难度大的模块给他, 来满足他追求挑战性的需求。至于小马, 他很喜欢做测试这份工作, 愿意花时间来深究一些测试的内容。

在工作之余, 我们 5 个人也有集体娱乐活动。那完全是 AA 制朋友间的关系, 每次都是在一个固定的地方吃火锅、喝啤酒、开玩笑。现在我们 5 个人分别在 5 家不同的公司, 做着不同的产品测试、开发工作。虽然我们这个测试团队在一起的时间不是很长, 但是在团队解散后, 我们还是互相有所来往, 而且完全是朋友间的来往。

这段短暂的测试管理工作最让我满意的是:

- (1) 大家都成为了朋友, 多年还保持联系。
- (2) 在团队中推行了周五沟通会的活动。
- (3) 其间没有人离职。

(4) 我们不但按期完成了上级的工作安排, 而且还集体提出了不少创造性的新测试观点。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

7.4 测试新人的培训建议



在大多数的 IT 公司中，测试新人的培训一般有 3 个阶段：

- HR 的企业文化培训阶段；
- 测试部门业务培训阶段；
- “老人带新人”阶段。

在 HR 的企业文化培训阶段，大致会告诉新人这家公司的企业文化是什么、福利政策等内容。不过说实话，大多数公司这一阶段的培训对新人基本无影响，真正关于公司 HR 的内容还是需要后面的“老人”来告诉新人，在公司应该怎么样来做事，怎么样来做人，才会得到怎么样的成长机会和物质回报。

在测试部门业务培训阶段，有些公司会在这个阶段对所有的新人进行全部产品线的介绍，对一些公司公用的基础知识进行一些培训；而有的公司会将这个阶段合并到“老人带新人”阶段。大多数公司在这一阶段的培训以基础业务技术知识的培训为主，对于一个刚进入公司的测试新人而言，这时也许才真正感觉在培训。

如果企业能在这个阶段不单单进行技术培训，而是在培训过程中穿插讲解一些测试部门内部的工作成功事例，并通过这些事例告诉新人，在我们这个测试部门什么样的行为是被鼓励的，什么样的行为是被禁止的，以及什么样的人能得到肯定，老员工们是如何取得成功事例的成就，还有新人快速成才的案例，新人们

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

将会通过这些真实的事例感受到这家企业真正的企业文化。如果在培训阶段不把测试部门鼓励的正确做人的标准、正确做事的态度和方式告诉新人，去主动影响新人，则不但很难去影响新人的心，而且技术培训的效果也要打一定的折扣。笔者一向很反对给新人的培训只传授技术而不传授做人、做事的态度和方式。只传授技术，对新人而言只有一种冷冰冰的感受，势必会延长他们真正融入集体的时间。

在“老人带新人”阶段，基本上所有公司都会实行所谓的“导师制”培训方式。这个阶段的培训应该是整个新人培训计划中最重要，也最关键的阶段，甚至直接影响到整个培训是否成功。一位网友曾说过：“新人进入公司，你在观察他们，他们也在观察你。所以，你要先给他们做好表率。”也许这位网友在当新人时，曾遇到过不太合格的导师，所以，对于新人也在同时观察导师的体会非常有感触。有的公司在选“导师”时过于草率，标准是看谁最近比较有时间就让谁来带新人。另有一些公司在选“导师”时，直接认为只有小组的首席技术骨干才能带新人，可惜的是有的技术骨干的确个人在技术上很强，但是他们并不擅长将自己的技术传授给新人，或许需要新人有超强的理解能力才能理解他用一个业务熟练者的思维说出的专业术语。可能也有少部分技术强人，以技术定位置，由于心里看不起这个“新鸟”，于是很少主动去关心新人，只丢一些技术文档给新人，让他自学，遇到问题再来问他。“丢文档，自学”是一部分新人的不幸培训经历，对于这种不良培训现象的发生，测试管理者要负一定的责任。

虽然，企业在培训新人时有着选拔淘汰的需求，但是毕竟招一个人是有成本的，而且这个新人是通过了企业几轮面试下来的胜利者，如果轻易就将其在试用期淘汰出局，则意味着整个招聘流程中的每位面试官都有不称职的表现。所以，既然花费了大量的人力和时间把一个人招聘进来，那么我们就应该在他的试用期和培训期尽可能地帮助他快速成长和上手工作。并在培训期让他感受到我们这个测试部门大家庭的热情和团结，不可留下只是冷冰冰的一个人在一边读文档就算是培训的印象。这种不良的培训风格很可能会让这个团队以后人情味欠缺，团队无法形成一个整体，而出现新老员工的代沟。

任何新人刚入公司时肯定有很多都不懂、不会。因此在工作之余，主动了解他们在想什么，愿望是什么，希望学习些什么，生活上有没有什么困难，看看能

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

不能力所能及地帮助他们。除了极个别的人之外，大家的心，常常是可以用心来换的。不光给予培训和指导，同时要让他慢慢喜欢测试。如果经过培训后，新人说不出做测试的好处和测试对公司的价值，甚至还不知道测试为何物，那么就是培训的失败。至于，培训新人如何进行测试执行、测试用例设计、自动化测试脚本开发等这些测试内容，应按照各公司历来的培训传统来实施。

综上所述，在测试新人技术培训方面，每家公司都有自己的培训计划和使用的技术特点。本节无法针对所有公司提出一个通用的测试培训计划，只是希望对测试新人培训中有可能被大家疏忽的导师选择标准，做人做事态度的培训，如何让新人尽快地融入测试团队这3个方面，提出了一些观点和建议供大家参考。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

第8章 测试杂谈

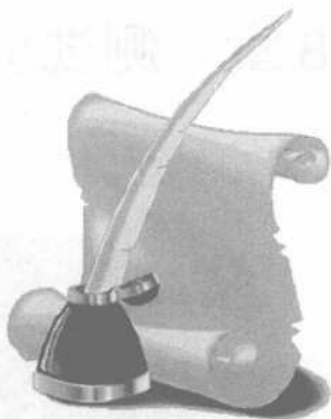


本章是把大多数测试人员所感兴趣的测试非技术话题集中起来，当读者在阅读前面几章内容感觉疲倦时，可以直接跳到本章来休息一下，换一下思路来品一些轻松的“测试咖啡”和“测试红牛饮料”。或者当读者在前几章依然没有找到自己心中期望解决的困惑时，可以到本章来试试，看是否能找到自己期望的答案。

最后，和大家分享一句话：“创新就像海绵里的水，只要你去挤，总会有。”软件测试是一个富有创造性的工作。

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

8.1 测试是科学也是艺术



测试工作到底是科学还是艺术？国内有的网友认为，测试只是科学，不是艺术；而有的网友认为测试是艺术，不是科学。

(1) 支持测试只是科学的观点

转载一段来自微软一位培训经理 William Rollison 的文章：软件测试曾在商业软件领域被认为是一种事后的亡羊补牢。初级的产品经理甚至开发人员认为软件测试是任何人都可以从事的工作。一些关于软件测试的书籍甚至冠以“艺术”、“手艺”之类的名字。这导致软件行业的某些读者误认为质量测试不是软件工程的一个学科。相反，通过缺陷鉴别、错误预防和合理风险分析的实测报告，有效的测试不仅是一门工程的学科，而且是提升软件质量和可靠性的关键部分。

(2) 支持测试只是艺术的观点

对于一个大规模的计算机程序，测试需要比设计程序更多的创造力。这个观点一直存在争议。Glenford J. Myers 在《软件测试的艺术》一书中支持了这个理论，他指出“测试是极度富有想象力和高智力的有挑战性的工作”。实际上，很多既设计又开发软件的人承认他们通过创造力的软件测试获得了更多艺术上的满足感。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

笔者在从事软件测试的前3年，从未想过测试是科学还是艺术，在笔者心中测试只是一个公司不可或缺的研发工作。但当几年前，第一次从一位有着15年美国测试经验的测试总监的培训中听到“测试是门艺术”的时候，心里受到了非常大的冲击，第一次听说软件测试是艺术，而之前身边从未有测试同行讨论过测试是科学还是艺术，更无法想象测试居然是门艺术。带着极度的好奇和兴奋，听完了这位美国测试专家的培训后，终于明白了测试是艺术的原因。从此，对测试工作的价值和意义有了重新认识，应该说对软件测试这种工作有了更高的兴趣和热情。

其实美国专家这句“测试是门艺术”的意思，并未将测试中具有的科学元素排除掉。软件测试过程中需要的数学基础和逻辑能力都是从事任何科学工作所必备的技术素质。计算机软件与科学假设类似，两者都固有易错性，软件测试过程的基本框架类似于科学假设的试验和错误实践。测试工程师通过严格的测试（主要用于证明缺陷存在）反驳无缺陷软件的假设，这种“造假”的过程和数据驱动的方法（如归纳和推理）形成鲜明的对照，数据驱动通过事实的重复来证明正确性。这些方式都是通过确认来支持论断的，通过使用特殊数据验证功能正常来证实不确定的无差错。这种方式仅能证明软件功能在某种情况下功能正常，它并不能证明软件无差错。而所有的这些思维方式也正是科学工作的思维方式。同时作者还有一种观点：诸如测试的规范、测试的流程、自动化测试技术这些知识，可以通过书籍达到自学提高的目的，而这也是大多数科学性质学科所具有的特点——任何人都可以轻易地通过大量相关书籍来学习和掌握大量的技能和知识。

可是测试中最核心的宝钻——测试设计方法，则基本没有一本书可以真正的教会你。虽然有的软件测试书中谈到一些所谓的测试理论如“等价类”等，但如果你真有3~5年软件测试经验，你会发现书上这些理论只会束缚你的思维，用处不大。事实上很少有公司会严格按照书中所谓的“等价类”等这些理论来设计测试方法。那么测试方法的设计靠什么呢？答案：测试经验和创造性思考。测试经验必须靠勤奋才能积累，而创造性的能力是不可能通过书本学到，或是听两次培训就具备了。下面还有一个很有力的证据来证明测试具有艺术的气质——管用的测试书好买吗？

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

无论是中国还是美国，如果你想自学任何开发的技能，基本上任何一类开发的书都可以买到很多，并顺利完成自学成为高手。但当你想学习测试，想了解测试，想买测试书时，才发现可以帮助你自学成才的书几乎买不到。因此，很少有人没有在正规公司做过测试前，对测试能有所真正的了解。那么为什么测试的书这么难买？我们是否可以这样来理解：能买到很多书的领域，通常人们都可以通过自学进入这个领域；而书很少的领域则注定很难自学成才，而必须依靠大量实践经验的积累和自己点点滴滴的思考积累才能有所收获。例如，音乐的作曲创作、作词创作是很难通过大量书籍让你自学成才、成名。测试方法和测试思维的精华就如同毕加索的作品、达芬奇的作品是无法看书能学到的，所以软件测试的工作性质基本和艺术创作是一个道理，其真谛只能意会，很难言传。

测试的流程和规范、自动化测试技术是测试科学的一面，它们能满足测试基本的需求。但如果想得到尽善尽美的产品质量，想不断地提高测试水平，达到高水平的测试效果，就需要通过挖掘测试艺术的潜质来实现，所以测试天生就具有艺术的细胞和基因。

因此，软件测试不仅仅具有科学工程的基因，也具有艺术的基因。现在一些国内公司在选拔和培养测试技术骨干时，都会强调测试骨干具有非常好的发散性思维，而发散性思维就是任何艺术活动必备的基本素质。所以，不管人们怎么看待软件测试的性质，如何定位软件测试人员的地位，任何从事软件测试的人都应该清楚地看到自己所从事的这份职业对人综合素质的要求，既需要我们有一个严谨的科学工程头脑，又需要我们有一个艺术创造的思维习惯。如果测试人员能理解软件测试是一门既具有科学性又有艺术性的工作，就不会再抱怨工作的无聊、无趣和没意义了。

8.2 测试人员真正做什么

当我们让测试工程师描述他们的工作的时候，大多数人的反应是“破坏”软件。也许这就是为什么一些人错误地认为测试人员的工作就是破坏的过程。其实，

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

测试人员并不真正“破坏”软件，而是通过观察测试的结果、逻辑推测和实验来暴露早已存在的 bug。Glenford J.Myers 在《软件测试的艺术》一书中对软件测试的描述如下：

- 测试用例中一个必需部分是对预期输出或结果进行定义。
- 程序员应当避免测试自己编写的程序。
- 编写软件的组织不应当测试自己编写的软件。
- 应当彻底检查每一个测试的执行结果。
- 测试用例的编写不仅应当根据有效和预料到的输入情况，而且也应当根据无效和未料到的输入情况。
- 检查程序是否“未做其应该做的”和“做了其不应该做的”。
- 应避免测试用例用后即弃，除非软件本身是一次性软件。
- 保留测试用例，当程序其他部件发生改动后重新执行，也就是所谓的“回归测试”。
- 计划测试工作时不应默许假定不会发现错误。
- 程序某部分存在更多错误的可能性，与该部分已发现错误的数量成正比。
- 测试总是倾向于聚集存在，故对容易存在错误的部分进行额外的测试。
- 软件测试是一项极富创造性、极具智力挑战性的工作。

由此可见，发现错误只是软件测试的一个方面。测试工程师通过多方面的努力，使得在开发过程中体现出测试的重要价值：验证软件，发现软件的错误，对开发人员提供信息避免 bug，给管理人员提供信息来评估风险，并且最重要的是扮演一个用户的角色。在软件中发现隐藏很深的错误是这份工作对智力的最高挑战，需要有很强创造力的人才能胜任。这就是为什么测试人员在经过富有创造力的攻击测试，发现一个隐藏很深的错误之后流露出兴奋神情的原因。

再来看两个比较流行的软件测试定义。

第一个：1983 年 IEEE 提出：软件测试是使用人工或自动手段来运行或测试某个系统的过程，其目的是检验它是否满足规定的需求，弄清预期结果与实际结果之间的差别。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

第二个：Glenford J. Myers 提出：程序测试是为了发现错误而执行程序的过程；好的测试方案是极可能发现迄今为止尚未发现的错误的测试方案；成功的测试是发现了迄今为止尚未发现的错误的测试。

这两个定义是现在对软件测试的比较普遍的认识，但是这两个定义对现在的测试来说都是不能很精确地解释测试的真正工作性质的。

首先，一个理想的软件测试贯穿在整个软件测试过程中，也就是说，在软件生命周期的初期软件需求分析阶段，软件测试就应该进入软件生命周期。

其次，就测试技术来说，包括动态测试与静态测试。动态测试进行时必须运行被测软件，这符合以上两个定义。但是对于静态测试，它主要包括人工静态分析和技术评审。而在人工静态分析中，主要包括两项工作：代码走读与正规检视。代码走读主要对象就是代码，主要关注的就是设计文档，有时可以是代码，然后对它们的规范化进行检查。技术评审，一般针对的对象就是 SRS、LLD、HLD、计划、测试方案等文档。由此看来，在静态测试的主要工作中，程序或系统的运行是没有必要的，也是不可能实现的。

最后，软件测试是通过采用一定的技术，运用一定的手段，结合科学的流程管理使软件产品缺陷尽可能减少，功能尽可能接近我们预期目标的过程。

通过上面这些内容，我们不难理解对于软件测试而言，技术是根本。我们只有具备了成熟的技术作为工作的依据，在做工作时才知道怎么做、如何做。无论手动测试或自动化测试，都是建立在技术基础之上的，目的只是为了运用已有的技术，最大化地发挥出技术的功效，例如，运用自动化工具可以有效提高工作效率。流程是基础，只有一个科学有效的流程管理，测试工作才能有效进行，才知道什么时候做什么，这时期该做什么，不该做什么。测试的最终目的就是产品缺陷尽可能少，尽可能接近目标。

测试的第一步应该怎样开始？

首先，测试人员先进行充分的技术准备和尽早进入项目开发早期阶段。如果我们缺乏充分的准备而是“临阵磨枪”，则往往会“手忙脚乱”，因而也无法达

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

到预期的测试效率和测试质量。为了提高测试效率和测试质量，测试人员应尽早参与项目，做好前期准备。前期准备包括熟悉需求、了解产品特性、准备测试数据、熟悉开发团队成员等方面。测试人员一定要提前规划好自己的时间，让自己早熟悉、多熟悉项目各方面的情况。实践经验表明，测试人员越早介入项目，后续测试工作就会越有序和顺利，测试效率和测试质量也就会越高。

其次，认真组织测试用例评审。产品测试实际上就是运行产品，执行已经准备好的测试用例。虽然每个测试人员也可能会根据自己的经验临时准备并执行一些用例，但是评审过的测试用例在很大程度上决定了缺陷被发现的数量和质量，即测试用例的质量直接影响到测试质量。保证测试用例的质量，最有效的办法就是对其进行认真而严格的评审。测试经理可以自己组织，也可以申请项目经理组织相关人员对测试用例进行评审。千万不要因为工作忙而不进行评审，这样只能是“欲速则不达”；也千万不要为应付公司的项目管理制度而走过场，这种“自欺欺人”的做法除浪费项目时间和成本外，无任何作用。

最后，在平时的工作中，积极配合开发人员工作，努力赢得开发人员支持。为了有利于测试工作，测试人员需要在手工测试正式开始前，主动和开发团队沟通项目的进度、项目存在的问题、项目的需求变更等情况。总之，测试人员与开发团队成员沟通越充分，对项目的信息收集和把握越及时、准确，我们的后续测试工作才可能做得越好，才可能提高测试效率和测试质量。

那么，在测试过程中，我们应该优先考虑哪些问题呢？

第一，测试工作最重要的一件事就是要考虑到所有的出错可能性。同时，还要做一些不是按常规做的、非常奇怪的事。说起来可能不太好听，测试的过程就像黑客攻击过程那样。像黑客这样的人是最好的软件安全测试员，他们专门找软件的漏洞，从而破坏这个软件，这样就可以修复这些漏洞来保证软件的功能。如果找不到这种漏洞，那就说明该软件质量已经很好了。

第二，除了软件漏洞之外，测试还应该考虑性能的问题，也就是一定要保证软件运行得很好，非常快，没有内存泄漏，不会出现越来越慢的情况。例如，

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

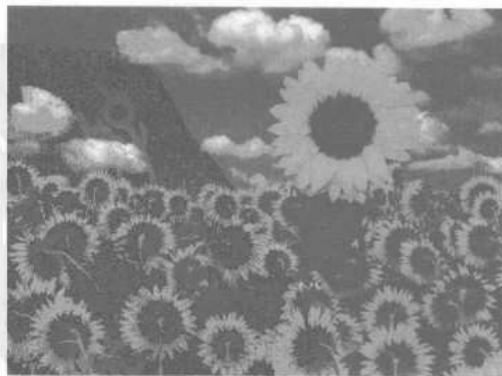
Windows 平台的软件可以在不关机的情况下,与其他软件一起持续运行一个多月,看看是否会出现越来越慢的情况(当然必须保证其他软件是没有问题的)。测试网页浏览器时,让它 72 小时连续不停地打开不同的网页,处理几万个不同的网页,而且速度不能减慢。有许多软件,当只有一两个人用的时候,可能感觉不到什么问题;而当几百个用户一起用的时候,有的网站就出现各种各样的异常,这就是测试工作还比较欠缺的缘故。

第三,软件测试还要考虑软件的兼容性。一般来说,一个软件是由许多小模块构成的,如果其中一个小模块与它的前一版本不兼容,那么这个软件就会出现错误,这种错误需要通过测试来发现和解决。

软件测试人员除了找 bug,还需要定位 bug 吗?

在一般的研发团队中,并不需要测试人员定位 bug 的具体位置,所以,对测试人员的要求并不高。只要你愿意学,测试工作是比较容易做的。但是,一些一流的公司就对测试人员的要求非常严格,不仅要找出 bug,还要定位引起此 bug 的本质原因。然后将这些信息交给开发人员,后者就可以很快更正,省去了他们找错误出处的时间。很显然,对于测试人员的要求是非常高的。不过幸运的是,大部分的测试人员只需要定位出重现 bug 的方式,不需要定位到 bug 原因。

8.3 软件测试的前景



软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

明天是不可预知的未来，而生命的精彩就在于它的不可预知性，所以许多人都喜欢在一些特定的日子去展望一下未来，软件测试的朋友们也喜欢给自己的明天描绘一下蓝图。“在国内软件测试作为一个崭新的行业发展还是很快的，大家对其发展也都抱着十分乐观的态度。毋庸置疑，从事软件测试工作的人员会有一个非常光明的明天。在看到美好前景的同时，我们也应该冷静地思考一些问题，比如软件测试行业如何发展？将会发展成怎样？制约它发展的因素有哪些？它的发展对每个从事软件测试工作的人将产生什么影响？等等。”

假如存在没有任何错误的程序，那么世界也会不复存在——因为修正错误而存在，这就是软件测试工程师的存在之道。虽然软件测试不是解决错误的根本举措，但却是必需的手段。也就是说，软件测试工程师在一家软件企业中担当的是“质量管理”角色，他的职责是及时纠错、及时更正，确保产品的正常运作。

著名软件测试专家、清华大学郑人杰教授说：“软件测试工程师是一个越老越吃香的职业”。郑人杰教授认为软件测试工作是对质量的把关，其中包含技术及管理等方面的工作，工作相对稳定，对年龄没有限制。而且随着项目经验的不断增长和对行业背景的深入了解，会越老越吃香。对于郑人杰教授对测试员职业寿命的看好，资深测试专家、北大测试负责人肖睿非常赞同。他认为软件测试人员的职业生涯如同一名医生的职业生涯，随着职业阅历和临床经验的丰富积累，到一定的年龄他们通过“望、闻、问、切”就能知道毛病出在什么地方。

因此，有人说软件测试员和医生是最不需要用“青春”来保证和延续自己职业寿命的职业。20多年来郑人杰教授一直在软件测试领域工作，如今70多岁的郑教授还在为软件企业“坐诊”。作为我国软件测试行业的“泰斗”，他见证了中国软件测试业20年来的发展历程。软件测试工程师地位翻天覆地的变化，源自信息产业的发展，以及软件企业自身、用户成熟度的不断加深。郑人杰教授表示：一方面，计算机使用越来越普及，越来越多的领域使用了计算机，特别是一些重要领域如国防、银行、金融、通讯、航天等，它们对软件质量要求很高。另一方面，一些重大事故的发生，也引发了人们对软件质量的关注。如2002年欧洲载重10吨的阿丽亚娜5型火箭发射失败，最后证实是软件

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

质量问题。还有国内的一些银行金融系统，因为软件质量问题不得不暂停营业。毋庸置疑，在经历了长期的不为人知和可有可无后，软件测试工程师目前已变得炙手可热。随着中国软件市场的发展，越来越多的国外资金投向中国软件行业。据报道，中国软件外包市场的潜力和机会已远远超过软件王国印度，不过由于软件人才的严重不足致使我国软件发展遭遇“瓶颈”。国家为了大力培养软件人才，不断采取积极有效的措施。

同时在大多数公司软件测试也是个可以很快入门的职业，门坎不高。而对于具有多年编程经验的程序员，如果改行做测试，更容易从系统的角度扩展自己的知识面。在北美，软件测试工程师因不同的级别，获取的薪资是不一样的，特别是近几年，由于越来越备受重视，所以薪资也节节高升。现在起薪已经在 5 万美元左右，若经验丰富的话，薪资可以拿到八九万美元。实际上，现在就业市场上最难找到的不是最佳程序员，而是最佳软件测试工程师！因为面对软件的规模越来越大，应用的复杂度和集成性越来越高，各种新的软件开发技术不断应用，特别是一些重要领域如国防、银行、金融、通讯、航天等，它们对软件质量要求很高，使得在这些行业软件测试工程师已变得炙手可热。巨大的市场空缺，使软件测试工程师作为中国近几年才兴起的职业，一下子成为了需求量非常大的新型职业。所以作为一名软件测试工程师，未来的发展空间是非常广阔的。

8.4 测试人员市场需求现状

2006 年的无忧指数显示，软件测试工程师已经成为最紧缺的人才之一，该类职位的需求主要集中在沿海发达城市，其中北京和上海的需求量分别占 33% 和 29%。而从企业分布来看，民企需求量最大，占 19%，外商独资欧美类企业需求排列第二，占 15%。目前，国内软件测试人才紧缺的现状已经凸显出来，如图 8-1 所示。曾经有媒体报道过近来软件测试工程师在职场需求中的风光景况，尽管 IT 行业的总体需求仍然疲软，但在北京和上海等地，测试员的需求量占到了招聘总量的近 1/3。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

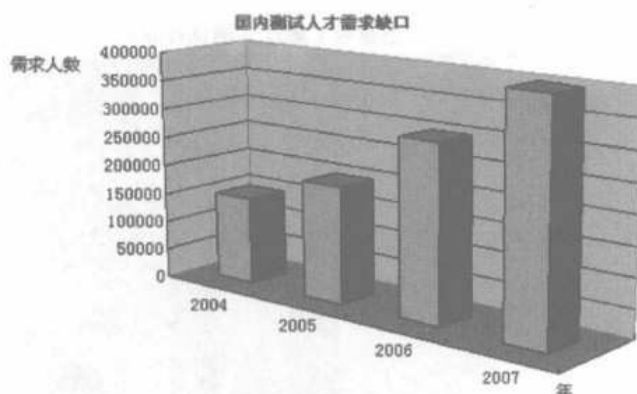


图 8-1 国内测试人才需求缺口

下面引用一份来自智联招聘 2007 年关于软件测试行业的报告。

软件测试行业认知态度报告

软件测试行业作为一个快速发展的新兴行业,在广大职场人心中的认知程度到底如何呢?智联招聘(www.zhaopin.com)特别推出了软件测试行业认知程度报告。

1. 行业认知程度一般

- 职场人对于软件测试行业认知程度一般

调查显示,职场人对于软件测试行业认知程度一般。选择非常了解的只有 8.8%,可见,对于该行业了解程度高的职场人不足一成,如图 8-2 所示。选择“知道一些”的不足四成,其余超过半数的职场人都选择不了解甚至没听说过。可见,软件测试行业作为一个新兴行业还没有受到广泛的社会认知,还有很多职场人对于这一行业不甚了解。

在对软件测试行业的认知程度上,理工类背景的职场人比文科和医学背景的职场人要高得多,而理工类计算机专业相关背景的职场人对于软件测试行业的认知程度又明显高于理工类非计算机专业背景的职场人。可见,目前对软件测试行业的了解还主要集中在 IT 相关背景的职场人中,而其他专业的人对此行业的了解程度还远远不够,如图 8-3 所示。

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

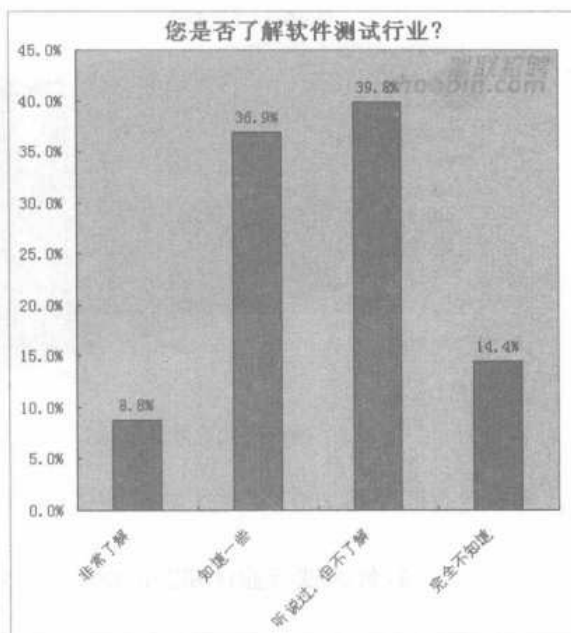


图 8-2 认知程度调查

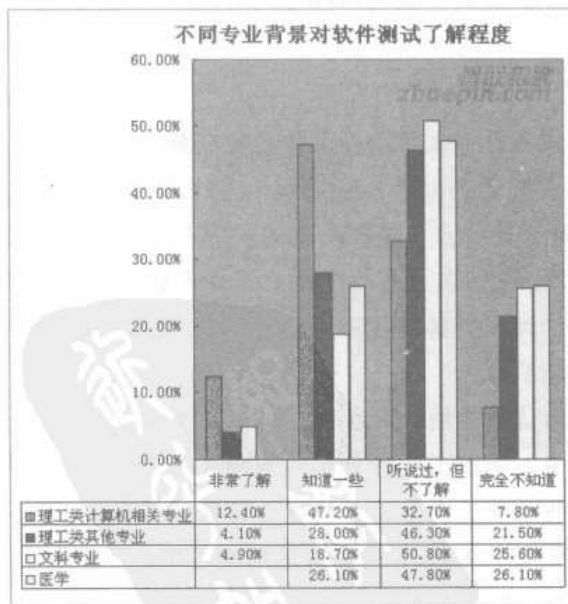


图 8-3 不同专业背景对软件测试了解程度

软件测试精要

每月及時觀看電子月刊書籍
就上溜客安全網 www.176ku.com

调查显示,伴随着学历的提高,对于软件测试行业的认知程度也明显提高,二者呈正相关关系。高中及以下学历者对于该行业非常了解和了解一些的只有两成,大专学历超过了四成,大本学历的职场人这两部分相加超过了半数,而研究生及以上这两部分人相加则超过六成,将近70%。可见,高学历背景职场人对于软件测试行业认知程度高,如图8-4所示。

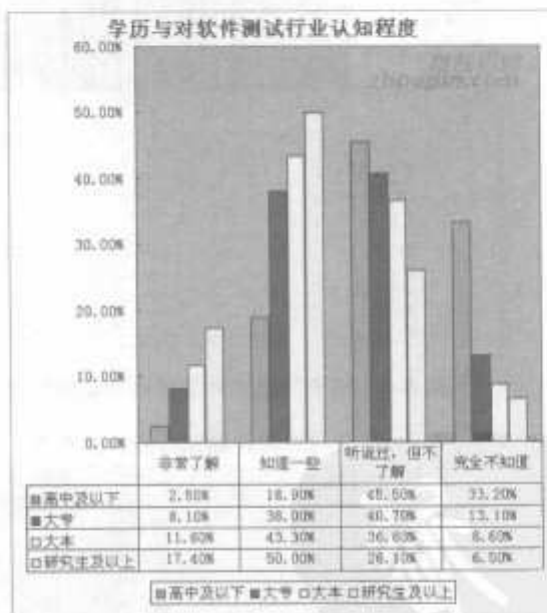


图8-4 学历与对软件测试行业认知程度

从城市分布来看,大城市和东部沿海城市对于软件测试行业认知程度较高。对于软件测试行业认知程度较高的城市除北京以外,主要集中在东部沿海地区,如南京、深圳、珠海、广州等,而在中部只有武汉比较突出,西部则西安对于此行业的认知程度较高,如图8-5所示。

关于行业,从事IT行业的职场人对于软件测试行业的了解程度远远高于非IT行业的职场人。但是值得注意的是,有近半数的非IT行业从业者对于软件测试行业听说过但不了解。可见这一行业目前越来越深入人心,只是更多的人还不了解该行业的具体内涵,也充分证明伴随软件测试行业的认知程度不断提高,会有越来越多的非IT行业从业者了解该行业,如图8-6所示。

每月及時觀看電子月刊
就上溜客安全網www.17

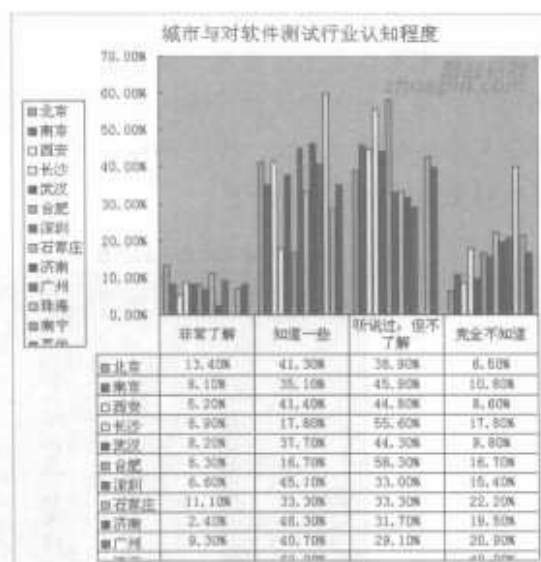


图 8-5 城市与对软件测试行业认知程度

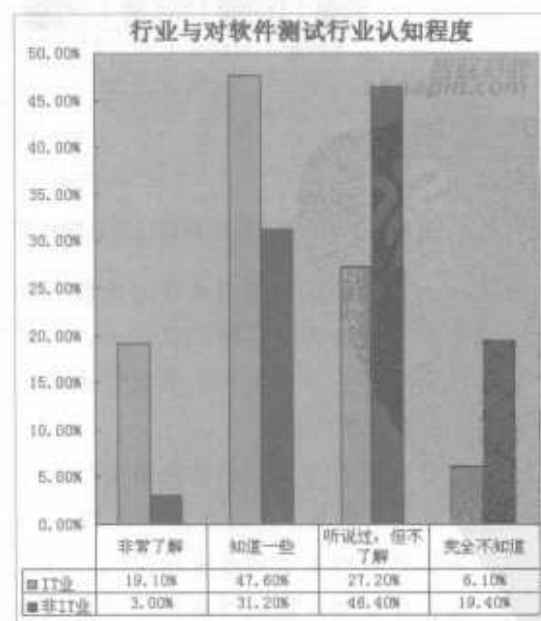


图 8-6 行业与对软件测试行业认知程度

软件测试精要

每月及時觀看電子月刊
就上溜客安全網www.17

2. 软件测试的专业性最被认可

对于听说过软件测试和了解软件测试的职场人来说，提到软件测试首先想到的就是“专业”，认为“专业”的职场人超过了半数，可见软件测试行业的专业性最被职场人认可。其次，被提及最多的特征是“高薪”，在一部分职场人心目中，软件测试行业拥有令人羡慕的薪酬水平。当然，也不乏会有人认为软件测试工作是一项枯燥的工作。尽管对于软件测试行业的态度褒贬不一，但是正面的评价明显多于负面。总体来说，职场人认为软件测试行业是一个专业、具有较高薪酬水平、较好发展空间的行业，如图 8-7 所示。

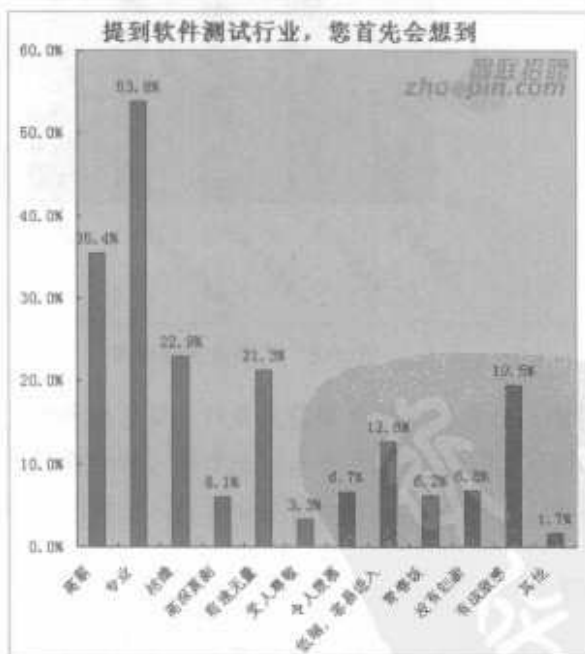


图 8-7 对测试行业态度调查

• 四成职场人 IT 行业首选软件测试

超过四成职场人表示，如果选择 IT 行业工作会首先考虑软件测试，软件测试在 IT 行业中排在首位。首选软件测试行业最主要的一个原因是认为软件测试行业前景好，选择这一原因的人超过了半数；其次是薪酬高也超过了四成；其他首选

软件测试精要
每月及時觀看電子月刊
就上溜客安全網 www.17173.com

软件测试行业的原因还包括发展空间大、个人兴趣，以及专业性强等，如图 8-8 所示。

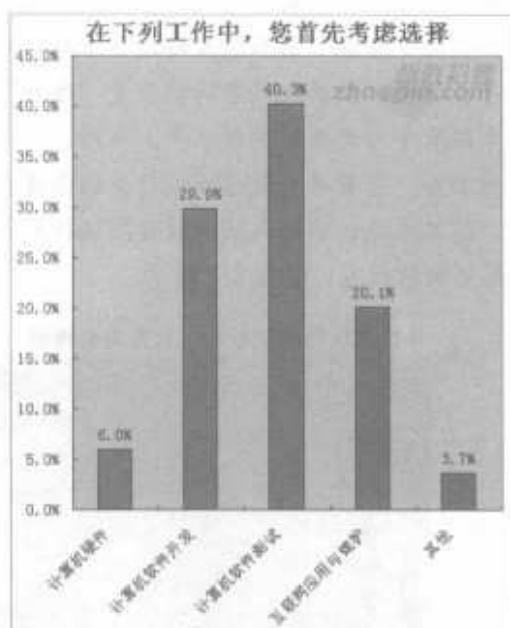


图 8-8 IT 职业选择调查

选择软件测试行业的主要原因是前景好、待遇好和空间大。尤其是关于行业的前景，在首选软件测试的人群中超过半数的人都选择了前景好。可见关于软件测试行业未来发展的前景，广大职场人是充分认可的，如图 8-9 所示。

3. 成就感

- 薪酬、技术含量与创新成果被认为是具有成就感工作的三大要素

职场人对于具有成就感的工作界定不尽相同，但是有三个要素的选择是相对集中的，其中最集中的要素为薪酬高，超过半数职场人认为这是工作具有成就感的重要要素。其次是技术含量高，选择也超过四成，接近半数。选择“不断创新”的职场人也超过了四成。可见，薪酬、技术含量和创新成果是职场人认为工作具有成就感的三大核心要素，如图 8-10 所示。

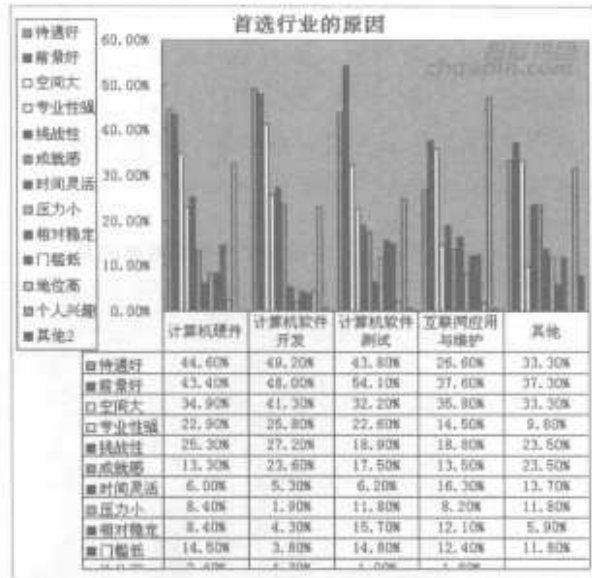


图 8-9 首选行业原因调查

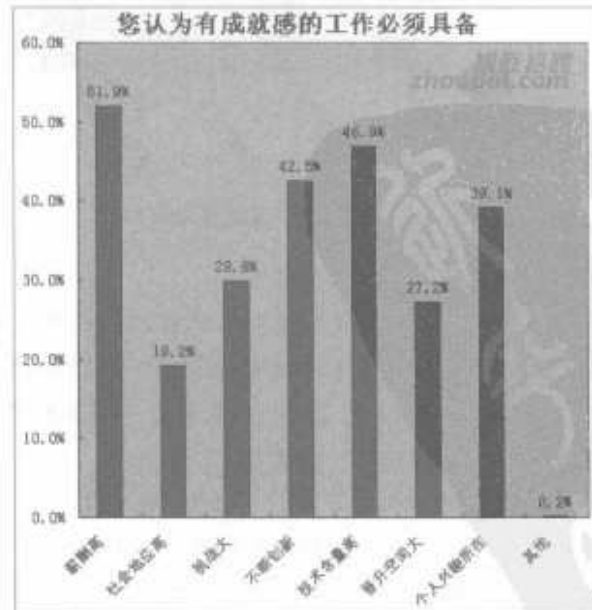


图 8-10 成就感调查 1

软件测试精英

每月及时观看电子月刊
就上溜客安全网www.17

- 超过半数职场人认为在 IT 行业中计算机软件开发是最具成就感的工作

在 IT 行业中，超过半数职场人认为最具成就感的工作是计算机软件开发，其次是计算机软件测试，但是比例一下子降到三成以下，与软件开发的选择比例还是相差较大。接下来是互联网应用与维护 and 计算机硬件。计算机软件开发在成就感方面的优势还是相当明显的，如图 8-11 所示。

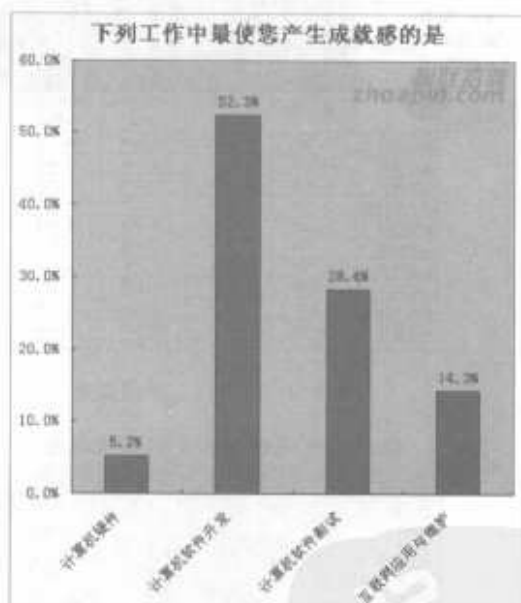


图 8-11 成就感调查 2

- 软件测试行业从业者成就感受

软件测试行业从业者对于软件测试行业的成就感还是比较认同的。在软件测试行业成就感量表中，对于 11 个成就感的描述每一项的最高分为 5 分，最低分为 1 分，其中最低的一项平均值也有 2.9 分，最高的一项达到了 3.7 分，成就感量表总体均值为 3.38 分，成就感较高，如图 8-12 所示。

在 11 项成就感描述中，均值最高的一项为“我在工作中有机会学习和成长”，其次是“我的工作对公司有价值”和“我的技能胜任目前工作”。可见，从业者对于软件测试行业的成就感也主要集中于成长空间、专业技能及发展前景上。

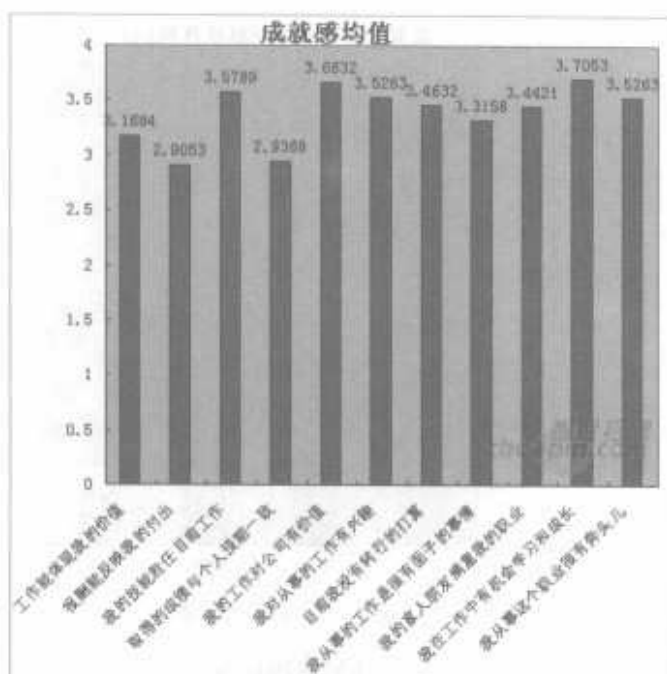


图 8-12 成就感均值

4. 软件测试行业对于人才具有较强吸引力

- 软件测试行业对 IT 行业从业者具有较强吸引力

调查显示,对于 IT 行业非软件测试从业者而言,软件测试行业依然具有较强的吸引力。在 IT 行业非软件测试从业者中,70%的人表示自己会考虑向软件测试行业转行,16.5%的人表示还没想过,只有 13.5%的人明确表示不会考虑。可见,软件测试对于 IT 行业内的其他从业者的确具有较强的吸引力,如图 8-13 所示。

- 专业门槛是阻碍更多 IT 行业从业者进入软件测试行业的障碍

对于不打算转行做软件测试的 IT 从业者来说,他们认为阻碍他们转行的主要原因是缺乏该职业专项技能。除此之外的另一个主要原因,则来自于个人兴趣,超过一成的 IT 从业者表示自己之所以不选择软件测试行业是因为没有兴趣,如图 8-14 所示。

软件测试精英

每月及時觀看電子月刊
就上溜客安全網 www.17173.com

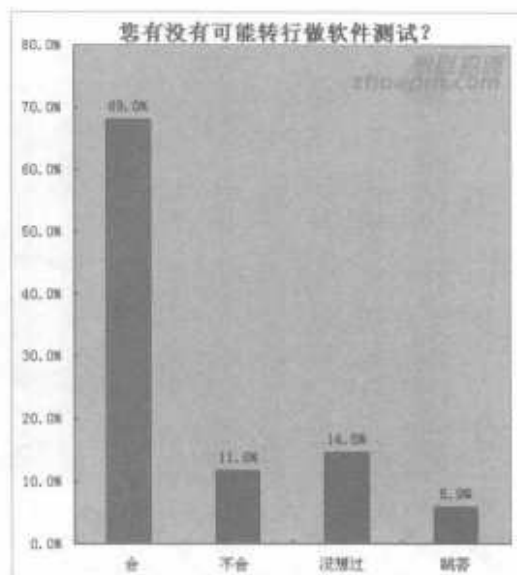


图 8-13 吸引力调查

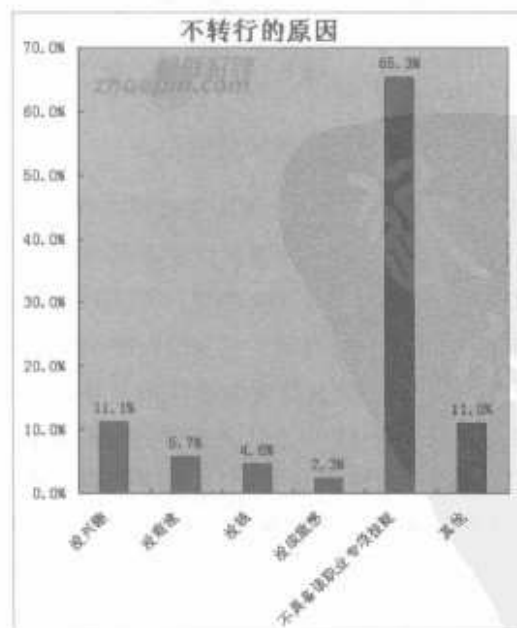


图 8-14 不转行原因调查

软件测试精要

每月及時觀看電子月刊
就上溜客安全網www.17

可见，软件测试行业是一个高速发展的新兴行业，尽管当下认知程度不高，但有着广阔的发展前景。伴随着广大职场人对于此行业认知的不断提高，必将带来行业人才流入的热潮，会有越来越多的人愿意投身于软件测试行业。

8.5 测试中的困难



本书在让读者逐渐认识到软件测试的价值、意义和前景的同时，还有必要再介绍一下在软件测试工作中可能会遇到的一些困难，让读者对软件测试这一职业会遇到的困难有一定的心理准备。

第一个困难是：因为有些组织的领导人不了解软件测试做什么，在他们的心中把软件测试定位于传统工业时代中工厂检测的地位，几乎从未想到要开发高水平的软件，必有高水平的测试人员。所以，对软件测试的认识产生了不少偏见和歧视，甚至认为是测试人员在耽误和阻挠软件产品的按时发布。如果发布的产品有缺陷，测试人员就应该负全责，结果很自然地导致了测试人员在这种企业工作时工作热情受到很大打击，造成人员流失，并直接影响到产品质量。

那么测试人员不幸遇到了这个困难应该怎么办？首先，应该坚定信心，相信软件测试工作一定会越来越多地得到重视。在平时的工作中端正心态，积极提高

软件测试精要
每月及時觀看電子月刊
就上溜客安全網www.17

技术水平，用实际行动证明我们的价值。其次，积极向领导讲述测试所带来的价值和意义，希望转变领导的错误观点。最后，如果实在得不到重视和认同，就重新找一个尊重软件测试的企业，而没有必要选择离开软件测试这个行业。

第二个困难是：经常由于测试任务繁重，进度压力大，以及不科学、不规范的计划，导致软件测试人员常常需要加班来完成工作任务。其直接后果就是有些测试人员长期处于过度疲劳的状态下进行工作，导致仓促交付工作，工作本身的质量下降。面对这个困难，软件测试人员应该怎么办？一方面，需要向上级及时进行沟通，反映实际困难和可能遇到的风险，协助制定出合理、科学的任务计划。另一方面，软件测试人员也要注意自我调节，注意劳逸结合。通过两方面共同努力，来减少测试人员加班的情况，避免长期处于过度疲劳的工作状态，不再疲于奔命，提升工作质量。

第三个困难是：软件测试人员期望按用户的真实需求和真实问题来测试，可是却没有真正做到用户之所想的测试。究其原因，是因为公司缺少提供测试人员直接与用户接触的时间和机会。这个困难所带来的结果是，测试人员虽然很希望急用户之所急，保障用户在真实应用中没有任何质量问题。但是却在测试过程中对用户期望行为产生了判断错误，遗漏了重要用户的使用场景。那么解决这个困难的方式应该是，测试人员主动积极要求公司进行市场调查，了解用户的真实应用模式；并提供资源让测试人员通过积极参加用户产品试用研究，研究用户发现的问题，帮助产品售后服务等方式来尽可能地了解用户的使用方式和常见问题。

第四个困难是：老板对自动化测试不够支持。因为老板一方面没有认识到实现自动化测试的必要性和重要性，认为手工测试都忙不完，人手不够，再搞自动化测试就会更加人手不够；另一方面担心实施自动化测试会投入过多的费用购买自动化测试的辅助工具。但是我们自己心里明白，不实施自动化测试，就会一直让手工测试忙不完，很难保证已测试过功能的回归测试质量。所以，为了解决自己的痛苦，同时提高产品测试质量，测试人员一定要争取越早得到老板对自动化测试的支持越好。

第五个困难是：如果遇到完成测试所需的工作文档不全的情况怎么办？如缺

少功能设计文档；功能设计文档不够详细；遗漏部分重要文档；缺少测试规范和案例；现有测试文档不够详细。对于测试人员而言，由于缺少某些重要的测试参考文档，将很难判断和估计测试范围、所需时间。而且由于没有完整的参考文档，就等于测试时没有参照可循，没有标准，测试的质量只能完全依赖于测试人员自己的专业水平和对产品的理解。对于这种公司文档管理规范的问题，测试人员只有主动积极地通过自己的主管去相关部门沟通，争取完善全面的测试文档，才能解决这个测试工作中重要的绊脚石。

8.6 测试经验杂谈

按测试用例完成了功能测试，只是测试走完的第一步。

测试人员在忙碌后，要适当休息一下，自学一些产品相关的开发知识。例如，如果产品是基于 VxWorks 平台实现的，则可以看一本介绍 VxWorks 开发入门相关的书；如果是基于 Linux 实现的 IP 包过滤，则可以看看 Linux OS 的代码结构、大致的数据结构，了解一些 IP 包转发的代码实现规律，修炼内功。

- 凡是有缺陷的产品，你不测试它，它就永远有 bug。
- 测试不是女同事的专属地，男女思维有别，测试最好能男女各项半边天。
- 无论测试设计还是测试执行，都要充分利用测试时间和测试设备资源，节约搞测试。
- 重视测试要年年讲、月月讲、天天讲，不要只是 2~3 个星期搞运动，过后大家又一潭死水。
- 测试人员互相帮助，共同进步，一起提高产品质量的同时一起提高测试水平的修炼。
- 测试和开发团结如一人，一起实现天下无敌的高质量产品。
- 测试人员若由能力差的人组成，则这样的测试团队很难得到开发人员的重视，也很难得到公司的重视和投入，更谈不上士气的提升、专业水平的提高。

软件测试精要

每月及時觀看電子月刊
就上溜客安全網 www.17

- 测试人员别空喊：“公司不重视测试，开发不尊重测试”。先拿出自己的所有本领，尽可能多地找出重要的 bug，并帮助开发人员快速地定位 bug。用你的输出和业绩让开发人员从业绩和能力上尊重你，正如同笔者当年一个人负责测试的帧中继 QoS 的模块，该模块的开发人员同时维护 3 个功能模块，在经过笔者相当一段时间的折磨后，某日他主动对我说：“因为严重 bug 修改得多，我目前最放心的就是帧中继 QoS 的模块”。往往开发人员修改的严重 bug 越多，他们反而更放心，更钦佩相应的测试人员。
- 坚定信念，bug 还会有的，先别归咎客观条件。
- 只有测试者的才智才是真正强大的测试力量。依赖测试工具，不可能完全保证产品质量的完美。



案例

深圳 S 公司的某款产品常在实际应用中出现一些无法在实验室测试时发现的 bug。在经过两款业内顶尖的测试仪器测试后，只发现了一个严重 bug。该 S 公司将这款设备交由一家专业的测试公司进行系统测试时，却在 4 天的时间内找到了 4 个严重 bug，让 S 公司不得不佩服这家专业的测试公司。而这家专业的测试公司，靠的不是所谓业内领先的测试仪器，而是依靠三四名有着 20 多年测试经验的优秀测试工程师，花了近半年的时间设计出的一个个高效的测试策略和测试方法，才达到这样的效果。

所以，只有人才是创造世界历史的动力。

- 人是要有一点精神的，测试人员至少需要一点信念，坚信 bug 一定还有。
- 测试就是找 bug，以 bug 输出为业绩。如果仅听开发人员说已不可能测出 bug，就不再测试的话，也就不需要专业的测试人员了。
- 实践出真知，所有的测试策略和测试方法的实际效果一定要靠实际动手实施论证后，才能知道是否真正有效。
- 世界上怕就怕“认真”二字，很多 bug 稍纵即逝，如果测试人员不和开发人员较真，就得不到 100% 的确认。
- 不要轻易放过任何一个测试人员自己主观怀疑的问题疑点。

软件测试精要

每月及時觀看電子月刊
就上溜客安全網 www.17

- 漏测是犯罪，过度测试同样是犯罪。过度测试会浪费开发人员宝贵的时间，会让所有人忽视其他或许更重要、更需要深入测试的地方。
- 测试人员不但要善于找 bug，还要加强定位 bug 重现的能力。找 bug 是艺术，需要的是创造性；定位 bug 则是科学，需要严谨的逻辑推理能力。所以，笔者爱说：好的测试人员一半是工程师一半是艺术家，是严谨与创造性共存于一身的人。
- 无限风光在险峰，当我们发现不了 bug 的时候，挑战自我、突破自我的时候就来了。
- 测试是一个从没有止境做到 100% 完美的工作。
- 虚心使人进步，骄傲使人落后。即使你是找 bug 数最多的测试人员，也要向身边的每一个测试人员学习他们的经验，毕竟每个人都有自己独特的思想和思路。而测试生存和发展的根基就是发散的思维和不断增加的新测试方法。
- 对于测试新人而言，最难的是找到第一个 bug，第一个严重 bug。因为测试新人在第一次执行测试的时候，很可能从事的是回归测试或一个已长期测试过的产品。测试目标本身能容易被找到的 bug 就比较少，再加之测试新人所依赖的测试用例，也早已发现了用例能发现的所有 bug 了。所以，新人一开始就直接遇到一个大的挑战。因此，新人如果第一次开始测试时，第一周没找到 bug，请不要灰心，坚定信念，发挥自己的创造性，去创造一些老员工没有用过的方法，至少先要乱想 10 个新的测试情景来一一尝试，试图找到一个 bug，哪怕是一个拼写 bug。只要有了第一个 bug 就如同销售人员开了第一单，你就可以星星之火，可以燎原了。
- 测试人员不要指望一周时间就找到所有的严重 bug，也不要期望一个测试人员或一种测试方案就可以保证整个产品的质量，饭要一口一口地吃，山要一点一点地移。一个产品的稳定，需要多人、多个版本、多个测试阶段、多种测试方案，一个一个的 bug 积累起来，才能最大化地保证产品的高品质。
- 测试策略是一个测试项目成功与否的生命。
- 测试部门真正的铜墙铁壁是从事各类测试相关工作的人，而且是真心实意拥护公司价值观和使命的人，并不是所谓的流程、仪器和工具。

软件测试精英
每月及時觀看電子月刊
就上溜客安全網 www.17

- 产品的质量稳定要靠公司中测试人员的主观能动性、智慧和潜能的挖掘、高昂的士气来保证和提高。切不能迷信靠几台测试仪器、几次技术培训、一些外部流程就能保证质量的提升，关键还是发挥和挖掘自身的潜力。

8.7 测试工程师的基本素质

要做一名合格的测试工程师，首先需要具备一定的基本素质，而去掉浮躁的心就是基本素质的第一步，也是决定以后发展高度的最关键因素。笔者曾在网上看过一些朋友发表的关于如何去掉浮躁心态的帖子和经验，谈到如何不做一个浮躁的技术人员，其观点很有价值，现与大家共享。

- 不要看到别人的回复第一句话就说：给个代码吧！你应该想想为什么。当你自己想出来再参考别人的提示，你就知道自己和别人思路的差异。
- 看帮助，不要因为很难而自己是初学者所以就不看；帮助永远是最好的参考手册，虽然帮助的文字有时候很难看懂，总觉得不够直观。
- 不要被对象、属性、方法等词汇所迷惑，最根本的是先了解最基础知识。
- 不要放过任何一个看上去很简单的小问题——它们往往并不那么简单，或者可以引伸出很多知识点；不会举一反三你就永远学不会。
- 知道一点东西，并不能说明你会写脚本，脚本是需要经验积累的。
- 学脚本并不难，JSP、ASP、PHP 等也不过如此——难的是长期坚持实践和不遗余力地博览群书。
- 看再多的书是学不全脚本的，要多实践。
- 把时髦的技术挂在嘴边，还不如把过时的技术记在心里。
- 学习脚本最好的方法之一就是多练习。
- 在任何时刻都不要认为自己手中的书已经足够了。
- 看得懂的书，请仔细看；看不懂的书，请硬着头皮看。
- 别指望看第一遍书就能记住和掌握什么——请看第二遍、第三遍。
- 请把书上的例子亲手拿到电脑上实践，即使配套光盘中有源文件。
- 扩充在书中看到的有意义的例子，并将其切实地运用到自己的工作中。

- 不要漏掉书中任何一个练习——请全部做完并记录下思路。
- 当使用脚本到一半却发现自己用的方法很拙劣时，请不要马上停手；请尽快将余下的部分粗略地完成以保证这个代码的完整性，然后分析自己的错误并重新编写和工作。
- 别心急，写脚本确实不容易；水平是在不断的实践中完善和发展的。
- 每学到一个脚本难点的时候，尝试着对别人讲解这个知识点并让他理解——你能讲清楚才说明你真的理解了。
- 记录下在和别人交流时发现的自己忽视或不理解的知识点。
- 保存好你做过的所有的源文件——那是最好的积累之一。
- 对于网络，还是希望大家能多利用一下，很多问题不是一定要论坛来问的，首先要学会自己找答案，比如 google、百度都是很好的搜索引擎，只要输入关键字就能找到很多相关资料，不要总是等待别人给你希望。
- 到一个论坛，要学会去看以前的帖子，不要什么都不看就发帖子问，也许你的问题早就有人问过了。

具有一颗不浮躁的心态后，优秀的测试工程师还应具备哪些基本素质呢？

1. 技术能力

作为一名测试工程师，不能仅从使用者的角度来测试软件产品，还要从技术的角度来设计测试用例，这里所说的技术包括基础的与专业的。基础方面应学习过软件技术基础、C语言、面向对象设计、C++、数据库理论、计算机网络技术、软件工程、数据结构与算法、离散数学等课程；专业方面应掌握软件测试技术概论等。

2. 具有一定的编程经验

测试工程师有时候需要对源码进行检查，有时候也会从程序结构的角度来测试软件，有时候需要写一些自动测试的工具软件，有时候需要写测试脚本。显而易见，会写简单代码，能读懂源码对测试人员来说是必需的，而且如果有一定的编程经验，可以帮助你软件开发过程有较深入的理解。

软件测试精要

每月及時觀看電子月刊
就上溜客安全網www.17

3. 沟通能力

测试人员需要与很多人员进行沟通，项目经理、开发人员、客户、市场人员等都是测试人员经常沟通的对象，而且在面对不同人员时，你需要不同的语气、不同的态度。与客户要谈得来，处处为客户着想，客户就是上帝，与上帝说话要和颜悦色；与开发人员交往就需要技巧了，测试人员与开发人员经常要在心理上进行较劲，因此在说话的语气或讲述一个问题的出发点时要特别注意了。

4. 要有严谨、敢于承担责任、稳重的做事风格

做测试工作，需要思维严密，什么问题都要考虑到。除了做事认真仔细外，也要有承担责任的勇气，在漫长的项目实施过程中，或大或小的错误在所难免，要敢于承认错误。

5. 具有怀疑与破坏的精神

测试人员不能总是以常规的思路来测试软件，要设计一些非常规的、相反的测试用例来不断地折磨软件产品，要破坏性地测试，并且不要停止怀疑。

6. 善于自我总结、自我督促

应该说软件测试是一种既烦琐又枯燥无味的工作，做多了会觉得似乎一成不变，对自己的能力没有提高，这时候就需要做自我督促，并经常作一些阶段性的总结。新的技术、新的方法、新的工具层出不穷，要让自己跟上技术发展的脚步，善于将新技术、新方法、新工具应用到测试工作当中。

最后用培根的话来结束：注意你的思想，它会变成你的行动；注意你的行动，它会变成你的习惯；注意你的习惯，它会变成你的性格，而性格将决定你的命运。

8.8 谁是测试团队中的核心技术人员

笔者个人认为公司测试团队中最重要的的人是设计优秀测试策略和设计优秀

软件测试精要

每月及時觀看電子月刊
就上溜客安全網www.17

ROI（投入产出比）测试方案的测试工程师，同时自动化测试脚本开发工程师和测试工具开发者对于测试部也是非常重要和不可或缺的。

为什么说测试部最重要的人是测试策略和测试用例的设计者呢？

1. 从任职资格来比较

自动化测试开发工程师可以由只毕业 3~4 个月或才进入项目 1~2 个月的工程师就能从事；测试工具开发甚至不需要对测试了解多少也能从事；而设计测试策略和好的测试方案设计者则需要至少 2~3 年的相关业务黑盒测试经验的工程师，才有可能做好。

2. 从相关人才发挥的作用分析

测试策略的制定就好比公司的市场部要制定大的市场战略。例如，这个测试方案的定位是什么？本轮测试的目的是什么？有多少资源可以进行测试？应集中资源先对哪些重要又紧急的功能进行测试？如何区分重要或紧急的功能？如何安排紧急而不重要的功能与重要而不紧急功能的测试顺序等。

测试方案的设计就好比公司的销售部针对每个具体的客户制定具体的销售计划和销售方案。例如，对于这个 Case 我们的目标期望是什么？如何达到这个目标？是否还有更好的方法来达到这个目标？如何利用现有的有限资源取得最大化的结果？该 Case 的 ROI 是否高效？

那么如何成为一名合格、优秀的测试策略的设计者呢？

- 一切测试策略的目的是为了满足公司市场销售策略。因此，我们应该设法保证测试策略既能按期完成研发计划，又能不影响销售的产品质量。
- 将有限的资源先用到紧急又重要的测试任务中，再评估其他任务的优先级。但一定要以市场销售计划为依据，一个脱离市场销售计划的测试策略是一个误国误民的策略。
- 测试策略不是测试计划。以市场销售计划为依据，先制定测试策略，再按任务的优先级和资源情况来制定测试计划。现在常有不少朋友误认为测试

软件测试精要

每月及時觀看電子月刊
就上溜客安全網 www.17

计划就是测试策略。

- 测试策略不是一个人设计出来的。它一定是先听取了市场人员的意见和要求，再结合研发的要求得出的一个平衡取舍的产物。
- 好的测试策略设计者是一个以公司利益为第一位，而不只局限在测试部门利益的员工。办公室政治玩多了，设计的测试策略有可能就是一个对公司利益有害的策略。
- 一定要最大化地接近客户的真实应用来定义重要的模块功能，而不是简单地依据模块的复杂度来定义。
- 能知己知彼。对自己的测试资源有多少，要有准确、清晰的认识才能量体裁衣。而不要打肿脸充胖子，最后实施时，却完成不了测试策略。

如何成为一名合格、优秀的测试用例设计者呢？

好的测试用例至少要满足如下要求（这里只讨论功能测试用例的要求）：

- 粒度一定要细，对功能需求中的每个小点、每个数据都要覆盖到，并尽可能多地想到更多的测试方法来对被测试项进行拷打。
- 测试方法不能过度测试。大家容易想到测试不充分对公司有害，但往往会忘记过度测试对公司也是有害的，这点在作者的测试经历中感受非常深刻。例如，因为你的过度测试，浪费了有效的测试时间，这个时间原本是可以拿来发现更多真正对销售有影响的 bug。因为过度测试发现的 bug，开发人员有时必须为这些意义不大的 bug 花时间来修改，浪费了修改其他更有意义 bug 的时间，影响了项目的进度，影响了产品的销售计划。说不定公司甚至因为你的几个过度测试，延迟了 1 个月推出产品，导致公司在这 1 个月中损失 1 亿元的订单。而这些都是真正有可能发生的。所以，我们需要避免在不漏测的前提下走向另一个极端——过度测试。
- 设计的测试用例是否容易转化为自动化测试。如果在进行测试用例设计时，根本没有把将来进行自动化测试的开发作为一个考虑因素。那么，你的测试用例永远只能用手工进行回归测试，将大大浪费公司的资源并影响产品发布时间。如果回归测试的手工执行是你自己来做的话，那么以后你的苦日子可就长了。

软件测试精要

每月及時觀看電子月刊
就上溜客安全網 www.17173.com

- 测试用例应该考虑执行时资源和时间的安排。例如，千万不要出现有的执行完一遍只需要1小时，而有的测试用例执行完一遍却需要20个小时。为了便于测试管理和计划安排，应设法让测试用例未来执行的时间保持在一个参考值左右。假设一个标准测试用例的执行时间为4小时，那么设计的测试用例应尽量保持在3~5小时内。这样非常容易量化并管理测试用例执行者的工作，更有利于测试计划的安排、测试策略的制定和研发计划的按时执行。
- 在理想状态下，最好测试用例也要有好的易读性。测试用例的执行者拿到你的用例后能在半小时或1个小时左右就能读懂并执行；否则，如果执行者还需要4~5个小时才能读懂或不能完全读懂你的测试方法和测试步骤，那么公司将会为这样的测试用例付出非常大的成本代价，甚至会影响测试计划的执行和产品的研发计划。

总结：对于部分迷茫的测试工程师，如果希望在测试领域发展而不是在代码开发领域发展，就不要误认为只有代码才是高手，只有代码写得好才是好的测试工程师。要做一个对公司有最大价值的测试工程师，就应该尽量向成为一个好的测试策略设计者和测试用例设计者方面成长和发展，但这一切工作又依赖一个很基础的因素——工作的责任心，只有把公司的大局发展放入自己的心中，才能真正做好工作的取舍，设计出有价值的测试策略和测试用例。

8.9 软件测试值得投资吗

软件测试的投入对企业是成本累赘，还是高回报的投资？

随着软件市场的成熟，人们对软件作用期望值也越来越高，软件的质量、性能、可靠性等方面也正逐渐成为人们关注的焦点。目前，中国软件产业在产品功能和性能测试领域都存在着严重不足，中国软件企业也已开始认识到软件测试的广度和深度将决定着软件企业的前途命运。因此，很多的软件公司也都认识到了软件测试的重要性，但现状是国内大部分的软件公司却只是象征性地请1~2名测试人员负责所有项目的测试。一方面，这些公司的老板认为程序能运行基本上就

每月及時觀看電子月刊
就上溜客安全網www.17

已经成功，没有必要成立专门的测试部门或设立测试岗位；另一方面软件开发企业在为软件开发支付费用后，也不希望再为软件的测试支付新的成本了，这也是软件测试没有真正得到重视实施的一个重要原因。

所以对于软件测试工作，很多公司说到底还是不够重视。如果管理层从控制成本考虑，不打算对测试投入太多，那么管理层就犯了一个软件成本认识的错误。因为对于一个企业的成本概念，不仅仅是生产过程中产生的费用是成本，未来因为产品缺陷产生的服务费用也是成本。如果产生了客户退货、产品召回，这也是成本！

在国外软件测试市场已成为软件产业中的一个独特市场。在美国硅谷地区，凡是软件开发企业或设有软件开发部门的公司，都有专门的软件测试部门，在这些公司或部门中，负责软件测试的质量保证经理的职位与软件开发的主管往往是平行的。因为，经过几十年软件行业的发展，外国的企业已经意识到重视软件测试的投入，不但不是一笔不划算的买卖，反而节约了大量的售后费用和软件开发成本。

在软件开发的早期，软件开发只是编码及部署。但随着软件行业的发展，开发企业开始在发布前进行测试。从这个意义上来说，软件测试的确是成本。但从更普遍的意义上来讲，软件测试其实是一种提高软件质量的投资，是一项可以减少将来若干不必要费用的投资，同时还可以为企业赢得良好的口碑。而达到这一目标的解决方案，主要有：采用测试驱动开发方式，确保每个小功能的模块都得到很好的测试，并建立可持续发展的资源整合环境进行可靠的整体测试。软件测试最终将转化成节约成本的开支，不论它是产品开发、软件升级还是软件维护。

许多统计资料表明，开发过程每前进一步，发现和修复一个缺陷的平均成本要提高 10 倍。软件测试的一个经济目标就是尽早发现缺陷，降低修复及售后服务成本。显然，每一个已发布产品中的缺陷除了会影响产品及企业的声誉外，还会直接增加产品的售后服务成本。无论是派人到现场调试还是研发、发布补丁程序，都要比发布前的修复成本昂贵数十倍，甚至数百倍。

因此，软件测试的必要性已经完全被业内人士所接受。因为软件工业已经发

软件测试精要

每月及時觀看電子月刊
就上溜客安全網 www.17

展到了与其他主要工业相当的规模，经过十几年的发展，少数天才手工作坊的世界已经转化为大工业生产的天下。在工业生产的传统行业里，测试就是生产的有机组成环节，各道工序的测试、组装测试、出厂测试及破坏性试验是实现生产目标的重要保证。

总结：纵然软件开发可能会面临很多的困难，诸如发布时间紧、市场压力大、投资过高等，但这些都不要成为挤压软件测试投入的借口。因为软件测试是软件开发生命周期中一个非常重要的部分，软件测试作为一种投资将大大节约售后成本、技术支持成本及软件修改成本。

8.10 软件测试的黄金阶段

软件测试作为一种专业的社会活动，一定有着它价值闪光的黄金阶段。如同在 20 世纪 70—80 年代，对于很多刚解决了温饱的家庭都未曾想过购买家用小汽车，认为没必要，也没足够的经济实力来购买、供养小车。但到 21 世纪后，中国很多的家庭富裕起来后开始购买小汽车了。小汽车之于中国家庭，正如软件测试之于中国软件企业，只有当企业解决了最基本的生存问题，在追求下一阶段高质量发展时，才会用满足了生存后的资源来投入软件测试中。而在一个软件企业有着做“基业常青”的目标时，软件测试对于软件企业，一定会随着企业的不断成长，而不断地得到越来越多的重视和资源倾斜。

因为任何一个产品都会经历：“从无到有，从有到好，从好到优”3 个阶段。不光指软件产品，还包括其他诸如汽车、家电等产品都有这样一个演进过程，这也是为什么许多老牌企业的成熟产品总要比新公司的新产品质量要高的原因。所以对于许多刚上市的第一版产品，企业为了最快时间推出第一版产品，抢占市场，在产品质量投入上都无法做到完全 100%精益求精。因此，在第一阶段，企业要解决的问题是“从无到有”，最重要的目标是尽早推向市场，“时间”是第一关注要素。在产品的这一阶段，肯定企业内部的所有资源和工作计划的优先级都是为了能抢时间，对于软件测试的投入和关注度肯定会有所降低。这一阶段的软件测试关注的重点应该是确保产品能满足最基本的产品市场发布的基准。何为软件测

每月及時觀看電子月刊
就上溜客安全網 www.17173.com

试的基准，以及如何确定软件测试的基准？将在下一节的内容中进行讲解。

当第一版产品成功推向市场，撕开了竞争对手的防线，并抢到了后续生存和支撑发展的“面包”后，为了站稳市场并不断拓展市场空间，产品的信誉就显得特别重要了。而产品的信誉中第一重要的要素则是“质量”，一旦推出的产品质量有重大问题或小问题不断，在用户口碑中流传这些质量问题，对于企业基本就是毁灭性的。不但刚刚打开的市场会丢失，恐怕日后还会更难重新进入市场。因此，这时企业都会在第二阶段把不断优化和提高产品，提高产品性能放在一个重要的地位。这时，对于软件研发的重视重心，开始逐渐从开发向测试转移，虽然开发依然会不断投入开发新功能，但是测试投入的比例会比第一阶段大大增加。

在产品第一阶段，软件测试确保产品满足了推向市场的最基本的基线标准，这一阶段对测试人员的技术要求最低，对测试人员基本不要求进行发散探索性测试，即最基本的功能测试和最简单的性能压力测试，是“防守性测试”，而不是“进攻性测试”。在第二阶段为了提高产品的质量，尽早暴露产品的问题，这时就需要对软件测试人员要求“进攻性测试”，相应地对软件测试人员的技术要求也就更高了，需要软件测试人员开始进行大量的发散探索性测试，创造和设计出更复杂的测试方案，来加大对产品功能测试、性能测试、压力测试的测试力度，发现产品异常的问题。这一阶段软件测试人员的压力会比第一阶段大得多，不但要求能完成基本的基线标准工作，而且还要求发挥主动性，进行更多积极主动的攻击测试，提前暴露产品的潜在问题。企业已在第一阶段告诉了市场，产品的存在，更多的潜在用户很有可能会在产品的第二阶段成为真正的用户，如果软件测试人员不能在产品大面积使用之前提前发现更多的产品问题，那么很可能企业会在产品第二阶段把以前的全部成果都毁于一旦。因此企业在第二阶段推出的产品质量通常都比第一阶段推出的产品质量提高了不少，不但企业修改了第一阶段暴露的问题，而且对产品的各种潜在质量问题也提前解决了，虽然有不少用户并未在这一阶段感受到产品内在质量的大大提升，但对企业而言却在第二阶段对产品质量的提升投入了更多的力量。

在第三阶段，随着企业开始进入行业第一军团，产品质量也成为行业第一流的标准自然是毋庸置疑的要求；否则，在与第一流竞争对手竞争用户时，会被对

软件测试精要

每月及時觀看電子月刊
就上溜客安全網www.17

手的质量比拼下去，而难形成第一流的品牌，这也是为什么知名大公司会更重视测试的原因。大公司要长期发展必须重视一流的品牌和质量信誉，一流的质量提升不光靠开发人员的高水平设计，还需要软件测试来帮助把握平衡。开发和测试在这个阶段如同一名运动员的双腿，只有双脚均衡有力才能决定最后整体的成绩。

第三阶段的软件质量提升方式已不仅仅是进行“大规模猛烈的进攻”就可以达到一流的质量水准。为什么有些企业把预算和测试人员数量增加了1倍，而产品质量却依然没有取得更大的提高？这是因为第三阶段软件测试技术层次的提升比第二阶段要难得多。这一阶段的软件测试水平的提升规律，已不仅仅是多投入、多投资资源就能保证出新成果的。即使投入更多的人，如果工作方向和思维方式不改变，则只会使更多的人依然停留在原水平进行重复劳作。第三阶段的质量提升方式就应该如同“美羹小火熬制”的方式，讲究精准的雕刻手法，讲究科学精巧地干。讲究“智”的投入，应选用和投入资源让真正的“测试明白人”开拓创新。一方面，投入高水平的测试研究型人才采用引入和自己创新研究的方式，不断引入更多提高测试技术水平的新思路和新方法来找到更深层次的质量问题。另一方面，在内部已有的测试流程和测试方法中不断优化测试效率，从内部要效益。采用“内外结合”的方式，加上少量高水准的测试研究型人才，不断对测试的各环节进行提升，让企业的产品质量满足最高质量标准要求的顶级用户的需求，成为市场一流产品品牌。

综上所述，中国软件测试的黄金阶段一定出现在中国的企业渡过生存期，开始进入高速发展期，和开始进入行业一流企业集团之时。中国过去不重视测试，导致测试长期处于低水平需求时代，是有其历史必然原因的，测试人员没有必要去埋怨过去和历史。但是现在中国越来越多的民族企业已度过了生存期，进入了高速发展期，对软件测试开始逐渐重视起来，最明显的表现就是测试人员的招聘数量开始大量提高。同时，中国的少量优秀企业已进入或马上进入所在行业的世界第一集团，对测试的需求和投入已开始追求高水准的测试技术阶段，高端测试人才的需求已悄然在猎头的列表中。测试人员的工作内容已不仅仅是编写测试用例、编写测试脚本的阶段，而是如何帮助企业在现有测试水平的基础上不断提高测试的技术方法，提高测试工作效率和科学性，挖掘更深的问题，从质量的角度来保证企业能进入第一集团并保住第一集团的市场地位。

软件测试精英

每月及時觀看電子月刊
就上溜客安全網www.17

8.11 软件测试验收基准

软件测试什么时候可以结束？这是一个对于有一定经验的测试人员，特别是测试管理者在测试项目中常常会困惑的问题，因为大家都知道 bug 永远也不可能 100% 被找到。所以，对测试工作结束标准的度量，成为了很多公司测试项目和测试绩效考核的疑难问题。错误的测试结束标准如下：

- 对产品进行一轮全面测试后，在发现的 bug 数中，一级 bug 数少于 1 个，二级 bug 数少于 2 个，三级 bug 数少于 3 个。
- 产品测试的 bug 总数达到了 N 个，如 500 个。

以上两种测试结束的度量标准都是不科学的标准。即使测试的最终结果达到了这些数字，并不能说明产品的软件质量已能满足市场需求了，甚至反而因为受累于这些数字指标影响了产品正常的市场发布时间。

在前一节中，曾提到在软件测试的第一阶段，需要制定一个基线测试标准。在正式提出建议的基线测试标准前，笔者先谈一种软件测试的理论：正面测试与负面测试。

正面测试就是在大多数情况下产品的正常使用方式，以及满足大多数情况下的抗压力能力和性能指标。严格按用户正常习惯来测试，不做发散探索性测试，满足最原始的 Marketing 或产品规划部门提出的产品交付需求。

负面测试则是大家所理解的对软件进行各类进攻性破坏测试，构造各种异常场景，甚至是高于多数真实场景的压力测试指标，目标就是尽可能地找到产品的所有 bug，即使这些 bug 可能在实际应用中 10 年也不会被用户遇到。负面测试对测试需求的要求标准高于原始的产品市场交付需求。

在产品第一阶段，为了最快地推出产品以满足市场需求，建议测试基线标准以正面测试为标准。测试人员尽力保证市场交付的原始需求，首先让产品在用户处使用起来，然后再通过升级的方式为用户提供第二阶段输出的更高质量的软件

版本。那么这一阶段的基线标准，就应该由公司的 Marketing 或市场人员，与测试人员共同从市场的视角和需求来编写测试用例，评审测试用例。由 Marketing 确认基线测试用例，测试人员执行测试用例时，只要保证所有测试用例没有再发现 bug，即证明产品的软件质量满足了最初的市场需求，便可以结束第一阶段的软件测试。

在产品第二阶段，为了提高产品的质量，提前暴露问题。满足用户更深层次的软件质量需求，必须在满足基线标准的基础上，加大负面测试的力量，包括各项测试人员进行自由发散探索性的攻击性测试，再加上安全性测试、负面测试、长期大压力测试等。尽可能去验证测试产品的健壮性，该阶段的测试用例可以先由开发人员和测试人员共同完成初稿评审。由于该阶段的测试工作中即时创造性内容较多，因此必须随时将找到了 bug 的测试方法补充固化到测试用例中。当使用所有的测试方法和测试方案进行一段时间测试后，bug 出现的数量开始呈经过数轮测试出现不断减小收敛的趋势后，测试经理们应该与开发人员、产品市场经理一起针对当前发现的 bug 情况和各测试目标 bug 收敛的趋势情况，决定产品是否可以向市场发布，但必须保证产品能 100% 通过基线测试标准和第二阶段所有测试用例的测试。

以上就是笔者关于软件测试验收基线标准的经验总结和对测试的认识。



软件测试精要

每月及時觀看電子月刊
就上溜客安全網 www.17173.com

附录 A 测试职业经历随谈

关键字：测试、bug、测试误解、创造力、坚定信念

第一部分：开场白

首先，作为一名一直在网络和电信设备测试领域工作了几年的老油条，我想把我这几年的工作经历，以及在我身边看到的故事讲出来与大家分享，希望能帮助一些刚进入测试的朋友更好地从学生角色转换为一个职业人，少一些刚毕业时的迷茫。

第二部分：“阴差阳错”

在两个月的上岗培训中，我第一次见到路由器、交换机、VoIP 网关、电信设备，在稀里糊涂的状态下进行着配置练习。我们那时真是魔鬼式培训，从早上 8:30 开始，一直持续到晚上 11 点，有一次还持续到了晚上 12 点。每天上午第一件事就是考试，将头天的学习内容进行笔试考试，基本上 TCP/IP 详解中每一个协议的报文结构都默写过。笔试成绩每天公布分数和排名。下午从 1 点开始一直到晚上 11 点就是上机操作，做完一个练习就做下一个练习，反正就是一直要做到晚上 11 点才下班回家。就这样一直维持了两个月的培训，每天都是新的理念知识、新的上机练习。压力非常大也非常累，进步也非常快。

7 月分配产品线，我来到了最不愿意去的电信产品线。因为在培训期间，电信产品线的理念最难理解，也最难完成上机练习，远比 TCP/IP 详解难得多，所以几乎没有一个人愿意到电信产品线。我和另外 3 名同事都来到了电信产品线，没办法再难也只有蒙着头冲了。8 月我再一次被迫接受了不愿意的工作分配，电信产品

每月及時觀看電子月刊
就上溜客安全網 www.17

线的测试经理将 FrameRelay 协议（因为这是一个老协议，所以我一直不愿意测试这个协议）给我负责，并要求在半年内必须独当一面，意思是以后公司遇到任何关于 FrameRelay 协议的问题都将由我来解决，我就是公司这个协议的专家。对于一个刚毕业两个月的学生，面对这样一个要求，既感到压力也感到了一种挑战。在必须成为公司内该协议的专家且没有任何退路的情况下，在随后半年里，对于该协议无论理论还是任何实际问题我都以几乎十全十美的态度来学习或解决。不但对该协议的每一个细节、每一个 bit 的组合意义烂熟于心，更对该协议在产品中从 FPGA 到 HDLC 再到驱动的所有实现原理进行了深入学习。再通过对实验室里和市场上各种问题的分析，最终在半年内渐渐做到了可以独当一面，可以独立面对和解决任何与 FrameRelay 协议有关的市场技术问题。

第三部分：“坚定信念，一定还有 bug”

也就在这半年时间里，我对测试的看法发生了两次大的变化。第一次变化，在进行手工测试 3 个月后，我心中还是认为开发比测试有意义、有挑战。而测试则是简单、无聊的工作，让我做测试就是大材小用。在这时，我内心开始讨厌测试工作，因为自己从心里看不起测试。有一次我测试了 5 天 FrameRelay 协议，没有发现一个 bug，之前我已发现了这个协议的上百个 bug。正当我沮丧时，一位老员工的一句话从此改变了我对测试的认识：“想想办法，肯定还有 bug”。于是，我怀着半信半疑的态度，利用周末梳理了自己的思路，总结自己已发现 bug 的方式和现象，到 FrameRelay 协议的所有实现原理，最后自己又提出了 10 个新的测试方法。周一上班，只用了 4 个新方法就发现了一个一级 bug、一个二级 bug。从此以后，在我心中对测试的看法发生了根本性的改变，不再认为测试只能由能力差的人去做。测试同样具有很强的挑战性和成就感，测试是一个必须需要创造性的工作，而且是每天都会用到创造性能力的工作。

当你发现不了 bug 的时候，请相信这句话：“一定还有 bug”。只要坚定这个信念，一定会发现 bug 的。“坚定信念”成为了我以后鼓励新员工时最爱用的一句话。对于测试人员，每天的工作都是对自己潜能的一次挑战。昨天，自己还认为没有 bug 了，今天通过自己坚定的信念和创造性的劳动，又发现了新的 bug。这

软件测试精英

每月及時觀看電子月刊
就上溜客安全網 www.17173.com

不恰当证明，好的测试态度能让测试成为一份非常有挑战、创造性和成就感的工作吗？

在其后 1 年的工作中，因为我同时具有开发和测试的能力，公司希望让我做测试工具开发工作，但我拒绝了调岗，同期也拒绝了华为（不是慧通）邀请我从事测试工具开发工作的 offer。我选择了测试，测试不但让我每天都能感受到自我挑战带来的成就感，进行创造性思考所带来的快乐，更能让我比普通的开发者用更短的时间了解到更多产品或功能实现的细节，能对系统了解得更广、更多，让自己不但看到树枝，更能看到整个森林。这使我只用短短 3 年多时间就对很多网络和电信协议的应用及设备原理有了比较系统的了解。结合我自己的特点，我喜欢在适度深度的基础上，对网络有尽可能广泛的了解和掌握。而不喜欢一个点或是几个点一路走下去，掌握每一个细节。所以，我喜欢测试比开发更多一点。

第四部分：“测试误解”

那么测试人员的水平怎么体现出差异呢？

我的看法是：不是你发现的 bug 数量多就很厉害，也不是你会写自动化脚本就比做手工测试的厉害，更不是你会用的测试工具多，你就厉害。好的测试人员应该首先具有严密的逻辑思路，同时还要有很强的发散性思维，能经常创造性地提出新的测试方案，并且具有非常强的分析问题、定位问题的能力。只要具备了很强的分析问题、定位问题的能力和创造力，即使你所负责的模块是一个小模块，你也能发现很多严重的 bug。

在测试中有些同事常有这样的误解：

情形 1：有的同事所负责的模块是个老模块、小模块，所以发现新 bug 的数量很少。这时很多人常常就容易松懈，认为这个模块没 bug 了，再认真测试是浪费时间。

我的看法是：

- (1) 任何模块永远都有 bug，只是还没找到触发条件。
- (2) 一般人的常规思路用完了，这时正是你挖掘自身创造力的好机会。如果

你这时能在自己思路严密分析的基础上提出一系列新测试方案，发现了新 bug，你的领导一定会对你刮目相看，而自己的自信心也一定会得到大大的提高。

情形 2：有的同事所负责的模块是个新模块、大模块，随便测试也能发现许多 bug。同样，这时测试者也很容易松懈，而不会进行有意识的分析和创造性的思考。

我的看法是：

(1) 虽然发现了很多 bug，但是你是否能通过严密的分析，使得找出一级、二级 bug 数能占到每轮测试所发现 bug 数的大部分。不要总是在产品快要发布前，忽然发现许多一级、二级 bug。

(2) 只有通过自己严密的分析和创造力发现的 bug，才比随便测试所发现的 bug 更具价值，对于产品的质量才真正有保障，对于测试者的水平提高才有意义。否则，测试者自身能力得不到提高，公司产品更是留下了许多隐患的 bug。

总结赠言

最后送大家几句关于测试的看法：

- (1) 坚定信念，一定还有 bug。
- (2) 测试是一份依赖严密思路分析和创造力的工作。
- (3) 测试能让你每天挑战自己。
- (4) 测试能让你的自信心不断得到提高。
- (5) 测试能锻炼你的创造力。
- (6) 测试能锤炼你的性格，让你成为一个不肯轻易放弃的人。
- (7) 测试能让你比普通开发者更快地了解系统全部。

软件测试精英

每月及時觀看電子月刊
就上溜客安全網 www.17

后 记

首先要感谢在 5 位美国实战测试专家身边工作的那段时光。虽无 5 位测试专家的亲身言传身教，但却有着耳濡目染的收获。加上自己几年一线的测试战斗经验，和参与几家国内企业测试咨询和交流的经历，以及善于总结的特长，才有了本书内容的原始材料。没有拷贝，没有抄袭，全来自于自己的亲身经历和感受。谈不上是经典理论，却希望能为大家带来一针振奋人心的测试强心剂。同时我也相信同样有着多年丰富经验的同行们如果有足够的时间，也能写出更好的关于软件测试的书籍。

其次要感谢工作中认识和结识的每一个人，无论是帮助我的，还是给我带来困难和挫折的，正是与大家的相逢相识才有了我这些年的人生经历和职场体会。

最后，推荐一本书《马云点评创业》。目的不是鼓励创业，恰好相反，是希望大家反思自己是否真的适合创业。如果不适合，则最好安心本职工作，去掉浮躁的心。同时，这个节目还可帮助我们如何选择一个好的职业发展平台。

我一直相信：“这个世界唯一不变的就是变化，而我们能做的就是坦然面对一切无法预计的意外变化。只要我们有一颗能承受这些变化的良好心态，我们就不会犹豫不定和恐惧。”

小插曲：就在本书快完稿前，作者在成都经历了“5.12”大地震，这一新中国最大的一次地震，让整个成都的每一个人都受到了巨大的惊吓，死亡离自己这么近。随后一段时间，每天都有连绵不绝的救护车声音和直升机声音在耳边回响，其场景不亚于美国大片。而每天的余震摇晃让成都人从最开始的恐惧害怕，到后来大家乐观的开玩笑“不就是摇摇而已吗”。这本书的手稿与作者是幸运的，没有被埋在废墟下，才能有机会和大家见面，希望能“福中有祸，祸中有福！”“中国，加油！”

每月及時觀看電子月刊
就上溜客安全網 www.17173.com