

如何成就一只懒散的团队？

软件质量与测试的新纪元

为什么要进行需求管理？

测试人员的思想理念和工作方法

我眼中的测试工程师

程序员不是包身工

新产品开发的两大教训

落地敏捷：计划会议中的估算，你纠结吗？

上海泽众软件电子期刊

2013 年 10 月 第二十二期

主办单位：上海泽众软件科技有限公司

联系电话：021-61079698

传真：021-61079698 转 8017

意见反馈：fangmh@spasvo.com

投稿：wangmf@spasvo.com

公司地址：上海市普陀区曹杨路 450 号绿地和创大厦 18 楼 1801 室

邮政编码：200063

公司主页：www.spasvo.com

论坛：bbs.spasvo.com

目录

如何成就一只懒散的团队?	4
软件质量与测试的新纪元.....	6
为什么要进行需求管理?	8
测试人员的思想理念和工作方法.....	11
我眼中的测试工程师.....	13
程序员不是包身工.....	14
新产品开发的两大教训.....	16
落地敏捷：计划会议中的估算，你纠结吗?	17

如何成就一只懒散的团队？

夜深人静了，回想之前经历的一些项目，感慨万千，有犯了低级错误的，有心情低落不堪的，有处于崩溃边缘的，也有咬牙坚持，把心拿出来燃烧的。资源也是各有特色，有不闻不问的，有心有余力不足的，还有说着明白，干着糊涂的。

后来干脆选择了放弃之前蛮不错的环境，就是为了去寻求更好的 PM 经验！说归说，其实有谁能不明白“家家有本难念的经”的道理呢？眼光放长远一些，还是选择讲究大规模开发的公司吧！那里有相对正规的软件开发，有正规的流程，有严格的制度。

很想总结一下如何成就一支有战斗力的团队的实战经验，总觉得好像是吃一张天大的饼，就算是很勉强的从中间咬开一口，又发现，周围又是无边无际的大饼等着你去咬。无奈只好换一个思路，从“如何成就一只懒散的团队”开始吧!!!

流水账开始：

1、指派一个朝令夕改、没有能力、没有眼光、没有责任心、没有领导力、没有控制力、没有操盘经验、小农意识浓厚的“包工头”式团队领导。

2、指派一个没有修养、自私自利不懂得照顾团队其他人情绪的成员。

3、当团队出现若干人游离于团队目标之外，出现消极工作，上网聊天，工作总结浮于表面的时候任其自由发展。

4、当然，在团队中，核心成员对团队的贡献是最大的，从贡献度上来说，其他的“铁打的营盘流水的兵”，就由着他们去了，第一，他们的工资不高，第二，涨工资不考虑他们就好了。

5、不需要有经验的领导，只需要一个能够承担领导责任的人，这个人可以是一个新手，可以是一个工作了十几年，但是只有两年工作经验的人。

6、不需要清晰的目标，工作了若干年，团队成员仍然不能够清晰的知道他们在做什么事情，甚至可笑的是，每个人都没办法清晰的说出来什么是企业文化？

7、计划？随便做做就好，不要把提交的计划当做一种承诺，就当作为一个随便写一下就好的草稿吧！分成四个阶段，现在还有八个月：需求分析、代码设计、代码编写、用户测试、上线！这样好了，两个月一个阶段，八个月后上线，简单修饰一下，提交给领导。

8、计划的执行？不许要太认真，只需要每天有事情可以做就好了。什么？今天该基线验收了？跟用户说一下，过几天吧！告诉用户领导，我们再需要十天就可以了。十天之后？那是十天之后的事情！

9、代码检查，随便翻翻就好，毕竟每个人都是花费了大量的心血写的，怎么忍心挑出他们的错误呢？退一步说了，如果仔细检查他们的代码，这不影响我下班么？

10、文档设计？在这个阶段完成了，或者这个项目验收之后再补吧！现在真的没有时间，也没有人愿意做这个事情，要不？拿别的项目的改一下？只要能够应付检查就可以了。不要跟我提文档的质量，别人的项目能够高到哪里去呢？

11、用户就是我们的测试人员，一定要充分利用用户资源，将错误信息完全展示在用户的面前，用户测试出来的，就是我们选择去改的，用户测试不出来的，那些根本就没有改动的必要。

12、鼓励团队中对技术孜孜以求的精英们，他们对产品的一切改进都是需要支持的，使用更好的技术，使用更好的设计方案，无休止的去做改动吧！

13、不需要理解业务，用户就是业务专家，用户提出的所有的要求，都是基于对我们生产出产品的不足，我们要答应他们所有的要求。

14、项目紧张，不需要申请更多的资源。在我们公司，所有的“资源都是紧张的”，我们只需要让现有的成员拼命的加班，拼命的修改，从而通过提高人天来缩短时间。

15、不需要对团队成员进行充足的培训，也不需要提高他们做事情的技巧，新员工直接拿来用就好了，也压根不用关心培训的效果。只要某件事情有负责人，这个事情就可以不用过多关注了。

16、我们要经常在团队中搞一些团队评价、给某些人升职、涨点薪水，让他们好好看看，只有好好干才能有出路！最好是隔两天就要做点小动作！

软件质量与测试的新纪元

软件质量是指软件的功能和性能满足用户需求和期望的程度，是软件的一种内在特性。软件测试则是通过技术、流程、工具、人员以及管理手段，检测软件文档、软件中间产品和最终产品，查找和报告软件缺陷、错误以及隐患的技艺。通过跟踪缺陷、错误及隐患的修正过程，确保软件产品、中间产品和文档符合软件工程过程需求和用户的最终需求。

软件应用的领域不断深入，设计的复杂程度逐步扩大，开发的周期不断缩短，质量的要求水涨船高，软件企业面临着巨大挑战。加强软件测试过程和技术，可以有效保证软件质量，这种观念正在被更多的软件行业人士理解、接受和实施。

现在和未来软件企业的竞争力不仅体现在产品类型的多样性，产品功能的先进性，更多的体现在产品质量的稳定性和可靠性。软件质量重于泰山，软件质量是核心竞争力之一。如何保证软件质量将成为软件工程领域研究深入研究的课题。

企业对提高软件质量的迫切要求，测试新手对测试技术的学习渴望，呼唤更多国内软件资深人士深入探讨、交流和共享最新质量保证技术和软件测试实践经验。提高软件质量的关键在哪里？提高软件测试的效率，保证测试的有效性，实现质量、过程和成本的最佳平衡，软件质量才能得到保证。

软件测试需要创新

软件测试在国内仍处于起步阶段，各种软件测试的方法、技术和标准都还在探索阶段。查阅国内出版的书籍、发布的论文、网络上的论坛，对于如何进行软件测试，不同的公司、不同的人士，观点各不相同，甚至互相矛盾。可以认为，当今中国的软件测试行业处于“春秋战国”时期，百家争鸣。这是一把双刃剑，一方面，这给行业的创新和发展提供了营养丰富的土壤；另一方面在测试行业一派“欣欣向荣”的气象背后，也隐藏着深深的危机：软件质量和测试观点“良莠不齐”，“泥石俱下”。“某些”所谓“测试专家”的错误观点正在误导刚刚进入软件质量和测试领域的新人，错误的方法论不利于提高测试过程和效率，更无法保证产品质量。

与此同时，创新不足日益成为困扰中国测试界、质量界的大难题，更多的人还在人云亦云。相同的文章被各个网站转载多次，由软件测试一线工程人员撰写的技术文章的数量严重不足。软件测试领域的一些人士视界不够开阔，创新意识不强，创新方法不足，直接照搬照抄国外的“最佳实践”，以为能包医百病。这是很严重的误解和误信，事实上，随着外包服务的纵深发展和继续深化，中国已成为对软件测试和质量领域最有发言权的国度。中国将逐步占据全球测试业务的绝大多数市场份额。中国测试和质量界需要系统地研究、总结和推广测试理论、测试实践和测试经验，为全球测试与质量行业的工程师提供知识服务。这是时代赋予的职责和使命，也是时不我待的迫切任务。

此外，中国软件行业的技术和市场现状，决定了中国软件质量和测试行业，必须根据国内行情，确定各个企业的软件质量和测试策略。为此，中国软件质量和测试领域需要创新！不仅需要方法创新，技术创新，还需要管理创新，流程创新和观念创新。

“软件质量研究”破土萌芽

“软件质量研究”网站是致力于软件质量与测试技术与推广的专业性网站。她的创建和发布是软件质量和测试领域寻求领域创新的尝试和努力的结果，也是适应提高国内软件质量要求的应运而生。

网站的创作和维护核心团队都是来自国内各大软件公司的技术人员，具有丰富的软件开发、测试和管理实践经验，对软件质量和测试具有深刻的理解和切身感受。

成立“软件质量研究”网站的宗旨在于“传播测试知识，研究测试技术，提供测试咨询，促进测试交流，培养测试人才”。

“软件质量研究”网站的主要内容包括：研究软件测试领域的前沿技术，传播国内外最新测试信息，创作出版测试专业书籍，开发测试系列培训课程，提供企业测试咨询方案。

“软件质量研究”网站的服务群体包括：软件企业软件质量与测试工程部门，高等院校计算机相关科系，软件学院和软件职业技术学院，社会软件职业教育与培训机构，软件技术学术研究机构。

“软件质量研究”网站的第一特色是坚持发布原创文章为主，也有选择的转载由国内外专业人士创作的优秀文章。向软件质量和测试行业宣传、推广国内外新技术和新动向是网站服务的内容之一。追求创新、注重实践、信息共享是网站坚持和追求的目标。

“软件质量研究”网站的另一特色是深入研究软件外包和外包测试。中国是全球软件行业的发展最为迅速的新兴外包国家，中国软件外包潜力无限。软件外包测试将成为国内测试行业新的力量。网站的专家组成员很多来自国际和国内的知名软件外包公司，他们将不断撰写外包测试领域的最新技术和管理文章。

随“软件质量研究”网站同时开通的“软件质量研究论坛”将是国内软件人士加强交流和沟通的另一个信息交换平台。论坛的栏目设置丰富、交互性强、使用方便，将为更多软件行业人士提供更为便捷、高效的技术和信息交流渠道。

软件质量正引起越来越多从业人员和专家的关注，研究测试技术，提高软件质量，成为推动软件发展的重要因素。“软件质量研究网站”的成功开通，揭开了软件质量和测试的大幕，她将立足中国软件企业的发展需要，放眼国际，研究真学问，解决真问题，为软件测试人员带来交流的平台，为促进软件质量事业不断努力和奋斗！

为什么要进行需求管理？

本文介绍了需求管理的必要性，并介绍了控制需求渐变的一些方法。

软件需求是整个软件项目的最关键的一个输入，和传统的生产企业相比较，软件的需求具有模糊性、不确定性、变化性和主观性的特点，他不像生产汽车、电脑 等硬件的需求，是有形的、客观的、可描述的、可检测的，软件需求是软件项目最难把握的问题，他的复杂性体现在以下方面：

需求的描述问题

笔者曾经被紧急委派主管一个已经进入了编码后期阶段的项目，该项目已经换过2次项目经理了，这是第3次更换项目经理，用户方的 IT 部经理找笔者抱 怨：“我已经是第3次来给你们讲补货申请的 处理规则了！”。我只能表示抱歉，因为我无法找到原来的需求描述，这是一个变更的需求，前任的项目 经理讲他只是 将当时与用户交流的需求记到2页草稿纸上，不幸的是，那2页珍贵的手稿现在已经找不 到了！更不幸的是，该 IT 部经理是在转述业务部门的需求，当软件开发完 毕后，业务部门讲“这不是 我们最初给 IT 部反映的需求，我们说的不是这样的！”。缺少正式的完整的需求文档浪费了大量的人力 物力，但是有了需求文档又出现 了新的问题。曾经有多个项目经理向我抱怨，在用户方进行的需求评 审会完全是走形式，因为用户根本不去听他读那上百页的需求文档。不同层次的客户（用户）关 心的 问题是不一样的，想要每个客户都成为需求专家是不现实的。

需求的完备程度问题

需求如何做到没有遗漏？如何准确划定系统的范围？这确实是一个两难问题，稍微大一点的系统要 想穷举需求几乎是不可能的，每次开需求评审会时，总会冒出 新的需求，以至于系统没有一个准确的 范围界定。即使是这样，系统还是要开发，没办法，系统的范围还要硬性的划定一个，从而建立一个基 线。

需求开发的工期问题

在需求上花费了大量的时间（而不是人*工时，因为需求阶段人多了也没有作用），客户、软件公司 是否能够忍受？为了确保需求的正确性，完备性，项目经理 往往坚持要在需求阶段花费大量的时间， 但是客户与公司的高层领导却会为项目迟迟看不到实际可运行的软件担心不已！他们往往会逼迫项目组 尽快往前推进，而项 目组的成员往往也会为系统复杂的善变的需求折腾的筋疲力尽，他们也希望尽快 结束此阶段。

需求的细致程度问题

需求到底描述到多细，才算可以结束了？仁者见仁，智者见智，并没有定论，如果时间允许，要想 细总可以细下去的。但是，需求的周期越长，可能的变化越 多，对设计的限制越严格，对需求的共性 提取要求越高，所以只要大家（客户、用户、需求分析人员、设计人员、测试人员）认为描述清楚了， 就可以进入设计阶段 了。

需求的变化问题

在软件开发过程中如果只有一条真理的话，那一定是：需求的变化是永恒的，需求不可能是完备的。

软件开发的过程实际上是同变化做斗争的过程，需求的变更不一定是坏事，也有可能是好事，是商业机会，对市场敏感的人可以从需求的变化中发现市场机会。

需求变化的原因很多，如：

一开始没有识别全，需要增加需求；

业务发生了变化，需求必须变化；

需求错误；

需求不清楚；

需求的变化问题是每个开发人员、每个项目经理都遇到的问题，也是最头痛的问题，一旦发生了需求变化，你不得不开修改你的设计、重写你的代码、修改你的测试用例、调整你的项目计划等等，需求的变化好比是万恶之源，为项目的正常的进展带来不尽的麻烦，怎么办？管理它！使需求在受控的状态下发生变化，而不是随意变化，需求管理就是要按照标准的流程来控制需求的变化。

难题随之而来，需求中的变化一般不是突发的革命性的变化，最常见的是"项目需求的渐变"(Project Scope Creep)问题，这种渐变很可能是客户与开发方都没有意识到的，当达到一定层度时，双方才蓦然回首，发现已经物是人非，换了一番天地。控制需求渐变需要注意以下几点：

(1) 需求一定要与投入有显示的联系，否则如果需求变更的成本由开发方来承担，则项目需求的变更就成为必然了。人们常说世上没有免费的午餐，同样也不应该有免费的需求变更。但是，接受需求变更目前却是软件开发商不得不咽下的苦果。所以，在项目的开始无论是开发方还是出资方都要明确这一条：需求变，软件开发的投入也要变。

(2) 需求的变更要经过出资者的认可，需求的变更引起投入的变化，所以要通过出资者的认可，这样才会对需求的变更有成本的概念，能够慎重地对待需求的变更。笔者曾经经历过一个项目，为了避免项目的风险，我们请了用户代表全程参与了开发过程，结果此用户代表在开发过程提出了大量"小"的需求变更，当开发人员按此需求变更修改了软件时，在项目进入现场实施阶段时，却有大量的这些变更需要改回去，问题就是出在我们的项目组成员视该用户代表的需求为圣旨，却忽略了需求是否经过了客户方真正有决策权的人员的认可。

(3) 小的需求变更也要经过正规的需求管理流程，否则会积少成多。在实践中，人们往往不愿意为小的需求变更去执行正规的需求管理过程，认为降低了开发效率，浪费了时间。正式由于这种观念才使需求的渐变不可控，最终导致项目的失败。

(4) 精确的需求与范围定义并不会阻止需求的变更。并非对需求定义的越细，越能避免需求的渐变，这是2个层面的问题。太细的需求定义对需求渐变没有任何效果。因为需求的变化是永恒的，并非由于需求写细了，它就不会变化了。

注意沟通的技巧。实际情况是用户、开发者都认识了到了上面的几点问题，但是由于需求的变更可能来自客户方、也可能来自开发方，作为客户他们可能不愿意为需求的变更付出更多的投资，开发方有可能是主动的变更了需求，他们的目的可能是使软件做的更"精致"，于是作为需求管理者、项目经理需要采用各种沟通技巧来使项目的各方各得其所。

软件需求的复用问题

笔者曾经遇到过一位领域专家，他在有20多年的领域工程经验，积累了大量的领域需求，可是在其每进行一次产品开发时，他总是感到他所理解的需求无法为与他配合的分析人员、设计人员所接受。当我们一起来讨论这个问题的时候，共同的一个观点就是：没有对需求进行有效的管理，已经形成的需求文档没有很好的复用。所以需求管理一个很重要的目标应是提高软件需求的复用率。

基于上述的问题，必须对需求进行管理，使需求能够真正成为软件工程和管理基线，使软件计划、活动和工作产品同软件需求保持一致，使需求可以复用。

软件测试的前提假设

测试人员进行软件测试的基本假设是“有罪推断”，即认为被测程序一定是有 bug 的，而且每个功能点的实现都存在 bug，而且一定存在严重的 bug。请牢记这个假设，一旦在日后的工作过程中产生了这样的认识：“这个功能很简单，肯定不会出现问题，就不再测试了。”或者“这个功能上一轮刚测试过，当时就没有问题，这一轮应该也不会有问题，就不用测试了。”等等诸如此类的意识，那么你就有90%的概率导致漏测，造成线上问题。其原因也正是这个测试工作的基本前提假设，一旦被违背就从开端导致了测试工作存在问题，所以真正出现问题的可能性也就很大了。正因为软件测试的这个前提假设，在导致了如果我们同开发人员看待程序的角度和出发点完全不同。因为，通常情况下一个有自信心的开发人员不会认为自己写的代码全部都有问题，他一定是认为自己的代码没有问题了才交付测试的。因此，如果要从事软件测试工作，那么就必须牢记并运用该假设。这个前提假设要求我们在实施测试的过程中不能放过任何一个细小问题。比如，某个程序运行时在控制台打印了一些错误信息，但是实际上该程序的运行和功能都没有问题，如果我们摒弃有罪推断的假设，从合理实现的角度去分析，那么就可以认为这是开发人员对于日志打印的输出控制没有做好导致的，属于微不足道的小问题，不需提出即可。于是，这就使你有90%的可能性错过了发现其编码上的异常分支判断错误导致的重大问题。此类案例更常见与那些小概率问题，即在测试过程中偶尔出现，但确实很难、甚至无法复现的问题，如果我们同样摒弃有罪推断的假设，这些问题也会从我们手边溜走，跑到线上由用户发现。相信诸如此类的教训在每一个测试人员那里都不是少数。所以，请转变思维，牢记这个假设。

测试工作的开展思路

从需求出发

无论什么样的软件产品，其设计开发的目的是为了为了满足一定的需求，这种需求或者是用户提出的，或者是某个关联系统提出的。因此，软件产品最终是为了交付给用户使用的，也因此可以满足需求是对于软件产品质量的基本保证，其它如扩展性、维护性等等其实也算是更为广义的需求。所以，我们开展软件测试工作必须从需求出发。首先要全面了解需求，包括其背景、关联性、用户特点等；其次要深入挖掘隐含的需求和关联，包括某个需求隐含了对于系统现有功能的修改等等。我们只有在全面、深入了解需求的基础上，才能设计全面、有效的测试用例来进行测试，以满足对于软件产品满足需求的基本质量保证。

测试依据是测试设计

我们进行测试设计的依据是对于软件产品需求的全面和深入分析，但是需求决不全是软件测试的依据。因为我们不仅要验证需求，而且要验证设计。比如程序实现的异常（指针越界、字符串 copy 越界等等）判断，是保证软件产品可以正常运行的必要实现，但是我们在需求中是无法描述和分析出来的。那么进行测试的依据是软件产品的设计或者是代码吗？当然也不是，因为如果依据开发人员的设计或代码来进行测试的话，设计或代码正确，但是不符合需求逻辑的错误就无法发现。而且，如果依据设计或代码进行测试的话，那么也就违背了我们进行软件测试的基本假设——有罪推断。所以，我们进行测试的依据应该是我们根据对需求的全面深入分析和对设计实现的了解，两相综合产生的测试设计。正因为如此，测试是否充分和有效的根源也是测试设计。所以，我们的工作重点也是测试用例设计。

测试人员只是验证质量

首先要明确的是测试人员无法保证软件产品的质量，软件产品的质量是通过参与软件过程的各方联合共同保证的。有两个原因：一、由于软件测试人员不是产品设计人员和开发人员，所以无法做到比他们更了解产品需求和产品设计，如果他们都无法保证需求和设计没有问题，那么测试人员就更无法保证了；二、软件测试位于软件开发生命周期的末端，如果依靠测试人员来发现所有的 bug 来保证质量的话，那么风险就会后置，导致问题修复的成本增加，同时也增加了修复问题带来新问题的风险，因为在项目末端是不可能投入过多的成本来进行那怕接近全面覆盖的测试的。也正因为如此，我们是无法决定一个软件产品质量的好坏的，只有 PM 设计出产品需求，RD 编码完成，我们才能够通过我们的工作，促进 PM、RD 改进他们的产品、系统，从而达到产品、系统的高质量。所以，我们才要参与需求评审和设计评审，大家一起努力，各个阶段由专业化的人员来保证阶段的质量，将问题尽量在前期发现。而测试人员只能根据前期分析的结果，设计出测试用例，来验证软件产品在事先设计或后续补充的测试用例上不存在问题。但是“测试人员只是验证质量”决不是指我们可以不为产品质量负责。因为大家（PM、RD、QA、OP 等）工作的最终目标是产品质量保证，这个目标是大家共同的目标，所以每个人都必须为这个目标负责。只是由于咱们处于软件生命周期的最后一个环节，所以目前看起来产品质量都是由我们来负责和把握的，实际上，如果最终发布的软件产品出现了问题，那么无论如何我们肯定是有责任的。

测试的内容一定是确定的

既然软件测试只能验证质量，那么所要验证的内容必然是确定的，或者说是概率确定的。否则也就无从验证了。因此，模糊不确定的需求我们无法验证，输出结果没有任何规律的程序设计我们也无法验证，这就是软件产品的可测性判断。也因此，我们进行需求评审和设计评审时是一定要保证这个基本点的。

测试的目标不是没有 bug

综上所述，进行软件测试的目标不是通过测试使得软件产品不存在 bug（这是不可能达成的），而是我们根据确定的需求、确定的设计和确定的测试用例验证出的结果不存在 bug。同样的，测试人员的目标也不是在测试执行过程中去找 bug，而是运用测试思维，使用一定的测试方法，尽可能早地在需求阶段、设计阶段将所有的 bug 找出来，真正测试执行阶段只是验证不存在用例所描述的那样的 bug，而不是通过用例去发现 bug。

测试人员的工作方法

通过前文的分析，我们可以总结出测试人员的工作方法是这样的：首先对需求进行全面深入地分析，接着去分析评审程序设计，假定每个需求的功能点开发人员的实现都是存在问题的，同时也假定每一个程序设计的编码实现（无论是方式还是代码写作）都是存在问题的，然后根据这些假定设计测试用例，最后执行这些测试用例，验证程序不存在那些问题。从中不难看出，我们同开发人员同时由需求出发，开发人员产生详细设计和代码，我们产生方案和测试用例，然后开发人员提交被测程序，由测试人员同时运行被测程序和测试用例，来动态验证程序质量。所以，测试方案和测试用例设计的过程等价于开发人员进行详细设计和代码开发的过程，两相对比可以看出，测试人员最重要也是最核心的工作就是测试设计。因此，测试人员的工作可以重点描述成：是一个运用测试的思维和各种测试理论及方法，将所测试的软件产品的每一个功能都改变成一组特定的输入和一组特定的输出——确定对应的形式，形成测试用例，然后待开发人员提交测试后，在测试环境部署被测程序，根据测试用例进行主动测试的过程。

我眼中的测试工程师

最近这几年，测试的工作一直在发展，工作的内容也有了较大的变化，很多技术之类的字眼被提了出来。现在也经常有朋友会迷茫自己到底是在做什么样的工作，该怎么理解测试这个工作。

其实，我自己也是非常困惑的，最近突然想了想，觉得有点理解了，记录一下，呵呵，如果有不对的地方，还希望大家多多包含。

把我们 IT 行业比喻成一个武林，那各种不同的职业就必然是各个门派+武林高手了，每个派别有各自擅长的绝技和修炼方式。当然我们也是属于其中的一个派别。

这些门派的共同点是：

一、初级人员

潜心修炼本门派的武功，基本无暇顾及及其他门派的

二、高手

本门武功已经修炼成熟，对其他门派的武功也有了较多的认识

三、宗师

不止对本门武功修炼到如火纯清，更加的了解其他门派的武功，并且优劣和如何拆招都有着独到的见解。

我认为测试人员就要修炼到宗师的这种程度，对测试的各种手段、策略都精通掌握，多开发、前端、视觉、交互、业务等其他的内容也要熟悉。有一个人物我觉得特别适合，简直是测试中宗师的典范，那就是王语嫣啦。虽然她不会任何武功招式，是完完全全的一个弱女子，但是她熟悉各个门派的武功招式，并且能准确的分析有优点和缺点，并且能在实战中，迅速的找到破绽，以及给出非常合适的破解招式。所以说，作为一个测试人员，不一定要在开发技能、用户体验、视觉上有多么的能干，关键是能够针对不同的场景给出适当的建议和意见。我觉得修炼到这种程度要以下几点：

一、扩大知识面

学习和我们相关的知识，不单从开发的技能上入手，还有产品设计、前端、视觉、交互、用户体验，尽可能多的去看。

二、深入研究优缺点

针对每个行业，每种技术或者方案，研究适用的范围，不适用的范围。这个学习研究，不是说是如何使用，是说能够准确的识别适用的场合、冲突的场景。

测试是一个军师的角色，对战不一定要很厉害，但是给出的意见和方案，必须能够中肯、有效、服众。

程序员不是包身工

程序员是活生生的人，他们有自己的情欲，兴趣和生活。老板们有了一个现实的愿望，想实现它，经过辛苦和几番努力让这个想法能够成形。要么是有了投资人，要么是自己的收入所得，你可以用这些钱来雇佣其他人帮助你。这些被雇佣的人把自己生活相当重要的一部分投入到帮助实现这个愿望中。静下心来想想，这些人值得你去感激。

在大卫的这篇博客中，他说员工在下班后做业余项目对老板不利，对公司有害。这种思想太落后了。

首先，一个人在业余时间应该做什么、不应该做什么，别人不该说三道四，这是一种极大的侮辱，这是对他人无礼的个人评判。这是一种极其短视的行为。

老板应该鼓励员工做业余项目

尤其是对一些在创业公司工作的中坚(例如——设计和开发老板的产品的这些人)，做业余项目对他们有格外的好处。下面就是一些：

试验：在建议公司使用某些新技术前可以先在业余项目上试验一下;做再多的理论研究也不比曾经使用过某个新框架能获得更真切的个人评价。

独立自主：这是一个他们能够独立自主做所有决定的地方(无论正确还是错误)，而不是像在公司里通过各种高谈妥协和才能出一套方案。你可以把这视作创造力的释放。

加强知识掌握：程序员有机会能在自己喜爱的方向上进一步研究某种技术，这比在公司上班的标准时间里学习要神速的多。

消遣解压：提供多样性的生活节奏，避免因为在公司长期从事某一种工作/项目而带来的疲倦的感觉。

专心：能够促使程序员更高效的完成工作，因为他们并不能把所有时间都放在工作上。有句老话说，“如果你希望快速的完成一个事情，那就去问问一个忙碌的人是怎么做的。”

为开源社区做贡献：如果不是因为有如此多的人在业余时间做项目，你将看不到这些优秀技术社区向开源项目捐赠代码。业余项目对此有很大帮助。

社交网络：程序员有机会经常和日常工作外的人群接触，这能帮助他们获取知识，扩大社交网络。也许某天他们能给你的创业公司介绍新的人才。

我相信还会有很多其它的好处。

如果我只是埋头在以前的那些工作上，我将不可能现在在 KISSmetrics 公司开发产品(我甚至不可能进入科技界，因为我的专业是电子工程)。对于目前的工作，我所有的技能都是来自例如 Greenhorn Connect 等业余项目，在业余时间去学习新技能，饥渴的阅读各种书籍，每个老板都希望能雇佣到对所做的事情有热情的员工，希望他们具有广泛的知识，精湛的技能。但这是有连带的附加条款的，你必须接受他们的业余活动。

做业余项目的程序员更容易辞职?

在回到大卫的那篇议论,他看起来认为,如果一个员工有了业余项目,他们很可能会辞职去自己创业。如果一个程序员真的准备好了去经营一个业务,那他真的应该立即辞职,但有很多原因他们不会这样做:

资金问题 他们缺乏资金,并且知道当一个创意成熟时去募资是愚蠢的。更重要的,并不是所有的业余项目都有清晰的盈利前景。

动机问题 很多的业余项目都是因为兴趣和爱好。有时这些项目可能会变成值得用全天时间去开发的业务,但通常,他们只是一种业余爱好,不值得投入所有时间。

生活稳定问题 公司工作很稳定。很多事情会影响一个人的生活,并不是所有程序员都喜欢去创业,同时也有时间因素。他们也许结婚了,也许会迁移到另外一个城市,很多事情都会导致他们不会把创业放在第一位考虑。

但所有的这些都不会阻挡他们把价值奉献到工作上。

现在是卖方市场,程序员是抢手货

如果你拥有很好的技能,比如设计,产品管理,技术工程,在创业公司工作是个不错的选择。各种公司一定会争相聘用你。除了薪水的扶摇直上,老板们会发现,吸引有才华的人,仅用钱是不够的。良好的公司文化,老板知人善用,尊重他们的自由,这都是重要的因素。

大卫的观点也许适合你,但我要告诫那些老板,抛弃你们那些对员工不信任的态度,支持他们做业余项目。你们从他们这些活动中获得的隐形好处远比大卫所说的那些损失要多的多,而且避免了阻拦有才华潜能的程序员的进步发展。

新产品开发的两大教训

我从开始用敏捷开发的模式带领团队快速交付，一路走来有很多的经验，也有很多的教训。这里分享两个曾经的失败教训：

1. 没有正确理解 PO 角色：PO 要和客户直接沟通，要能真正代表客户意图，否则产品很可能失败

关键的客户沟通不到位，导致快上线了，客户认为的关键功能 PO 都不知道。产品开发之初，公司没有产品负责人（PO），也没有敏捷开发模式，完全是业务部门和客户直接沟通，然后反馈给研发部门。当时的问题是业务部门没有时间和研发部门沟通，往往是一页纸的需求，细节也没有和客户沟通清楚。后来公司成立了产品部，业务部门把需求反馈给产品部，由专职的 PO 和研发部沟通需求。请注意信息流转的方式：客户 -> 业务部门 -> PO -> 研发部门。确实，需求非常清晰，研发部的质量也不断提高。但是，最大的问题出现了：客户的需求经业务部门转述后偏离了客户的本意，产品部又没有和客户直接沟通，Sprint 之后的 Demo 业务部门也没有和客户沟通，需求的偏离在不断的积累...在敏捷开发中，最终客户必须要参与到开发过程中，特别是 Backlog 和 Demo。PO 不代表客户，特别是 PO 不是和客户直接联系时，这个问题就会出现。

2. 不要认为会给研发团队很好重构的时间

产品开发之初，为了尽快上线，产品的技术架构、性能、负载均衡、高可靠性等均没有考量，总认为可以不断的重构，但实际上公司根本不会给团队一个完整的时间去做重构，不断的业务需求占据了团队几乎全部的时间。于是脆弱的架构越来越弱，完成一个新的需求需要越来越长的时间。现在总结一下，架构的高扩展性和稳定性 是最高的优先级，可以满足业务需求不断扩展。网站的性能是次优先级，特别是登录前的静态页面。CMS（content Management System）也是很重要的，最好在项目的前期就做，可以选择外包的方式，否则大量的页面修改，要根据发布流程，业务部门会不断提出申述...所有这些需要在 Sprint0 阶段就要考量清楚，实践中 Agile Modeling 确实是实用的。

落地敏捷：计划会议中的估算，你纠结吗？

在 Scrum 中的计划会议中，估算是较为让人纠结的环节。主要的问题是：1) 花费的时间长；2) 估算不准确 3) 估算是否有必要

原因是什么？我认为主要是：1) 对需求的理解不到位；2) 每一个需求严格按照 planning poker 的方式，对于简单的需求和强分工的任务意义不大，但是耽误了时间；3) 设计的风险没有被发现；4) 团队受到“兑现承诺”的压力，一般会“悲观估算”

如何解决这些问题，我的建议是：

①PO 在 grooming 会议之前提前将需求发出，开发团队是带着问题进行需求讨论，否则刚刚拿到需求，讨论的效果并不好

②需求的描述，不拘泥于用户故事的形式，详细程度由团队的成熟度决定。团队越成熟，在简单的需求描述情况下依然可以把需求讨论清楚，可以简单描述；否则还是详细描述为佳。

③需求和设计的讨论，异常处理、用户使用场景、非功能需求往往是被遗漏和疏忽的地方

④风险前移：复杂的设计还是通过时序图等在白板上进行，不要把设计的风险带到实现中。

⑤PO 和团队都要承认“估算是 inaccurate 的”，估算的目的是为了帮助团队消除需求和设计的理解风险，并得到哪些需求可能被交付。PO 和团队关注的焦点在于“迭代的目标是否被实现”，而不是“团队承诺的需求是否全部被实现”。

⑥忘掉前面几点，在回顾会议中找到适合自己的做法，持续改进。可能这样的方式最适合：估算和实际校准的成员，估算不需要扑克的方式，直接他估算即可；反之则需要发现原因，是需求理解还是设计风险，扑克、时序图、活动图、反讲都要用上。也可能另一种方式最适合：放弃估算，根据最后需求完成的情况不断分析问题所在，然后在迭代过程中加强这些问题的解决...

泽众软件工具使用技术支持

电话：021-61079698

Email: sales@spasvo.com

QQ: 1404189128

MSN: spasvo_support@hotmail.com

	产品租用		
	下载	在线申请	详细
	<p>AutoRunner 是一款自动化测试工具。AutoRunner 可以用来执行重复的手工测试。主要用于：功能测试、回归测试的自动化。它采用数据驱动和参数化的理念，通过录制用户对被测系统的操作，生成自动化脚本，然后让计算机执行自动化脚本，达到提高测试效率，降低人工测试成本。</p>		

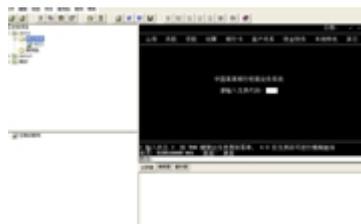
	在线体验		产品租用	
	企业版	免费版	在线申请	详情
	<p>TestCenter 是一款功能强大的测试管理工具，它实现了：测试需求管理、测试用例管理、测试业务组件管理、测试计划管理、测试执行、测试结果日志察看、测试结果分析、缺陷管理，并且支持测试需求和测试用例之间的关联关系，可以通过测试需求索引测试用例。</p>			

其他测试工具

Precise Project Management



Terminal AutoRunner



PerformanceRunner



有关培训、产品购买及试用授权方法等事宜

电话：021-61079698

Email: sales@spasvo.com

QQ: 1404189128

MSN: jennyding0829@hotmail.com

