

一次中间件结合业务的性能测试

软件测试需要学习些什么技能

五年软件开发的一点自我总结

优秀的开发人员和测试人员应有的态度

软件测试管理中存在的问题及分析

从传统ERP测试向主流互联网测试转型

浅析缺陷责任期与保修期

如何少走弯路的10条忠告

上海泽众软件电子期刊

2014 年 1 月 第二十五期

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

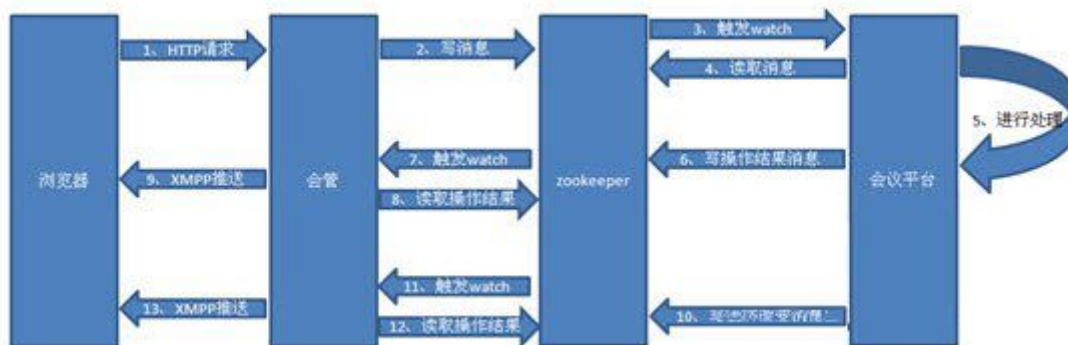
论坛：bbs.spasvo.com

目录

一次中间件结合业务的性能测试.....	4
软件测试需要学习些什么技能.....	6
五年软件开发的一点自我总结.....	8
优秀的开发人员和测试人员应有的态度.....	10
软件测试管理中存在的问题及分析.....	12
从传统 ERP 测试向主流互联网测试转型.....	22
浅析缺陷责任期与保修期.....	24
如何少走弯路的 10 条忠告.....	26

一次中间件结合业务的性能测试

开篇先扔一张图，下图是我本次测试对象的简单架构图：



先简单介绍下整个流程吧，我们根据图中画的来说（下文中说到的节点之后会讲）：

- 1、首先浏览器发出一个 http 请求至会管后台
- 2、会管收到请求后向 zookeeper 的一个节点（数据节点）中写入消息（一般就是某种请求的消息）
- 3、会议平台 watch 到 zookeeper 中该节点数据变化便从中把这条消息取出，并进行处理。注意：异步消息（我们应用大部分是异步消息）
- 4、平台在取出 zookeeper 节点上的消息后会立即往 zookeeper 中的另外一个节点（数据节点）上写入消息（该消息表示我平台已经收到来自你会管的消息了）
- 5、会管也会 watch 到这个节点的变化进而去读取这个消息，然后将该消息放入 xmpp 服务器，由 xmpp 推送给浏览器（当然，其实这个消息并没有什么实际意义）
- 6、另外还有就是等平台处理完一开始会管的请求消息后（3中的消息），会再次推送处理结果后的消息（比如某个人被静音了，就会推送一串 json 数据，其中有标识说这个人被静音了）到 zookeeper 的状态节点（OK，先不要管他什么数据节点和状态节点）
- 7、然后会管 watch 到后读取该数据并进行一系列封装和处理
- 8、数据处理和封装完成后将数据放入 xmpp 服务器
- 9、由 xmpp 服务器推送给浏览器，从而实现浏览器的实时状态

OK，说到这里，基本的架构以及流程说完了。现在，有一个任务就是，会议管理界面上的某个按钮点击之后，“实时状态”感觉变化很慢。（比如我点击全部静音按钮，页面上与会人员的图标会显示出一个图标表示被静音了，对，就是这个操作后，图标很久才显示出来，测试找出原因）

如果交给各位读者这样的测试需求（当然，这个根本算不上需求，我就这么被忽悠了），读者朋友们会怎么进行这项测试呢？

好了，各位，下面我给出我本次测试中采用的方法（如果觉得有更好的，请留言联系我!），先看图：



解决方案

接下来我简单介绍一下我这个类似“框架”的东西吧。

- 1、首先初始化一个线程池
- 2、线程一：使用 smack 包登录 xmpp，监听所有消息，并过滤出我此次测试中需要的消息
- 3、线程二：使用 httpclient 来模拟 http 请求并接收响应，并提取响应中的某些重要数据
- 4、线程三：使用 zookeeper 的 java 客户端，连接我们的 zookeeper 服务器并设置节点的 watcher
- 5、线程四：收集前三个线程的数据，并附加时间戳

只要将我们四个线程放入线程池中运行，即可。

基本都已经介绍完了，如果有需要代码的童鞋也可留言回复哦（注：本次测试脚本有效代码量在2K行左右）~

软件测试需要学习些什么技能

- 1.测试环境（网络环境， windows 环境等）
- 2.数据库管理
- 3.编程技巧（java 编程设计， 脚本语言， 设计工具， XML 编程）
- 4.软件测试技术（测试理论， 方法， 流程， 文档写作， 测试工具， 自动测试）
- 5.测试技术实践

软件测试的基础知识（黑盒测试， 白盒测试， 单元测试， 系统测试）

软件测试的基本工具（测试管理工具， 自动化测试工具， 性能测试工具）

其他：

一定的编程知识是需要的还需要数据库， 中间件， 网络协议 CMMI 等软件工程的理论也是重要的软件测试要求知识面广， 但不一定精， 编程语言的话， 最好是什么都学过， 至少几种主流的程序语言要学过， 如 java、 .net、 C 。 还有会一些脚本语言 vba (vb)、 sql、 C 等。

网络的话： TCP/IP 协议， 局域网广域网相关知识等

数据库： 当前主流的 mysql、 ms-sql、 oracle

常用测试工具： HP 的三剑客首当其冲了： QTP（自动化功能测试工具）、 LoadRunner（自动化性能测试工具）、 QC（TD 前身， 测试管理追踪工具）。当然这些都是收费产品。网上有破解版的可供学习。（一套）

开源免费的测试工具： QTP（自动化功能测试工具， 破解版的可以满足绝大部分测试需求）、 jmeter（自动化性能测试工具）、 bugzilla（测试管理追踪工具）。

信息产业部门发布的《2006年度软件测试行业专项调查报告》显示， 国内 IT 企业招聘软件测试人员时， 比较重视的依次是： 软件测试技术、 相关经验、 缜密的逻辑思维能力、 责任心和团队合作精神、 出色的沟通能力、 良好的学习能力、 耐心、 细心、 信心、 优秀的书面表达能力等。雅虎中国品质保证部经理鲍女士将作

为一个合格的软件测试工程师所应具备的能力总结如下：

第一： 专业系统的软件测试知识。软件测试作为一门专业的技术有其自身的特点， 如果想在软件测试领域有所发展， 专业系统的软件测试知识是必不可少的。例如： 搭建测试环境、 编写测试计划和用例、 进行黑白盒测试、 使用和开发测试工具等。

第二： 一定的项目经验。软件测试职业在意测试经验的积累， 特别是大型的软件测试项目经验对于测试人员水平的提升是至关重要的。同时要求测试入门者具备一定的项目经验也是对软件测试负责责任的体现。统需求报告中的每一个需求点都会在测试需求中成为几个测试需求点， 如果不够耐心， 不够细心

是很容易遗漏测试需求点的，而这些遗漏的地方往往是问题点 所以说认真、细心、不怕麻烦是一个好的测试人员必备的素质要求。

第三：善于沟通。测试工作是一个问题的爆发点，特别是对于那些开发流程不规范的单位。作为一个测试负责人要经常和开发人员、测试人员、公司领导沟通，而作为一名好的测试人员需要掌握如何处理、协调好人际关系的方法和技巧。

第四：学习能力强，善于总结。软件测试技术随着时间的变化也在不断地发展，作为一名优秀的测试人员要善于利用书籍、网站、论坛、交流等各种途径不断提高自己的软件测试水平。不断的学习新技术，不断总结在实际工作遇到的问题 and 解决的方法，并将其整理归纳，是一个测试人员提高自己的技术水平的最好方法。

五年软件开发的一点自我总结

马上就要去新公司报道了，换城市（离开北京）+换职业（不做开发），这个决定下得其实还是蛮艰难的。最近一直在反思自己毕业后的这五年工作，自感惭愧，找个地方记录下（主要是这里谁都不认识我），也算是自己的一种反醒以及给过来人提点建议，少走点弯路吧。

08年顶着名校硕士的光环加入了一家非常有名的非软件公司做软件开发，刚开始一切其实都很美好。大外企的各种好在头一年给自己带来了许多光环，当然自己也学到了很多(主要是非技术的东西)。可是从第二年开始，当自己被各种邮件，开会和扯皮的事包围后，技术能力急转直下。然而自己当时还没意识到这个问题，感觉钱还行，也不忙，再加上本来就很迷茫，就得过且过了。直到去年，很多清华北大同事的离职，日复一日的简单重复工作，明显的天花板，不涨的工资，以及家庭原因的集中爆发才让自己后知后觉，才开始反醒，才开始下决心做转变。

然而转变是痛苦的，这五年技术上主要是在 windows 平台上做一些企业内部业务的处理和展示，主要用一点 c++/c#，还有 MFC，Winform，WPF，WCF。技术基本上是做的皮毛，一般问题用 MSDN，google 和 stackoverflow 就能基本解决。用不到数据库，也用不到什么数据结构，用到一点点网络知识，主要精力都在业务展现上。这一切在我看来招个应届生用一年也能有和我一样的开发能力，唯一懂的多业务，也长进不多，都是繁杂的重复。所以在自己开始面试和找工作时被 bs 了很多次，顶着光环人家一般都给你面试机会，但是一旦聊到技术细节，自己很多都答不上来，也曾经一度非常沮丧。庆幸最后找了个技术相关的职位，能够兼顾到家庭和自己后面发展的想法，还算可以，这是后话，暂不讨论。

总结这五年不成功的开发经验，现在想想主要问题如下：

目标不明确，由于学的专业不是纯软件，自己误打误撞的进来做软件开发，自己一直没想清楚自己的目标，结果导致自己浑浑噩噩的做了 5 年开发还停留在原地。

缺乏坚持，没能坚持一件事情做到底，自己其实在这五年也自学了一些东西，包括 HTML，JavaScript，CSS，Python，Java.....也看了不少技术的东西。可都很肤浅，一样也没有深入坚持下去，什么都懂一点皮毛，结果就是什么都不懂。

不够勤奋，我从小成绩优秀，学习能力很强，缺乏需要非常勤奋投入才能做成事情的锻炼，所以养成勤奋的习惯，到了工作中才发现要取得成功无不需要加倍的努力才行。

此外从一个非科班出生搞软件的角度，谈谈自己对软件开发的了解：

做上层开发学了这么多所谓的语言，框架，技术，其实现在想想万变不离的却是底层的数据结构，操作系统，网络通讯和一些其他的基础的东西。自己也算半路出家，以前总是不屑基础的东西，总认为我懂了 WPF，能画出很炫的界面什么就很好了。其实却是认识太肤浅，只有掌握了不变的东西才能应付好上层的技术，否则只会被微软牵着鼻子走。所以对新人来说，打好基础是多么重要，可惜我后知后觉了。

除了基础的东西，我认为其次重要的是软件架构，好的架构可以让事情事半功倍，让下面实现具体代码的感觉很轻松。说得过分点，一个好的架构师可以带一群实习生的干活都能出好东西。软件开发越来越不需要我们懂底层的东西，很多现成的轮子可以用，但是要知道怎么搭积木就需要好的架构师了。

所以我觉得平常要多看好的开源架构，多学习框框的东西，当然轮子和积木知道哪里能拿到。最怕的是像我一样，只会 google 找积木搭，这就是所谓的纯码农了：（

其次是业务，开发很多时候要懂业务，很多开发的很不屑业务，认为技术才最重要。其实很多时候业务才是王道，软件开发在我看来大部分时间都是给业务做服务的，做支持的，公司都是靠业务才能赚钱的。所以开发人员也要懂业务，特别对那种不是很想一直做开发的来说，懂业务又懂技术也许会给你更多的机会。

最后是软件开发的职业发展，个人的看法是码农能爬上去做架构师和经理的其实很少，基本一个萝卜一个坑，我见到开发能一直做到退休的都是老外，人家基本没有 title 的，这样的我观察无非两种人。一种是对业务很精通，同时对技术系统又很懂的，公司少不了他。一种是对技术很热情很钻研，能力很强，又不想去要什么 title 的，基本可以挑大梁。我想国内的情况虽然不太一样，不过基本可以借鉴下老外。当然不做开发还有很多可以做，我想说的是开发是可以做一辈子的，就看个人的想法了。这个问题我自己也一直在想，目前只有一条模糊的路，也得摸石头过。

以上是作为一个经验尚浅的码农对工作五年的反思，基本是想到什么就写什么，思路有点乱，敬请谅解。

优秀的开发人员和测试人员应有的态度

RajaramanRaghuraman 是一位具有丰富经验的软件从业者，是 AgileDevTest 博客网站的站长，以及电子书《Programmer's Motivation for Beginners》的作者。针对优秀的软件开发人员和测试人员应该具有什么样的职业态度，他专门撰文进行了阐述。

在针对软件开发人员的文章中，Raghuraman 认为软件开发是一种艺术，而不是一门科学。你可以学习软件开发的各项技术，不过还需要对编码有着充分的热情。如果你是这样的人，那么就可以成为“伟大的开发者”。Raghuraman 列出了优秀的开发人员应该具有的态度：

态度1：缺陷表明我编写优秀代码的能力出现了问题

修改缺陷是软件开发人员的日常活动。显然，缺陷是开发人员最大的敌人。当修改缺陷的时候，又有多少人能想到以下几点呢：

起初的时候，我能做些什么来避免这个缺陷呢？

我怎么能让这个缺陷逃过我的眼睛呢？

好了，既然问题已经产生了。我下次该如何避免相同的错误呢？下一步该做些什么呢？

事实上，很少有开发人员这么想。

愿意成为优秀开发者的人应该将缺陷视为其位置的一种威胁、个人可信性的一种威胁以及编程技巧的一种威胁。正是这种态度能够使其成为伟大的开发者。

态度2：测试人员，有本事就在我的代码中发现缺陷

有多少人具有这样的态度呢？很多开发人员认为测试的任务就是发现缺陷。的确如此，但是这并不意味着开发人员就可以将缺陷视为理所应当的事情。优秀的开发者愿意邀请测试人员在他的代码中寻找缺陷。他们对自己的代码有这样的自信。

态度3：在代码质量问题上不妥协

对于开发者来说，最重要的事情就是代码质量了。这包括正确的编码标准、使用恰当的设计和重构保证代码更加可维护等等。但是，我们通常会因为一些众所周知的原因在代码质量问题上妥协，比如面临时间和进度的压力。优秀的开发者永远不应该在代码质量上妥协，不管所面临问题是什么。

态度4：自信但不傲慢

优秀的开发者应该对自己的能力有自信，但是不应该对其他开发人员和测试人员傲慢。他应该始终记住他是团队中的一员，这个团队要准时高质量地完成项目的共同目标。

态度5：感激测试人员

尽管优秀的开发人员工作非常努力认真，但是测试人员依然可能在他的代码中发现缺陷。如果这样

的话，感谢这些优秀的测试人员。优秀的开发人员应该感谢测试人员所发现的缺陷。他应该记住这一点，他的敌人是缺陷，而不是做测试的人。

在针对软件测试人员的文章中，Raghuraman 认为开发人员和测试人员的工作方向正好相反，但是有着共同的目标，那就是生产高质量的软件。他认为优秀的测试人员应该具有如下的态度：

态度1：我要不惜一切代价破坏软件

不要产生误解！测试人员的工作就是要发现缺陷，在这个过程中，要随着时间的推移，使软件变得越来越好。缺陷是测试人员的好朋友，因此他最重要的事情就是不惜一切代价破坏软件，发现漏洞。这个过程中，可以系统性地执行测试用例，也可以进行临时测试或探索性测试。

如果你想成为优秀的测试人员，那么你的态度应该是“不惜任何代价破坏软件并查找缺陷”。

态度2：优秀的开发人员，我一定要在你的代码中找到缺陷

不知道有多少测试人员能有这样的态度？

但是如果成为优秀的测试人员，就需要有这样的态度：不管开发人员多优秀，也要努力在他的代码中找到缺陷。

态度3：在测试质量问题上不要妥协

就像代码质量一样，测试也需要高质量。优秀的测试人员不应该在测试质量上做任何妥协。

态度4：感激开发人员

有可能优秀的开发人员编写的代码不包含任何缺陷。难以置信吧？如果是这样的话，那就感谢开发人员吧。

如果开发人员所编写的代码没有缺陷，优秀的测试人员应该感谢并尊重他们。应该记住这一点，他的朋友不仅仅是缺陷，开发人员也应该是他的朋友。

态度5：我绝不比开发人员差

在测试领域，是不是有很多人觉得比开发人员低一等？相信有很多这样的人。如果测试人员觉得比开发人员低一等的话，那么他实际上并没有恰当地做好测试工作。正如前面所述，开发人员和测试人员工作方向相反，但是有着共同的目标，那就是生产高质量的软件。

如果你想成为优秀的测试人员，那应该首先为自己的工作感到骄傲，不要觉得低人一等。

态度6：要比开发人员懂得更多

有多少测试人员在技术和架构方面懂得比开发人员还多呢？可能并不太多吧。作为测试人员，工作内容不应当仅仅是编写和执行测试用例并报告缺陷。他应该在技术上比开发人员了解得更多，只有这样才能知道在什么地方能够破坏软件并找到缺陷。

优秀的开发人员和测试人员除了高超的技术技能外，还都具备专业的职业态度，希望 Raghuraman 的总结对您有所帮助。

软件测试管理中存在的问题及分析

摘要：本文结合实践，主要探讨了在中小型软件企业中，在测试资源不是很充足的情况下的软件测试管理。文中前两部分简要介绍了软件测试管理及测试的范围，方法及重要性，之后对当前国内中小型软件企业在测试及测试管理中可能存在的问题进行了简单的介绍与分析，最后介绍了一些较好的解决方法。

关键词：软件测试；测试管理；测试问题；管理体系

1、引言

随着 IT 技术的迅速发展，计算机在各行各业日益广泛的应用，软件产品的不断推出，计算机软件已经越来越深入到人们的生活中，人们对计算机软件质量的要求也就越来越高。如果软件存在故障，将可能造成人力、物力和财力的巨大浪费；如果软件的质量不高，其维护费用不仅将大大超过其开发费用，而且会使维护变得很困难，甚至将可能造成不可弥补的损失。

软件测试是软件质量保证的关键步骤。美国质量保证研究所对软件测试的研究结果表明：越早发现软件中存在的问题，开发费用就越低；在编码后修改软件缺陷的成本是编码前的10倍，在产品交付后修改软件缺陷的成本是交付前的10倍；软件质量越高，软件发布后的维护费用越低。另外，根据对国际著名 IT 企业的统计，它们的软件测试费用占整个软件工程所有研发费用的50%以上。由此可见，为了保证软件产品的质量，必须对计算机软件进行测试。

随着计算机硬件成本的不断下降，软件在整个计算机系统的成本中占有越来越高的比例，如何提高软件质量是整个计算机软件行业的重大课题。软件测试作为软件开发的一个重要环节，日益受到人们的重视。为了尽可能多地找出程序中的错误，生产出高质量的软件产品，加强对测试工作的组织和管理就显得尤为重要。

由于软件测试至今仍令人捉摸不定，为确保测试工作的顺利进行，就要对其进行有效地管理。软件测试管理是一种活动，可以对各阶段的测试计划、测试案例、测试流程进行整理、跟踪、记录其结果，并将其结果反馈给系统的开发者和管理者。同时将测试人员发现的错误立刻记录下来，生成问题报告并对之进行管理。所以采用软件测试管理方法可以为软件企业提供一个多阶段、逐步递进的实施方案。通过此管理方法，软件企业还可以用有限的时间和成本完成软件开发确保软件产品的质最，进一步提高计算机软件在市场上的竞争能力。

一般应用过程方法和系统方法来建立软件测试管理体系，也就是把测试管理作为一个系统，对组成这个系统的各个过程加以识别和管理，以实现设定的系统目标。同时要使这些过程协同作用、互相促进，从而使它们的总体作用大于各过程作用之和。其主要目标是在设定的条件限制下，尽可能发现和排除软件缺陷。

但是当前，中国软件企业在软件测试方面与国际水准仍存在较大差距。首先，在认识上重开发、轻测试，没有认识到软件项目的如期完成不仅取决于开发人员，更取决于测试人员；其次，在管理上随意、简单，没有建立有效、规范的软件测试管理体系；另外，缺少自动化工具的支持，大多数企业在软件测试时并没有采用软件测试管理系统。所以对国内软件企业来说，不仅要提高对软件测试的认识，同时要建立起完善的软件测试管理体系。

2、软件测试及测试管理的范围

2.1测试的范围

下面主要就测试的参与者，测试要素，测试开始时应确定的工作，测试过程简要介绍软件测试的工作范围。

参与者

- 用户方代表
- 软件最终使用者
- 软件开发人员
- 软件测试人员
- 高层经理的支持
- 过程保证人员（SQA）

测试要素

- 正确性：数据输入，过程处理和输出的正确性（IPO）。
- 文件完整性：文件被正确使用，恢复和存储的数据正确。
- 授权：特殊的授权可以执行一个特殊的操作。
- 进程追踪：当进程运行中，程序有能力证实进程在正常工作。
- 系统运行的连续性：当有非致命性问题发生后，系统有能力继续运行关键的任务。
- 服务水平：系统有紧急情况发生时，要求程序的输出结果不经或进行简单的处理后就可以直接使用。
- 权限控制：防止系统被误用（意外或者有意的）。
- 一致性：确保最终设计和用户需求完全一致。
- 可靠性：在规定的时间内都可以正常运转。
- 易于使用：多数人均感觉易于使用。
- 可维护性：可以很容易的定位问题，并且进行修改。
- 可移植性：数据或者程序易于移至到其它系统上。
- 耦合性：系统中的组件可以很容易的联接。
- 性能：系统资源的占用率，响应时间，并发处理。

●操作性：易于操作（Operator）。

测试开始时应确定的工作

●需求阶段

→确定测试策略

→确定收集了足够的需求

→产生功能性的测试用例

●设计阶段

→确定设计和需求之间的联系

→确定进行了足够的设计

→产生结构和功能的测试用例

●编码阶段

确定和设计之间的联系

确定拥有执行的足够条件

产生结构和功能的测试用例

●测试阶段

确定设计了足够的测试用例

测试应用系统已经完成

关键资源已经到位

●安装阶段

将测试完成的系统变为产品

●维护阶段

修改和重新测试

软件的测试过程

●估算：对软件工作量的估算；对软件系统的状况的评估。

●测试计划：详细的描述怎样能成功的完成测试工作，其中应包含必须的资源 and 实施计划。

- 需求测试：在软件开发的所有阶段进行测试，测试应该尽早，在需求和设计阶段发现的缺陷修正的花费最小。

- 设计测试：给测试要素打分；分析测试要素；对设计进行评审；检查修改的部分。

- 编码测试：编码是否按照既有的标准进行，过程是否易于实践；是否编制了足够的文档。

- 测试总结：表示出目前项目的实际状况；明确测试所做的工作，给出系统的操作性能的评价，明确什么时候系统可以进行产品化的工作。

- 安装，交付测试：检验检查表和产品的正确性；使用测试标准去检验发生的问题。

- 维护阶段的测试：开发一些测试用例，预先发现一些问题；对用户进行培训。

2.2测试管理的范围

软件测试管理要解决的课题是如何确保软件测试技术能在软件项目在软件生命内得到顺利实施，并产生预期的效果。按照管理的对象不同，软件测试管理大致分为软件测试团队组织管理、软件测试计划管理、软件缺陷（错误）跟踪管理以及软件测试件管理四大部分。

软件测试团队组织管理通俗的讲就是测试团队应该如何组建以及测试人员管理。在实际项目开发中，我们常常看到有些单位忽视测试团队存在的意义，当要实施测试时，往往临时找几个程序员充当测试人员；也有些单位尽管认识到了组建测试团队的重要性，但在具体落实的时候往往安排一些毫无开发经验的行业新手去做测试工作，这常常导致测试效率的低下，测试人员对测试工作索然无味。一个好的测试团队首先要有好的带头人，他必须具有极为丰富的开发经验，对开发过程中常见的缺陷或错误了然于胸，此外，他还应具有亲和力和人格魅力。其次，测试团队还应有具备一技之长的成员，如对某些自动化测试工具运用娴熟或能轻而易举地编写自动化测试脚本。另外，测试团队还应有兼职成员。如验证测试实施过程中，同行评审是最常使用的一种形式，这些同行专家就属于兼职测试团队成员的范畴。测试团队里往往不乏缺乏开发经验软件新手，这部分人可以安排去从事交付验证或黑盒测试之类的工作。

软件测试计划管理通俗地讲就是安排好测试流程。这部分内容具体涵盖软件测试策划、软件测试技术剪裁、测试进度管理、成本管理等几个部分。其中测试策划工作主要是指具体测试活动实施之前做好策划工作，如起草测试大纲以及测试计划；软件测试技术剪裁工作主要是指测试团队应根据软件项目的具体实际剪裁出所要实施的测试技术；测试进度管理工作主要是指排出各项测试的时间进度及人员安排，如有变动时应做相应调整；测试成本管理工作的内容即开列出测试活动中会涉及到的资源需求。

软件缺陷（错误）跟踪管理通俗地讲就是确保发现的缺陷（错误）已经被开发团队纠正或处理过并且没有引入新的缺陷（错误）。具体来讲，当测试团队通过各种途径发现了文档或代码中的缺陷或错误以后，并不是交一份测试报告就草草了事，而是在递交报告以后继续督促开发团队及时关闭已知缺陷或错误（当然，如有必要应对这些缺陷、错误做严重程度排序，以便开发团队能视轻重缓急安排处理顺序）。当开发团队关闭了测试报告中的缺陷（错误）以后，测试团队还需验证开发团队在关闭过程中有没有引入新的错误。通常，这个过程称为回归测试。回归测试如发现问题，继续报开发团队，按上述流程循环，直至回归测试最终通过。

软件测试件管理通俗地讲就是指努力建设好测试团队的财富库并对测试团队成员进行技能培训以帮助他们能使用好这个财富库。这里，财富库是指软件测试件。测试件（Testware，指测试工作形成的产品）是一个不常见到的词汇，它包括是测试团队在长期实践过程中逐步积累起来的经验教训、测试技巧、测试工具、规格文档以及一些经过少量修改能推广至通用的测试脚本程序。测试件管理工作做得越

好，测试团队在实际测试过程中就能越少走弯路，测试团队内部的知识交流和传递就越充分，测试脚本或规格文档的重复开发工作也就能被有效地避免。软件测试件管理工作包括两部分，一是建设，另一个是培训。建设工作大抵是收集各类测试外文档、测试工具、测试脚本，也包括收集整理测试人员的会议发言、总结报告、技术心得等等。培训工作大抵是通过技术讲座、正式或非正式团队会议、印发学习资料等形式进行。

2.3 软件测试管理内容

具体的测试管理内容有：

- 测试方案管理：单元测试、集成测试和产品测试的测试计划的录入、修改、删除、查询和打印。

- 测试案例管理

测试案例的增、删、改、拷贝和查询；

测试案例测试情况的管理，如测试状态包括：未测试、测试中、已测试；

测试结果分为：通过、未实现、存在问题等；

测试案例输入、编号和归档。

- 测试流程管理：测试进度管理；测试流程标识；测试日志及状态报告。

- 问题报告管理：问题报告处理流程（问题报告、整改报告）、实现问题报告与测试案例的关联。

- 测试报告管理：生成单元测试、集成测试和产品测试的测试报告。

除了以上这些，在测试管理过程中还应对人员和环境资源进行管理。

3、测试及测试管理中的问题及分析

通过以上的简单总结与分析，可以看到软件测试及测试管理的重要性，及其复杂、广泛的组织管理工作，所以在实施起来，难免与理论有些出入。另外，国内的软件企业大多起步晚，技术基础薄弱，应用与管理经验缺乏，在测试上更是如此。于是国内的一些中小型的软件企业，在软件测试方面存在诸多问题，不仅与理论要求相差甚远，与实际的应用需求也相差很多。下面将简要介绍与分析当前国内中小型软件企业在测试及测试管理中存在的问题和问题原因，并在之后提出一些解决办法。

3.1 软件本身的复杂性与企业自身的不足

这里复杂性包括软件用户需求的复杂与难确定性，软件开发过程的组织管理的难控制性等，使得软件开发过程必然会存在诸多问题，开发出的产品也必然存在一些缺陷与不足。而由于生产与管理经验的不足，缺乏高效的开发与测试团队，往往是开发人员又是测试人员，或测试人员质量管理；缺乏有效的测试技术，代码走查室最常用的方法；测试开始较晚，往往在开发完成之后；对用户反馈信息缺乏整理总结等；使得不仅难以控制产品的缺陷数量，而且对于缺陷的定位与修补也很难到位。

3.2 测试的特性

3.2.1 测试是不完全的（测试不完全）

由于软件需求的不完整性、软件逻辑路径的组合性、输入数据的大量性及结果多样性等因素，哪怕是一个极其简单的程序，要想穷尽所有逻辑路径，所有输入数据和验证所有结果是非常困难的一件事情。

3.2.2测试具有免疫性（软件缺陷免疫性）

软件缺陷与病毒一样具有可怕的“免疫性”，测试人员对其采用的测试越多，其免疫能力就越强，寻找更多软件缺陷就更加困难。在软件测试中采用单一的方法不能高效和完全的针对所有软件缺陷，因此软件测试必须采用不同的测试方式和测试数据，应该尽可能的多采用多种途径进行测试。

3.2.3测试是“泛型概念”（全程测试）

如果单纯的只将程序设计阶段后的阶段称之为软件测试的话，需求阶段和设计阶段的缺陷产生的放大效应会加大。这非常不利于保证软件质量。需求缺陷、设计缺陷也是软件缺陷，记住“软件缺陷具有生育能力”。软件测试应该跨越整个软件开发流程。需求验证（自检）和设计验证（自检）也可以算作软件测试（建议称为：需求测试和设计测试）的一种。软件测试应该是一个泛型概念，涵盖整个软件生命周期，这样才能确保周期的每个阶段禁得起考验。同时测试本身也需要有第三者进行评估（信息系统审计和软件工程监理），即测试本身也应当被测试，从而确保测试自身的可靠性和高效性。

3.2.4软件缺陷具有空间聚集性（80-20原则）

80%的软件缺陷常常生存在软件20%的空间里。这个原则告诉我们，如果你想使软件测试有效地话，记住常常光临其高危多发“地段”。在那里发现软件缺陷的可能性会大的多。这一原则对于软件测试人员提高测试效率及缺陷发现率有着重大的意义。聪明的测试人员会根据这个原则很快找出较多的缺陷而愚蠢的测试人员却仍在漫无目的地到处搜寻。

80-20原则的另外一种情况是，我们在系统分析、系统设计、系统实现阶段的复审，测试工作中能够发现和避免80%的软件缺陷，此后的系统测试能够帮助我们找出剩余缺陷中的80%，最后的5%的软件缺陷可能只有在系统交付使用后用户经过大范围、长时间使用后才会曝露出来。因为软件测试只能够保证尽可能多地发现软件缺陷，却无法保证能够发现所有的软件缺陷。

3.2.5为效益而测试

为什么我们要实施软件测试，是为了提高项目的质量效益最终以提高项目的总体效益。为此我们不难得出我们在实施软件测试应该掌握的度。软件测试应该在软件测试成本和软件质量效益两者间找到一个平衡点。这个平衡点就是我们在实施软件测试时应该遵守的度。单方面的追求都必然损害软件测试存在的价值和意义。一般说来，在软件测试中我们应该尽量地保持软件测试简单性，切勿将软件测试过度复杂化。

3.2.6缺陷的必然性

软件测试中，由于错误的关联性，并不是所有的软件缺陷都能够得以修复。某些软件缺陷虽然能够得以修复但在修复的过程中我们会难免引入新的软件缺陷。很多软件缺陷之间是相互矛盾的，一个矛盾消失必然会引发另外一个矛盾的产生。比如我们在解决通用性的缺陷后往往会带来执行效率上的缺陷。更何况在缺陷的修复过程中，我们常常还会受时间、成本等方面的限制因此无法有效、完整地修复所有的软件缺陷。因此评估软件缺陷的重要度、影响范围，选择一个折中的方案或是从非软件的因素（比如提升硬件性能）考虑软件缺陷成为我们在面对软件缺陷时一个必须直面的事实。

3.3测试组织管理不专业

1、测试人员不独立于开发者，测试人员独立于开发者的程度将影响测试结果。

人很容易用自己已经非常仔细地做过测试来欺骗自己，开发人员做测试容易受到个人的习惯性想法的影响，程序中可能包含由于程序员对问题的叙述或说明的误解而产生的错误。如果是这种情况，当开发人员测试自己的程序时，往往还会带着同样的误解致使问题难以发现。开发和测试是两种不同的活动，开发人员对自己的程序进行一定的审查、调试是必要的，但是一般情况下还需要另外的专业测试者进行测试。不过，由于有的企业中，人力有限，或者认为没有足够的资源或理由支持一支测试队伍，有时不得不由开发人员测试；那么，开发者对自己的程序的测试需要注意许多问题，或者应由另外的开发者对自己的程序进行测试。

2、测试时间安排往往计划不周，测试计划有时受制于开发计划。

3、测试等级以及在那个等级上的测试时间往往选择不当。

4、测试辅助设备和测试工具将影响开发者的测试效率及测试彻底性。

5、测试策略文档的普遍缺失。

6、测试范围的确认经常被其他文档或经验所取代。

7、测试任务应该像 BUG 一样有明确的分级，不同类型的测试应该有相应的测试用例集合与之对应。

8、关键路径概念在测试规划时容易被项目经理弱化。

9、测试用例不科学，测试用例在实际中没有起多大作用；在实际测试时根本没有按用例执行；往往测试执行后没有把新的用例补充到用例库中。

3.4测试人员的影响

1) 测试人员入门容易学习困难，无章可循；人员增加可能有重复工作。

2) 测试人员对现实应用与需求的理解可能有偏差。

3) 测试人员可能对测试存在一些不正确的看法和错误的态度，如下：

(1) 认为测试工作不如设计和编码那样容易取得进展难以给测试人员某种成就感；

(2) 以发现软件错误为目标的测试是非建设性的，甚至是破坏性的，测试中发现错位是对责任者工作的一种否定；

(3) 测试工作枯燥无味，不能引起人们的兴趣；

(4) 测试工作是艰苦而细致的工作；

(5) 对自己编写的程序盲目自信，在发现错误后，顾虑别人对自己的开发能力的看法。

4) 提交以后对用户反馈信息缺乏及对缺乏足够的重视，对于有大量用户有持久生命力的软件产品（如 MicrosoftOffice），用户反馈信息较全面，便于开发和测试人员进行软件的修补和维护；而一些中小软件企业的产品却远远无法和 MicrosoftOffice 相比；于是可能缺乏足够的用户反馈信息，或没有足够

的时间或人力处理用户反馈信息。

5) 开发及测试人员工作习惯, 编程习惯, 测试习惯等也影响测试的效果; 由于测试人员短期的学习与培训, 一般能提高的只是方法和技巧; 而其自身能力与习惯可能的负面影响却一时难以消除。

4、测试管理问题的解决

4.1建立软件测试管理体系

建立软件测试管理体系的主要目的是确保软件测试在软件质量保证中发挥应有的关键作用, 包括以下工作:

软件产品的监视和测量: 对软件产品的特性进行监视和测量, 主要依据软件需求规格说明书, 验证产品是否满足要求。所开发的软件产品是否可以交付, 要预先设定质量指标, 并进行测试, 只有符合预先设定的指标, 才可以交付。

对不符合要求的产品的识别和控制: 对于软件测试中发现的软件缺陷, 要认真记录它们的属性和处理措施, 并进行跟踪, 直至最终解决。在排除软件缺陷之后, 要再次进行验证。

产品设计和开发的验证: 通过设计测试用例对需求分析、软件设计、程序代码进行验证, 确保程序代码与软件设计说明书的一致, 以及软件设计说明书与需求规格说明书的一致。对于验证中发现的不合格现象, 同样要认真记录和处理, 并跟踪解决。解决之后, 也要再次进行验证。

软件过程的监视和测量: 从软件测试中可以获取大量关于软件过程及其结果的数据和信息, 它们可用于判断这些过程的有效性, 为软件过程的正常运行和持续改进提供决策依据。

一般应用过程方法和系统方法来建立软件测试管理体系, 也就是把测试管理作为一个系统, 对组成这个系统的各个过程加以识别和管理, 以实现设定的系统目标。同时要使这些过程协同作用、互相促进, 从而使它们的总体作用大于各过程作用之和。其主要目标是在设定的条件限制下, 尽可能发现和排除软件缺陷。测试系统主要由下面6个相互关联、相互作用的过程组成: 测试规划、测试设计、测试实施、配置管理、资源管理和测试管理; 确定这些过程的顺序和相互作用, 前一过程的输出是后一过程的输入。其中, 配置管理和资源管理是这些过程的支持性过程, 测试管理则对其他测试过程进行监视、测试和管理; 确定这些过程所需的准则和方法, 一般应制订这些过程形成文件的程序, 以及监视、测量和控制的准则和方法; 确保可以获得必要的资源和信息, 以支持这些过程的运行和对它们的监测; 监视、测量和分析这些过程; 实施必要的改进措施。

4.2建立配置管理系统, 规范项目管理流程

建立配置管理系统 CVS, CVS 的全称是 CurrentVersionControl。在软件质量体系的诸多支持活动中, 配置管理系统处在支持活动的中心位置, 它有机地把其它支持活动结合起来, 形成一个整体, 相互促进, 相互影响, 有力地保证了质量体系的实施。建立公司配置管理系统很容易得到公司领导层的支持, 几乎没人反对。更重要的是建立配置管理系统后测试人员的工作有了系统保证, 测试工作的“矿藏资源”有了明确的位置, 可以主动积极开展测试工作。

4.3测试过程分阶段执行

将测试过程分成几个阶段执行, 即: 代码审查、单元测试、集成测试、确认测试和系统测试。

单元测试是针对软件设计的最小单位-模块进行正确性检验的测试工作，其目的在于发现各模块内部可能存在的各种差错。在单元测试之后，需要按照设计时做出的结构图，将它们联结起来，进行集成测试。是检验所开发的软件是否按用户要求运行。确认测试应检查软件能否按合同要求进行工作，即是否满足软件需求说明书中的确认标准。软件开发完成后，还要与系统中其他部分配套运行，进行系统测试，包括恢复测试、安全测试、强度测试和性能测试等。

4.4做好过程管理

过程管理须做好以下工作：分阶段设立里程碑，按里程碑计划工作和总结工作；加强审核，测试过程的中间结果要进行充分的审核；注重风险管理和规避风险，任何决定和过程都存在风险，尤其是质量好坏的风险，通过审核管理风险。

4.5制定成功的测试管理计划及测试计划

一个成功的测试开始于一个全面的测试管理计划。因此在每次测试之前应做好详细的测试管理计划：

首先，应该了解被测对象的基本信息，选择测试的标准级别，明确测试管理计划标识和测试管理项。在定义了被测对象的测试管理目标、范围后必须确定测试管理所使用的方法，即提供技术性的测试管理策略和测试管理过程。在测试管理计划中管理者应该全面了解被测对象的系统方法、语言特征、结构特点、操作方法和特殊需求等，以便确定必要的测试环境，包括测试硬件/软件及测试环境的建立等等。而且在测试管理计划中还应该制订一份详细的进度计划如：测试管理的开始段、中间段、结束段及测试管理过程每个部分的负责人等。由于任何一个软件不可能没有缺陷、系统运行时不出现故障，所以在测试管理计划中还必须考虑到一些意外情况、也就是说，当问题发生时应如何处理。因为测试管理具有一定难度，所以对测试管理者应进行必要的测试设计、工具、环境等的培训。最后，还必须确定认可和审议测试管理计划的负责人员。

还需要一个成功的测试计划，专业的测试必须以一个好的测试计划作为基础。尽管测试的每一个步骤都是独立的，但是必定要有一个起到框架结构作用的测试计划。测试的计划应该作为测试的起始步骤和重要环节。一个测试计划应包括：产品基本情况调研、测试需求说明、测试策略和记录、测试资源配置、计划表、问题跟踪报告、测试计划的评审、结果等等。

4.6测试人员及早介入

测试人员应从软件生命周期的需求阶段就开始介入，这样可以在这些需求基础上生成一份测试计划，并将测试用例对应于需求。这样便于提高测试用例的有效性和可用性，并且方便测试用例的设计和管理。

4.7测试文件的使用

在软件的需求分析阶段，就开始测试文件的编制工作，各种测试文件的编写应按一定的格式进行。测试文件的重要性表现在以下几个方面：

a、验证需求的正确性：测试文件中规定了用以验证软件需求的测试条件，研究这些测试条件对弄清用户需求的意图是十分有益的。

b、检验测试资源：测试计划不仅要用文件的形式把测试过程规定下来，还应说明测试工作必不可少的资源，进而检验这些资源是否可以得到，即它的可用性如何。如果某个测试计划已经编写出来，但

所需资源仍未落实，那就必须及早解决。

c、明确任务的风险：有了测试计划，就可以弄清楚测试可以做什么，不能做什么。了解测试任务的风险有助于对潜伏的可能出现的问题事先作好思想上和物质上的准备。

d、生成测试用例：测试用例的好坏决定着测试工作的效率，选择合适的测试用例是作好测试工作的关键。在测试文件编制过程中，按规定的要求精心设计测试用例有重要的意义。

e、评价测试结果：测试文件包括测试用例，即若干测试数据及对应的预期测试结果。完成测试后，将测试结果与预期的结果进行比较，便可对已进行的测试提出评价意见。

f、再测试：测试文件规定的和说明的内容对维护阶段由于各种原因的需求进行再测试时，是非常有用的。

g、决定测试的有效性：完成测试后，把测试结果写入文件，这对分析测试的有效性，甚至整个软件的可用性提供了依据。同时还可以证实有关方面的结论。

4.8测试团体的建设与测试人员的学习及培训

要注重建立一支高效的测试团队，不利用各种方法对测试人员进行培训。测试人员须加强自身学习，测试人员必须和程序员、用户、市场人员、技术作者以及任何的可能为实现更好测试提供线索的人进行交流。充分利用网络，及时地不断地学习，充分利用软件测试的技巧等。

4.9经济的测试

作为软件开发企业来说，投入人力，资金搞软件测试的最终目的还是离不开经济效益。而对与测试项目的管理也不能离开这个大前提。软件测试的经济效益主要来自以下两个方面。一是满足用户需求，提高产品的竞争力，最终提高产品的销售量。二是尽早发现缺陷，降低售后服务成本。而软件测试的最终目的就是使它带来的经济效益最大化。

有时候用户应用机率越大的那些软件业务功能，应该越少有差错；对某些关键功能，应该不允许有差错；经常不用的，甚至基本不用的那些软件业务功能，则可以容忍它有差错，甚至有较多的差错。

5、总结

软件测试工作在软件开发过程中占有重要地位，在各种规模的软件的开发中，测试工作都必不可少。关于软件测试的研究与技术探讨可以说都有成熟的成果，但是在实际应用中，由于软件开发的时间的影响或软件项目中人员的不足等等各种因素的影响，使得在国内的中小型软件企业中软件测试工作存在一些问题，本文对这些问题进行了总结和分析，并提出了一些解决方法。

从传统 ERP 测试向主流互联网测试转型

昨天晚上微博上有测试同仁 @testGM 想了解下如何从传统 ERP 测试向主流互联网测试转型。我本人在来淘宝之前就是做传统 ERP 测试的，现在在淘宝做了这么长时间，对互联网测试也有一定的了解，就想谈谈自己的看法。

特意说明下，我这边了解到的传统 ERP 测试就是4或5年前的 ERP 测试情况，不代表目前的整体情况，说的不对的地方 望谅解。

整体策略：

谈过如何转型，那就是知己知彼了。首先自己要了解传统 ERP 测试的优点和缺点以及特点，也要了解互联网测试的相关特点，从而给自己设定一个计划来快速转型，目的很简单，发挥在传统 ERP 测试的优势来弥补互联网测试的劣势，或者达到一种较好的平衡。

传统 ERP 测试 VS 互联网测试

(1) C/S 架构：来淘宝之前，我是在 MS 做传统 ERP 的测试，这个 ERP 是 MS 从国外收购过来的，主要给中小企业提供 ERP 服务，大家应该知道，那时候，ERP 是 C/S 架构的企业软件，现在很多 ERP 系统都是走 B/S 架构的。说实话，我不了解现在 ERP 系统的 B/S 架构和互联网产品的 B/S 架构的区别，所以这方面我就不继续说了。C/S 架构带来的是什么呢？很显然，就是版本升级的非实时性。我记得我们当时每次修复 bug 后，都是将修复后的代码文件交给技术支持，然后由技术支持来重新 import 到版本中。

(2) 流程严格：这个就不多说了，在传统流程中，把文档和 review 和前后接入准则做的都是非常好的，我在 MS 那边的时候，提交 bug、修复 bug、回归 bug 都是很严格要求的，包括验证 bug 的 TC，都是写的很详细的。还有就是 fix bug 的 old code/new code 的细节，反看互联网测试这些都是很少去做的。

(3) 业务逻辑复杂：记得在 MS 我们要学习很多关于库存管理、财务管理，报表管理等很多行业知识，一方面给测试的复杂度带来了挑战，另一方面对业务逻辑的沉淀就显得非常重要了，记得当时有很多这样的业务文档要学习，当然也有一些人不做开发或测试，去做 ERP 的实施了。反看互联网测试上层业务变化还是非常快，业务逻辑变化也快，而且也没有 ERP 那么复杂的业务逻辑，从而测试接触到的都是非行业核心的东西。

(4) 测试效率：前面几个特点也说明了传统 ERP 的测试效率应该很一般，的确是，包括 ERP 传统测试技术的发展，这个原因也是多方面的，我觉得很大一方面是开发架构和技术的变化不大，导致测试技术的变化也不大。大家可以看看互联网的开发技术变化多快，很多新技术和架构都是在互联网产品上应用起来的，同样的映射就是测试也必须在测试技术上提高从而跟上开发的步伐。包括接口自动化测试、分层自动化、持续集成、A/B Test 等等，很多测试技术在互联网上应用较多。很多测试同仁也认为互联网的测试技术很牛 B，一方面互联网测试必须找到很多较先进和流行的测试方法或技术或框架来提高效率。其实做传统 ERP 测试的人也是很牛 B 的，在国内的特点，就是太封闭了，不去外面交流来寻找新的方法和技术，国内估计很难找，那就去国外找啊，相信总能找到的。

其实对比这个还有一些特点：比如传统测试，职责明确；而互联网测试就是责任共享；传统测试就

是大步迈进，而互联网测试强调的是小步快走。

说了那么多，大家可能会问，我到底怎么转型啊，怎么更快的啊。

目标确定了，既然要做互联网测试，那就了解互联网产品测试的特点呗，有哪些核心的技术和框架呗。建议想转型的人，从下面几个角度考虑：

(1) 测试方法：大家可能会说，测试设计方法应该差别不多，是的，的确不大，但还是有差别的，建议各位看看互联网产品的公共功能的测试设计方法，我之前的 blog 也有提到。多了解下同样的边界值和组合测试法在互联网产品上是如何测试设计的。

(2) 自动化测试：一方面，把自己做 ERP 自动化测试的经验带过来，我记得我在 MS 做 ERP 自动化测试的时候，那是 MS 的测试自己开发的测试框架，很牛 X，我仔细的了解了下该框架的架构，发现 Client 的界面都是建模的等特点。后来到淘宝，做页面自动化时，就提出了类似的建议并一直使用至今。另一方面，就是学习分层自动化测试和持续集成，多学习下 java 语言以及互联网产品的开发架构，包括相关的基础技术，数据库和 Linux 命令啥的。一步一步来，先从页面自动化框架开始玩玩，从使用到熟悉到精通，搞起来，然后就是接口自动化，类似的过程。持续集成也是一样。

(3) 测试流程：需要了解互联网产品的测试流程，大部分是走敏捷的，虽然很多走的乱七八糟的，但是就是那个样子，流程乱，测试开发信息脱节，交互设计信息脱节，太多的问题存在了，回归测试和冒烟测试的重要性的严谨性等等都是大打折扣的。这些东西在传统里面就做的非常好，你可以把这些实践拿过来在互联网产品的测试上实践，虽然因为各种各样的原因，有很多难以做起来，但是不要忘记这些好的东西，慢慢的坚持，总会有进步的。

(4) 测试心态：这个大家可能觉得奇怪了，为啥要看测试心态。只是提前打个预防针而已，互联网测试 很枯燥，很乏味，你需要测试很多浏览器，兼容看看；你还会发现很多 bug，虽然很多是前端 bug，但是你发现和回归都很痛苦；你会不停的关注性能测试以及线上反馈；你还会担心你测试的页面是否有安全漏洞；你还会痛苦的发现我做了那么多，测试技术还这鸟样。不管怎么样，你还是要坚持下去，目前来看，就是互联网的测试技术发展较快，能把你的眼界打开，看到更多的彩虹。

我接触了很多做传统 ERP 测试的同仁，当然也有非 ERP 的测试，大部分人都是很仔细，很有耐心的，在互联网产品上继续保持这个，你肯定有前途，当然测试技术上也要跟上大家的步伐。

浅析缺陷责任期与保修期

近来做的比较多的工作就是拟合同，主要是工程合同，并且还都是涉及金额比较大的，因此对于这项工作我是很认真负责地去做。其中也遇到了很多问题，当然也学到了很多知识。让我学到最多的就是如何用词来准确地表达己方的意思同时又能不使的权利缺失。其实对于合同中的一些用词也是很讲究的，或许一词甚至一字之差会有很大区别的，当然涉及到的就是双方的利益问题。

我重点研究了有关缺陷责任期的问题，其实是通过与保修期的比较才对缺陷责任期有了更多的了解。

缺陷责任期，中文词条名：缺陷责任期；英文词条名：Flaw responsibility tenure in office.

缺陷责任期的概念源于2005年的《建设工程质量保证金管理暂行办法》。该办法规定如下：

“第二条本办法所称建设工程质量保证金（保修金）（以下简称保证金）是指发包人与承包人在建设工程承包合同中约定，从应付的工程款中预留，用以保证承包人在缺陷责任期内对建设工程出现的缺陷进行维修的资金。”

缺陷是指建设工程质量不符合工程建设强制性标准、设计文件，以及承包合同的约定。

缺陷责任期一般为六个月、十二个月或二十四个月，具体可由发、承包双方在合同中约定。

“第五条缺陷责任期从工程通过竣（交）工验收之日起计。由于承包人原因导致工程无法按规定期限进行竣（交）工验收的，缺陷责任期从实际通过竣（交）工验收之日起计。由于发包人原因导致工程无法按规定期限进行竣（交）工验收的，在承包人提交竣（交）工验收报告90天后，工程自动进入缺陷责任期。”

“第八条缺陷责任期内，由承包人原因造成的缺陷，承包人应负责维修，并承担鉴定及维修费用。如承包人不维修也不承担费用，发包人可按合同约定扣除保证金，并由承包人承担违约责任。承包人维修并承担相应费用后，不免除对工程的一般损失赔偿责任。”

在该办法中不仅对缺陷责任期的概念进行了阐述和界定，而且还对其期限做了描述。实质上就是指预留质保金的一个期限，具体期限由双方合同约定。

而保修期却是完全不同的一个概念了。对此2000年6月26日《房屋建筑工程质量保修办法》已有明确的规定。

“第七条在正常使用下，房屋建筑工程的最低保修期限为：

- (一)地基基础和主体结构工程，为设计文件规定的该工程的合理使用年限；
- (二)屋面防水工程、有防水要求的卫生间、房间和外墙面的防渗漏，为5年；
- (三)供热与供冷系统，为2个采暖期、供冷期；
- (四)电气系统、给排水管道、设备安装为2年；

(五)装修工程为2年。

其他项目的保修期限由建设单位和施工单位约定。

第八条房屋建筑工程保修期从工程竣工验收合格之日起计算。

第九条房屋建筑工程在保修期限内出现质量缺陷，建设单位或者房屋建筑所有人应当向施工单位发出保修通知。”

可见保修期指的是承包人对工程承担保修责任的一个期限。该期限虽然可由双方在合同中约定，但不能违背以上第七条的规定，有一定的法律强制性。

通过上文对缺陷责任期的比较，从本质上来说其区别是很明显的，前者是发包方预留质保金的一个期限，而保修期是承包人对工程承担保修责任的一个期限。不过从一个角度，即利益，来考虑，如果缺陷发生在缺陷责任期内，对于承包人没有进行维修的行为，发包方是可以从质保金中扣除的；而如果缺陷发生在保修期内缺陷责任期外，那么对于承包商不维修的行为或许只能通过法律途径解决了。

通常来讲保修期是长于缺陷责任期的，特别是对于大型工程。而我们在日常的一些小项目工程中，常常将缺陷责任期等同于保修期，而且在合同的拟定中混用，或许不会造成大的影响，不过这对我们进行大的工程来说可能会造成很大损失的。当然对于一些小的工程，也可以将缺陷责任期的期限等于保修期的期限，不过在用词的时候应该准确，至少作为拟定合同的己方应该明白其代表的含义。

两种常见的缺陷责任期条款为：承包商须修理、矫正及改正（repair, rectify and make good）任何在缺陷责任期内出现的缺陷；或承包商须在缺陷责任期内保养（maintain）有关工程。

第一种缺陷责任期条款是规定承包商有责任改正在有关期间内出现的缺陷。任何在这期间届满后出现的缺陷都不会在这种条款范围内，而且承包商的责任并不包括改正任何工程上出现的意外损害。而第二种缺陷责任期条款是规定承包商有责任维护工程，在缺陷责任期内使其保持在缺陷责任期开始时的状况。所以一般而言，第二种条款所规定的责任会比单单改正缺陷的为大。不过这种条款通常都会连带给予承包商权利，以获得进行任何并非因为承包商出错所引致的缺陷改正工程费用。

如何少走弯路的 10 条忠告

1、买个闹钟，以便按时叫醒你

贪睡和不守时，都将成为你工作和事业上的绊脚石，任何时候都一样。不仅要学会准时，更要学会提前。

就如你坐车去某地，沿途的风景很美，你忍不下车看一看，后来虽然你还是赶到了某地，却不是准时到达。闹钟只是一种简单的标志和提示，真正灵活、实用的时间，掌握在每个人的心中。

2、如果你不喜欢现在的工作，要么辞职不干，要么就闭嘴不言

初出茅庐，往往眼高手低，心高气傲，大事做不了，小事不愿做。

不要养成挑三拣四的习惯。不要雨天烦打伞，不带伞又怕淋雨，处处表现出不满的情绪。记住，不做则已，要做就要做好。

3、每个人都有孤独的时候

要学会忍受孤独，这样才会成熟起来。

年轻人嘻嘻哈哈、打打闹闹惯了，到了一个陌生的环境，面对形形色色的任何事，一下子不知所措起来，有时连一个可以倾心说话的地方也没有。这时，千万别浮躁，学会静心，学会忍受孤独。在孤独中思考，在思考中成熟，在成熟中升华。不要因为寂寞而乱了方寸，而去做无聊无益的事情，白白浪费了宝贵的时间。

4、走运时要做好倒霉的准备

有一天，一只狐狸走到一个葡萄园外，看见里面水灵灵的葡萄垂涎欲滴。可会死外面有栅栏挡着，无法进去。于是它一狠心绝食三日，减肥之后，终于钻进葡萄园内饱餐一顿。当它心满意足想离开葡萄园时，发觉自己吃得太饱，怎么也钻不出栅栏了。

相信任何人都不愿做这样的狐狸。

退路同样重要。饱带干粮，晴带雨伞，点滴积累，水到渠成。有的东西今天似乎一文不值，但有朝一日也许就会身价百倍。

5、不要像玻璃那样脆弱

有的人眼睛总盯着自己，所以长不高看不远;总是喜欢怨天尤人，也使别人无比厌烦。没有苦中苦，哪来甜中甜?不要像玻璃那样脆弱，而应像水晶一样透明，太阳一样辉煌，腊梅一样坚强。既然睁开眼睛享受风的清凉，就不要埋怨风中细小的沙粒。

6、管住自己的嘴巴

不要谈论自己，更不要议论别人。谈论自己往往会自大虚伪，在名不副实中失去自己。议论别人

往往陷入鸡毛蒜皮的是非口舌中纠缠不清。

每天下班后和你的那些同事朋友喝酒聊天可不是件好事，因为，这中间往往会把议论同事、朋友当做话题。背后议论别人总是不好的，尤其是议论别人的短处，这些会降低你的人格。

7、机会从不会“失掉”，你失掉了，自有别人会得到

不要凡事在天，守株待兔，更不要寄希望于“机会”。机会只不过是相对于充分准备而有善于创造机会的人而言的。

也许，你正为失去一个机会而懊悔、埋怨的时候，机会正被你对面那个同样的“倒霉鬼”给抓住了。没有机会，就要创造机会，有了机会，就要巧妙地抓住。

8、若电话老是不响，你该打出去

很多时候，电话会给你带来意想不到的收获，它不是花瓶仅仅成为一种摆设。交了新朋友，别忘了老朋友，朋友多了路好走。交际的一大诀窍就是主动。好的人缘好的口碑，往往助你的事业更上一个台阶。

9、千万不要因为自己已经到了结婚年龄而草率结婚

想结婚，就要找一个能和你心心相印、相辅相携的伴侣。不要因为放纵和游戏而恋爱，不要因为恋爱而影响工作和事业，更不要因一桩草率而失败的婚姻而使人生受阻。感情用事往往会因小失大。

10、写出你一生要做的事情，把单子放在皮夹里，经常拿出来看看

人生要有目标，要有计划，要有提醒，要有紧迫感。一个又一个小目标串起来，就成了你一生的大目标。生活富足了，环境改善了，不要忘了皮夹里那张看似薄薄的单子。

由于博客空间在国外，所以有时会造成不稳定(你懂滴)，如果您觉得我的博文对您有帮助，建议大家多使用 RSS 访问阅读，本站所有文章均已全文输出。


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

