

我从可用性测试中学到的五件事

如何给一个私有方法做单元测试

从测试员到测试负责人

怎样才能做好技术团队管理

Android应用测试总结

如何对待测试开发团队中那个拖后腿的人？

如何领导员工成为项目管理者

IT经理，你可以管理多少人

我是如何打败拖延症的

使用缺陷管理工具不应有的坏习惯

上海泽众软件电子期刊
2014年6月 第二十八期

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：xudan@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

www.spasvo.com（泽众软件测试网）

目录

我从可用性测试中学到的五件事.....	4
如何给一个私有方法做单元测试.....	6
从测试员到测试负责人.....	7
怎样才能做好技术团队管理.....	9
Android 应用测试总结.....	10
如何对待测试开发团队中那个拖后腿的人?	11
如何领导员工成为项目管理者.....	13
IT 经理, 你可以管理多少人.....	17
我是如何打败拖延症的.....	19
使用缺陷管理工具不应有的坏习惯.....	20

我从可用性测试中学到的五件事

我喜欢做可用性测试。

没有比把假设放在用户面前来测试更有效的方法了。你不仅可以在开发环境之外看你的工作，还可以从用户那里得到很多创新的想法，因为他们每天都在用这个系统。

这件事你必须尽快安排，但是让人吃惊的是很多开发者并没有这样做。他们应该少花点时间开发，多花点时间和用户交流。也就是说，他们更应该走出去。

我还学会了如何获取更有效的反馈。如果专注于特定的模式，那么你可以提高自己发现隐藏观点的能力。

最好在环境上下文中进行测试

我第一次参与的几个可用性测试就是市场人员喜欢做的：主持人坐一边，另外五六个人坐在另一边。而我们就应该坐在另一边做记录，在观察用户表现的同时还要经常和他们开玩笑互动。

观察用户的表现很有趣。但有时完全没用。所有的可用性测试都有偏差，更严重的是，即使你给用户付了钱，你就能保证他们的表现是真实的？所以，你的目标是要尽可能限制这个偏差。在这个过程中我们可以学到很多东西，这就是为什么我们会继续测试，无论有没有偏差。

最好的方法是模拟用户使用应用程序时的环境。比如，如果用户坐着，我不会对其做移动地图应用的测试，因为用户一般都是在走路寻找某个地方时才使用地图。我已经对手持电话的用户做过了测试，测试指标是“他们找到目的地了吗？”

对于其它应用，我会在办公桌上和用户做可用性测试。这样可以更好地理解用户日常生活中使用应用的环境。

我和客户最近做了一个测试，用户日常生活中必须与 3-4 个应用进行交互。我可以从中了解用户使用的其它应用（比如 Salesforce、Excel、Outlook 以及其它六个浏览器窗口）。我一边看用户与软件的交互，一边做笔记，并列出了与应用相关的工作流程建议表，而不是我最初设计的那样。

看他们做什么而不是说什么

很多人都喜欢帮助别人，所以他们会积极地面对你摆在他们面前的问题，这种情况被称为社会赞许（Social Desirability）。我进行过几次可用性测试，用户都表达他们如何喜欢这款应用。他们会说：“我喜欢它！我完全会使用它。”

很明显，他们并不知道怎么使用这个应用。

当你称其为“可用性测试”时，终端用户可能会简单地认为是对知识的测试，而不是 app 的可用性测试。他们不想别人认为他们很愚蠢，所以他们会说他们有多了解如何使用网站或他们很专注这个工作，这样他们就不会犯错。很多人都喜欢帮忙，所以他们对摆在他们面前的问题非常积极。仅凭这一点，

我可能就有一系列内容可以写。

看他们做的，以及他们如何对屏幕上的应用做出反应，并试着和他们的评论联系起来。有时，我记录鼠标的运动就可以精确地知道用户在干什么，但是我通常只会粗略的记录一下交互方面。

让用户说话比你想象的要容易

在 Jobvite 工作时，我采访过很多招聘专员和招聘经理。没有比从招聘专员那里更容易得到反馈的了。

你问他们三个问题，他们就会告诉你所有事情。他们中午吃的什么，采访了谁，喜不喜欢他们的工作等等。更重要的是，如果引导他们，他们会告诉你如何使用技术。招聘专员每天会说很多，但是没有人会认真听他们的话。他们一旦有机会，会很乐意谈。

他们和像大多数人一样，希望有人倾听，因为我们是一个开放的社会。作为设计师和开发者，我们给他们对话的机会，也给我们一个获取真实意见的机会。这不仅仅因为他们使用你测试的技术，他们的环境也适合。

你需要做的就是问为什么？

这些反馈对设计出伟大的产品非常重要，因为你了解用户的痛点，你可以解决他们的问题。

最好的想法来源于脱稿流程

我看到过这种可用性测试：主持人列举了一系列问题，做笔记，轻易获取毫无价值的反馈，甚至可能都不知道谈了些什么。

我也见过这种可用性测试：主持人只列举了几个简短的问题，让用户按照几个重要的观点来谈，顺着产品变化的反馈来走。

最好的情况是有一个宽松的剧本，这样你可以从自己的角度进行思考。如果你学会如何做，那么测试就会有更多偶然的问题解决办法。这时真正价值的内容就出现了，因为你可以深入探讨一个话题，并获取用户的真实想法。

不需要用一个完整的原型来测试概念

我已经记不清有多少次我使用半成品原型来测试想法了。在进行测试前，我会告诉用户这是测试试验中的想法。我也会告诉他们这是个原型，所以有些东西可能不凑效。由于原型通常是高度逼真的，以致用户会忘了这是测试。

我给他们发送一个链接。他们一点击就会崩溃。

为什么会崩溃？

因为这是一个原型，但它并不完整。我们正在尝试其它方法。所以，当你点击这个链接，你认为应该发生什么情况？你能一步一步地解释出来吗？

大多数情况下，他们会描述下一步会怎么做。当然，如果刚好在用户描述每个步骤的时候，你能确定他们的真实想法，那就很酷了。更好的是，他们可能会给你提供一个你从来没想到过的想法。

可用性测试并不仅仅是测试你当前的设计，它可以利用用户的集体体验来改善它。这些想法不仅能验证你现在所做的工作，并且还可能进入你的产品线路中。

你向用户学习了吗？

抓紧时间。

多问为什么。

倾听。

这并不难，但是你需要坚持每天做，无论什么时候。你所学到的可能出乎你的意料。

如何给一个私有方法做单元测试

给类的公有方法做单元测试，很简单，new 一个对象，设好入口参数后，调用这个方法，比较期望值和实际值即可。给一个类的私有方法做单元测试，也不麻烦，先通过反射获取这个方法，然后将这个方法的可访问性强制设为 true，这样的话，这个私有方法就可以被调用了。

代码如下：

```
//被测试方法
private Double format(Double fileSize){
    Double size = fileSize;
    size = size / 1024 / 1024;
    size = (int)(size.doubleValue() * 10 + 0.5) / 10.0;
    return size;
}

@Test
public void testFormat() {
    ClientDownloadAction action = new ClientDownloadAction();
    double size = 3732930;
    Class class1 = action.getClass();
    try {
```

```
Method format = class1.getDeclaredMethod("format", Double.class);
format.setAccessible(true); // 设为可见
Double result = (Double)format.invoke(action, size);
Double expect = 3.6;
Assert.assertEquals(expect, result);
} catch (Exception e) {
// TODO Auto-generated catch block
e.printStackTrace();
}
}
```

从测试员到测试负责人

从测试员到测试负责人的本质改变是开始承担管理责任，测试负责人作为组织中的最基层管理者，除了执行相关能力的继续提升外，需要开始担任部分管理职能。从一个执行者开始转变为一个管理者，主要的变化有以下几点：

1：责任范围的变化

纯粹的执行者原则上只需要为自己的执行工作负责即可；而管理者需要对自己管理范围内的所有工作负责，即使不是自己执行的工作，也要负管理责任。

对于执行者，我们会希望他们有超出自己职责范围的责任心，这会有助于其个人能力的发展和进步，也会提升部门整体的工作效率和绩效，但这并非职责要求必须达到的，所以在评价员工的时候，会有不合格、合格、超出预期几个档次。

对于管理者，对责任心的要求则成为基本要求。通常也只有获得超出预期评价的员工才可能会被提拔为管理者。

2：责任范围的变化直接带来具体工作方式的变化

作为基层管理者，实际工作中的执行工作仍然占了相当大的比例，但新增的管理工作内容虽然比例相对比较小，但却是评价一个管理者的重要部分。

首先是内部工作分配，要做好内部工作分配有以下几点需要注意：

A：首先是发现工作，管理者要自主发现工作内容，而不是等待上级分配；

B：然后是对下属的了解，充分了解下属才能把合适的工作分配给合适的人；

C：维持部门工作士气，限制负面情绪和负能量的产生和传播，创造和传播正面情绪和正能量。

D：工作成果的验收和检验，考验管理者对自己部门的掌控力，也是部门绩效的最终成果。

其次是外部协调，为本部门的工作开展创造一个良好的外部环境：

A：调节自身测试部门和其他部门的关系

B：调解其他部门之间的关系

C：注意以上两条的最终原则：团队的稳定和项目的成功

然后是协助上级，完成上级指派的任务：

基层管理者除了自身的管理者身份，同时很大程度上也是被管理者，需要完成上级指派的各种任务，和纯粹的执行者的差别只是可以利用自己管理范围内的资源去完成上级指派的任务，而不一定必须亲自完成。

包括但不限于，例如：项目间的测试资源调配帮忙。

3：必要的工作技巧和方法：

1：在部门内部建立威信

A：展示自身工作能力

B：展示自身职业素养和专业素养

C：帮助下属提升

D：任何时候保持沉稳、冷静；主管的心乱了，整个部门就乱了。

2：获得项目组其他部门特别是部门主管的认可

A：展示自身的责任心

B：展示自身工作能力

C：展示自身职业素养和专业素养

D：帮助其他部门解决问题

3：眼界的开阔

A：关注行业和专业动态

B：把 A 转化为对项目的实质性帮助

怎样才能做好技术团队管理

经常看到有人问程序员适合做管理吗，或者手底下有牛人比我技术更好怎么办，或者感叹一下做管理好难啊之类的。同时，相当大的一部分程序员都梦想着走所谓专家路线——并不是因为对技术特别有兴趣或者觉得自己特别适合走技术路线，真正的原因是对管理工作的恐惧，觉得自己搞不定。做管理真的很难吗，程序员出身到底适不适合做管理，我可以斩钉截铁的告诉你：不难!适合!

上面的答案显然并不完全正确。不过我们今天讨论的并不是管理一个国家那样的管理，也不是管理一个公司或者半个公司这样的管理——绝大部分程序员同志们短时间内都不会有这样的机会，这样

的话题也完全超出了我的知识范畴。我们今天讨论的只是基础的简单的管理，小到几个人的小组大到十几二十个人的团队，再大的都不在讨论之列，而且仅限于软件行业。所有的爷爷都是从孙子走过来的，做管理也一样都是从小小管理一点一点慢慢做大的。

第一个话题：团队里有牛人怎么办

这个问题问的很多，但是实际上管理一个团队更容易碰到的也是更头疼的问题是“团队里没有牛人怎么办”，所以用流行的话说要“怀着感恩的心”看待这个问题。有牛人意味着你可以在一定程度上脱离繁重的开发或者设计工作，把更多的时间放在做好管理和决策等“清闲”的工作上，意味着你有精兵强将可以完成更有挑战性的项目，意味着你的团队可能创造更多的效益使你的管理工作“看起来更出色”，等等。这都是有牛人的好处。

但是，但凡牛人多少可能有点牛脾气，不好管。但是这个不好管究竟多大程度上是因为牛人的问题，又有多大程度是因为管理者的问题是必须要搞清楚。很多人觉得牛人不好管可能只是心理上的，是因为不够自信，虽然身为管理者在牛人面前却觉得矮半截唯唯诺诺，或者是由不自信演变成自负在自己擅长的领域刁难、打击报复等等，反复几次以后管理者的权威荡然无存，管理工作也就永远做不好了。

管理牛人与管理普通员工并无太大的区别，只是要更慎重更懂得平衡和技巧。因为牛人通常在团队里的影响力比较大，做好牛人的工作管理就已经成功大半了，以下是一些要点，其他的自己任意添加：

最重要的一点是要保持自信，既然能做到这个位置必然有自己的过人之处，找到它们并充分发挥；

倚重但不依赖牛人，并且让牛人自己也知道这一点；

注意培养新人，只有一个牛人并且有野心才是最危险的事，自己带起来的兵最可靠；

第二个话题：道与术

很多讲人讲管理喜欢扯到“道”与“术”上去，特别是半吊子的管理者更喜欢讲“道”，似乎“道”总是比“术”高一个层次。例如很多人看到我们上一个话题的讨论就会跳出来大叫，你只讲到了“术”的层次，还要从“道”的层次考虑；还有人会说，这样是玩弄权术，不行，做管理要靠“心”，以心换心。

讲“以心换心”的同学未免太过天真了，在社会上混几年的人都应该知道这一条并不总是成立的，哪怕我们宁愿相信它成立。做管理当然要待人以诚、以心换心，但是这并不构成做好管理的充分条件，一个在战略方向上总是举棋不定的领导哪怕人再好也会导致下属丧失安全感从而不愿意追随。

道与术的关系本质上就是一个指导思想与具体手段的关系，是心法与招式的关系，固然脱离了心法的招式容易走向混乱，但也从来没有一种心法能够凌驾于所有的招式之上并且脱离招式而独立存在。所以，所谓的“道”并不是很高深很神秘的东西，它只是给我们的管理工作提供一个总体的方向，所有的具体工作都是围绕着这个方向进行。我自己总结出来的管理好一个团队必须要做好的勉强能称之为“道”的几个方面如下：

保持团队的方向感，让团队成员知道自己在做什么将来又要做什么，能感觉到自己的价值所在；

保持团队的进步感，让团队成员感觉到自己每隔一段时间都能学到新的东西，值得为之付出的努力；保证团队成员的归属感和自豪感，这样的团队才有凝聚力。

第三个话题：无为而治是很扯淡的事

《英雄》中的始皇帝说“用剑的最高境界是不杀”，这句话直接导致了无名丢掉了自己的剑。有人喜欢把这话套用到管理上——“管理的最高境界是不管”，我想说的是千万不要被这样的话忽悠了，也千万不要拿这样的话忽悠自己。

Android 应用测试总结

Android 应用程序测试注意要点：

- 1、应用程序功能是否都实现
- 2、给用户的提示信息是否简单、易于理解，且符合当前事件
- 3、交互性事件的影响：
 - (1) 硬件：USB 插拔、电池没电及插拔；
 - (2) 软件：电话来电、短信、QQ，音乐盒等后台运行的软件。
- 4、网络影响：wifi、2G/3G/4G
- 5、非法操作对应用程序的影响

下面是针对容易忽视测试路径的功能测试模块的测试用例做一个小结：

一、登录

- 1、wifi 对登录的影响
- 2、数字、字母、特殊字符、汉字在登录输入框中的响应
- 3、被删除的用户能否登录成功
- 4、被禁止的用户能否登录成功
- 5、不存在的用户能否登录成功

二、下载

- 1、wifi 正常，下载资源期间，账号退出/切换的影响
- 2、wifi 正常，下载资源期间，暂停、取消的使用
- 3、wifi 正常，下载选择的切换
- 4、wifi 正常，账号 A 的资源下载选择后台下载，账号 B 登录共同下载相同的资源，是否发生异常
- 5、wifi 正常，后台有资源正在下载（不同账号或同一个账号），使用“一键清理”是否能清除完全
- 6、资源正在下载过程中，wifi 断开，下载给出的响应，连接 wifi 之后能否继续下载当前资源
- 7、资源正在下载过程中，切换 wifi 账号，对下载的影响

8、wifi 正常，弹出下载对话框，断开 wifi，点击下载，应用给出的响应，以及连接 wifi 之后，是否能下载成功。

9、wifi 正常，下载队列中的资源下载完成之后是否自动继续等待队列中资源的下载

10、下载数据能否查看，且内容是否完整

11、删除本地缓存的下载文件，再次点击下载，是否能下载成功

12、手机内存已满时，下载开始或中途是否有相应的提示信息

13、下载过程中没电关机、USB 的插拔，突然关机对下载的影响

三、调查问卷

1、问卷内容的布局、内容显示是否符合要求

2、手机 HOME 键、返回键、MENU 键的响应

3、暂存、交卷多次点击的响应

4、wifi 对暂存和交卷的影响，缓存时间的长短

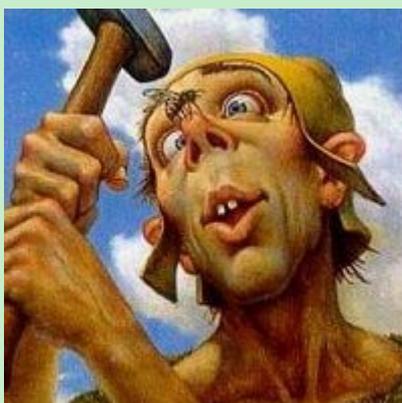
如何对待测试开发团队中那个拖后腿的人？

每个开发团队中总有一个人是最差的，老跟不上整体水平。据我观察，队友们对待这个差生的态度是团队健康状况的重要标志。（注：原文用“贝尔曲线（bell curve）”，也就是“正态分布”来衡量团队的整体水平分布，这里意译成“整体水平”。）

我运气一直不错，在过去的几十年里，干过各种的行业和职业，在不少团队中待过，都是气氛融洽和技能娴熟的开发团队。并不是说每个项目都是成功的，（外部因素无法控制），但是经验永远有深刻的价值。

在成功的开发团队里，最显著的特点是队员们互相尊重——“他擅长做这个，也喜欢做这个。好极了。因为我不喜欢做这个”）。每个人都明白人多力量大的道理，并懂得珍惜队友的技能。如果你在团队中待过（或者你曾管理过这样的一群家伙），你懂我的意思。

不论这些黄金组合团队是网络社区、公司部门还是志愿者组织，都有个共同特点：每个团队中都有这样的一个人——我们可以无恶意的称之为笨蛋，但重点是，大家却对这个人很好。



在我待过的一个志愿软件开发团队中，Elliot 就是那个笨蛋。准确形容的话，这家伙是个热心但没脑子的人。像一只大小狗一样，他是个热心肠，但往往好心办坏事。他乐于助人，而且很羡慕别人做得比他好。（实际上所有人都做得比他好。）

Elliot 总把事情办砸，哪怕他已非常的认真和努力。没人会把关乎项目成败的关键任务交给他办。他也从未意识到自己交付的活有问题，比方说我总是要帮他重写文档，而发布后的文档与他的大意一致，以至于他根本都没有意识到被人重写了。Elliot 常常对大伙做的事摸不着头脑，但他依然兴高采烈地跟在那些他认为知道要做什么以及怎样去做的人的屁股后面。

但每个人都体谅他。运作志愿者团队的人想方设法地帮助他，让他能真正贡献自己的一分力量，并把他的高昂的工作热情和乐于奉献的精神看做他的优点。就算是搞社区活动也需要人手去摆放折叠椅或者在前台迎宾吧。我们把所有类似的活都交给他干。

他是有点笨，但他是我们的笨笨。我们私下里也会抱怨，大家翻白眼，叹气或埋怨：“没办法，Elliot 就是这样的。”可是如果有外人胆敢欺负他，我们会扭折他的脖子。我想说的是，Elliot 几年前去世了。组织里所有认识他的人都参加了他的葬礼。生活并没有像我们那般关怀他。对于我们每一个人，他都是值得怀念的。

自从我认识 Elliot 以后，我开始观察这种现象。每个开发团队，不管多么优秀或者蹩脚，总有个在垫底的（不论是从智商，工作效率或其他的方面看）。团队的整体水平根据行业的种类而定 - 衡量一组科学家的工作价值和效率的标准跟衡量一组园林景观设计师完全不同。

你们团队里垫底的那个 Elliot，换到另外一个场合，也许是个高手。但总是会有那么一个垫底的人。

在不和谐的团队里，这个笨笨会被当成白痴。人们欺负他，瞧不起他，对他不友好，你能猜想到这会多少不良影响。但对于懂得相互理解体谅能力差距的团队而言，情况就不一样了。

这种积极的特性在重于参与的团队中尤为明显，比如用户群，开源项目组，社区组织。这些情况下，你不可能因为某个成员工作质量有所欠缺而开除他，所以你 必须积极地跟那些志愿者们打交道。商业组织也要以此为鉴，我们都看到过组织开除某个过能力稍微欠缺的人而不是帮助他们找到能够胜任的工作岗位。

问题是，你 - 作为项目经理 - 是如何对待你手下的这个笨笨的？你的团队又是如何对待她的？

我很想能为你提供一些以“怎样才能…”打头的建议…但实际上这些都没用。不是简单地知道要做什么就行了，而要考虑到你团队的文化氛围。团队和领导 如何对待后进员工代表了团队的文化，标志了团队的和谐度。如果你对他们好，他们会报答你的，结果会在最终的产品里体现出来。

如果你知道关于团队如何对待最差成员的例子，对公司文化的团队的功能产生了怎样的影响，不管是好是坏，请告诉我。

如何领导员工成为项目管理者

如何更好地激发关键员工的积极性和创造性，是管理者必须研究的课题。通过笔者的观察，把员工的工作丰富化，把一些重要的工作作为项目开展，对于激励员工是个不错的办法。

项目管理作为一个新管理门类正在越来越多的受到推崇。项目管理不同于任务管理，相比较，它更能锻炼员工的技能，激发员工的进取心和潜能，更利于管理决策的实现和员工的成长。目前，能否独立承担一个项目并把它做好已成为衡量一个人能力水平高低的重要标志，同时，能否带领员工做更多的项目，能否通过做项目 来获得更大收益也已经成为衡量一个经理管理水平高低的重要指标。

汤姆·彼得斯强调说，“现代管理，项目就是一切，每个人都是一个项目管理者”。对团队领导者来说，这句话值得认真玩味和思考。

一个高效能的经理不能仅仅满足于把员工管好，把事情做正确，更应激励员工多做事情，正确地做事，让员工富有工作热情和创造力，让员工享受工作的成就感，在工作中超越和实现自我。为此，团队领导者应更多考虑领导员工成为项目管理者，致力于打造高绩效的工作团队。

领导员工成为项目管理者就是要求经理在更高层次上对员工进行管理，在团队管理中，经理应更多地扮演领导者的角色，领导员工做项目，让员工学会用项目管理自己，学会对项目负责，在项目的定立和实现的过程中不断超越自我，在压力下成长进步。

一、用项目管理员工的好处

1、有效激励员工

给员工分配一个项目比简单地告诉员工把工作做好更为有效。经验表明，用项目管理员工可以激发员工的潜能，让员工在压力下获得高绩效。

在项目面前，员工更愿意主动地工作，积极地寻求领导的支持和资源配备。一个项目有严格的验收标准和完工日期，这无形中就给了员工一种压力，这种压力不是来自经理的大声恐吓而是来自项目的本身。当员工获得了关于项目的全部信息以后，他就会主动给自己一个压力，他会提醒自己要把工作做好，要通过高效完成项目使自己的能力和价值得到提升，这种自动自发的责任感比经理天天叮在员工的耳边要强的多。

员工更愿意把项目当成自己的事情，做项目的主人，因为这样他们可以体验到驾御工作的感觉，可

以在项目实现的过程中获得更多的自我满足感和超越感。员工会希望通过努力工作获得经理的赏识和认可，当然也希望因此获得更多的奖励和晋升的机会。

2、有利于建立经理和员工之间合作伙伴关系。

在项目管理下，经理更像是一个辅导员和教练，经理的主要职责是作为员工的合作伙伴帮助员工规划工作，与员工一起合作开发项目，帮助员工预见可能的变化和困难，使员工在项目中加强学习与锻炼，为员工协调人员与资源，清除工作中的障碍，以员工的高级合人的方式实施管理，着手工作。

在这种工作关系下，经理所要做的工作更多地是领导员工，而非简单的管理，这是更高层次的工作方式，持续强化这种工作方式将有助于经理更加高效地工作，有助于经理自己的职业规划和发展的。

3、有利于创建团队合作的关系

项目不是一个人的事情，通常需要一个密切合作的团队来完成。用项目管理员工，就是让员工既可能是自己项目的队长，又可能是别人项目的队员，通过这种区别渗透的管理方式，使管理向扁平化发展，建立和发展团队的合作关系，促使员工非常规地成长。

4、有利于对员工进行在职培训（OJT）

从项目的设立到项目的完成，经理都和员工保持着持续的沟通与联系，经理可以根据员工的工作情况进行针对性的辅导，员工也经常就工作中的困难与经理探讨，在这个过程中，经理和员工的绩效能力都能获得提高。

实际上，这就是一种在职培训（OJT），通过关注项目的实现来关注员工的成长，而这种培训的效果要比脱产培训更佳。

二、项目管理的受益者

马先生就是这样一个项目管理的受益者。目前，他是一家中型高科技企业的信息部经理，手中握有思科、微软几家权威IT认证机构的证书。

三年前跳槽来到这家企业的时候，他只是一个企业信息化的爱好者，刚刚入门。他的汽车专业根本与信息化不搭边，四年的设计工作经历也与信息化毫无关系。

但是，马先生对信息化有着高涨的热情和特别的爱好，业余时间自己读了一些信息化方面的书籍，工作之余留心对以前所在企业的信息化工作做了一定的研究。

凭借这些基础的技能和高超的沟通技巧，马先生说服了招聘考官，尽管考官认为马先生比较适合与客户沟通，做市场开发工作将更有利于他的职业发展，但还是同意马先生来公司试试信息化建设的工作。

刚到公司的时候，负责企业信息化建设的经理只给了马先生一份简单的岗位描述，让马先生尽快熟悉公司的情况，熟悉部门的业务。

过了不到三个月，经理把马先生叫到了办公室，表达了他对马先生的期望，他希望马先生能通过自己的努力给公司的信息化建设做些贡献。他给了马先生一份公司信息化建设的规划，里面列了三个项目：

局域网建设、因特网建设、ERP 系统的建设。他告诉马先生，这就是他未来三年的工作目标，这三个项目将由马先生负责主持完成。

身负重托的马先生深感责任重大，同时他也窃喜自己终于有机会在 IT 领域习好好练一把，他告诫自己不要让机会从手中流走。马先生深知自己在企业信息化方面的知识、技能和经验不足，与面前的三个项目相比，自己未来还有很长的时间需要学习充电，于是在接受项目的同时他也给自己列出了未来的学习计划。

之后的两年多的时间里，马先生一门心思地投入到项目当中，劲头十足地学习工作、工作学习，逐步地进入了工作状态。当然他为此也付出了很多，当别人都已经下班了两个小时他还在电脑前紧张地工作，当别人节假日休闲度假的时候他却依然呆在机房查找数据。

在经理的支持和帮助下，他工作的思路越来越清晰，项目的进度也按计划地逐步进行。这三年，他不仅仅做了这三项工作，他还不断地为公司的文化建设提一些建设性的意见，为员工提供微机操作的培训，为公司配备微机，购买办公用品等。

随着工作的开展，马先生的工作思路越来越开阔，越来越受到公司领导的赏识和认可，到去年年底，分管他的经理因个人原因（升任另一家公司的总经理）辞职，他顺理成章地被提升为企业信息部的经理。最近公司高层正有意把公司知识管理的任务交给他，他也正在考虑将信息化和知识管理结合的可能性，顺利的话，他将成为公司的信息化和知识管理的总监。

现在，他不仅工作上小有成就，而且对未来自己要从事的事业已经有了一个逐渐清晰的蓝图和规划，也许不久的将来，他就要开始他自己的事业了。

马先生的经历很好地说明了项目管理在团队管理中的作用。用项目管理员工不仅能实现经理对未来的设想和规划，更能激发员工潜能，使员工持续得到激励，不断成长，提高员工的自我管理水平和提高绩效能力。

三、如何领导员工成为项目管理者？

领导员工成为项目管理者主要有以下几点需要考虑：

1、职位分析，确立团队成员的职位说明书

职位说明书是一切管理的开始，用项目管理员工不代表可以撇开职位说明书不管，相反，经理更应该多花时间研究员工的职位说明书，更加清楚地描述员工的职位，以此作为与员工订立项目的基础性文件，为项目提供可操作的信息，保证项目的有效性。

2、与员工分享公司的远景和战略目标

工作团队的所有工作都是为公司的整体战略目标服务的，工作团队的每一步行动都要与公司的决策相一致，所以在给员工订立项目之初，经理必须和员工充分分享公司的战略和目标任务，使团队的任务与公司的目标相匹配，使工作项目都能产生实效。

3、与员工分享工作团队的规划

分享公司的远景的同时，经理也应把工作团队的发展规划分享给员工，以便于员工更加明确未来的工作方向，配合经理做好团队的工作。

4、规划团队成员的职业发展

任何的远景与目标都必须落实到员工，如果员工对自己的职业规划有非常清晰的认识，他们将更加有干劲，更加积极主动，他们愿意为了实现自己的职业发展与团队共同进步，与企业共谋发展，所以经理必须给员工的职位发展一个清晰的定位，让员工看到最近和未来一段时间的发展目标。

5、与团队成员一起定立项目

在前面工作的基础上，经理应着手与员工就项目的创建进行沟通并达成一致。

项目应该是在经理和员工双方充分沟通的基础上达成的绩效合约，而不是经理单独对员工做指示。领导员工成为项目管理者应该强调员工的主动性，把员工作为项目的主人，更多地倾听员工的想法。充分沟通应该是定立项目的一项重要原则，没有充分沟通的项目势必影响员工对项目的理解，影响员工完成项目的效果。

所以，所有有关项目的信息都必须进行沟通，在取得员工的认可之后再行定稿，经理和员工双方都应在项目书上签字，以示重视。

6、在项目的实施过程中对员工进行辅导

项目定立完成，经理的工作才刚刚开始，还有更多的工作在后面。经理必须在项目实施过程对员工进行辅导，为员工提供人力、物力、财力的支持，帮助员工清除工作中的障碍，使他们更加高效地工作，获得更多的在职培训机会，不断提高绩效能力和业绩水平。

6、及时对员工的表现进行反馈

反馈是激励员工更努力工作的行之有效的手段，反馈得越及时，员工工作的积极性就会越高，反馈得越具体，员工的工作就会做得越好。

反馈包括正面反馈和负面反馈。

经理应时刻关注员工项目的进展，及时真诚地对员工在实施项目的进程中所表现出来的正面行为进行表扬，表扬必须及时，不要等整个项目都结束了才想起来要表扬或奖励员工，那个时候可能员工本人都忘了自己曾经做过什么事了。

对于员工所表现出来的消极行为，经理不能视而不见，而是应及时予以指出，使员工及时发现及时改正，以便在以后的工作中做得更好。

7、评估项目实施的效果

在项目完成之后，经理应对项目的实施效果进行全面有效的评估，给出评估结果，将评估的结果应用到相关的人事决策中，如奖励、调薪、晋升等。

8、与员工沟通评估结果

评估完之后，经理应将评估的结果正式反馈给员工，并与员工一起总结项目实施过程中的成败得失，为员工提供建设性的改进意见，帮助员工树立信心，引导他们正确评估自我。

环境时刻都在变化，管理时刻都在创新，没有一成不变的模式，也没有放之四海而皆准的方法，经理不妨试试用项目管理员工，领导员工成为项目管理者，打造高绩效的工作团队！

核心提示：

激励员工的手段多种多样，每个企业都有不同的方法，每个管理者都有不同的方式，采用奖金的激励办法最为常见，但这种办法所带来的效应并不持久，到了一定的阶段，这种激励办法将失去作用。还有一些企业把工作丰富化作为激励员工的办法，这里的项目管理就是工作丰富化的一种，与金钱的激励相比，这种办法对于员工的激励作用更大，更能激发员工的工作热情，激发员工追求自我实现。

领导员工成为项目管理者并不普遍适用，只有当员工拥有追求自我实现的热情，而企业又愿意提供一定的施展空间的时候，这种方法才发挥作用。

领导员工成为项目管理者是绩效合作伙伴的最佳实践，因为在项目管理和实现的过程中，员工并不是单打独斗，管理者始终是与员工站在一起的，为之呐喊，为之提供支持和帮助。

IT 经理，你可以管理多少人

Johanna Rothman 是一位经验丰富的管理咨询师，负责向 IT 经理和项目团队提供管理方面的咨询，并出版了多本管理领域的著作。在最近的一篇文章中，她提出了一个比较尖锐和值得深思的问题：作为一个管理者，你能管理多少人？

在 Johanna 的第一任管理者角色中，她只“管理”一个人。被管理的那人并不需要太多的管理。

他指导我如何管理的时候比我管理他的时候还要多。他及时阻止了我犯许多错误，这对我来说真是很棒的一次实践。

之后，在我的管理生涯中，我管理了一个 15 个测试工程师的“团队”。他们并不算是个团队，而是一个群体。我不知道为什么我上司要坚持称他们为一个团队，但是我的上司做到了。我的角色是使测试人员与项目良好的配合。我不认为我的角色是将工程师分配到项目中去，因为大家会长期呆在项目中。我不属于那种可以把人员如同棋子一样任意移动的管理误区中的一员，这也是为什么我认为我的角色是“牵线搭桥”的作用。

Johanna 对于这个测试工程师群体还做了些什么呢？

与每个人进行一对一的沟通。有些一对一的沟通是每周都进行的，有些是每两周一次的。但我至少每两周都和每个人进行一次私人谈话。每两周我都一对一的与每个人进行职业发展对话。

召开每周一次的会议并且每个测试团队的人都会在组会上学到一些东西。这是一个社区实践的会议。

有时是有关于技术实践的，有时是关于工具的，有时是关于项目管理技术的。由这个小组决定他们想要学习什么，由我来促进并推动会议的开展并促进他们学习。人们在会上轮番展示一些成果。是的，我也会去展示，有时候，我还会邀请公司中的其他人来展示。我们有个很长的清单，列满了要学的东西。

我确保每个人都知道其他人在做什么。但并不是通过一连串的进度会议来了解。我记录每个人的电子邮件的状态，整理它们，并通过电子邮件发给所有人。如果有人感兴趣的话，可以阅读它，如果不感兴趣，他们可能将它移到垃圾桶。我当时的想法是，既然他们都工作在不同的项目，他们可能会发现一些对方可能想知道的事情，我知道没有足够的技术能力，无法为他们解决技术问题。如果必要的话，我可以协助他们解决问题。可以提供信息，他们可以跟进。

我确保正确的人会被邀请到合适的会议中。这件事听起来更加困难。有人不想去参加所有的会议，但却被邀请了。我必须确保正确的测试人员被邀请。

在最近一次会议上，Johanna 认识了一个约有 100 名工程师的公司的 CTO（首席技术官），他想要一个扁平化的组织，他说“每个工程经理都应该能够管理 15-20 名工程师，以及他们所做的项目”。Johanna 认为这很难实现：

你会发现我并不是在管理我清单上面的那些项目。除了上述的管理责任之外，我拼了命的工作，并且时间满满当当。我不能再这样做了，如果你问我老公的话，我敢打赌他会说我过去很忙。晚饭后我完成了手机屏幕上所有的事。当我需要的时候，我也会在晚饭后写报告，因为我白天没有时间。

我做了任何有用的项目管理吗？没有。我还能管理更多的人吗？不能了。而且肯定没到 20 个人。为什么？因为我需要时间每周和大家碰面。

为什么我能管理 14 个人？因为在这时我已经是一位有经验的管理者了。我一直在练习。首先管理一个人，之后三个人或四个人。再之后七个、八个。当我的群体里有九个人的时候，我发现对于一些人我不得不每两周进行一次一对一面谈了。我问那些更资深的员工他们是否介意这件事，他们觉得可以接受。但是如果他们资历尚浅并且需要辅导呢？这就会是一场灾难。

Johanna 指出，不要有这样的理解误区：作为一个管理者，你能管理任意数量的人。如果你不关心管理的多优秀，你不能管理任何人，也没法做的很好。这个原则在在写代码和做项目中是同样适用的。如果你不在意代码的质量如何，你可以想写多少就写多少。但如果这些代码不能起作用，它们没有任何意义。如果你不关心自己对于项目管理的如何，那么你可以想管理多少项目就管理多少项目。

我不是在谈论微观管理。我是在谈论如果他人需要的话，我们应当提供指导。我们应该为人们营造一个学习环境，一个让人们可以学习的地方，一次每个人每周都能相互信任的机会。就是这样，我期待我群体里的人们能够把他们余下的时间都用来学习，对自己负责。

但是因为我们都是员工，而且我要对整个团队都负起责任，所以我们很忙。如果我没有抽出时间来一对一面谈，我可能已经很多个星期没有和人们见面了，那将是错误的。作为一个管理者，如果你

管理超过九个人，重新思考一下你可以做什么。如果你不是每周或每隔一周进行一对一谈话的话，那么除此之外你又做了什 么？

管理不仅仅是微观管理，而是创造一个每个人都能做到最好的工作环境。如果你太忙以至于不能做到这点，你是真的在管理吗？

我是如何打败拖延症的

关于拖延症的话题我在 Hacker News 上不断的看到有人提出来(你也读了，不是吗？)，感觉有必要将我如何跟拖延症做斗争的方法分享给大家。然而，我这里说的主要是针对程序员/美工，但其实任何人都可以使用。首先最重要的...

它不是那些老套陈旧的动机心理学扯淡。

我并不是说那些传统的应对拖延症的方法理论不对，只是对我无效。当正经历极度消沉的时候，我通常听到的理论的最后一句话是”You just DO IT!”。我有很多的事情要去做。但我不会去阅读你那 400 页的治疗拖延症手册，也不会执行你那一套自我调整的行动方案。

OK，下面看看我的方法。

第一、你想完成什么？

以程序员为例：

我需要在我的程序里添加讨论和留言功能。

以美工为例：

我需要给这个客户做出 logo。

回答这个问题。有答案了吗？非常好，下一步。

第二、寻找下一步你需要马上去做、而且不带压力的一个任务。

这里真正的艺术就在于我是如何实施这一步(注意，是我如何实施，而不是科学上如何实施)。纵览一下所有你的任务，把它们拆分成小任务，然后再次拆分，直到任务小到你认为可以毫无压力的完成为止。这样的任务此时应该可以用“too easy”来形容。

以程序员为例：

建立一个路径，这样我可以访问”myapplication.com/discussion”

以美工为例：

选出 3 种最适合这个 logo 的字体

第三、完成第二步中选出的任务。

第四、重新从第一步开始。

为什么这种方法好用

首先第一个原因是，这种方法让你没有压力，让你有能很容易完成的心理准备。这个任务如此简单，你会想干脆做了它。如果你觉得自己有拖延症或意志消沉，那这就是你需要的...而不是一个写满了 100 个任务的 to-do 列表。

第二个原因才是真正的原因——不需要担心下一步需要做的事情。还是以开发或美工设计为例。每次你完成了一个用这个方法为你选出的任务，余下的任务的总体情况很有可能发生变化。完成 X 之前，你会认为 Y 应该是接下来最该做的事情。但当 X 完成之后，我用这个方法选出的任务却是 Z...比如：

以程序员为例：

建立了路径后，下一步该做的应该是查询数据库操作... 但在我完成后，用这个方法选出的却是去开发 view 层。

一个美工为例：

我要挑选 3 种字体，接下来应该做的是从中选定一个，但是，因为这三个字体都是这么好看... 我最后决定做的是在 Adobe Illustrator 里用 3 中颜色分别演示它们。

最后一个这种方法对我有效的原因是，它能激发启动动力。让你开始去做。一旦做起来、行动起来，你就不会停下来，当完成任务时，你会发现一步一步的这样做下去是如此的容易。

我已经开发了 4 个大型的应用，都是使用的这种对抗拖延症的方法。这意味着我不是按常规顺序先去计划好 X 和 Y，而是直接进入了设计和编码。所以，这种方法并不只是用在业余项目中的小技巧。

使用缺陷管理工具不应有的坏习惯

Bugfree 是一款优秀的 bug 管理和追踪工具，因此受到不少公司的青睐。但实际的工作中，我发现不少开发或是测试的同事有一些不好的使用习惯，使得我们对 Bugfree 的利用不够高效。我下面列出使用 Bugfree 的一些坏习惯，以此与各位测试同仁切磋使用这个工具的高效的方法。

对开发的同事而言，可能会有下面几条坏习惯。

坏习惯一：只采用默认的解决方案。

周围不少开发的同事，在解决掉一个 bug 的时候，往往只采用默认的解决方案：fixed。事实上，Bugfree 提供了 7 种解决 bug 的方案供程序员选择。它们分别是：By Design、Duplicate、External、Fixed、Not Repro、Postponed、Won't Fix。这 7 种解决方案反应了程序员解决 bug 的理由。By Design 的意思是，设计上就是这么定的，bug 无效；Duplicate 的意思是，这个 bug 已经有人提过，重复了；External 的意思是，软件本身没有问题，是外部因素（比如操作系统）造成的问题；Fixed 的意思是：bug 被解决掉了；Not Repro 的意思是，这个 bug 无法重现；Postponed 的意思是，这个 bug 推迟到以后解决；Won't Fix 的意思是，是

个问题，但是不值得解决。为什么会有这种习惯呢？我询问过一位开发的同事，他说看不懂英文，又懒得去查。另一位开发同事说，一开始没注意到这一项是可选的，时间久了，自然而然就视而不见了。

改掉这个习惯不是更好吗？我的理由：给 bug 设置正确的解决方案，一方面可以减少开发和测试的沟通障碍，让测试员知道程序员为什么要关掉这个 bug；另一方面可以给 bug 归类，便于查找 bug 和开发后期集中解决 bug。

坏习惯二：只在详细信息里写上：已解决。

由 Bugfree 提供的 7 种解决方案，不难看出详细信息这一栏多数情况下是为第四种解决方案 Fixed 提供补充的。很多 bug 在被 fixed 掉以后，如果只在这一栏注明已解决字样会有不好的影响。因为时间久了，或许程序员自己都不清楚这个 bug 是怎么被 fixed 掉的，如果再碰到类似的问题又要花很长时间去想办法解决，影响工作的效率。

改掉这个习惯不是更好吗？我的理由：在详细信息栏里注明 bug 被 fixed 掉的理由，一方面像上面所说的可以给开发人员提个醒，便于解决类似的 bug；另一方面对测试员也有好处。测试员在碰到类似的 bug 以后，能够知道哪儿出了问题，这样就可以准确及时地提醒开发人员，便于 bug 的修改。

对测试的同事而言，可能会有下面的几条坏习惯。

坏习惯一：创建 bug 时，选错了项目。

周围有测试同事，在发现了 bug 后，就急急忙忙去 bugfree 里描述 bug 和指派 bug，往往会忽略其他的选项。就拿这个项目来说，登录 bugfree 后里面就有默认的内容，但未必是和 bug 相对应的。如果测试人员因为发现了 bug，有点兴奋，再加上一点粗心就会忽略这一项。

改掉这个习惯不是更好吗？我的理由：设置正确的项目可以给 bug 归类，便于 bug 的查找。若选错了项目，难免会抱怨找不到自己创建的 bug，还得通过其他方式查找这个沉入“大海”的 bug，影响了工作的心情和效率。

坏习惯二：创建 bug 时，没有选/选错了模块。

创建 bug 时，模块这个选填项，不仅看起来不起眼，而且会让人误以为它没有用，所以测试人员往往会忽略它。

改掉这个习惯不是更好吗？我的理由：设置正确的模块，可以给 bug 分门别类。这样，开发人员就能很方便知道 A 模块有哪些 bug，B 模块有哪些 bug，让开发人员对自己负责的项目模块心里有底。所以，还请测试人员辛苦下，把模块这一项设置好。

坏习惯三：设置错误的严重等级。

周围有同事，往往只注重对 bug 地描述，不去关注对 bug 等级的设置，不利于开发人员优先解决严重的 bug。其实 Bugfree 提供可选的 4 种严重等级：1、2、3、4。1 是最高等级，意思是这个 bug 导致系统死机，数据丢失或者与需求不符合；2 是严重等级，意思是这个 bug 导致计算出现错误，功能实现出现错误；3 是一般等级，意思是这个 bug 是个合法性问题，界面问题或是文档问题等；4 是最低等级，意思

是这个 bug 影响易用性。不少情况下是测试人员不清楚各种级别的含义，导致的分类错误。

改掉这个习惯不是更好吗？

我的理由：设置正确的严重等级，可以让开发人员优先解决 1、2bug，在项目时间允许的情况下，再着手解决 3、4 类 bug，以保证产品的质量。啰嗦一句，首先要保证产品能用，再去保证产品好用。

其实最好的习惯是按照 bugfree 的格式，把每一项该填的内容填好！！

软件产品



AutoRunner (自动化测试工具)



TestCenter (测试管理工具)



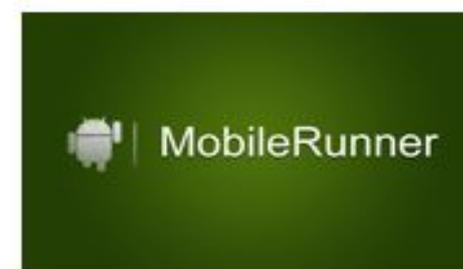
TAR (银行终端测试工具)



PPM (项目管理软件)



PerformanceRunner (性能测试工具)



MobileRunner (App测试工具)



CodeAnalyzer (静态分析工具)

泽众软件工具使用技术支持

电话：021-61079698

Email：sales@spasvo.com

QQ：1404189128

培训、产品购买及试用授权等事宜

电话：021-61079698

QQ：1404189128

Email： sales@spasvo.com