

西西吹雪：从程序员到项目经理（一）

来源：[西西吹雪](#)

“从[程序员](#)到项目经理”，这个标题让我想起了很久以前一本书的名字《从Javascript到Java》。然而，从Javascript到Java充其量只是工具的更新，而从程序员到项目经理，却是一个脱胎换骨的过程。从Javascript到Java，是一个取巧的方法；而从程序员到项目经理，却并无捷径可走，必须从内而外的改变和提升。

一．为什么要当项目经理

1. 问题本质

如果我对一个老程序员说：“有必要转项目经理啦”，很多人第一反应是“为什么一定要当项目经理？！”，反问很给力，甚至会让人哑口无言。但反问成功的结果可能只是使自己麻醉，暂时忘却现实中面临的烦恼和压力，这无异于把头埋进沙子中的鸵鸟。只有理智的分析，才能作为自己行动的指南。

首先申明，不是每个程序员都**需要**当项目经理，也不是每个程序员都**想**当项目经理，更不是每个程序员都**能**当项目经理。因此，当不当项目经理，可以说是一个“需不需要、想不想、能不能”的问题。

想不想，是一个意愿的问题。这是前提，毕竟强扭的瓜不甜嘛。显然，富二代一般是不想当项目经理的，因为他们想直接当总裁。还有些人，只想钻研技术，不想钻研人，他们也是不会想当项目经理的。如果你没有意愿当项目经理，也就没有讨论的必要了。什么，你不知道想不想？呃，那就继续往下读吧，也许读着读着，你就想当了。

能不能，是能力的问题。这是不关键，因为只要有意愿，能力是可以培养的。程序员连复杂得让人琢磨不透的软件都能[搞定](#)，还有什么搞不定的？

因此最后落实在需不需要这个问题上。这个问题很棘手，需要从程序员自身以及外部环境等方面进行分析。要讨论这个问题，就要弄清楚它和想不想的关系。想和需要是紧密相关的，但并不是一回事。想不想，主是感情的因素，而需不需要则要进行理智的分析的了。理智与感情，并不总是一致的。有些东西，是你需要的，但你未必想要。比如，被困沙漠的时候，有时被逼喝自己的尿液，这是理智战胜了感情。电影《色戒》中的汤唯，则是感情战胜了理智，爱上了敌人，最后造成了悲剧的结局。因此，我们还是少说气话了，不要冲动，冷静的分析自己的处境吧。

2. 鸭梨山大

当我从网上看到码农这个词时，觉得网民很有自嘲精神，后来我看到了码畜和码奴这两个词，不禁从心底涌起了深深的悲哀，为这个行业，也为这个社会。

看看智慧的网民对 IT 人士级别的划分：

IT 领袖：年入过亿（例如任正非、马化腾、李彦宏、丁磊、马云等，包括期权股票以及投资理财等收入。）

IT 大哥：年入千万（级别次于以上几位大佬的公司老板，不缺钱，普遍对上一条里的人物羡慕嫉妒恨。）

IT 精英：年入百万（各 IT 公司副总裁级别人物，包括 COO、CTO 等，大多为职业经理人，赚够钱就跑。）

IT 人才：年入 50 万（各 IT 公司总监级别人物，有房有车，生活压力相对较小）

IT 工程师：年入 20 万（高级经理级别，有房贷，生活压力大）

IT 民工：年入 10 万（经理级别，基本无房，学会装波一，生活压力大）

码农：年入 6 万到 10 万（工作三四年，租房，继续混日子）

码奴：年入 3 万到 6 万（工作一两年，租房，混日子）

码畜：年入低于 3 万（刚毕业的，租房，傻乐）

我知道你想问什么问题了。不要问哥赚多少，哥只是一个普通的 IT 人士而已。前面三级都是牛人，是成功人士，他们的作用不是让去成为他们，而是激励我们自己。你现在读到的也不是一篇成功学的文章，而是和你一起分析程序员的处境、以及怎样缓解压力的文章罢了。

言归正传。看到这个表，是不是有鸭梨山大的感觉。找到了自己的位置吗？什么，不好意思？没关系啦，园子里面不是很多人称自己为程序猿或者猴子吗？那大概也就是相当于码畜吧。我想能读到这篇文章的，大概都是“IT 工程师（高级经理）”以下，他们的主要特征是“生活压力大”和“混日子”。如是你是前面四级，建议你果断退出本文。

我在上一篇博文中提到 30 岁现象，有些人认为车到山前必有路，这是杞人忧天。不错，程序员确实可以干到 30 多岁，甚至四五十岁，但他们面临的压力却可能是“不足与外人道也”。

我经常与 30 岁以上的程序员交流，他们流露出来的对现状的不满、无奈、无力、对安全感的缺乏，让我感同身受。

虽然谈压力并不是一件愉快的事情，但我仍然必须要说出来，因为我宁可清醒的痛着，也不要再在麻醉中睡去。那就让我们拿着手术刀，对自己进行痛苦的解剖吧。

下面是一个简单的“危机评估表”，总共有 30 项。在“是否认同”后面打出分数，每一项如果认同为 1 分，不认同为 0 分。

类别	评估项	是否认同
身体	悄然发现已经没有以前经折腾了。	
	没有定期的体育运动。	
	中餐午餐都是在外面吃快餐。	
	确信自己是亚健康。	
家庭	每月开支不算不知道，一算吓一跳。	
	有房贷或房租。	
	有孩子了，上幼儿园是一笔大开支。或者超过 30 岁了还没结婚。	

时间	买不起车，或有车子，开不起。
	家里时有摩擦，经常有不开心的事。
	每个月存不了多少钱。
	要花很多时间陪家庭成员。
	加班时间越来越少。
激情	社交时间较少。
	只想休息，不想工作。
	对新技术、新工具不甚了解，有心无力。
	没有制度明确的短期、中期和长期目标。
	理想已经模糊了。
社会	只有交税，没有回报。
	担心老了病无所依，老无所养。
	担心国家经济衰退，陷入失业。
	收入增长跟不上通货膨胀的速度。
	新人比我更具有性价比。
行业	行业竞争激烈，低价抢标现象严重。
	行业被某些公司垄断。
	行业正在慢慢衰落。
	公司发展前景不是很明朗。
	公司薪资福利一般。
公司	公司没有企业文化。
	公司员工关系比较紧张，有内斗现象。
	公司缺乏活力。
	总分

（说明：此表并不精确，仅供参考）

如果总分小于 10 分，那要恭喜你，说明你生活稳定幸福，让人羡慕。我觉得这篇文章你也不用往下看了。

如果你的总分大于 20 分，说明你承受的压力过大，可能面临职业方面的危机，应当寻求改变了。

如果总分在 10-20 分，说明你生活比较稳定，收入方面可能是中上等水平，但职业发展方面仍有风险。

3. 另一片天地

所谓“穷则变、变则通”，如果你还是普通的老程序员，并且还在为自己的职业彷徨和苦闷，那就应该寻求变化之道了。

如果你愿意，转向项目管理乃是上上之策。

当然转项目管理只是程序员很多选择中的一个。显然不是每个程序员都需要当项目经理。一般每个公司都最少提供了技术和管理两条职业发展通道，如果你技术超牛，你完全可以从程序员做到系统分析师，一直做到技术总监。如果技术方面你信心不足，转项目管理就是一件自然而然的事情了。

技术和管理，这是两条绝然不同的路，虽然“条条大路通罗马”，但沿途的风景却是完全不一样。一旦你从事了项目管理，你将看到不同的另一片天地。

（1）在管理的天地里，你将不再有职业瓶颈。

程序员虽然也可以干一辈子，但工资水平是有天花板的，不要问我为什么，行业就是这样。项目经理则有无限上升的空间，不但工资更高，职位上也可以升至部门经理、副总经理甚至总经理职位。

（2）促进项目经理内在成长，心智更加成熟。

美国项目管理协会 PMI 认为，项目经理 75%-90% 的时间应该用在沟通上。沟通的对象显然是人，因此，项目管理主要是一项与人打交道的工作。如果说解决技术问题人主要是靠一个人的智商，那么与人打交道，则是要靠一个人的情商。

虽然不当项目经理也可以发展情商，但在项目中锻炼是自我成长、自我完善的捷径。

（3）项目管理知识可以用在生活中的各个方面。

生活中的许多事情，我们并没有称之为一个项目，但可以用项目管理的方法来对待。例如一次婚礼的组织，或一次自助旅游。你在项目管理中培养起来的情商，更是让你面对生活中的各种问题游刃有余，你的家庭也会更和谐，就像范范的一首歌里唱的：“好像什么困境都知道该怎么办”。当到达这种境界时，你会有一种海阔天高，一览众山小的感觉。

因此，即使你不想从事项目管理，也建议你学习一下项目管理知识。有一本书叫《不懂项目管理，还敢拼职场》，虽然觉得内容一般，但对标题深以为然。

二. 项目管理到底难不难

程序员问：“我现在想当项目经理，但心里没底，不知道项目管理到底难不难？”这个问题确实不好回答。俗话说，“会者不难、难者不会”，很多事情都是如此。

有些人觉得不难，他们好像天生就具有管理的才能，他们举止得体、八面玲珑，具有很强的个人魅力，可以把大事化成小事，把坏事变成好事。这样的人，想不成功都难。

大部分人还是会觉得难。在 PMI 的知识体系里，项目管理有九大领域，五大过程组，44 个过程，有数不清的工具和方法。项目执行中方方面面出了问题，都是项目经理的责任，项目经理又不是超人，怎么应付得过来。项目管理确实有点难。

你若问我，我会说项目管理既难，又不难。对于愿意改变自己的人而言，它不难；对于性格偏执的人而言，项目管理确实太难了。

很多人无法意识到自己的偏执。上级只要提出一点批评，他们就要拼命的辩解和反驳。他们的保护壳太厚了。

项目经理最重要的素质，就是心智的成熟，一个心智成熟的人，不会是一个偏执的人。

毕竟，人无完人，项目经理必须从善如流，才能完成自己角色的转变。对于从程序员转过来的项目经理，做事的方法与以前应是翻天覆地的不同，必须迅速审时夺势，改变自己。否则，那你还不只是个有项目经理职位的程序员么？

因此可以说，项目管理难就难在项目经理要改变自己。这个改变，不只是知识体系的扩充，更可能是性格的改变，而一个人要改变性格是极其困难的。

程序员习惯于与机器打交道，通过严密的代码和逻辑来控制机器；而项目经理是跟人打交道，人是有感情的，绝对不是你给他输入 1+1，他就给你输出 2。项目经理必须时时用心去思考、体会，然后改进。几番回合下来，项目经理会惊喜的发现自己变了，有种脱胎换骨的感觉——那是当然的，因为变得更成熟了。

只要你愿意改变自己，假以时日，你一定会成为一个优秀的项目经理。

三．程序员应克服的障碍

程序员与项目经理之间，往往有一条鸿沟。对技术钻研越深的程序员，这条鸿沟可能越大。这是由程序员的性格特征决定的。

程序员普遍有非常多的优点：例如聪明、逻辑思维强、学习能力强、创新能力强、直率等。但优点往往也是弱点之所在，例如：

（1）太讲逻辑：与人相处时容易忽视人际关系、感情等方面的因素。

（2）过于直率：说话直来直去，容易伤害他人感情。

(3) 自傲：总觉得自己技术不错、比周围的人要强一点。好比一只鸡看到同类觉得自己最大，看到鹅觉得跟自己差不多，看到火鸡才觉得比自己大一点。

(4) 固执：在自己的逻辑中不能自拔，无法听取别人的意见。

(5) 沟通能力较弱：大部分程序员在口头表达、写作、汇报、交流等方面存在不足。

而这些缺点，也是心智不够成熟有表现，这是项目经理的大忌，往往会成为程序员晋升项目经理的障碍。因此，必须要克服这些障碍，给自己制定符合项目经理要求的行为准则，时时提醒自己，每日进行反省，坚持下去，必然会成功。

西西吹雪：从程序员到项目经理（二）

来源：[西西吹雪](#)

被任命为项目经理，是职业生涯的第一次飞跃，既惊喜又紧张。从现在开始，你要思考怎样才能胜任项目管理的工作，否则等着你的，很可能是一场悲剧。

一. 升职之辨

1. 为什么是我

不是每个人都能当项目经理，[程序员](#)中只有一小部分能成为项目经理，大部分人会随着岁月的流逝，成为了“资深程序员”。

那为什么领导要选择我呢？一般人对自己所拥有的东西都会很快习以为常，认为这是自己应得的。一点也没错，这就是你应得的，原因也很简单，那是因为你**比别人优秀一点**。

其实领导挑选人才的标准很简单，那就是你比别人优秀，而且只需一点点。你不需要“鹤立鸡群”，“鸭立鸡群”已经足够了。俗话说：“群众的眼睛是雪亮的”，其实领导眼睛才是真正雪亮的，如果他还没有发现你，那是因为你还不够优秀，没有引起他的注意。

因此，如果你工作多年仍然没有职位上升，不要埋怨公司不给你机会，而应该从自己身上找原因，机会只会给有准备的人。如果你不知道自己准备好了没有，就试着回答下面的问题吧：

- 工作是不是比别人积极主动一点；
- 加班是不是比别人多一点（如果贵公司喜欢员工加班的话）；
- 提交成果是不是比别人提前一点；
- 成果质量是不是比别人要好一点；
- 学习是不是比别人勤奋一点；
- 面对问题是不是比别人勇敢和执着一点；
- 人际关系是不是更和谐一点。

如果你能做到这些，相信机会迟早会属于你的。

2. 彼得定律的启发

心理学中有一个词，叫“光环效应”，是说当我们对一个人某个方面有好的印象时，我们会倾向于认为他的其他方面也是好的。因此，当你能胜任你现有职位、比别人优秀一点时，领导会认为你是下一个职位的最佳人选。然而实际上，你不一定是最合适的，但有什么关系呢，你已经是项目经理了，你有很多时间，可以边做边学。但是，如是你长期不胜任项目管理工作，项目经理就会成为你职业生涯的最高职位。

这也就是彼得定律的内涵：“在一个等级制度中，每个员工趋向于上升到他所不能胜任的职位”。

从彼得定律中，我们可以得到以下启发：

（1）在公司里面，大部分人都干着他不能胜任的事情。这听起来真是一个悲剧，好在我们暂时还不用操心。

（2）金子是一定会发光的，人才绝对不会被埋没的。这是由于人才的稀缺性造成的，只要是胜任当前职位，晋升是迟早的事。因此，无论是程序员还是项目经理，都要做好你的本职工作，这才是最重要的。试想，如果本职工作都没做好，怎么可能提拔到更高职位呢？别告诉我还可以走后门。

（3）当上了项目经理，只是说明你可以胜任程序员职位，而不意味着你可以胜任项目经理。因此，别急着庆祝，还是多想想怎么来管项目的事情吧，否则你就可能是下一场悲剧的主角。

（4）如果你已经担任项目经理很长时间，还没有得到升迁，不要骂老板，这只是说明你没有完全胜任项目经理的职位，还是赶快想想怎样完善自我，提升内功吧。

二、新任项目经理的误区

新任项目经理，由于经验和知识储备的不足，往往会出现相同类型的问题。

1. 农夫的一天

有一个小故事，讲的是一个农夫的一天：

有一个农夫一早起来，告诉妻子说要去耕田，当他走到 40 号田地时，却发现耕耘机没有油了；原本打算立刻要去加油的，突然想到家里的三四只猪还没有喂，于是转回家去；经过仓库时，望见旁边有几只马铃薯，他想起马铃薯可能正在发芽，于是又走到马铃薯田去；路途中经过木材堆，又记起家中需要一些柴火；正当要去取柴的时候，看见了一只生病的鸡躺在地上……这样来来回回跑了几趟，这个农夫从早上到夕阳西下，油也没有加，猪也没有喂，田也没耕，最后什么事也没做好。

故事看上去很可笑，但笑过之后，回过头思索一下，故事里是不是也有我们项目的影子呢？我们将《农夫的一天》换成《项目经理的一天》：

软件项目经理小赵打算今天完成本周五项目阶段汇报的材料，他打开电脑，想起了还有一个重要的技术问题没有确定最终方案；于是他召集项目技术骨干准备继续讨论，一个钟过去了，还没有

结论，这时老板来电话，要去老板办公室汇报工作，原来昨天老板跟客户吃饭，客户说到系统有一项功能无法使用，两周了还没解决；从老板房里出来，小赵继续写汇报材料，没多久，项目组的小张找来要反映项目组绩效考核结果以及加班工资的问题；快下班的时候，销售部经理匆匆忙忙地找到小赵：“快帮我估算一下这个项目的实施成本，明天我要给客户报价”……就这样，小赵一天都忙得不可开交，终于下班了，汇报材料没写多少，重要技术问题也没有解决，客户的问题也没安排处理，绩效考核的问题还要跟部门经理以及人力资源部沟通。唯一完成的一件工作，就是帮销售部估算成本，可惜跟自己负责的项目却没什么关系……经过一天的奋战，问题不但没有减少，反倒变多了。

这样的一天无疑令人沮丧，但却经常出现在我们的现实中。当高级经理询问怎么还没有提交项目计划的时候，项目经理无可奈何又理直气壮的说：“我很忙啊！”

项目经理确实很忙，但这是没有效率的忙。其实何止是忙，还“茫”，而且“盲”，“忙、茫、盲”是许多新任项目经理的写照。

- 忙：一天到晚都在忙过不停，是为忙碌；
- 茫：碰到什么做什么，像个无头的苍蝇，没有计划性，或者无法坚持计划，是为茫然；
- 盲：项目经理这一天初始目标究竟要做什么，做着做着就丢了，没有目标性，是为盲目；

2. 思维转换

有时候我们会说一个项目经理，不像一个项目经理，那像什么呢？当然是像程序员罗。也就是说，他的职位虽然变化了，但并没有完成相应的角色转换，仍然像程序员那样工作。项目经理之所以会出现“忙、茫、盲”状态，归根到底也是因为他没有实现自己的角色转换。

角色转换本质上是思维转换。思维决定一个人的行为，项目经理不像项目经理，那是因为他的思维仍然是以前的技术思维，而不是管理者应当具备的管理思维。这就好比一个人在陌生的城市，拿着过时的地图，寻找自己的目标，结果只会是四处碰壁，无所适从。

表 1 技术思维 vs 管理思维

比较方面		技术思维	管理思维
关注中心	以过程为中心的思维	关心每项任务本身，而不是整体目标。不重视计划，对任务缺乏控制。	以目标为中心的思维
			以终为始。关注整体目标、实现的路线、影响目标实现的因素、各种事件对目标的影响，区分重点。
事物结构	局部思维	过于关注细节，对整个项目工作的内容、完成路线没有概念。上来就干，工作缺乏计划性、条理性。	整体思维
			采用结构化分析方法，自顶向下，先整体后局部。有时亦采用头脑风暴，先将细节展开再归纳。

逻辑思维	以机器为中心的思维	思想单纯，性格直率。在人际关系上过于讲究逻辑。	以人为中心的思维	人是执行项目的主体，关注事情本身，更关注人的价值。学会包容，能与各种不同情格的人打交道。
决策依据	完美思维	不关心进度和成本，只关心完美的功能和代码，并视之为艺术。经常对上一任的工作推倒重来。	平衡思维	拒绝镀金，项目不需要艺术。在进度和质量之间取得平衡，在员工个性与团队凝聚力之间取得平衡，在员工、项目、公司和客户之间取得平衡。
人际关系	个人思维	以个人为中心，单兵作战，依赖个人能力。个性固执，工作方法简单。	团队思维	你不是一个人在战斗，发挥每个成员的作用比个人埋头苦干重要得多。关注团队分工、配合以及士气和凝聚力。

实现思维转换需要时间，这期间是一个懵懂的、左右为难的、痛苦难熬的阶段。有些人可以在很短的时间内完成蜕变，有些人却可能一辈子都在这个阶段，这跟一个人能不能改变自己有关。这些不能改变自己的人，理论知识往往也很丰富，说什么都头头是道，可惜的是，这是无效的知识，因为不能用在自己的实践中。这样的人，往往有一定的人格分裂倾向，因为他的知识和他的行为不统一，甚至是矛盾的。知行合一才是学习的最高境界。

新任的项目经理，别忘了时刻提醒自己，像一个项目经理一样去当项目经理！

3. 项目经理行为分析

第一次当项目经理，往往会由于经验不足、项目管理知识的不足以及角色转换等原因，表现出种种不胜任的迹象。

不胜任的项目经理，通常有以下几种类型：

（1）刺猬型

刺猬型的人非常敏感，随时都保持警惕，只要一感觉受到威胁，便会用豪猪般的刺扎向对手，让人避之不及。他们通常自我封闭，坚守自己的地盘，处处表现出来自己是对的，虽然其实他自己也并没有底气。他不会让别人看到自己的脆弱。

刺猬型项目经理不允许别人干涉自己的项目，哪怕是自己的上级。如果领导询问项目中的某个问题时，他会非常明确的告诉你，那不是我的问题，那是客户的问题，或者是公司制度引起的问题，或者是领导你干预项目造成的问题。总之，我一切都做得很好。

刺猬型项目经理的这种反应通常是不自信的反应。小猫在害怕时，总是拱起背，把全身的毛都竖起来，让自己看起来更强大，但老虎永远不会这样。

（2）绵羊型

绵羊型项目经理的性格非常温顺，他们语气平和，慢条斯理，不急不躁。对待下属非常友好，在他们心里，似乎没有好和不好、对和不对，这些对他们都不重要。项目每天都很平静，似乎永远不会有暴风雨的到来。当上级提出要求时，他们永远都是好的，至于做成怎么样，只要尽力了，那有什么关系呢？

绵羊型项目经理通常工作缺乏计划性，即使有计划，也只是应付上级而已。看到什么事情，就去做什么事情，除此之外，还有什么其它的办法吗？

（3）猴子型

想像一下孙悟空的行为就对猴子型项目经理有了大致的认识。他们技术能力强，很有激情，非常聪明，非常自信。但他们往往性格冲动，做起事来横冲直撞，不讲究方法。

猴子型项目经理悟性很强，进步会很快，他们最终会克服自己的不足，像孙悟空一样，取得正果。这一刻，他已经不是猴子了。

刺猬型和绵羊型项目经理，他们往往缺乏自信，其管理模式一般是被动式的，做事没有计划性，有什么事就做什么事，就像条件反射一样，只会对外界刺激做出反应。

猴子型项目经理则是主动式的管理，他们充满自信，但往往由于经验不足，过于盲目，对问题考虑不周。同时由于冲动的性格，在团队中并不受欢迎。

这三种类型都是不胜任的表现，那怎样才是胜任的类型呢？如果还是用一种动物来比喻，我觉得应该是“头狼”，也就是狼群的首领。

暂时的不胜任不要紧，关键是要有进步。如果一个项目下来，除了很疲惫，你没有感觉到自己有一些积极的变化，那你的危机也要来了。要知道，项目经理并不是“铁饭碗”，虽然公司倾向于选用有经验的项目经理，但当你明显不胜任时，领导不会再在你身上押上赌注，他们宁可重新冒险一次，因为他们不想“两次踏进同一条河流”。

4. 心态

新任项目经理没有管理经验，不胜任是可以理解的。也许你认为公司应该给你更多的培训再上岗，但往往形势是箭在弦上，在没有更多资源的情况下，领导把这个成长的机会给了你。

可怜的是公司老板，他的项目成了你的试验田。实际上，公司提拔你做项目经理，就是花巨资送你去培训学校，不是吗？我一直认为，由一个不合格项目经理负责的项目，相比由优秀的项目经理来带，实施成本可能多出 50%，甚至更多。不合格的项目经理就像一个给项目减肥的机器，使得肥肉变瘦肉，瘦肉变骨头，骨头变渣滓。

项目经理应该学会感恩。要成为优秀的项目经理，应该有好心态，而感恩是一切好心态的基础。你只知道自己压力大，却不知道你让老板少赚了多少钱！是老板交学费帮你从一个初出茅庐的项目经理，培养成了一个合格乃至优秀的项目经理。

我见过不少新任项目经理，对公司满肚子怨气，好像是公司一手造成他的项目问题百出，仿佛领导和老板成了他的敌人，刚做完项目甚至还没有做完项目就果断匆匆辞职，带着公司用无形成本换来的宝贵经验，绝决的离去，换取更快的升职加薪。设想一下你是老板，不知会作何感想？

感恩是一个人最重要、最美好的品质之一。网上有一个经典感恩的段子：“...感谢鞭打你的人，因为他激发了你的斗志，感谢遗弃你的人，因为他教导你该独立，...凡事感激,学会感激，感激一切使你成长的人!” 而你的领导和你的老板，他们既不是鞭打你的人，也不是遗弃你的人，而是培养你成长的恩人，我们有什么理由不感谢他们呢？（<http://watsonyin.cnblogs.com>）

从程序员到项目经理（三）：认识项目经理

来源：[西西吹雪](#)

（阅读提示：本文比较长，如赶时间，可直接跳到第二节的第四点，即“西西吹雪的六种能力模型”）

在希腊德尔斐的阿波罗神庙上，刻得着一句神秘的箴言：“认识你自己”。从某种程度上来说，我们都是自己的“最熟悉的陌生人”。认识自己的位置，是每个人获得成长的第一堂课。一个人的位置，对其言行的影响是至关重要的，俗话说：“屁股决定脑袋”，虽然听着粗俗，却饱含人生哲理。既然我们屁股在项目经理的位置上，就应该像项目经理一样去思考问题，做事情。

一．项目经理的处境

经过数年的打拼，怀着美好的向往，我们终于成了他——项目经理。然而，梦做到最真的时候，往往也是梦醒的时候。

项目经理其实也是悲情人物。从“程序猿”到项目经理，可以说是刚出虎穴，又入狼窝。要知道，做一个合格的项目经理，比成为一个优秀的[程序员](#)，还要难得多。

本来以为当上了项目经理，王子和公主从此就可以幸福的生活在一起了，没想到，跋涉的路才刚刚开始。我实在不想打碎这美好的梦想，这有些残忍，但清醒的痛着，总好过麻木的睡着。更何况人生本来就是一个接一个的杯具，每个角色都有他的难处，我们只能接受这个现实。人生就像登山，当你到达一个山头时，发现还有更高峰，一山还比一山高。

王子和公主，一直在路上。

1. 高和低

没有成为项目经理之前，期望着当上了项目经理，可以拿着更高的工资，被别人尊敬的称呼为某某经理，还可以干着更少、更简单的活——指挥别人干活，这谁不会啊？

然而，人生不如意十之八九。更高的工资，应该是有的，但往往还不会达到让你眼前一亮的数字。被尊称为经理，也是应该的，Project Manager，名正言顺的经理。然而，在大部分公司里，项目经理也就是像弼马温一样的小官，明白真相之后，又难免有一些失落。至于干更少、更简单的活，那就只能说是痴人说梦了。

事实上，在兴奋过后，等你翻到硬币的另一面，你会看到和你想像不一样的高和低：能力要求高、职位低。

(1) 能力要求高

能力要求高不高，口说无凭，我在网上随便找了一个软件项目经理的招聘信息，要求如下：

职责范围：1、负责软件项目管理及计划实施；2、具备较强管理、协调及沟通能力，帮助开发人员解决开发过程中遇到的技术问题，做好日常的团队管理工作；3、与各团队协同工作，确保开发工作正常顺利的开展；4、具备较强的分析问题、解决问题的能力，能够解决项目团队在开发过程中遇到的技术难题；

任职要求：

- 1、计算机相关专业，4年以上 JAVA 软件从业经验，2 年以上开发经理或团队管理经验；
- 2、精通 java、jsp、HTML、JS、xml、AJAX[编程语言](#)，精通 Struts、Hibernate、Spring、IBatis 等常用框架技术；
- 3、精通中间件技术，对 Websphere、WebLogic 等有很深的了解；
- 4、快速适应工作环境，应变能力强，抗压能力强；
- 5、重视成本和进度控制，合理有效利用资源，有较强的责任心；
- 6、熟悉 Android 开发、Hadoop 技术者优先考虑；

上面的要求写得比较随意，我帮他整理一下，并点评一番：

表 1 项目经理职责要求

类别	职责/要求	点评
专业技术	精通多种编程语言和技术框架；精通中间件技术；熟悉 Android 及 Hadoop。	项目经理必须是技术专家，也许你自己不用写代码，但你必须能指导下属，解决技术问题。必要时，还得参与做系统架构和系统分析。
管理技能	项目整体管理；成本管理；进度管理；资源管理；团队管理；沟	难道风险管理、质量管理、采购管理就不需要了吗？九

通协调能力。

大领域一个都不能少。

① 适应能力：像变色龙。能适用不同公司文化和氛围，不同性格的同事，特别是上司。

② 应变能力：像变形虫。项目过程中会出现各状况，必须能调整自己、调整计划，以适应变化。

适应能力；应变能力；抗压能力；③ 抗压能力：像驴子。项目管理压力很大哦，天塌下来要也扛着。
个人内在 责任心；分析问题解决问题的能力。

④ 责任心：项目出问题，基本上责任都是你的，决不可推卸责任，勇敢的去解决问题吧，不要辜负领导的重托。

⑤ 逻辑思维：项目经常会出问题的，所以你必须思维清晰，能够客观的分析问题和解决问题。

相关经验 4 年开发经验+2 年管理经验

老板可不想冒险，把项目给你去做试验田。

怎么样，要求很高吧？能完全达到这样的要求，我想去铁道部当个 CIO 应该是没什么问题了。即便如此，对于项目经理而言，这些要求也没有哪一项是多余的，也就是说，项目经理必须成为一个超人，最好是像《蜘蛛侠》里面沙人那样，可以随心所欲的变化自己，穿越一切障碍，拥有无穷的威力。

（2）职位低

说职位低，有以偏概全之嫌。在项目型组织结构的公司中，项目经理的职权还是很大的，项目经理一般直接向总经理汇报工作。但在 IT 行业中，比较少采用项目型组织结构，大部分是矩阵型或职能型的组织结构。在这种架构中，项目经理基本上就是最小的官了。

2. 大和小

项目经理之所以需要很强的个人能力，归根到底是由项目经理的责任所决定的。项目是一种个人责任制的管理方式，项目经理是项目组的核心，责任无疑很大；与之相对应，其权力又是比较小的，这让项目经理的处境更加困难。

（1）责任大

项目经理作为项目组的最高领导，对项目的成败起着至关重要的作用。对项目的目标和实施过程中的一切问题，负有最终的责任。影响项目成败的因素也许有许多，但不管是什么原因，最终的责任会落实在项目经理身上，领导会说，项目经理不给力。

(2) 权力小

项目经理的正式权力包括指挥权、人事权、财权、技术决策权以及采购权等，项目经理一般在某一限度内具有完全的权力，无需沟通汇报即可自行做出决定；在超出限度时，则需要与高级经理或职能经理商议决定。在一个矩阵型组织结构的公司中，项目经理的权力大致如下表所示：

表 1 矩阵型组织中项目经理权力情况

权力类型	完全的权力	部分权力
指挥权	对项目内的人、财、物的调度安排，可以自主决定。	对项目结果产生较大影响时，需与高级经理讨论。
人事权	可以依据公司制度对员工进行考核、奖惩。	人员的聘用、辞退等决定一般由职能经理安排，项目经理可以作出建议。
财权	小额活动经费一般可以自主决定	达到一定金额需要申请，由高级经理直至总经理审批
技术决策权	一般技术措施可以自主决定	重大技术措施，必须通过外部评审，并请上级领导拍板
采购权	小额采购项目必须品	达到一定金额需要申请，由高级经理直至总经理审批

乍看上去项目经理权力并不小。但在实际操作中，项目经理权力范围的这个限度往往比较小，并不足以保证项目经理推动项目顺利开展，项目经理必须花去大量的时间去与上级领导沟通、汇报、提出建议、争取支持。在有些公司，甚至连项目组聚餐也要向上汇报请示。项目经理的这种处境往往会导致其工作畏首畏尾，做事犹豫不决，久而久之，失去了对工作的激情。

3. 夹心饼

项目经理的位置是比较尴尬的。下面的兄弟要你多争取一些奖金；领导要你经费更省一些；客户要你更快一些；用户要你的产品更好用一些。在员工面前，你代表老板；在老板面前，你代表项目组员工；在客户面前，你代表公司。你代表了很多人的时候，就是没有代表自己的时候。

项目经理就是一个不折不扣的夹心饼。做人难，做项目经理更难啊。

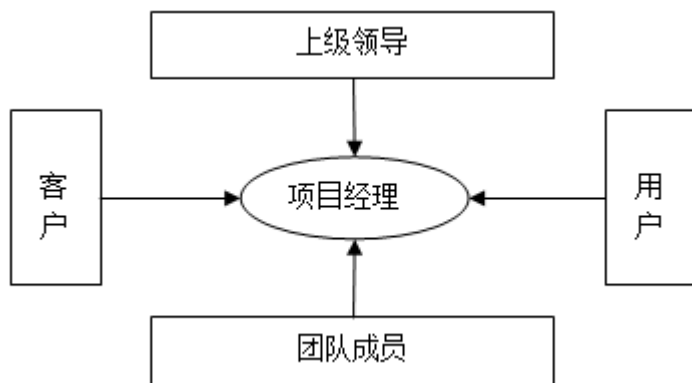


图 1 项目经理成了夹心饼

4. 为什么还要做项目经理

也许你会问，既然项目经理这么难、这么惨，好像比“程序猿”还要苦逼，那我为什么还要做项目经理呢？这看上去不是个问题，“人往高处走，水往低处流”嘛，高处虽然艰险，向上追求的脚步却不能停止。无限风光在险峰，还是别埋怨攀登的辛苦，好好享受一路的风景吧。

当然，人的一生有不同过法，有些人喜欢在泳池中游水，有些人在热衷于在大海的激流中冲浪，还有些人，一辈子也不会游泳，他们只是偶尔到河边洗洗手，用冷漠或者好奇的目光看着那些乘风破浪的人们。每种活法的选择权在自己手上，一旦选择，无怨无悔。

二. 项目经理素质模型

1. 素质模型的作用

谈素质模型是一件很严肃的事情。因为素质模型就像一面镜子，项目经理拿来一照，可以发现自己的优势和弱点。只有扬长补短，才能在职业发展之路上步步高升。

管理方面的素质模型很多，但不是每一个都是客观的镜子，如果不能在镜中看到一个真实的自己，那它也就失去了应有的价值：

如果它是一面哈哈镜，那看到的将是一个变形的自己，无法作为自己的参照；

如果镜子太小，就只能照到自己的局部，会导致产生盲目的悲观或乐观；

如果镜子太大，可能会看到太多无关的东西，反倒干扰了自己的视线。

2. 他山之石

(1) PMI 知识体系模型

PMI 将项目经理应具备的知识和技术分为五类，即：项目管理知识体系，应用领域知识、标准与规章制度，理解项目环境，通用管理知识与技能，人际关系技能，如下图所示：

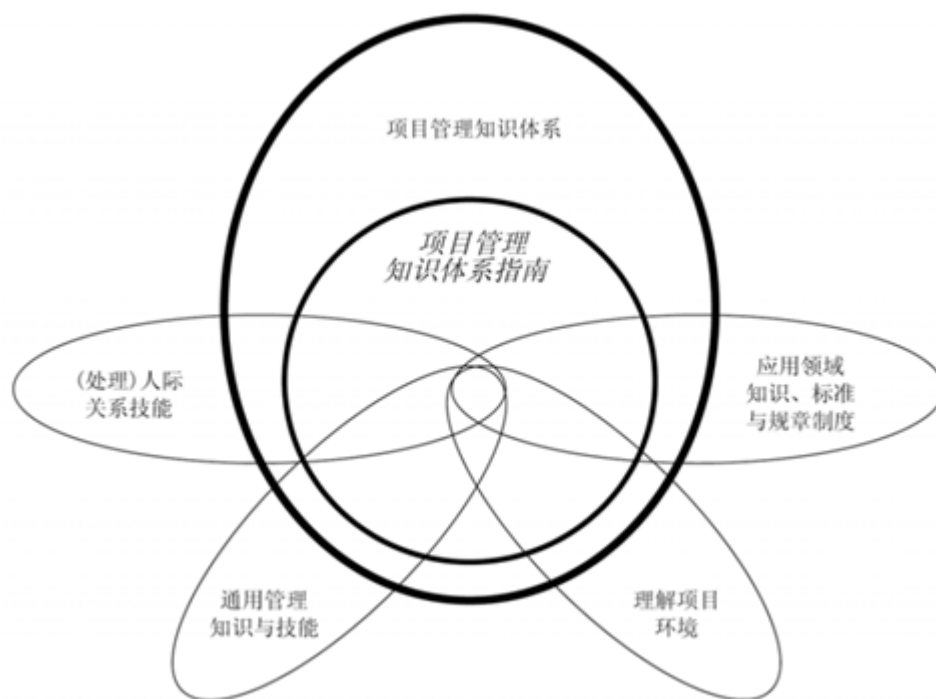


图 2 PMI 的项目经理知识技术体系

（2）麦克利兰的素质模型

美国心理学家麦克利兰经过研究提炼并形成了 21 项通用素质要项，并将 21 项素质要项划分为 6 个具体的素质族，同时依据每个素质族中对行为与绩效差异产生影响的显著程度划分为 2~5 项具体的素质。6 个素质族及其包含的具体素质如下：

- ①管理族，包括团队合作、培养人才、监控能力、领导能力等；
- ②认知族，包括演绎思维、归纳思维、专业知识与技能等；
- ③自我概念族，包括自信等；
- ④影响力族，包括影响力、关系建立等；
- ⑤目标与行动族，包括成就导向、主动性、信息收集等；
- ⑥帮助与服务族，包括人际理解力、客户服务等。

（3）管理者胜任特征模型

胜任力是指任何直接与工作绩效有关的个体特质、特点或技能等，在本质上也就是应该具备的素质组合。有学者利用物元分析和可拓评价方法建立了基于管理技能、个人特质和人际关系 3 个维度的胜任特征物元模型。

- ①管理技能的维度，包括团队领导、决策能力、信息寻求和市场意识等；
- ②个人特质的维度，包括影响力、自信、成就欲、主动性、分析思维和概括性思维等；
- ③人际关系的维度，包括人际洞察力、发展他人、关系建立、社会责任感和团队协作等。

（4）四种能力论

Robert hogan 和 Rodney B.Warrenfeltz 研究指出管理人员的素质可以分为 4 种，分别为：自我管理能力和人际关系能力、领导能力和商业能力。

- ①自我管理能力，包括自我尊重、正确对待权利的态度和自我控制等；
- ②人际关系能力，包括换位思考、正确预计他人的需要、考虑他人的行动等；
- ③领导能力，包括建立团队、维持团队、激励团队、建立共同愿景和巩固团队等；
- ④商业能力，包括制订计划、管理预算、绩效评估、成本管理和战略管理等。

3. 几种素质模型的分析

上面这些模型，都是被广泛认可的模型，我本人对四种能力论，更是情况独钟。为了找出一个适合项目经理学习修炼的模型，我们有必要对这几种模型进行评价。

首先确定评价的指标：

- （1）针对性：是否适合于项目管理领域；
- （2）完整性：是否太过宽泛或狭窄；
- （3）实用性：是否适合于项目经理修炼。

表 2 几种素质模型的评价

模型	针对性	完整性	实用性
PMI 的项目经理知识技术体系	好	太小	好

麦克利兰的素质模型	较差	太宽	较差
管理者胜任特征模型	中	太宽	较差
四种能力论	中	太宽	中

那我们能不能找到一种这三个指标都吻合的模型呢？

4. 西西吹雪的六种能力模型

“六种能力模型”力图在针对性、完整性和实用性方面达到最佳。六种能力分别是：知识、技能、逻辑思维、执行力、心智成熟和领导力。这六种能力是一个有机的整体，如下图所示：

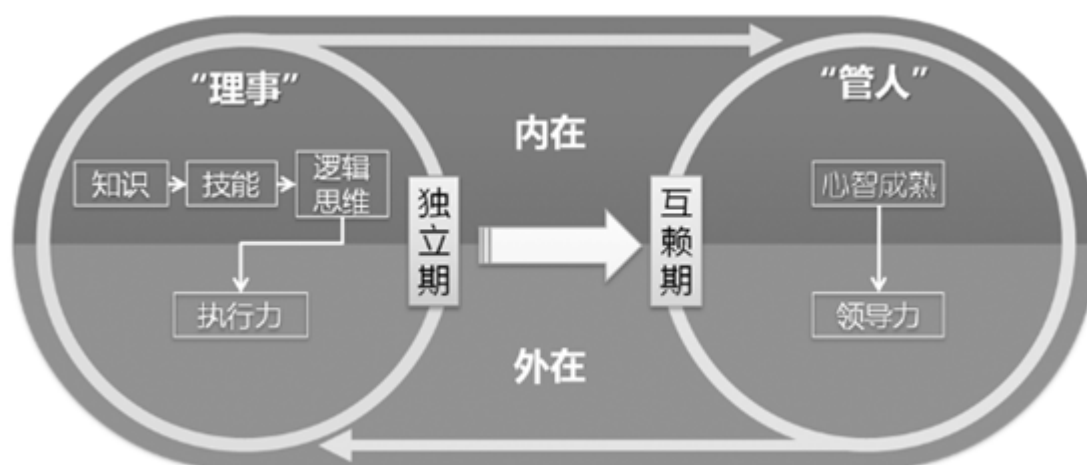


图 3 项目经理的六种能力模型

（1）人、事结合

管理，就是管人理事，这个理念已经深入人心。这个模型首先就是一个管人理事的素质模型。

从“理事”的角度来讲看，项目经理应当具备四大素质：

- 知识

必须具备项目管理的理论知识，所处的行业知识， 以及专业知识；

- 技能

光有知识是不够的，还要能知道怎么做。主要有项目管理技能、沟通表达技能、写作技能、专业技能等。

- 逻辑思维

项目经理必须具有较强的逻辑能力、思维清晰，对项目任务和要做的工作，随时都有清晰的分类和列表。逻辑思维能力有很多种，如果要挑出两种对项目经理最重要，我觉得是归纳能力、判断力。

- 执行力

项目经理本人必须具有很强的执行力。如果项目经理像个蔫老头，整个项目的执行结果可想而知。

从“管人”的解度来讲，项目经理应当具备两大素质：

- 心智成熟

要管人，首先必须学会与人相处，心智不成熟的人，与人相处往往会无所适从。心智成熟，也就是要管好自己的内心。自己都管不好，怎么管别人呢？

- 领导力

项目不是一个人的战斗，有些项目经理，只顾自己埋头干活，乐不滋滋，下面的同事却不知道每人要做什么，这是缺乏领导力的表现。余世维说：“管理就是让别人完成事情”，“真正厉害的人不是自己累死，而是要让手下做事情累死，这个才叫本事”，“优秀的管理者不会让员工觉得他在管人”。这三句话，可以说是领导力的三种境界。

简而言之，项目经理就像一个贤妻良母，要上得厅堂，下得厨房。上得厅堂意味着，项目经理要擅长与人打交道，也就是“管人”的要求。下得厨房则意味着项目经理懂技术、懂业务，能把复杂的事情理清楚，并解决各种问题，这就是“理事”的要求。理事主要靠智商，而管人则主要靠情商。

（2）内、外兼修

这个模型还是一个内外兼修的模型。古人云：“胜人者力，自胜者强”，说的其实就是一个人的外在修养与内在修养的关系。

战胜外在的事物，你需要是“力”，因此模型也有两个力：执行力和领导力。有这两种力，我们可以在管人、理事都做得很好。

要战胜自己，则非要靠一个人的内在修养不可。因此模型中，有四项个人内在素质的修炼：知识、技能、逻辑思维和心智。

从表面上看，“自胜”似乎比“胜人”更牛一些。但是从一个人成长的角度来看，我们主张要先“自胜”，再“胜人”。如果以树类比，“自胜”是根，“胜人”则旧枝干，一棵没有发达根系的树，是不可能长成参天大树的。所以不要让自己一开始就显得很牛，而是首先让自己成为一个真正的牛人，否则大树会过早夭折。

（3）从独立到互赖

一个人有成长过程可以分为三个阶段：依赖期、独立期和互赖期。每到一个新的阶段，都是一次巨大的飞跃。

- 依赖期：围绕着“你”这个观念——你照顾我；你为我的成败得失负责；事情若有差错，我便怪罪于你。
- 独立期：着眼于“我”的观念——我可以自立；我为自己负责；我可以自由选择。
- 互赖期：从“我们”的观念出发——我们可以自主、合作、统合综效，共创伟大前程。

也许你已经注意到了，在素质模型里面没有依赖期，这是因为在依赖期的人是无论如何也成不了项目经理的。这个模型，是一个从独立期走向互赖期的素质模型。

在独立期，我们主要擅长做“理事”的工作。我们是技术英雄，可以把每件事都做得很完美；

在互赖期，我们的精力转向了“管人”。我们懂得如何与各种不同类型的人相处，如果驱动团队为一个共同的目标而努力。

（4）层次分明

这个模型是还是一个层次分明的、渐进的模型。从知识到执行力，实际上是一个从“知道”到“去做”的过程，而从心智成熟到领导力，是发挥团队力量的两个阶段。

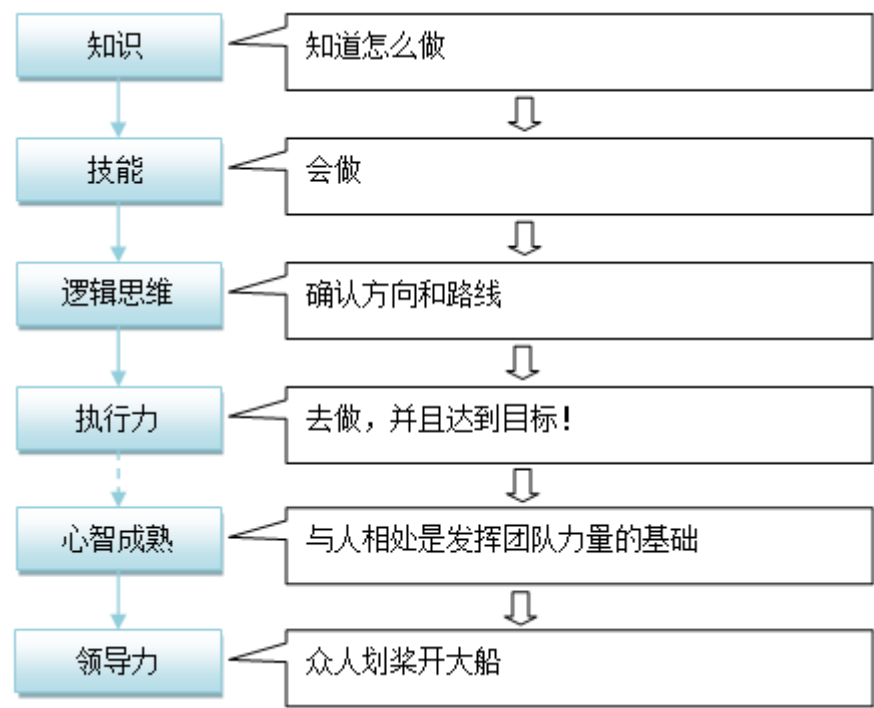


图 3 六种能力的层次

从程序员到项目经理（四）：外行可以领导内行吗

来源：[西西吹雪](#)

一. 从几个招聘要求说起

在[上一篇](#)中，我举出了一个招聘需求，引起一些朋友的争论。既然招聘的是项目经理，为什么需要那么多专业技能呢？

在百度上招聘频道搜索“软件项目经理招聘”，可以查到 8500 多条类似的招聘信息。我们看看国内软件行业老大东软集团的招聘条件：

工作职责：

带领团队完成需求分析，开发计划制定与跟踪，项目组关键技术问题解决，负责项目 QCD。

岗位要求：

- 1、3 年以上软件开发项目经验，2 年以上项目管理经验；
- 2、熟练掌握 JAVA、WEB 开发，精通基于 Oracle/Mysql 数据库的 MIS 系统开发；
- 3、具有较强的沟通、组织能力和较好的文字表达、写作能力；
- 5、有医疗业务开发经验者优先。

显然，东软公司也是要求具有较强的专业技能的。当然，也许东软公司太大了，不具有代表性，那么我们再看一个比较小的公司，你绝对没听过(我也没听过)，广东广风隆电子科技有限公司：

任职要求：

1. 能很好的把握开发质量和项目进度，规避风险。
2. 具有较强的语言和文字表达能力、沟通协调能力、良好的团队合作精神。
3. 具备至少 3 年项目管理经验或大型系统开发实施经验的优先。
4. 掌握 JAVA 技术，能熟练应用 J2EE, Spring, Struts, Hibernate 等开发和测试。
5. 熟悉基于 java 的 B/S 架构应用技术。
6. 熟悉基于 Tomcat、WebSphere、weblogic 等应用服务器的开发；
7. 熟悉大型数据 Oracle/SQL Server 等，熟练掌握存储过程编写、数据库表设计。
8. 熟悉 unix/Linux 操作系统。
9. 具备软件团队管理经验，熟悉软件开发流程，能够独立完成项目实施的优先。
10. 具备一定的系统[框架设计](#)、熟悉开发流程，具有的良好需求分析、项目设计、规划能力。
13. 有如下经验者优先考虑：
 - a.熟悉 BIEE，或有 BI 项目开发实施经验
 - b.对 BI/DW 的概念和架构有比较深入的了解，熟悉维度模型架构
 - c.熟悉 Oracle 数据库开发，或有 ETL 工具 ODI 经验，精通 SQL
 - d.有基于 java 技术项目管理经验的优先，教育行业背景优先

哇啦啦，这个更不得了。这究竟是招[程序员](#)还是招项目经理，我也快被弄迷糊了。看来中小公司比大公司更看重专业技能。

当然，我再多举一千条也代表不了所有的企业。但诸位如果有时间一条条看，会发现绝大部分公司对“软件项目经理”这个职位，都对专业技能有较高的要求。那么，传说中的“外行领导内行”究竟是不是真的？外行真的可以领导内行，带领项目走向成功吗？

二. 外行 vs 内行

1. 优势劣势分析

外行和内行究竟谁更适合当项目经理？那些招聘要求似乎已经为我们给出了答案，最少在软件行业内行项目经理更占据优势。然而，外行的项目经理往往也有其独特的优势，比如，他们往往更有大局观，能跳出技术本身看待问题，有更强的领导力等等。事实上，外行领导内行的现象，在国家大型建设工程或科研项目中要屡见不鲜。据说，我国的原子弹工程就是聂荣臻元帅领导的，而聂帅是不懂核物理的。

如果拿外行和内行项目经理来 PK，并不是一件容易的事情，因为每一项都不是绝对的，这就如同比较男人和女人谁更适合做厨师一样。当我们拿两者 PK 的时候，其实包含了一些隐含的信息，就是这个外行的项目经理比内行项目经理，更加懂得管理、情商更高，否则的话，内行项目经理会毫无悬念的胜出，也就没有比较的必要了。

基于这些隐含的信息，我们试着比较一下两种项目经理的优秀和劣势：

项目经理类型	优势	劣势
外行	一般具有更强的领导力，更善于激发员工的士气、战斗力； 一般具有更强的谈判能力、资源协调能力，客户和上级领导满意度会更高； 一般具有更强的沟通汇报的能力； 更容易跳出技术本身，看清问题本质； 一般更善于权衡轻重缓急，更善于取舍。	项目详细计划要依靠技术骨干，对其评估的准确性无法做出自己的判断； 无法对技术人员进行辅导； 无法对技术问题做出分析判断，帮助解决棘手问题； 无法对技术人员进度拖延原因做出准确分析、不能很好控制项目； 容易造成瞎指挥； 容易和技术人员互相看不起对方。
内行	外行的劣势往往是内行的优势	外行的优势往往是内行的劣势

2. 技术决定论的误区

所谓内行与外行是纯粹从技术的角度来看问题，单纯讨论内行好还是外行好，其实也暗含着一个前提，就是技术决定项目的成败。而实际上，一个项目能否成功的影响因素，远不止是技术，对一个项目经理的素质要求也远不止技术。同是外行或内行来带一个项目，会由于个人修养与经验在差异，项目结果可能相差很远。因此单纯说外行好，还是内行好，是没有意义的。

3. 综合素质决定论

问题的关键其实不在项目经理是内行还是外行，而在于他的综合素质。无论是外行还是内行，只要谁的综合素质更高，谁就是更优秀的项目经理。

上一篇我们讲到项目经理的六种能力模型，也就是说，一个优秀的项目经理，应当具备六个方面的素质，即：知识、技能、逻辑思维、执行力、心智成熟和领导力。

在知识层面，包括专业知识、行业知识和管理知识。外行项目经理在专业知识和行业知识方面已经输了，但在管理知识方面按默认值，外行赢了。

在技能层面，包括专业技能和管理技能。外行项目经理在专业技能也又输了，同样管理技能方面，又略胜一筹。

现在打成了平手。剩下的，要拼逻辑思维、拼执行力、拼心智、拼领导力，这就和内行外行无关了，鹿死谁手，要看个人的修养。

因此，项目经理的比拼，拼的不只是管理知识或专业知识这一个方面，而是综合素质的比拼。

三. 外行，你凭什么

1. 唐僧的团队

外行，也就是不懂专业知识技术，显然不但不是什么优点，反而是一个项目经理的极大缺陷。那为什么领导还会置这么大的缺陷于不顾，任命一个外行为项目经理呢？换一个角度，也就是说，一个外行，在什么情况下，可以成功的管理一个软件项目呢？

一件事情的发生，总有他的内部原因和外部原因。具体到这个问题上，也有它的内因和外因。

（1）在内部因素上，外行项目经理必须具有更高的综合素质。

现在流行分析西游记中的取经团队，其实也是一个典型的外行领导内行的团队。到西天取经，靠的是降妖服魔的本领，显然唐僧是个外行。但是，唐僧并不是一无是处，相反，他的综合素质很高。他外柔内刚，意志坚定，目标明确，还精研佛法，具有很强的人格魅力，因此他的那些徒弟才能凝聚在他周围，虽历尽千难万险而无悔。

（2）在外部因素上，必须有合理的人才结构作为支撑。

唐僧虽然不会打怪，但是孙悟空可以，补齐了唐僧在这方面的不足。试想，如果他的徒弟都不能降妖，任凭唐僧的领导能力再强，也注定最终只会被妖怪吃掉。同样一个外行的项目经理，在他的团队中，必须可以信赖的技术骨干，像孙悟空一样能在关键时候解决问题，这些骨干一般就是项目中的组长、系统架构师或者系统分析师，必要时可能要设置项目副经理之职。如果团队中没有技术骨干，都是一些经验不足还不求进取的程序员，那除非项目超级简单，否则项目经理纵然有诸葛亮的才华，也无济于事。

2. 规模决定一切

在上面两项条件都具备的情况下，只能说明外行可以担任项目经理了。站在项目本身的角度，除了这两项因素，往往还跟以下方面有着紧密的关系。

（1）项目规模：规模越大，采用外行项目经理的机率越高。

（2）项目所在行业：在建筑、施工、水利等传统行业，采用外行项目经理的机率更高。

（3）项目的技术难度：在项目规模不大时，如果技术难度越大，采用内行项目经理风险更小。

（4）项目进度要求：时间要求越紧，更倾向于采用内行项目经理。

（5）项目管理的层次：有些项目层层分包，对于上面次层的公司，项目不需自己实施，只需对项目进行监管，项目经理自然也不需要很专业了。但对于底层实施单位而言，项目经理懂技术就很有必要了。同样，有些大型项目分成若干个工程，每个工程又包括若干个子项目，也是类似的情况。

在这些因素中，项目规模是具有决定性的因素。项目规模足够大的时候，也就有足够的经费来配备充分的人才。至于其实方面，其实只是表现而已。

三. 透过瓶子看软件行业

为什么软件行业外行项目经理相对较少呢？这与软件项目本身的特殊性有一定关系，但在一定程度上也折射出软件行业的现状：

（1）软件项目规模不够大

在软件行业，几十万的项目很常见，几百万上千万就是大项目了，项目的利润率很低，很多中小型企业都生存在赢利的边缘。据工信部统计，2011年上半年我国软件行业利润仅占软件业务收入的1.28%。这么低的利润率，估计比东莞的制鞋厂还不如吧。而几百万上千万的金额，对建设、国防这些行业来说，简直不值一提啊。前几天太极集团1.99亿中标铁道部IT项目，大家都不服气。也是，人人都在喝汤，你凭什么搞特权吃肉？

（2）成熟的项目经理相对紧缺

软件行业小项目太多，对项目经理的需求量是非常大的，与此同时，成熟的项目经理相对很少。所谓“千军易得，一将难求”啊。当然，即使牛B的项目经理有了，其收入要求也不会低，这是小型项目难以承受的，只能退而求其次，找一个性价比更高的项目经理，或者干脆拔苗助长，找一个不错的程序员来带吧。

从程序员到项目经理（5）：程序员加油站，不是人人都懂的学习要点

来源：[西西吹雪](#)

学习是一种基础性的能力。然而，“吾生也有涯，而知也无涯。”，如果学习不注意方法，则会“以有涯随无涯，殆矣”。

一．学习也是一种能力

看到这个标题，有人会说：“学习，谁不会？”的确，学习就像吃饭睡觉一样，是人的一种本能，人人都有学习的能力。我们在刚出生的时候，什么也不知道，是一张真正的白纸，我们靠学习的本能，学会了走路、说话、穿衣服...后来，我们上学了，老师把书本上的知识一点一点灌输到我们的脑子里，我们掌握的知识越来越多，与此同时，我们学习能力却好像越来越差了，习惯了被别人喂饱，似乎忘记了怎么来喂自己了。

学习本来只是一种本能，算不上什么能力，然而，经过二十多年的不断学习，学习反而成为了一种真正的能力，因为我们慢慢失去了它，它就更显得珍贵。

在学校里我们基本上被动式学习，然而走出了象牙塔之后，不会再有人对你负责，不会有人主动教你，我们需要主动的学习。所谓的学习能力，其实就是自主学习的能力。

几年前，曾有一本风靡管理界的书，叫《[第五项修炼](#)》，这本书倡导建立学习型组织，因为从长远来看，一个组织唯一可持续的竞争优势，就是比竞争对手更快更好的学习能力。

一个公司如此，一个人又何尝不是如此？众所周知现在是一个知识爆炸的时代，知识更新非常快。据说，一个大学毕业生所学习到的知识，在毕业之后的 2 年内，有效的不过剩下 5%，更何况我们的学校与社会需要严重脱轨。**我们赖以立足的，不在于我们现在掌握了多少知识，而是我们有多强的学习能力！**

学习不但是能力，而且是一种至关重要的能力，而这种能力的核心，就是学习的方法和心态。

二．买书是最划算的投资

古人云：“书中自有黄金屋，书中自有颜如玉。”这说明先贤们早就认识到，买书是最划算的投资了。

当我刚出道的时候，拿着非常微薄的工资，有一次我向主管抱怨道：“现在的书真贵啊，这点工资连饭都吃不起，更别说买书了！”主管对我说：“不要吝惜买书的钱，宁可忍着不吃饭，也不要忍着不买书，因为买书是回报率的最高的投资了。”

主管的话让我非常震动。后来，我看到喜欢的书时，再有没有手软过。我不断的学习，开发能力也不断的提高，工资水平也获得了大幅度的提高。一年后，我一个月工资的涨幅，就足够买两年的书了。你说，还有比这更划算的投资吗？

一本书，哪怕只有一页纸是有用的，它将所产生的潜在价值，也会远远超过书本身的价格。当然，书不在多，能踏踏实实消化掉一本好书，可能比泛泛而读 10 本普通书，要更有价值得多。

三. 多读经典书

十年前，我刚进入IT行业的时候，真是求知渴，每星期都要往购书中心跑，可惜的是，那时给[程序员](#)看的书不像现在这么多，高质量的书就更少了。当时我印象中比较经典的书籍就是

《[Windows程序设计](#)》、《COM本质论》、《[Java编程思想](#)》，还有就是谭浩强的《C语言程序设计》。其它充斥书架的，就是类似于《21 天精通XXX》、《XXX从入门到精通》、《XX宝典》这样的书籍。

回首往昔，令我比较郁闷的一件事就是在我最有学习动力的时候，看的高质量书籍太少，就好像是在长身体的时候，天天吃的是没营养的泡面。当然，这跟没有人指导也有很大的关系，独自一个人学习，让我走了很多的弯路。

软件开发方面的书籍，我大致将其分为三类：

（1）浅显的入门类书籍。

这类书的标题往往是《XX 天精通 XXX》、《XXX 从入门到精通》、《XX 开发实战》等，这类书往往从软件的安装讲起，喜欢翻译帮助文件。有人批评这类书为烂书、毫无价值，这并不公平。至少我本人，也曾从这些书中学到一些东西。即使是 21 天系列书，也有适合看的人群，只不过，它一般也就只能看 21 天而已，过后就可以扔到垃圾堆。这类书只适于还没有入门的初学者，从中学到一些入门的招式。这种书在刚起步的时候一般买上一本就可以了。如果你善于使用搜索引擎，这一本书也可以省了。

（2）国内外高手写的实战类书籍。

这类书实战性很强，把技术及原理讲得很透彻。比如《windows 环境下 32 位汇编语言程序设计》、《深入解析 MFC》、《Delphi 深度探索》、《深入浅出 WPF》、《深入剖析 Asp.net 组件设计》等。以前这类书都是从国外翻译或从台湾引进，现在国内高手越来越多，出自国内作者的也越来越多。这类书如果你学习的每个方向看个两本，并且通过实践消化掉，那么毫无疑问，你会成为一个优秀的程序员。

（3）国外大牛写的、揭露本质、有丰富思想的书。

这类书就是所谓的经典书了，例如《[代码大全](#)》、《[编程珠玑](#)》、《[设计模式](#)》、《[重构](#)》、《[代码整洁之道](#)》等。经典书就像一个有深度、有思想的朋友，他会给你启发、每次阅读都会有新的收获，这类书具有真正的收藏价值。看经典书永远是正确的选择，它绝不会浪费你的时间，因为经典书是无数人沙里淘金、帮你挑选过的结果。

然而，阅读这类书并不是一件容易的事情，读者需要有丰富的开发经验，才能与作者产生共鸣。真正能消化经典书的人其实不多，这就好像饮酒，一个新手无论如何也品不出葡萄美酒的醇香。在酒桌上，人人都把杯中酒一饮而尽，当有人点评“这个酒不错”的时候，我只能无奈的苦笑一番，真的是甘苦自知。

如果一本经典书你看得很辛苦，很有可能就是因为你功力未够，这种情况下不要着急，慢点来，不妨先将其先束之高阁，多看看第二类实战型书籍，过一段时间再回头来看，也许你会有新的惊喜。

（[伯乐在线补充推荐](#)：→ [《国外程序员推荐：每个程序员都应读的书》](#)）

四、不要在上班时间看书

一个善于学习的人，首先要善于利用一切时间来学习。不知是伟大的雷锋叔叔还是鲁迅爷爷曾经说过：“时间就像海绵里的水，只要愿挤，总还是有的。”然而，当我们从上班时间内挤时间学习时，就千万要注意了，不要在上班时间看书！

上班时间看书不但不是一件很敏感的事情，而且非常吸引眼球，很快会引起周遭的不爽。首先老板心里不爽，他想：“我给你钱是让你来工作的，不是来学习的！”；其次同事们也不爽：“我们工作都做不完，瞧，这小子真闲哪！”用不了多久，你就会成为被众人排斥的异类。

当然，你可能会说，“我工作已经做完了，经理没有安排，当然可以学习了”，其实不然。你完成了一件事情，不等于所有的事情都完成了。一个优秀的员工，应该是主动要工作，而不是被动的等工作。工作完成以后，你至少还可以：

- （1）主动汇报给你的经理，请他来检查你的成果，并安排新的任务；
- （2）如果公司这一段时间确实比较闲，没有什么具体的任务，可以进行代码重构、优化；
- （3）你还可以主动请缨，承担额外的工作或更艰巨的任务。
- （4）如果一定要学习，也只能对着电脑屏幕来学习，纸质书最多只能拿来翻阅一下，而不能一直捧着，以免影响到其他人的情绪。

五、只学习与工作相关的东西

我曾发现不少程序员在学习方面找不到方向，一会学学C#，一会学学Java，看了最新的[编程语言](#)排行榜，又觉得该学C++。这样左抓抓，右挠挠，只会让你觉得更痒。

学习最忌三心二意。俗话说：“伤其十指不如断其一指”，每门都学一点，还不如专心学好一个方向。这个道理谁都懂，可是又该学哪个方向呢？难道只能跟着感觉走吗？

不！最实际的方向，应该跟着工作走，工作需要什么，我们就学什么，把工作需要的技能熟练掌握。我们为什么要学习和工作弱相关的东西呢？是为了转行或跳槽吗？可是，如果我们连现在本职工作都不能做好，又怎么能保证到新的岗位、用新学的技能就可以做得更好呢？

学习与工作需要的东西，有很多好处：

首先，可以集中精力，在某一方面钻研得更加深入。所谓“百招会不如一招绝”，有了绝招，你恐怕不能在“武林”立足吗？《天龙八部》中的慕容复武功博学无比，最后还不是被只会一招六脉神剑的段誉打得落花流水？

其次，可以学得更快、更深入，因为学习更具有针对性，而且可以立即在工作中运用，可以马上检验出学习的效果，对存在的问题可以进行深入的研究，因此掌握的知识也会更加的牢固。

第三，学习与工作结合在一起，工作时间也就成了学习时间，这样突破了三个 8 小时的限制。有人说，我们每天所有拥有的时间可以分为三个 8 小时，工作 8 小时，睡觉 8 小时，另外还有 8 小时自己可以自由支配的时间。工作和睡觉的两个 8 小时大家都一样，决定人生高度的是另外这个 8 小时。当我们把学习的焦点放到与工作相关的知识上时，工作时间中的很大一部分，同时也就成了宝贵的学习时间，这真是一举两得的美事啊。

六．织网式的学习

知识的广度和深度都很重要。作为一个程序员，深入把握技术细节，是写出优质代码的保证。但对于一个项目经理而言，知识的广度更显重要。项目中碰到的问题往往是综合性的，只有具有广博的知识，才能快速的对问题进行分析和定位。在程序员通往项目经理的道路上，我们必须有意识的扩大自己的知识面，形成更完善的知识体系。

每个人的知识体系就好比是一张网，我们学习其实就是要织这样一张网。我曾看过渔网的编织过程，渔网虽大，也是一个结点起步，一个点一个点的编出来的，编织的过程中，始终只有一根主线。

学习又何尝不是这样，知识体系的大网也是由许多小的结点组成，要结这样一张网，只能由一个点起步。牵住一条主线，织出一个个的点，由点带出面，最后才能形成这张大网。

我曾经编写过一个网络信息采集软件，这个软件可以从具有列表页网站中按字段设置采集信息，支持自定义字段、页面多级关联、下载附件、支持多种数据库、可视化定义等特性。刚开始时，觉得这个软件也是一个比较大的功能点而已，后来发现这个不起眼的功能关联着大量的知识点，在开发过程中，我顺藤摸瓜，各个击破，对很多知识点进行了细致的学习研究，软件开发完成后，个人的知识体系网也进一步得到了补充和完善。

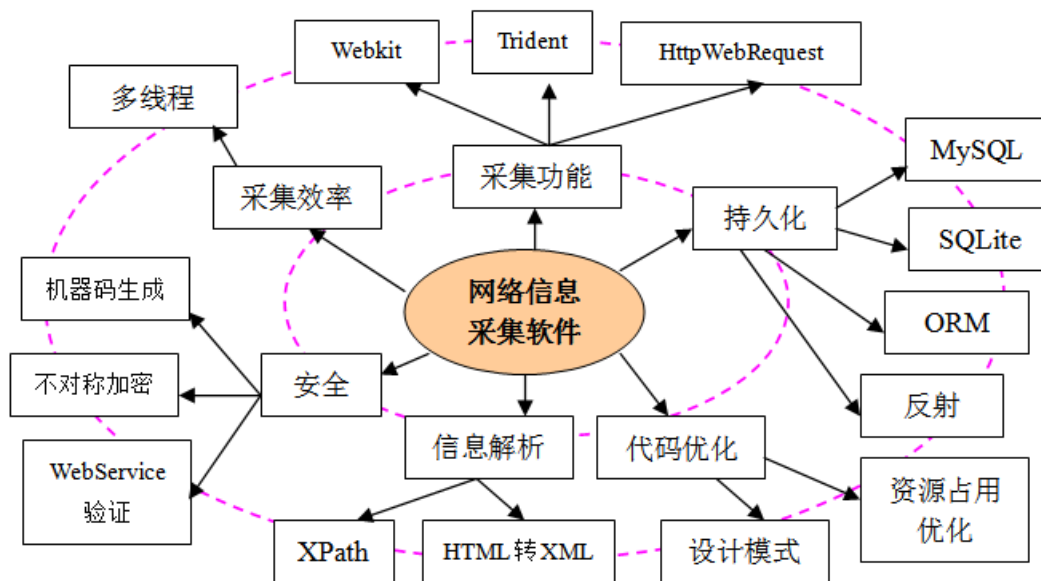


图 1 由知识点形成知识网

七. 问题是最好的学习机会

日本经营之神松下幸之助曾经说过：“工作就是不断发现问题、分析问题、最终解决问题的一个过程，晋升之门将永远为那些随时解决问题的人敞开着。”可见，工作过程中有问题是正常，没有问题那才是真正的问题。在发生问题能时，能勇于面对问题、解决问题的人，才是公司真正的核心骨干。

现实中，很多人总是千方百计回避问题，当上司安排一项艰巨的任务时，也是想尽办法推托。殊不知，对于个人而言，其实问题是最好的学习机会。往往那些愿意接受困难工作的人，能力会变得越来越强，那就是因为他们在克服困难的过程中取得了巨大的进步。

有一次，一位项目经理对我说：“有一个问题，客户有一台 HP 服务器要装磁盘阵列，没人会做，怎么办啊？”“可以学啊，没有人愿意去吗？”“我都问了，没人想去。”“哦，正好明天我有时间，我也没装过磁盘阵列，那我明天去学着弄一下。”我说的是真心话。第二天早上，当我准备出发时，项目经理告诉我不用我去了，因为项目组好几个同事都想去“学着弄一下”。

结果服务器很快就装好了，远远没有之前大家想像的那么困难嘛。更重要的是，在解决这个问题过程中，大家都学会了怎么装磁盘阵列。

碰到困难时，迎难而上吧，千万不要拒绝这个最好的学习机会！

八. 经常思考总结

子曰：“学而不思则罔”。只学习不思考，就会迷惑，难以把握事情的本质。这就好比一个学武之人，只习得其形，而未得其神，难以成为真正的高手。

一个程序员从入门，到成为高手的过程中，往往要经过几次顿悟。顿悟会让你跳出知识的丛林，一切豁然开朗，仿佛打通了全身的奇经八脉一般奇妙。记得我有一次，顿悟到了一个很简单的结论：“原来高级编程语言中的类库是封装了 Windows API 来实现的。”后来碰到一些自带类库无法实现的功能时，我就会想到，其实可以通过调用 Windows API 来实现。利用这个思路，我解决了一些看起来很难的问题，得到老板的赏识，从而很快获得提升。

顿悟非常可贵，然而它不是随便发生的，而是经过一次次苦苦思索之后、灵光闪现的结果。思考的过程，其实就是将外在的知识内化为自己的知识的过程，而顿悟，则是批量的实现这种内化，将无数个知识点连接在一起，达到融会贯通的境界。

九、克服“高原现象”

爱学习的人都会有这样的经历，学习持续了一段时间之后，往往会有一个瓶颈期，长时间似乎很久没有什么进步，于是内心非常着急。

这种情况实际上这是由人的学习规律决定的一种“高原现象”。据研究，学习者在刚开始进步快，随后有一个明显的或长或短的进步停顿期，后期进步慢，中间的停顿期叫高原期。

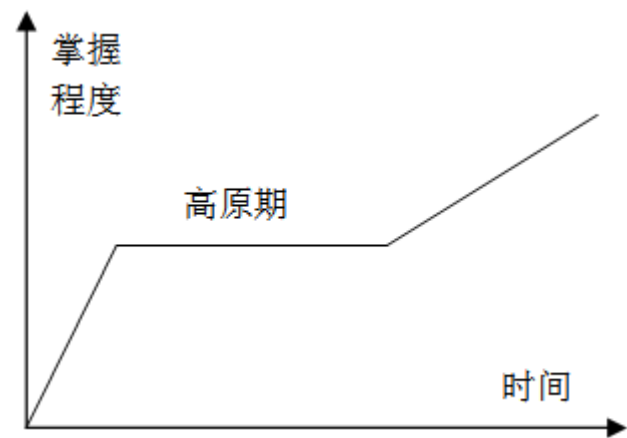


图 2 技能学习练习曲线

在我看来，高原期实质是一个消化期，由于前期的学习积累了太多的知识点，这些知识点在大脑中乱作一团，还没有形成一个知识体系。这时需要一定的时间来消化它，将它融会贯通，经常思考总结可以快速帮你跨过高原期。

在处于高原期的时候，还可以换一个相关的方向来学习，例如编程语言学不下去了，你可以学习一下设计模式，设计模式也学不下去了，再换成数据库。通过学习这些相关的知识，不但补齐了知识体系中的短板，而且各个知识点之间可以互相启发，帮助你实现顿悟，跨过高原期。

十、学习要有好心态

（1）学习要静心

急于求成是学习过程中普遍存在的一种心态。这可以理解，毕竟作为一个程序员，要学的东西实在太多了，而社会又是那样的浮躁，让人觉得一切都是那样的不安全、不确定，似乎只有学得快一点，才能跟上社会的脚步。

可是“欲速则不达”，想快快的学，往往会形成东一榔头、西一棒槌的学习方式，每一个点都没有吃透。心沉不下去，知识也会沉不下去。要想成为真正的高手，只能静下心来，一步一个脚印的攀登。

（2）学习是一个持续一生的过程

人生的过程，就是一个自我完善过程。

孔子曾经说：“吾十有五而志于学，三十而立，四十而不惑，五十而知天命，六十而耳顺，七十而从心所欲，不逾矩。”可见孔子也不是天生的圣人，也在不停的学习、进步，从“志于学”到最后“从心所欲，不逾矩”，孔子一共花了 55 年的时间。

作为一个程序员，更是需要不断更新自己的知识。我们所知道的东西，就像一个白色的圆圈，圈外则是黑暗的未知的世界。当圆圈越大，所接触到的黑暗部分就越多。我们只有不停的学习，打破更多的黑暗，找到更多光明。

（3）保持饥饿，保持愚蠢

看了《乔布斯传》之后，我最喜欢的一句话是“求知若饥，虚心若愚”（Stay Hungry, Stay Foolish），其实我更喜欢它更原生态的翻译“保持饥饿，保持愚蠢”。我们只有认识到自己还很饥饿和愚蠢，才会像没吃饱一样，由衷的需要学习、爱上学习。

当然，知易行难，知行合一才是学习的最高境界。我也始终是一个学习者，一直在路上。

从程序员到项目经理（6）：程序员加油站 — 懂电脑更要懂人脑

来源：[西门吹雪](#)

说起[程序员](#)三个字，我觉得既骄傲又可悲。骄傲的是，我们曾经是时代骄子，是一群真正改变世界的人；可悲的是，我们很多致力于改变世界的程序员，却生活在自己的世界里，无法自拔，成为了继“书呆子”之后的“电脑呆子”。电脑本来只是一个工具，我们竟然被其所限制、甚至同化，悲夫！

一、警惕成为“电脑呆子”

（1）程序员眼中的自己

程序员是怎样看待自己的呢？看看园子里的发言，码农、码畜、IT 民工、苦逼、程序猿...这样的字眼俯拾皆是。

在网上曾经广泛流传一首关于程序员的诗，模仿的是唐伯虎的《桃花庵歌》，我们暂且称之为《程序员之歌》吧：

写字楼里写字间，写字间里程序员；程序人员写程序，又拿程序换酒钱。

酒醒只在网上坐，酒醉还来网下眠；酒醉酒醒日复日，网上网下年复年。

但愿老死电脑间，不愿鞠躬老板前；奔驰宝马贵者趣，公交自行程序员。

别人笑我忒疯癫，我笑自己命太贱；不见满街漂亮妹，哪个归得程序员。

这首诗的作者不知姓甚名谁，但可以肯定的是，他是一名程序员，因为只有程序员才能这样生动的写出程序员的酸甜苦辣。从诗中看出程序员眼中的自己的形象：敬业、辛苦，每天的时间全部花在写程序和上网；思想单纯；清高不合群，自傲自恋；清贫不得志，自悲自叹。

（2）别人眼中的程序员

在别人眼中程序员又是怎样的一个群体呢？在 360 网站有一个关于程序员形象的热帖，其中回帖的大部分都不是程序员，很多回复都非常生动，没有骂街，可以说比较客观。

总结一下，大家回复的情况大致如下：

工作方面

富有钻研精神，是技术方面的高手，没有时间概念，加班多，辛苦劳累，工作认真严谨，懂制作软件却不懂这软件如何运行更好。

外在形象

黑眼圈，红眼圈，睡眠不足，瘦小，邋遢，带眼镜。

生活方面

电脑前潇洒自如，世人前胆小腼腆。聪明，思维敏捷，生活刻板。

性格方面

“闷骚”这个词不好听，但还是蛮准确的：程序员大多沉默寡言，不善与人交往，但内心却很丰富。性格腼腆甚至孤僻，圈子小，爱憎分明，有点不食人间烟火的样子。

思维方式

是一种面向问题的思维方式，逻辑灵敏而严谨，无时无刻不在思考攻克解决问题，善于找别人的问题，却对自己的问题视而不见，不善于解决生活中的问题。

综合起来，程序员在世人眼中大抵是一个聪明而又迂腐、善良而又刻板的形象，是不是有点像鲁迅笔下的“孔乙己”先生呢？

（3）“电脑呆子”是怎样炼成的

上面描述让我想起了一个词：“书呆子”。书呆子是指那些死读书、读死书、不通人情世故、不会用书上的知识变通的人。书呆子是与书待一起的时间太久了，以至于生活在书的世界里，用书里的道理来评价和要求真实的世界。而我们程序员呢，日复一日，年复一年在代码间摸爬滚打。每天用在与电脑交流的时间，比大部分书呆子看书的时间有过之无不及。每天基本上就是“电脑一开，一关就过去了，嚎”。

俗话说：“带着锤子三年，看什么都是钉子”。当程序员三年，看到谁都当作是电脑。于是产生了计算机时代的“书呆子”，不妨称之为“电脑呆子”。电脑呆子用电脑的时间太久了，生活在电脑的世界里，用电脑的逻辑来要求别人，不懂生活，不懂人情世故。可能你对这样的措辞感到不满，但对多我们身边有些程序员，是不是有几分神似呢？

悲夫！程序员曾是时代骄子，有非常细腻内心、非常丰富的感情世界、非常聪明的大脑，在世人眼里的形象却是如此不堪！

孔子说：“君子御物而不御于物”。电脑只是被我们利用工具而已，而我们的思维却被电脑所限制，甚至变得和电脑一样。

程序员，是该求变的时候了！

我们再也不要闷骚，将我们的内心美好善良的一面勇敢的表达出来吧！

我们再也不要苦逼，我们要金钱，更要快乐，我们要工作，更要生活！

我们再也不要死板，我们可以做出漂亮的程序，同样也可以漂漂亮亮的做人！

（4）一个老程序员的肺腑之言

也有大家会觉得“电脑呆子”这样的词是在骂程序员，是对程序员的不敬，但也许激烈的言辞更能令人警醒。有一个成语叫当头棒喝，据说佛教禅宗和尚接待初学的人常常用棒一击或大喝一声，促他醒悟。

我曾经是一个程序员，现在仍是。我也曾经是一个真正的“电脑呆子”，我曾独自在黑暗中摸索，花了多年的时间才摸着石头过河——也许我还远未过河。那些曾狠狠骂我的人，我把他们当作我的恩人，因为他们激励和启发了我的成长。当我逐渐走向成熟时，已经错过了无数的机会。

二、懂电脑更要成为人脑

（1）电脑逻辑 vs 人脑逻辑

程序员写代码离不开电脑，沟通、交际又要与人脑打交道，然而电脑与人脑的逻辑在很多方面却是大相径庭。

比较方面	电脑的逻辑	人脑的逻辑
差异性	同一个程序在每台电脑上的运行结果都一样	任务交给不同的人，结果可能大相径庭
多样性	每台电脑都一样（换一台电脑编程完全没问题）	每个人都不一样，人千差万别，因此要适应不同性格的人
确定性	程序正确，电脑一定能得到正确结果	任务明确，做出来的结果与预期可能相去甚远
思维	电脑无自主思维。	个人存在理解力，执行力，判断力等方面的问题
情感	电脑没有感情、情绪等因素的影响	人受感情、情绪的影响
自主性	电脑无自主性，完全受程序的控制	人具有自主性，但行为由很多因素决定
社会性	电脑与电脑之间只在严格的逻辑交互，无社会性	人与人之间的关系微妙
合作	1 台电脑+1 台电脑，运算能力更强	1 人+1 人，结果无法预知，团队合作至关重要

电脑的逻辑简单，所以我们愿意与电脑打交道。如果我们把电脑的逻辑带到与人交往的过程中，那就太“简单化”了，当然也就给人以迂腐、刻板、不懂变通的印象。我们毕竟是生活在人的世界中，我们要懂电脑，更要懂人脑。我们不是只懂电脑异类，而只是更懂电脑的正常人。

（2）做回正常人

我曾经很看不起那些不懂技术却八面玲珑的人，看到他们身居高位更是感到愤愤不平，甚至感叹要是生活在西方国家就好了，什么事情都直截了当，不用拐弯抹角。

然而，经历了无数的挫折之后，我明白了一个道理：“世事洞明皆学问，人情练达即文章”。人家能说会道、八面玲珑也是一种本事啊。不然，我们怎么做不到啊？

其实并不是这样做很难，而是我们不愿意这样做而已，不愿意为世俗的观念改变自己。没错，现实是世俗的，但现实也是无法改变的，我们只能承认现实，臣服于现实。我在 360 的那个帖子中看到有一个对程序员的绝妙评价，“程序员是七仙女中的织女”，难道我们真正的要像仙女一样不食人间烟火吗？

我们不用做仙女，只需要做一个普通的正常人。要顺应人的逻辑，懂人情，明事理，做一个正常人该做的事情，这样并不难。

莫言在领诺贝尔奖时有一段精彩的发言：

最后，我讲一个小故事。听说法兰克福是歌德的出生地。在中国，流传着一个非常有名的关于歌德的故事。有一次，歌德和贝多芬在路上并肩行走。突然，对面来了国王的仪仗。贝多芬昂首挺胸，从国王的仪仗队面前挺身而出。歌德退到路边，摘下帽子，在仪仗队面前恭敬肃立。我想，这个故事向我们传达的就是对贝多芬的尊敬和对歌德的蔑视。在年轻的时候，我也认为贝多芬了不起，歌德太不象话了。但随着年龄的增长，我慢慢意识到，**在某种意义上，像贝多芬那样做也许并不困难。但像歌德那样，退到路边，摘下帽子，尊重世俗，对着国王的仪仗恭恭敬敬地行礼反而需要巨大的勇气。**

处处与世俗为敌，并不会让世俗变得清高。尊重世俗，也并不意味着失去清高，失去自我。

不要比拼清高，而要自己生活得幸福。当你能自由的游走于世俗的现实与内心卓尔不群的原则之间时，你也就实现在个人修炼的圆满，成为了一个从内心里幸福的人。

我们不需要成为清高之人，也不需要成为世俗之人，我们只要成为普通的正常人，一个外圆内方的人。



从程序员到项目经理（7）：程序员加油站 — 完美主义也是一种错

来源：[西西吹雪](#)

追求完美是一种可贵的精神，完美主义也历来被认为是一种优秀的品格。可是在项目中，完美主义也是一种错，虽然是一种“美丽的错误”。项目讲求平衡，要的是合格，而不是优秀；要的是 70 分，而不是 100 分！

1. 两极分化的程序员

相信在很多人眼里，程序员都是工作一丝不苟、对代码精雕细琢、精益求精的人。瞧，他们在电脑前面一坐就是大半天，如果不是追求完美之人，谁能这样坐得住板凳？

可是依我所见，在“追求完美”这个问题上，程序员其实是严重的两极分化。有一部分程序员确实对自己的代码要求很高，他们在编程时，非常注意逻辑是否严谨、运行效率高不高、代码是不是优雅，经常进行代码[重构](#)与优化。他们就像有洁癖农村老太，整天扫把不离手，在哪里看到不顺眼的代码，就要改到哪里，如果让他来维护一个系统，多半最后会让他把整个系统的代码全部重构或者重写了一遍。他们是真正的完美主义者。

还有一部分人，他们写代码似乎只是为了完成任务。他们对自己负责的功能，缺乏计划和设计的过程，想到哪里就写到哪里，最后按下 F5，编译通过，欧耶！他们甚至不愿意多点一下自己创建的按钮，更加不会在一个身份证号码文本框中输入一个电话号码来测试一下，最关键的是终于可以向经理交差了，至于代码中多少隐藏的问题，以后再说吧。这让我想起了程序员部落酋长 Joel 所说的，他们编写的程序“看上去像是给狗吃的早餐，只经狗能吃饱就行了，何必再花钱让食物变得色香味俱全呢？”我们甚至可以想象一下，他们的电脑屏幕上是不是铺满了灰尘，键盘缝里是不是塞满了头发和食物碎屑，电脑桌面的图标是不是如七彩拼图一般，让人眼花缭乱。

后一类程序员，在数量上似乎占据绝对的优势，对于一个不擅于控制项目质量的项目经理来说，他们的代码最终会成为项目的噩梦。系统一旦投入运行，虫子就会像美国恐怖片中的外星生物一样，源源不断的从鼻孔、嘴巴和耳朵缝里冒出来。

第二种程序员这种低标准低要求、随随便便的做法，很容易被识别出来是一种不好的倾向，而完美主义不是这样，因为我们从小就被教导要追求完美，完美主义一般被认为是一种优秀的品格，是每个人应追求的目标。

然而完美主义虽然听上去不错，却并不适合于项目，因为项目的目标是用最少的成本来完成项目，让各方满意，而不是制造一个完美无瑕的产品，以证明个人或公司的能力。显然，完美主义更适合于个人能力的修炼，或者一项没有限期出成果的科学实验，在项目中，完美主义也是一种错，虽然是一种“美丽的错误”。

完美主义者和随随便便的人，这两种程序员都不是项目的最佳人选，他们是恰好是两个相反的极端，如果让他们负责项目，估计就像玩跷跷板一样，要么压到地底下，要么翘到天空上。但是项目经不起这样的折腾，项目中需要有平衡能力的人，他们很好的把握追求完美的“度”，使得软件

功能既能满足客户的应用需求，又不至于要花费过多的精力。可惜的是，这种程序员实在是不多，因为度的把握对程序员而言，确实不是一件容易的事情。



2.完美不等于质量 100 分

程序员心中的完美和项目经理心中的完美并不是一回事，因为两者关注中心不一样。程序员关注的是自己的软件功能本身，力争将软件产品质量做到最好，因此程序员的完美实际上是质量的完美。

而项目经理眼中，看到的是整个项目，包括质量、进度、成本、范围、风险等方方面面，需要进行平衡，花最少的成本、用最少的时间、达到各方满意、实现项目验收，这就是完美。单纯产品质量一流，而进度拖延、成本超支，这显然不是什么完美的项目。

其实现代质量管理理论普遍认为，质量并不是越高越好。事实上，市场已经对此无数次给出了证明。很多人骂过微软公司的产品烂，据说乔布斯也曾经大骂 windows 是坨屎，但微软公司后来却成了软件行业的霸主。

ISO9000 对质量的权威定义是：一组固有特性满足要求的程度。看到了吧，是满足，而不是超出，这非常重要。不要少，少了通不过；但也不用多，多了便是浪费。我们需要的不是 100 分的质量，甚至也不是一流的质量，而只是满足要求的质量。

在项目管理中有一个名词叫“镀金”，也就是在产品达到客户要求后，再多做一些额外的工作，让产品更加完美，以进一步提升客户满意度，这在 PMBok 中是一种被明确禁止的行为。软件质量 100 分，在项目中不但是一种巨大的浪费，而且几乎是一个不可达到的目标，只会让项目不堪重负，最后陷入灾难的境地。

3.合格就是完美

追求完美本身并没有错，但如果上升到完美主义，时时处处要做到最好，却不一定符合当时当地的条件限制。一个“最”字会害死人，因为“没有最好，只有更好”，如果一味追求更好，其结果可能就如陷入焦油坑的怪兽一般，无法自拔。在这样一个讲求效率的时代，完美主义更是可能会造成机会的丧失。因此，要保持追求完美的心，但又要懂得权衡，不要陷入极端的完美主义的陷阱。



要完美不要完美主义，本质上是一个度的问题，项目应讲求平衡，避免极端。学过项目管理理论的人都知道，项目管理中有一个“铁三角”，也就是在一定的项目范围的约束下，成本、进度和质量构成三角形的三个端点，为了让三角形面积保持不变，任何一个端点的变动，都会引起其他一个或两个端点的同步变动。这个铁三角本质上就是一种平衡和制约的关系，而完美主义，则只单纯的强调质量，而忽略了其它方面的因素，这显然是一种极端的行为。

那项目中质量的“度”倒底是什么呢？其实就“合格”二字。合格意味着被认可，却不需要达到优秀的代价。客户认可、领导高兴、员工轻松，这不就是完美吗？可以说项目中没有最好，只有合格，合格就是完美。

4.“70 分主义”

从小老师和书本就教育我们要追求完美，考试要考 100 分，90 分都嫌太低，那 70 分还拿得出手吗？

其实 70 分不低了，要知道现在大学生的口号是“60 分万岁，多一分浪费，少一分作废”。当然这种口号容易被批评为不思进取，但万物存在就有其合理的一面，“60 分万岁”也是事出有因。

在学习方面，我是主张完美主义的，前提是学的是个人感兴趣、有用的、切合实际的东西，可以我们大学的大部分课程，基本上是背道而驰。上课、考试，无非是为了不挂科，顺利拿到毕业证和学位证。这种情况下，60 分万岁也就容易理解了。何必要考 100 分，节省下来的时间，完全可以用来学习自己更感兴趣东西。

从某种程度来说，做项目也是一种考试：有考试内容（项目范围）、考试时间（进度要求），还有及格线（质量要求和验收标准）。项目的及格线如果用分数来表示，也是 60 分，既然 60 分

就够了，为什么还果提出“70分主义”呢？其实很简单，因为要想刚好考60分，实在太不容易了，搞不好就会弄个不及格。所以我提出“70分主义”，一种超越完美主义的新主义，力求在及格和完美之间达到平衡。

项目如果以70分为目标，适当留出缓冲，就可以做到游刃有余，更容易把控。70分意味着已经达到客户的验收要求，已经能投入正常使用，但可能存在一些影响较小的Bug，个别页面效率有待提升，个别操作不是很顺手，系统扩展性一般，代码组织有等进一步优化……这些不完美的地方，就让他们在那里待着吧，毕竟客户已经觉得已经达到目标，何苦自己跟自己较劲，非要达到100分呢？早验收、早收钱，这才是王道！吃饭只用7分饱，项目也是只要70分，“70分万岁”！

从程序员到项目经理（8）：程序员加油站 — 不要死于直率

来源：[西西吹雪](#)

直率听上去是一种美好的品德，然而如果不注意区分实际情况，直率可能会成为一把伤人害己的“双刃剑”！

1. 直率是关于说话的问题

公司曾有一位人力资源经理是从传统行业转过来的，有一次她跟我说：“[程序员](#)真有意思，他们全都是一根肠子通到底，大脑不会转弯！”

还真是这样的，估计没有哪个行业的人员像程序员这样，具有同样的鲜明的性格特征：直率。

直率很容易理解，其实就是一个关于说话的问题，准确的说这是一个关于说还是不说、以及说多少的问题。过于直率的人，在说话方面往往有两个特征：

(1) 想到什么说什么

这是一个说还是不说的问题。显然，不是什么东西都可以随时随地的说，或者对任何人说。可是对于过于直率的人而言，想到了就要说出来，就像俗语所说的：“嘴上没有把门的”，不管好话坏话，不区分场合，不论说话对象是谁，心里想着什么，嘴里马上就出来了。如果是好听话还好，皆大欢喜；如果是让人难堪的话，那就会伤害了别人了。如果是在公众场合让人难堪，人家可能会记住你一辈子。

(2) 知无不言、言无不尽

这是一个说多少的问题。“知无不言、言无不尽”毛主席老人家曾大力倡导的，这在讨论具体事情时无疑大有帮助，可是如果作为一种品格，在中国国情（文化氛围）下，还是不宜提倡。话并不是说得越多越好，说得越多，错的机会就会越多，反倒容易被人家抓住弱点或把柄。俗话说：“逢人只说三分话”，对一个项目经理而言，更应是如此。

在情绪方面，过于直率的人，往往不能很好的掌控自己，在说话时容易显得消极或冲动。

(1) 消极型

言语显示出消极的态度，例如给人泼冷水。给项目组成员安排工作时，我最怕听到两种话，一种是说“这个我不干了”，还有就是说“我不想干这个”，每次听到这两句话，心里就觉得凉飕飕的，虽然不爽，但作为团队的核心，我只能选择耐心的分析和开导。如果项目经理也以消极的方式来应对，整个团队的士气将会受到打击。

(2)冲动型

虽然人人知道“冲动是魔鬼”，但魔鬼不是那么容易掌控的，人难免有时会说出冲动的话来，以直率著称的程序员更是如此。冲动的主要表现是言辞激烈固执、情绪激动，一旦过头就会变成人身攻击。我经常看到程序员与项目经理讨论问题，后来争论不休，项目经理被迫使用自己的职位权力，强制执行，但这样可能会导致矛盾激化，搞不好程序员就会拂袖而去：“我不干了！”

无论是消极还是冲动，这都是职场的大忌。这些外在的情绪，在领导的眼中，就是体现出了一个人的工作态度，而在职场上，态度决定一切。对于领导而言，没有积极、合作的态度，就意味着你不能为我所用，那留你何用？！

2.直率的悖论

对于直率的人有很多有意思的词语，好听一点的比如“直性子”、“直来直去”、“心直口快”、“快人快语”、“率性而为”，不好听的如“口无遮拦”、“大嘴巴”、“一根筋”、甚至“缺心眼”。可见在要不要直率这个问题上，中国人真的过得很矛盾、很痛苦，有时甚至无所适从。“说，还是不说，这是个问题”。长此以往，对于修养不够的人，造成人格分裂的倾向也不足为奇。

(1)书上提倡直率，现实鼓励含蓄

其实好像没有哪本书正儿八经的告诉我们做人要直率，那为什么很多人眼里直率是一种美德呢？我想这是从小到大的教科书对我们潜移默化的结果。书上教育我们世界上主要有两种人：好人和坏人。好人多是直率的人，他们流芳青史，而坏人多是阴险狡诈之徒，他们遗臭万年。对比之下，我们当然想做直率的人了，光明正大，而且要直率太简单了，人人都做到，做的过程中还很爽。

另一方面，中国人的含蓄又是出了名的。中国是一个讲人情的社会，什么东西最重要？人情和面子。一个成熟的人不会当面伤害别人的感情，不太好的事情喜欢用暗示，以免伤了面子。不但说话，连中国的诗歌、中国的医学、中国人谈恋爱，也是含蓄的、有点模糊的不清的。

(2)觉得直率好，却很少有人喜欢别人对自己直率

假如做一个调查，你是喜欢别人直率，还是含蓄，我相信大部分会选择直率。当别人含蓄的时候，我们会催着让他“有什么话直接说”，等别人说了，如果事情难办，心里又可能犯嘀咕，“这人说话也太直接了，连退一步的余地也没有了”。

其实我们很少有人希望别人每件事情都对自己直来直去，试想一下，当别人当面揭露我们的缺点时，我们会是什么样的心情？与此相对照的是，我们又希望自己对别人直来直去的时候，别人会高兴的接纳，这可真是奇怪的事情。

(3) 网上“直率”，现实中彬彬有礼

有些人在网上非常的“直率”，直率到了一不高兴随时骂娘的地步，骂了又怎么样，反正见不着你。我相信在现实生活中，他们大部分都是讲道理有礼貌之人，因为在现实中，我从来没有碰到像网上那么多人动不动就爆粗口、骂娘，祖宗十八代什么都可以骂。这样的人，有人格分裂倾向。一个人格完整健全的人，应该是一个不管什么场合都言行一致的人。

(4)表面直率，心里打算盘

在中国什么类型电视剧最热？我想无非是武侠剧、宫廷剧、历史剧，这些电视剧无一不充斥着人与人之间勾心斗角的故事。有一位伟人说：“与人斗，其乐无穷”，试想一个什么都藏不住的人，与人斗，还不被别人玩死啊？因此，无耻的政客们不论心里面打着多坏的算盘，表面上也得笑着，一副给人家交了老底了样子，好叫别人不要防着。当然，职场中还没有这么夸张，那是因为我们没有那大的利益需要去争斗，但中国人长期受这种文化的熏陶，表里不一的人到处都是。

上面这些所谓的“悖论”只为了引起大家的思考，对于直率是多是坏这样的问题，并没有标准答案。但凡事[过犹不及](#)，过于直率无疑不但会伤害别人，最后也会反过来伤害到自己。

3.直来直去伤人害己

直率这种性格听上去还不错，显得一个人光明正大，无所畏惧。有些人在自我介绍时候会说：“我这个人就是个直肠子”，言语之间还透着一些小小的得意。

很多人将直率视为一种美德。美德应当是对别人对自己都有好处的事情，可是“直率”并不是这样，如果把握不了度，可能反而会伤人害己而不自知。

(1)伤人

要说直率伤人，相信人人都懂，很多人还会有切身的体会。国人最讲面子，一旦面子被伤，很难挽回，正所谓“刀伤易痊，舌伤难愈”。武侠小说中江湖中人，为什么整天打打杀杀的？直率就是一个重要的原因，那些人个个都是直率之人，又不讲礼节，经常出言不逊，一言不合，就兵刃相见。好在祖先教我们注意礼节，用礼节约束我们的言语和行为，社会就会和谐多了。

有一次我收到一个程序员的辞职申请，让我奇怪的是，这名程序员一向工作踏实，为什么突然辞职呢？跟他谈过之后，这才明白原因很简单，就是因为项目经理多次批评他“怎么这么简单的事情都做不好”，而他认为事情并不简单，是项目经理不了解实际情况。但他不想解释，因为他的自尊心被伤害了，不想再待在这里工作。我又找项目经理沟通，项目经理说这是无心之言，只怪现在的员工太脆弱。最后的结果就是员工离职了，项目进度也受到一定的影响。试想，如果项目经理在说话的时候更加注意措辞，怎么会有这样的结果呢？

(2)害己

还记得《三国演义》中的杨修是怎么死的吗？杨修有过人之才，总是能看穿曹操的意图，然后得意洋洋的告诉别人，最后被曹操以扰乱军心的罪名处死。在历史上，许许多多的忠臣最后都没有好下场，因为他们往往是刚直之辈，言语伤了领导的面子，而领导一旦报复，后果是很可怕的。

直率就像没有成熟的柿子一样，好看不好吃。现实中，只听说过因为直率吃大亏，没有听说过谁因为直率升职加薪，不是这样的吗？一个管不住自己嘴巴的人，小则在团队中难以与人和谐相处，大则可能会得罪客户、甚至泄露公司机密，这样的人，领导怎么敢委以重任呢？

一个人能力很强，却因为说话的问题而吃亏，那实在不划算。技术的成长需要日积月累，而一句未经思考的话就有可能毁掉你在公司的前程。因此在把话说出去之前，我们应该多思考，三思而后言，避免犯下言语上的低级错误，后悔莫及。

4.三思而后言

在计算机中，我们常用 IPO（输入-处理-输出）图来描述一个过程或一个功能模块的设计，其实人说话也是一个输入-处理-输出的过程。首先我们接收到信息，比如别人说的话、文件、任务指令或其它外部事件，然后大脑对这些信息进行处理、思考，决定要说什么，最后嘴巴将大脑思考的结果输出，也就是说话了。

人与人之间说话的过程，主要区别就在于大脑思考处理这个环节。过于直率从某种程度来说，是一种有失理智的行为，因为他没有经过“理智”的思考，全凭个人的直觉反应来应对外部事件。

(1) 两种说话模型

根据人们说话的“输入-处理-输出”过程的不同，可以将说话的分为直率型和谨慎型两种方式。

对于直率型的说话方式，其主要特征是思考问题的方式比较简单，只是根据大脑的直觉做出反应，得到结论，然后直截了当的将结论输出（说出来），而且往往是一个输入对应一个输出，整个过程看上去就像一根“直肠子”，如下图所示：

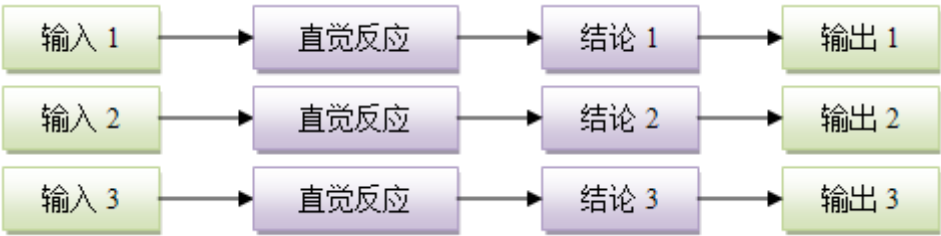


图 1 直率型说话模型

这个图让人不禁想起了巴浦洛夫的条件反射理论，显然直率型的人在说话方面没有充分利用人的思维能力，发挥人的主观能动性。

而谨慎型的人，他们的思考过程比较复杂，在直觉反应之后，还要经过大脑的分析总结，在说话之前要判断是否可以说，因为除了说之外，他们还可以选择沉默。另外与直率型不同的是，他们往往多个输入对应一个输出，这意味着他们说之前要听对方把话说完，而不是匆匆忙忙下结论。

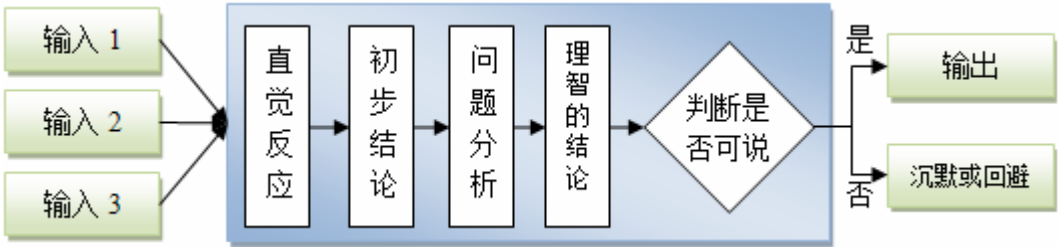


图 2 谨慎型说话模型

两种方式比较如下表所示：

环节	直率型	谨慎型
----	-----	-----

	说明	点评	说明	点评
输入	每个输入一个输出	快人快语	多个相关输入一起处理	把话听完，察言观色，结合事件背景，边听边看边想
处理	直觉反应→结论，缺乏思考与分析的过程	说话不经过大脑	多了思考分析的过程，同时话说还有判断是否可说	说前要思考，嘴巴多了一个把门的
输出	每个输入一个输出	直肠子	多个输入，一个输出（必要时沉默）	心有城府

（2）三思而后言

从上面的模型可以看到，谨慎型的人主要特征是说话理智，真正做到了三思而后言。说话一定要经考大脑思考，没有思考，我们就如同一个被自己的性格的直觉控制的牵线木偶。因此，当我们与人交流时，务必要时时思考说什么以及怎么说的问题。

● 说什么

说什么就是要判断我们想说的内容可不可以说。说话不能逞一时之快，千万不可说极端的话。还有就是要学会察言观色，学会从事件的背景、对方面的性格、脸色、手势等“输入”信息来揣测对方意图，从而做出合适的反应。

● 怎么说

这是说话方式的问题。要注意对事不对人，看透不说透，立场要客观，不能偏激，好话可以直接说，不好的话要委婉的说。其实程序员直率一点，尚无大碍，因为程序员最重要的事情就是高效地写出合格的代码，即使你伤害了你和经理，他一般也不会计较；但对项目经理而言，沟通是其最重要的工作，他需要与各方、代表不同利益的人打交道，如果想到什么说什么，不但会被人家认为肤浅，而且你的底牌、你的真实想法全在人家的掌握之中，无论是在处理内部冲突的时候，还是与客户谈判的时候，都会陷入被动的境地。“见人说人话，见鬼说鬼话”，这话虽然“逆耳”，但是“利于行”啊。

养成思考的习惯，刚开始可能会很不适应，觉得很累，但只要坚持，久而久之，就会习惯成自然，用大脑管住嘴巴。

当然慎言也不能过度，否则可能会造成胆怯，什么都不敢说，反而不利于问题的解决。把握好度，不要走向反面，好与坏之间往往只隔着一层纸。

需要说明的是，本人也绝不不是什么说话的高手，我对自己的要求不高，就是尽量照顾别人的感受，不捅漏子！当然即使是这个看上去简单的要求，也必须要改掉过于直率的毛病，凡事三思而后言！

5.守住真我

（1）直率也有可取之处

三思而后言，并不是要将直率一棒子打死。之所以要避免过于直率，是为了避免伤害别人和自己受伤，而不是要把自己藏得很深以进行利益的争斗，更不是要去算计别人，记住这个出发点。

因此该直率的时候还是要直率，否则就容易变成虚伪。比如在讨论技术问题时，就应该畅所欲言，让别人准确了解你的想法，注意就事论事、不搞人身攻击就可以了。在分配工作时也是这样，应该力求清晰明确。再比如，项目遭遇危机的时候，有什么想法要全盘说出来，就算只是头脑风暴也有价值。如果这时候还在考虑要直率还是委婉的说，那就好比一个人都快饿晕了，还在考虑是要吃米饭还是面包一样。在生活中，也有需要直率处理的问题，比如谈恋爱，如果总是含蓄，始终不感说出“我爱你”三个字，说不定你的心上人就要飞了。

（2）真我永存

周星驰在《喜剧之王》中有一句台词：“其实我并不是一个直率的人，我只是个演员！”说出这句话，多少有些无奈，谁不想过恣意纵横的生活呢？每个人都是生活这个舞台上的演员，需要涂脂抹粉，甚至要戴上面具，不然容易伤到别人，或为别人所伤。但这并不意味着要泯灭真我，因为真我就如“挪威的森林”一样，别人无法触碰得到，但永远存在于我们的内心深处。

有人说，直率的人生活更加真实、更有个性，如果人人都像演员，便那岂不人人都失了性格？其实性格是一种内在的秉性，并不会因为你管住了自己的嘴巴就没有了性格，难道含蓄和委婉的人就没有了性格吗？我们说话做事应该以原则为导向，而不是被性格所牵引和控制，只要你内心有自己的原则，你就是一个性格的人。

不要过于直率，但也不要走向了另外一个极端。如果心机太深，或者虚情假意，那就变成了虚伪和狡诈，没有人会愿意与这样的人交往，因为他们的行为是不友好的，已经脱离了善的范畴。只要心存善念，真我就会永存。

从程序员到项目经理（9）：程序员加油站 — 再牛也要合群

来源：[西西吹雪](#)

“丛林法则”从未离我们远去，“适者生存”仍然是支配社会运行的一般法则。对于一群社会性动物而言，所谓“适者”，不只是体格的强壮，更重要的是能参与群体的公共生活。即使是最强大的狮子，只要离群，也只有死路一条！

1.好汉也要三个帮

我喜欢看动物世界，感受那些发生在非洲大草原上的那些美丽或者哀伤的故事。那里生活着成群的狮子和鬣狗、还有数以百万计的野牛和角马。无论是凶猛的狮子，还是温驯的角马，都属于是群居动物，个体一旦离群，就会离死亡不远了。

其实人也是一样。人是一种社会性动物，我们只能生活在社会群体中，离开了群体，我们的人生价值也就无所依附。在社会心理学名著《社会性动物》的扉页上，印着一段亚里士多德的名言：“从

本质上讲人是一种社会性动物；那些生来离群索居的个体，要么不值得我们关注，要么不是人类。社会从本质上看是先于个体而存在的。那些不能过公共生活，或者可以自给自足不需要过公共生活，因而不参与社会的，要么是禽兽，要么是上帝。”

其实这段话应该修正一下，许多动物也是要过公共生活的，至于上帝，我们都不曾见过，想必也是差不多的。无论是希腊神话中的宙斯，还是中国神话中的玉皇大帝，他们身边不也是都有一班大小天神簇拥左右吗？

可见下至动物、上至上帝都需要合群，更何况是人？

可是在[程序员](#)这样一个群体中，确实还是有不少人不喜欢与别人打交道，喜欢独来独往，过着自我封闭、离群索居的生活。

一个人不合群的原因有很多种，比如：价值观不一致、胆小害羞、不善言辞、性格内向等。而对于一个技术牛人来说，其不合群的原因还要加上一条：看不起别人，觉得“竖子不足与谋”。

中国素有文人相轻的习惯，其实程序员相轻的现象一点也不比文坛少。程序员多有自傲的性格，容易看高自己，看扁别人。觉得自己一个人也能[搞定](#)所有事情，多几个人来弄反倒碍手碍脚。当今社会是一个高度分工、讲求合作的社会，每个人都是团队中的一员，总想着个人单干的小农思想，已经无法与现实相容。个人英雄主义的时代已经远去，在一个项目中更是如此。俗话说：“一个好汉要三个帮”，一个人再牛，也应该学会欣赏别人的优点、与人和睦相处，因为没有这“三个帮”，他便当不成英雄好汉，空有一身武功，四处碰壁，一事无成。



2.合群谁都可以做得到

每个人都内心里对外在的事物都有一道防线，这是一种自我保护的本能。对于不合群中的人，这道防线显得格外的高大和坚固，以至于将他与其他人隔离成了两个世界。其实合群并不是一件难

事，关键是要敞开心扉，卸掉内心的防线，主动与别人交往，融入到所在的团队中。当然，合群也需要注意一些问题，避免盲目交往，或者言行失范。

（1）合什么样的群

合什么样的群，也就是说我们应该与什么样的人交往。所谓“近朱者赤，近墨者黑”，因此有必要对自己交往的对象加以界定。

如果是一帮举止不端或格调不高的人，应该果断退出，平时也应保持适当距离，以不得对方为限。

对于自己不感兴趣或者对自己助益较少的群体，不要一概拒绝，否则会给人以不近人情的印象。可以适当参与他们的活动，但不能过多，否则会占用自己过多时间。

交往的重点应该是与自己兴趣相投、对自己有帮助的人。与他们相处，不但可以互相学习，而且人生的快乐和价值可以找到落脚点。

（2）言行的把握

在与人交往中，言行得体是非常重要的。2009年河南有个局长叫逯军，因为一句“你是准备替党说话，还是准备替老百姓说话”名扬天下，沦为笑柄。最近，“表叔”杨达才因为在车祸现场诡异一笑，不但引得丢官弃爵，恐怕还要陷入牢狱之灾！

在我们普通人的生活中，因为言行不慎，招来误解、怨恨的例子同样非常多。

对于言行的把握最重要的是要谦和、通融、合规、适度。例如大家玩的时候你也玩，不要做异类；开玩笑不要过分、让人难堪；举止不要怪异等。

（3）尊重他人，保持平等

这是对牛人的忠告，因为牛人技能超群，更容易觉得自己高人一等，看不起别人。人与人交往最重要的是获得尊重和认同，如果他不能从你这里获得这些，你就是比牛顿还牛，对他而言也是没有价值的。须知，尊重是双向的，合群的首要便是尊重对方，以平等之心相待，不卑不亢，这样才能赢得别人的尊重与认同。

从程序员到项目经理（10）：程序员加油站 一要执着但不要固执

来源：[西西吹雪](#)

[程序员](#)的成长之路，没有捷径可走，只有坚持不懈的执着追求，才能成为一名优秀的程序员。执着诚然可贵，但如果不能经常自省，则有可能会陷入固执的境地。

1.程序员需要一点执着精神

《士兵突击》中许三多有一句名言：“不抛弃、不放弃”，这是一种可贵的执着精神。正是靠着这种不抛弃、不放弃的执着追求，许三多从一个普通的小兵，成长为团部的精英。在现实生活中也是这样，可以说大凡取得一定成就的人，在工作中都是一个执着的人。

对程序员则言，执着精神尤为可贵。在编程过程中，我们难免会碰到各种问题，如果没有一点执着精神，一碰到问题就抱怨、回避，怎么可能取得技术上的突破呢？又怎么能体会到解决问题的快感呢？

回想起我刚入门学习GIS（地理信息系统）编程时，经理就给我安排了一个之前让不少人望而却步的难题，用MapObjects实现地图符号化，要求具有自定义符号库的功能。以我当时的经验，根本不知道从何下手，但也只能硬着头皮上。首先我把MapObjects的帮助文件全部仔仔细细看了一遍后，找到一个CustomDraw接口。但是只是一个接口而已，离完整的符号化功能还相差很远。怎样利用这个接口呢？当时网络还很落后，网上的编程资料更少，关于MapObjects的中文开发资料则几乎没有，于是我又通过蜗牛速度的网络，查阅国外的相关英文资料，在片言只语中寻求灵感。那一段时间我无论是吃饭、睡觉，还是走了路上，无时无刻不在思考技术上的问题，由于坚持不懈的努力，我一次次获得小小的启发，一步步接近问题的解决之道。6个月艰苦摸索之后，我终于彻底搞定了这个在公司内公认的难题，我本人也从一个门外汉，一举成为了公司的核心技术人员。这一段时间，我不但把MapObjects每个接口弄得烂熟，还学会了一百多个Windows API的使用，无论是技术方面，还是个人的职业生涯，都取得了一次飞跃。程序员都需要一些执着的精神，来磨炼自己、发展自己，要有水滴石穿的决心和勇气，才能够成为真正优秀的程序员。

2. 自省消除固执

固执和执着一样，都是一种坚持不放弃的精神，既然如此，那为什么人们总是赞美执着的人，对固执却嗤之以鼻呢？

其实两者的差别全在于坚持的方向。执着和固执，就像一根绳子的两端，虽然是在同一根绳子上，方向却相反。执着是沿着正确的方向前进，是一种理智的坚持，而固执则恰好相反。既然都是坚持，那怎么判断方向是否正确呢？

其实，何为正确，何为错误，两者之间并不是泾渭分明，不然，也就不会有那么多“执迷不悟”的人了。方向是否正确，往往是以结果来衡量的。因此是执着还是固执，其实主要是结果导向，结果好就是执着，结果不好，就是固执。爱迪生发明灯泡的时候，经历了无数次的失败仍然坚持不懈，最后终于找到了用钨丝作为灯丝方法，取得了成功，他的坚持我们称之为执着。后来，爱迪生创立了通用电气公司，坚持用直流电供电，无视交流电在远距离传输方向的巨大优势，最后输给了采用交流电方案的西屋电气公司，他自己也只黯淡离开自己创立的公司，这时候，我们只能说发明大王也有固执的时候。

如此说来，难道我们非要等要结果发生，才能知道自己的坚持是对是错吗？有没有办法让我们在进行过程中就能出判断呢？这只能靠我们的自省。孔子曰：“吾日三省吾身”，大凡善于自省的人，都不会是固执的人。他们能随时察觉自身的问题，具有理智的否定自己的勇气。

自省需要常识。对于一个不具备常识、不明白对错、不理解基本规则的人，怎么能正确判断方向呢？这样的人再怎么自省也是无济于事的，他只有在不断的碰壁中才能获得真正的成长。



我曾经见到一些程序员，在自己的想法与项目经理发生冲突时，总是一味的坚持，不肯让步，甚至与项目经理陷入无休止的争吵，还以为自己掌握了真理。殊不知，与上司顶撞是一种愚蠢的行为，这种过分的坚持，会在上司心目中形成不听话的印象。更何况，服从上级工作安排是基本的职场规则，你可以提意见，但必须尊重上司的决定。毫无疑问，在这场对峙中，不管理项目经理对错，程序员都是固执的一方。如果程序员具备这些基本的常识，并且保持自省，也就不会发生这样的事情了。

自省还需要具有突破思维舒适区的勇气。每个人的都有其思维舒适区，这里一切受潜意识的保护，一切都似乎理所当然，我们的大脑无需对事物做过多的思考，爽爽的享受这种自我封闭带来的轻松和愉悦。毫无疑问，思维舒适区阻挡了我们对事物深层次的探求，以及我们对不同观点的接纳，因而也就无法对自己所坚持的东西做出真正客观的分析。

在程序员与项目经理的争吵中，其实双方都应该勇敢跳出自己的舒适区，心平气和地考虑，对方的观点是否也具有可以接纳的成分，做一个理智的坚持者，这样才能做到双赢。执着还是固执，往往也就只是在一念之间的差别。

从程序员到项目经理（11）：每个人都是管理者

来源：[西西吹雪](#)

从[程序员](#)转为项目经理，这是一个巨大的跨越。一个新任的项目经理，对项目管理找不到感觉，一般也被认为是一件正常的事情。这是否意味着，一定要等到当上了项目经理才能学习项目管理吗？一定要做砸一个项目才能成长为合格的项目经理吗？其实未必，项目管理所素质和技能

并不是什么独门秘籍，而是在生活中时时用到、处处可以锻炼的。只要有心，程序员一样可以学习和实践项目管理知识。从某种程度来说，我们每个人都是管理者。

1.管理是职能而不是职位

管理学之父彼德·德鲁克曾说：“任何一位做决策的人，其工作性质和董事长，和行政领导相同。即使他的管辖范围有限，甚至于他的职能或他的大名，不见于组织系统里，办公室连专线电话也没有，但他确实也是一位管理者。”

可见管理并不是经理、老总的专权，管理不是个职位，而是个职能。无论你在什么岗位，也不论你有没有下属，只要你需要做出决策，需要对结果负责，那你就是个管理者。从这个角度来说，我们每个人都是管理者，因为每个人都需要对自己的生活的工作负责，对碰到问题进行权衡决策，只不过决策的内容不一样而已。

程序员显然也需要对工作进行决策。当接受任务时，程序员需要对工作量、工作难度、时间限制进行评估，以确定能否实现项目经理的目标；开发一个功能点时，我们需要思考哪些实现方式，哪种方式开发速度、运行效率、对资源的占用几个方面综合最优；最进度滞后时，是要加班赶回来，还是要调整工作方法，提高开发效率……这些不都是决策的过程吗？在每一个决策点，程序员完全可以像一个真正的项目经理一样，发挥其主观能动性，主动进行管理，保证任务又快又好的完成。我们的管理才能，就在这一次一次的决策过程中，逐步积累、逐渐提高。

管理只是一项职能，人人都可以随时随地履行这项职能。可惜的是，很多人没有意识到这一点，不自觉的放弃了这项可以做而且应该做的工作，这不能说不是“失职”啊。

2.自我管理是一切管理的基础

管理有一个流行的定义，叫做“管人理事”，既然是管人，那必须得有人可管。有人说，我没有一个下属，只是一个“光杆司令”，要说我是管理者，那我都管了谁呢？

其实只要在社会中，没有谁是真的光杆司令，你管理的不一定是下属，每一个你需要打交道的人，包括你的领导，都是你的管理对象。退一步讲，即使你不需要跟任何人打交道，你也可以、而且必须管好一个人——那就是你自己。

彼德·德鲁克说过，“有伟大成就的人，向来善于自我管理。然而，这些人毕竟是凤毛麟角。但在今天，即使是资质平庸的人，也必须学习自我管理。”试想一个连自己都管不好的人，怎么能管得好别人呢？更别说管好一个大的团队了。

那自我管理该管些什么呢？李嘉诚先生曾说：“自我管理是一种静态管理，是培养理性力量的基本功，是人把知识和经验转化为能力的催化剂。”如果更加直白的说，**自我管理实际是一个修身的过程，是一个自我约束、自我磨炼、自我精进的过程。**作为一个普通人，哪些方面需要磨炼和精进呢？我想无非是一个人的身心和素质技能两个方面，相应的，自我管理的内容也应该是包括身心管理和个人素质技能管理两个方面。

（1）身心管理：包括身体、心态、情绪、世界观、人生观、价值观、人生目标、职业目标等不同层次；

（2）素质技能管理：包括学习管理、时间管理等。其中时间管理时自我管理非常重要的一环，因为它与项目管理、企业管理等内容直接交织在一起。要成为一个卓有成效的管理者，首先就是要能管好你的时间。

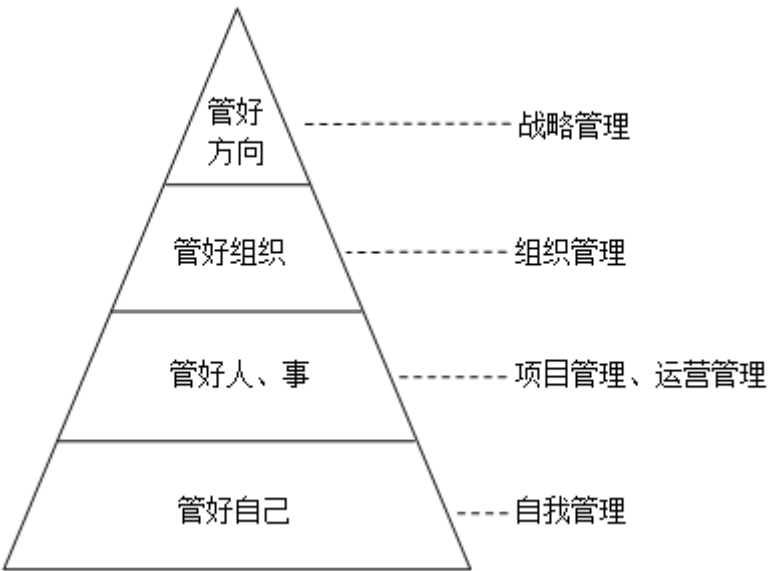


图 自我管理是其它管理的基础

既然自我管理是一种修身，那也就可以说，自我管理是其它一切管理的基础，因为不论是什么管理，都离不开管理者自身的身心和技能。一个企业中的所有管理工作，从管理的对象来说，可以分为管理者自己、企业中的人和事、企业组织本身以及企业战略方向几个层次，其中管好自己属于最为基础的层次。一个能管好自己的人，才有能力、有精力管好别人，处理好复杂的事务，才能够通透人性，把握组织和市场的规律，成为一个真正卓越有管理者。

3.每个开发任务都是一个微型项目

作为一个程序员，也许你从来没有把自己放在项目经理的角度来考虑过问题，但实际上，你不只是一个程序员，同样是一个项目经理，因为每次接受了一项开发任务，实际上就是接受了一个小项目。

一项开发任务，同样具备项目的典型特征：临时性、独特性和渐进明细。临时性是显然的，因为每一项开发任务都有开发时限，而不是重复无休止的工作，当目标达到，任也就结束了。同时每一项开发任务又是独特的，时间、地点、完成人、成果、项目环境等，总有一样是不同的，就便是其独特性。越往后开发，对细节的把握越具体，这渐进明细。

由此可见，一项开发任务就是真真实实的一个微型项目。只不过这个项目，只是由你一个人来完成而已。在完成任务的过程中，同样需要像管理项目那样，进行计划、时间安排、偏差控制和领导（自我领导）。

把自己当项目经理的程序员，才能成为真正优秀的程序员。优秀的程序员，也更容易成长为优秀的项目经理，因为在被正式任命为项目经理之前，他已经负责开发过了无数个微型项目。

从程序员到项目经理（12）：如何管理自己的时间

来源：[西西吹雪](#)

对很多项目经理而言，是没有什么所谓的“我的时间”的，因为他们不是在管项目，而是被工作的潮水带着跑而已，他们的时间被工作主宰了。项目经理必须要主动的管理自己的时间，合理安排自己的工作，才能真正“翻身”做自己时间主人。

1.谁动了我的时间

时间对于每个人而言，都是最稀缺的资源，对于一个管理者更是如此，时间不够用成为几乎所有管理者共同的问题。如果要对项目经理常说的话做一个调查的话，想信“我很忙”一定可以名列前茅。以我的经验，当要求项目经理按时提交项目材料，或者临时支援某件紧急事务的时候，经常会听到同样的回答：“我很忙”。

多年以前，我就从经理那里听说，厉害的管理者都是很轻松的，因为他的工作全部交出去了，根本不用自己操心，所以他们出去度假十天半个月，一切工作都会如常进行。从那时起，我就充满了对管理的神往，可是后来我才发现原来这只是个传说，现实中忙忙碌碌的经理比比皆是，而轻松自如的管理者则是众里难寻。

为什么管理者都这么忙呢？是谁动了他们的时间？实际上，这是一个综合性的问题，既有内部原因，也有外部原因，既有主观原因，也有客观原因。总的来说，让经理们不堪重负的因素有三：

(1)工作

对于一个[程序员](#)来说，他的工作是比较单纯的，基本上是单线程运作，只需要项目经理交待开发任务即可，可是当上了项目经理就不一样了。以前好比在游泳池中游泳，现在是在大海里冲浪，各种事情如潮水一般向你涌来，让你顾此失彼，手足无措。

(2)下属

下属也是一种资源，即人力资源，这种资源与时间一样，同样具有稀缺性。其实我们可以设想一下极端情况，如果你的下属人数足够，能力也很强的话，你完全可以像我的经理说的一样，把你的全部工作授权给你的下属，你自己也就不用整天焦头烂额了。

因为你的下属不给力，所以你总是要自己来制定计划、自己来做系统架构、自己来监控进度、自己来检查质量、自己来写文档、自己来汇报工作、自己来解决重要问题、甚至自己来编写代码，你整天忙忙碌碌，就是在忙这样的事情。

然而，千万不要怪你的下属，因为他们不给力正是老板雇佣你的原因，况且资源的稀缺性是永远存在的——从原始社会到将来的共产主义社会。要知道，老板做项目为了赚钱，而不是让管理者更轻松，如果每个项目都是精兵强将，你只要一声令下工作就会自动完成，你倒是轻松了，但老板还要你来做什么？

(3)自己

既然资源受限是一定的，项目经理还是应该反求诸己，从自己身上找到解决之道。这就好比天下雨了，你怪老天是没有用的，只能怪你自己没有带雨伞。经常问一问自己，我对工作安排合理吗？我抓住了主要问题吗？我在旁枝末节的事情上浪费时间了吗？我有充分发挥下属的能力吗？我

自己工作拖拖拉拉吗？...通过不断的自省，改善自己的管理方法和行为习惯，我们对时间利用也必然会变得越来越高效。

2.时间管理的本质是对工作的梳理

要破解忙的难题，必须要有意识的对时间进行管理。其实时间本身是没法管理的，因为无论怎样管理，时间既不会变多，也不会变少，既不会变快，也不会变慢。所谓的时间管理，其实就是如何更有效的利用时间的问题，更加直白地说，其本质就是工作管理，即通过对工作的梳理，让我们在有限的时间内，使得工作更有条理、更有成效。

必须要主动、有目标地对工作进行梳理，这是对一个管理者的基本要求。工作梳理就好比整理房间，你不去整理它，杂物就会堆积得越来越多，你房子最终会变得不适合人类居住。一个好的家庭主妇，必定善于将各位物品分门别类，并且适时扔掉一些用处不大的物品。一个好的项目经理也一样，同样需要对工作进行分类，对不同类型工作采用不同的策略，有些工作要现在就做，有些可以晚点做，或者不做；有些工作一定要自己做，有些工作则可以请其他人来完成。

通常对工作梳理，可以采用 5W1H 法，即：

Why——为什么干这件事？（目的）；

What——什么事情？（对象）；

Where——在什么地方执行？（地点）；

When——什么时间执行？什么时间完成？（时间）；

Who——由谁执行？（人员）；

How——怎样执行？采取哪些有效措施？（方法）。

在一般的项目中，**Why** 和 **where** 往往不是什么问题，或者说对项目经理的时间管理影响较小，因此我们不妨将其简化为 **3W1H**，也就是确定要做什么，不做什么；先做什么，后做什么；谁来做；怎样做才更有效。基于此，项目经理可以按以下三个步骤来梳理工作：

（1）分析要做什么、不做什么，以及先做什么、后做什么

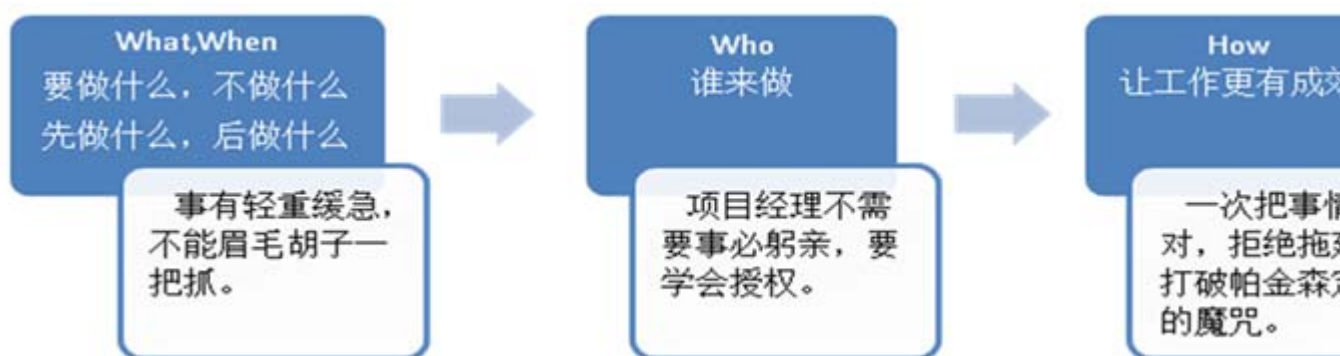
解决 **What** 和 **When** 的问题。事有轻重缓急，事情的重要程度和紧急程序直接决定其处理的优先级。虽然很多事情来势汹汹，但并不表示一定要当即处理，有些事情只是静静的躺在那儿，也并不意味着要“等有了时间再做”。

（2）分析由谁来做

解决 **Who** 的问题。虽然我们提倡项目经理要以身作则、亲力亲为，但并不是说每件事项目经理要亲自去做。对于下属可以胜任的事情，就把它分配出去。如果出现项目经理很忙、下属很闲的情况，那就说明项目经理你做得太多了，不要和你的下属抢事情做。

（3） 如何让工作更有成效

做不做、什么时候做以及谁来做的问题都解决了，剩下就要解决怎么做才能让工作更有成效的问题了。在这里我们不是要讨论编码或写文档的技巧，而是个人的习惯和认识，这对工作成效的影响更是本质上的。

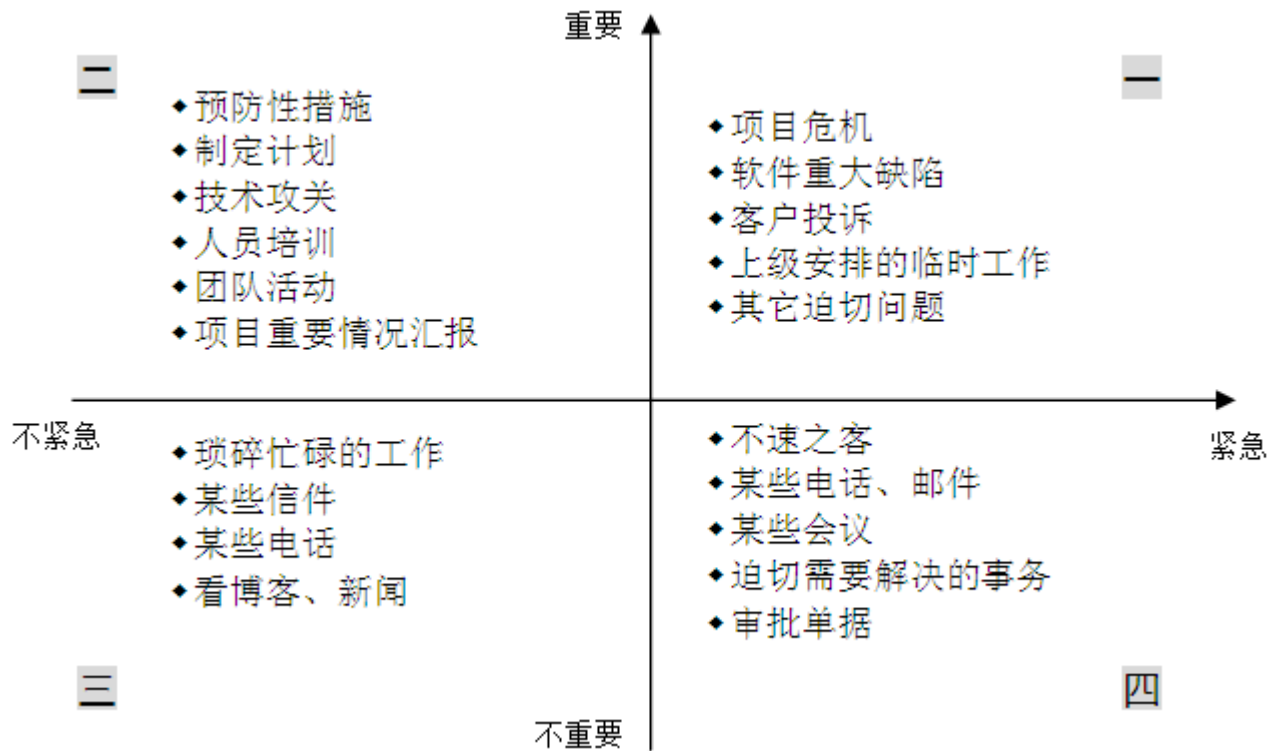


3.做事要分轻重缓急

老外就是善于总结，中国有词语叫“轻重缓急”，可是到了国外摇身一变，变成了“时间管理四象限法”——自从美国总统艾森豪威尔提出以来，人人将其奉为圭臬，成为时间管理领域最重要的方法论。

所谓的“四象限法”，就是将工作按照重要程度和紧急程度两个维度进行分类。我们找一张白纸，以紧急程度为纵轴，以重要程度为横轴，在纸上划上一个十字，将纸面分为四个象限，然后将当前所有要做的工作放到这个四个象限中。

一个典型的项目经理四象限图如下所示：



(1) 第一象限：重要紧急

这一类往往是火烧眉毛的事情，需要马上去处理，否则项目会受到重大影响，比如客户服务器崩溃。

(2) 第二象限：重要不紧急

这类事情一般是预防型的工作，例如制定项目计划、团队建设等，它们不需要你停下手上的工作马上去做，但如果没做好的话，可能会导致产生项目危机。许多第一象限工作产生的原因，正是因为第二象限的工作没有去做。

(3) 第三象限：不紧急也不重要

这类事情看上去最不需要做了，例如上网偷菜、看新闻、写博客等，但如果你在办公室走上一圈，就会发现很多人正在干着这些不需要干的事情。

(4) 第四象限：紧急不重要

这类事情虽然不重要，却需要马上去处理。一个典型的例子就是桌上的电话响了，你接还是不接？当然要接，因为你不知道是谁。接通后，发现是推销保险的，你又不不好意思立即挂掉，只好被对方折磨一番了。

我们到底该怎样安排四个象限的工作呢？对于一个普通的管理者，其工作的优先级一般是这样的：第一象限>第四象限>第二象限>第三象限。可是，等做完了第一、四象限的工作，根本就没有时间来人做第二象限的工作，于是项目到了后期项目经理只好四处救火。

管理大师彼德·德鲁克十分推崇“时间管理四象限法”，并将其总结为“要事第一”的原则。根据这个原则，每个象限的工作处理策略是不一样的。

（1）重要紧急

优先级最高，需要尽快处理。很多人都玩过《植物大战僵尸》的游戏吧，那你一定知道“一大波僵尸正在逼近”的感觉，是的，你必须要马上打死它们，不然它们就会冲进你的房子，吃掉你的大脑！

（2）重要不紧急

这类事情看上去可以暂缓，但考虑到其重要性，应当与第一象限的工作并行去做。如果不及时去做，它们就会转移到让你头疼的第一象限中去，或者在第一象限产生更多新的“僵尸”。所以，要在僵尸还没有逼近的时候，就好防御工事，并尽快打死它们，如果等到它们冲了过来，你还能不能保住大脑，就要看你的运气了。

（3）紧急不重要

它们就像是在你耳边“嗡嗡嗡”地叫着的苍蝇，你必须要花时间去赶走它们。这多少让人有些无奈，但这些事情确实层出不穷。有些公司在实施紧急项目时，经常采用封闭式开发，这样做的一个重要原因就是回避那些紧急不重要的事情。很多管理专家建议我们在必要的时候勇敢说“不”，其实就是针对这类事情。如果实在无法说不，建议安排或委托其他人来做。

（4）不紧急也不重要

如果不是时间充裕的话，建议不要去做。如果碍于人情的话，建议安排或委托其他人来做。它们就像一群在几百米远处飞的苍蝇而已，你完全不必要放下手中的饭碗，举起苍蝇拍跑过去和它们决斗。

因此，对于一个卓有成效的管理者，其优先级应该是这样的：第一象限=第二象限>>第四象限。第三象限就像数学中的无穷小一样，被舍弃了。

写到这里，我想起了前不久一位项目经理的故事：

项目定于当天上线，项目组决定搬到客户现场办公，以应付可能出现在的突发事件。项目成员电脑已经全部打包好，都围在项目经理周围等待。原来项目经理正在理一大堆发票准备报销，于是发生了这下面这样的对话：

我：“大家都在等你，怎么还在填报销单呢？”

项目经理：“今天是公司的报销日，不填好单子，又得推后很久。”

我：“你的电脑打包了没有？”

项目经理：“没有”

我：“放行条开了没有？”

项目经理：“没有”

我：“申请用车了没有？”

项目经理：“没有”

我不知道说什么好了。要知道公司的报销单粘贴和填写非常严格，经常被打回重新弄，那一堆发票，显然不是十几分钟可以[搞定](#)的事情。还有公司的用车也比较紧张，不赶紧申请，说不定就没有了，到时就只能租车或打的，这无疑又会耽误更多的时间。更何况六七个同事都在等项目经理一个人，耽误的时间还得要乘以他们的人数。万一系统上线，状况频出，客户火烧眉毛，项目组却仍然在路上，这样的后果是很严重的。

贴报销单看上去一件重要紧急的事情，实际上它既不重要也不紧急，因为今天不报销，以后还是可以报销，可是因此耽误的宝贵时间，却无法再要回来。如果项目经理更加理智一些，分清楚什么才是真正紧急重要的事，也就不会出现这样的状况了。

对很多项目经理而言，是没有什么所谓的“我的时间”的，因为他们不是在管项目，而是被工作的潮水带着跑而已，他们的时间被工作主宰了。项目经理必须要主动的管理自己的时间，合理安排自己的工作，才能真正“翻身”做自己时间主人。

4.管理者无需事必躬亲

有一种类型的管理者，他们不论什么事一定要亲自去做，至少也是亲自过问。人们习惯用一个成语来赞美他们，叫“事必躬亲”，仿佛诸葛亮再世一般。凡事亲自去做未必真的可取，为什么诸葛亮只活了 53 岁，恐怕跟他这种事必躬亲的精神也有莫大的关系吧——他是把自己累死的。

（1）不要和下属抢事做

管理者相对于操作层员工，多了一项法宝，就是授权。理论上，只要员工可以胜任，所有的工作都可以授权。事实上，总经理为什么能对全公司发号施令、对工作进行变革，那是因为董事会授予了这个权限。连这么高层的工作都可以授权，一个项目里面的工作还有什么不可以授权的呢？

因此，当你疲惫不堪的时候，就应该问问自己，我是不是管得太多了？如果一件事情下属能做，就应该让下属去做，不然等于是你抢了下属的工作。项目经理最可悲的事情就是，自己累得半死，项目组成员却闲得发慌。

管理者必须学会授权。授权不只是为项目经理分担工作，也是项目培养下属成长的必要方法。如果项目经理总觉得下属能力不行，不给他分配具有挑战性的工作，这显然不利于下属的能力成长，从长远看，对项目、对公司也是有百害而无一利。

（2）授权绝不是简单的把工作交出去

授权两个字说起来简单，但做起来效果却会因为而异。有效的授权必须把握以下几个要点：

！目标明确：要做什么内容、达到什么质量要求、什么时候完成等等，必须要清晰具体。管理学家们认为目标必须要 SMART（S=Specific、M=Measurable、A=Attainable、R=Realistic、T=Time-based），这是很有道理的。

！跟踪反馈：项目经理应当经常性对任务完成情况进行检查，这是很多项目经理非常欠缺的一个重要环节。只授权不检查，最后的情况可能就是进度大大延迟，或者与你想要的东西大相径庭，下属进行种种解释，但为时已晚。

！能力辅导：项目经理要对下属的能力有比较准确的把握，安排工作也应该在其力所能及的范围。如果跳一跳能够得着，就比较理想，但项目经理仍然需要主动辅导，加强监控，当发现偏差时，应及时采取应对措施。如果工作大大超出其能力范围，再怎么跳也够不着，项目经理就要另想高招了。

（3）不做甩手掌柜

是不是任何事情都可以授权呢？理论上是可以，但由于资源的稀缺性，这种条件往往并不具备。至于什么可以授权，什么不可以，这要因项目而异，根据项目工作与资源的实际情况，两厢权衡之后才能决定。不管怎么说，授权不可过度，否则项目经理就成了甩手掌柜，实际也等于放弃对项目的控制权。

项目经理应该做的工作：

！系统性工作由项目经理做，比如制定计划、安排任务、鼓舞士气、项目检查等，具体事务由下属去做。

！重要的事情项目经理来做，紧急的事情让下属去做。

！决策由项目经理来做，执行由下属去做。

！下属能做的事由下属去做，否则由项目经理自己做或带着做。

5.好习惯让工作更有成效

高尔基曾这样来描述时间：“世界上最快而又最慢，最长而又最短，最平凡而又最珍贵，最易被忽视而又最令人后悔的就是时间。”的确，时间是快还是慢，是长还是短，不在于钟表是的指针转了多少圈，而是在于在我们如何使用时间。一个人的习惯，对如何利用时间具有至关重要的作用。

（1）尽力避免返工

项目中最浪费时间的事情是什么？

是返工！

一旦发生返工，不但所耗时间将会成倍增加，而且会大大降低员工的成就感，打击员工士气，降低工作效率，使得项目时间进一步滞后。

我见过一个城市三维模型制作的项目，经过一年多的辛苦工作，终于提交成果了，但是由于客户认为模型不够漂亮，最后几十平方公里的模型全部重做！项目组员工身心俱疲，公司遭受严重损失，客户也非常不满，一个三输的结局。

返工并不总是这样严重，其实在一般的软件项目中，返工现象也是大量存在的，只不过我们借着迭代的名义将其掩盖了。例如软件试运行后，客户要求将某项业务流程中的两个环节进行整合，或者将某个环节中的输入信息，转移下一个环节中。单个修改的工作量也许并不算大，但累积起来就相当可观了。很多项目在试运行后要修改几个月，甚至半年以上，这就是返工的代价。

迭代设计还是返工之间，并没有明确的界限。要区分二者，有两条标准：

一是迭代是计划之中的完善，而返工则是计划之外、迫不得已而为之的事情；

二是在工作量的层面，如果抛弃或被重做的功能工作量很大，那只能认为是返工，如果你非要认为这是设计就是要这样干的，那我只好给它取个新名字：“返工式迭代”。这也给我们一个启发，做系统原型的时候，千万不要写大量的代码，否则的话，迭代最后会变成返工。

（2）打破帕金森定律的魔咒

英国学者帕金森通过多年的调查研究，发现一个规律：“工作会自动地膨胀占满所有可用的时间。”一个人可以在十分钟内看完一份报纸，也可以看半天；一个程序员开发一个功能，可以两小时完成，也可能花上一周的时间；项目经理制定计划，可以半天完成，也可能一个月还不见影子……总之，只要还有时间，工作就会不停的扩展。

帕金森定律就像一个魔咒一个样，困扰着很多人。它之所以起作用，表面上原因在于时间充裕，外部压力太小。因赖床而上班迟到的人常有，但因赖床而误飞机的则很少，因为误机的后果很严重。因此，有必要对每件工作确定一个时间期限——**dead line**，一过这条线 **dead**！给下属安排工作时，这的确是一个好办法，但对于管理者而言，约束别人容易，约束自己则很困难。即使工作到期，还可以告诉自己，再推迟几天也没关系，这件事情还可以让某某来完成，即使到了 **dead line** 还可以说这件事其实不重要，少做一点没关系。



图 帕金森定律的魔咒

归根到底，还是在于我们的内心力量不够强大，面对一点点的外部阻力，就变得消极懒散，不能自我驱动。截止日期是靠不住的，要靠只能靠自己，养成良好的习惯，主动给自己压力和动力，战胜心中的“懒惰小人”，才能真正解除这个“帕金森魔咒”。

（3）合理利用时间

每个人都希望工作不被打扰，但作为一个管理者，你的时间不是自己的，你的上级和你的下属都有权来随时打扰你。你坐在那里，就会有人过来找你签字，找你谈工资，找你讨论技术问题，找你支援其他工作……每天的时间就这样被打成了无数的碎片，所以经理们常不由自主的感慨：“白天真的做不了事，只能晚上和周末才能工作”——加班才能做事，你说经理能不累吗？

的确，项目经理很多工作都需要大块时间，比如制定计划、编写文档、分析风险、关键技术实现等，都需要较长时间的思考。一个人要让心静下来，进入工作状态是时间的，一旦被打断，再次进入这种状态会花很多时间。这就好比炒菜，把锅烧热是需要时间的，你刚放下油，来了电话，等你接完电话，锅又冷了。

时间碎片的问题对管理者而言是不可避免的，但可以采取方法更加合理的利用时间，将其影响降到最低。

I 制定规则

例如约定在指定的时间签单、讨论技术问题、反馈进展等，而不是随时进行。

I 琐碎事情一起做

对于工作中的琐碎问题，不用急着处理，可以启动“碎片整理程序”，将其记录下来，在你不需要“炒菜”的时候一起处理。

I 利用碎片时间

碎片时间并非不可利用，而是要安排合理的工作。几块大石头中间的缝隙，肯定塞不下另一块大石头，但放一些小石子或沙子还是没问题。例如与员工沟通、向领导汇报工作、检查员工工作、辅导员工、项目风险分析、项目目标回顾、发传真、收邮件等，这些工作就是小石子一样，利用小块小块的时间就可以完成。

