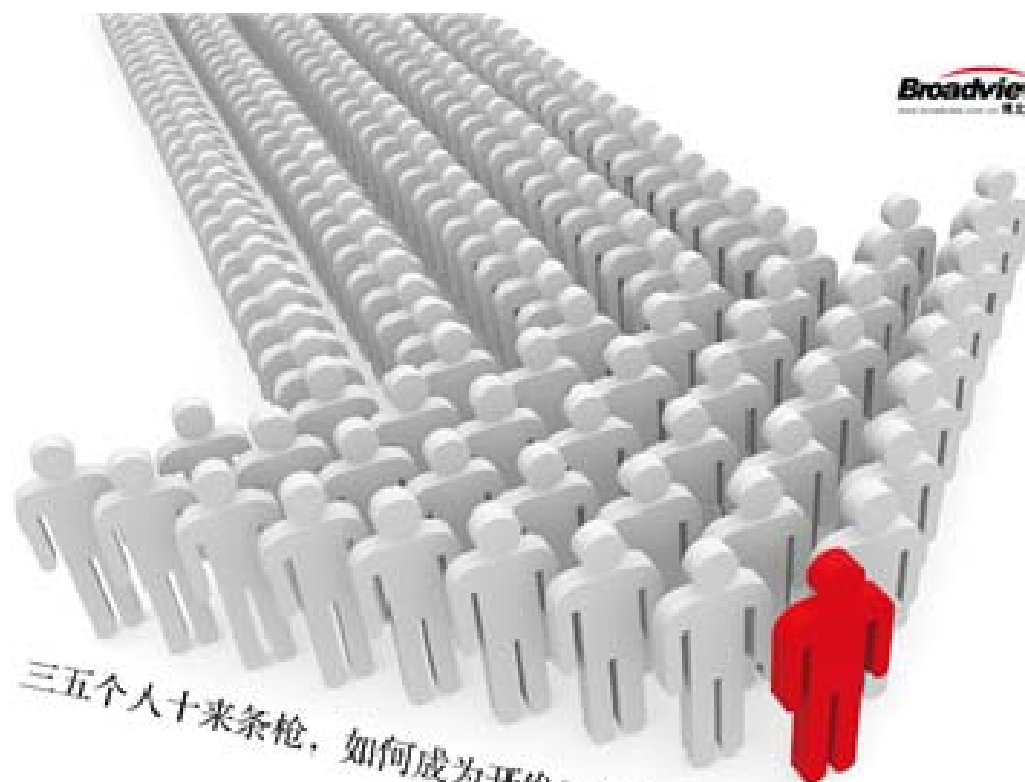


Broadview
博文视点



三五个人十来条枪，如何成为开发正规军

走出 软件作坊

阿朱 \ 著

电子工业出版社
www.pitp.com.cn

目 录

1、三五个人十来条枪 如何成为开发正规军（一）	4
2、三五个人十来条枪 如何成为开发正规军（二）	7
3、三五个人十来条枪 如何成为开发正规军（三）	11
4、人，是人，真的是人	14
5、习惯决定性格，性格决定命运，细节决定成败--实施经理的工具箱	17
6、客服顾问的工具箱	21
7、你这该死的销售	24
8、水清则无鱼	28
9、实施费用也能DIY	32
10、将服务费用DIY到底	34
11、物以类聚，人以群分	35
12、DIY报价	39
13、敢问路在何方	42
14、懈寄生	46
15、那根胡萝卜	52
16、七里香	56
17、走钢索的人	60
18、焦油坑	64
19、一个人的战斗	67
20、累斗累	72
21、我要飞的更高	75
22、波、波、波	78
23、八部众	81
24、葵花点穴手	86
25、文档知多少	88
26、狮面人	92
27、沙场秋点兵	96
28、代码那些事儿	99

29、风语者	102
30、蛋白质女孩	106
32、一分钟先生	110
33、灯塔客户	117

=====本书由dolf整理<http://www.idolf.cn>, 仅供试阅, 支持作者, 请购买正版=====

1、三五个人十来条枪 如何成为开发正规军（一）

自从发了上一篇博文，这几天收到很多朋友的来信。

大家从各个开发语言的优缺点和适用领域，一直讨论到设计模式、框架、重构、单元测试，乃至敏捷编程，最后都讨论到了软件开发过程管理，甚至都谈到了盈利模式和中国软件的悲哀。

最后不了了之，都觉得改善中国内地现在的软件生产状况不可能。

为什么呢？

我重新把这几天大家的讨论留言翻了一遍，发现大家的软件团队都存在着这样一种普遍现象大部分人所在的公司，开发人员仅 3-5 人，多的在 10 人。别看就这几条枪，还从售前支持，软件开发，测试、打包发布、文档编写、实施安装、培训、技术支持都做。

这还不算什么，而且几乎是一个人负责一个产品或一个项目，一个人从头跟到尾，而且负责多个客户的维护工作。

这还不算什么，而且随时老板会找来八竿子打不着的新活，要的还挺紧，突然要开发，打乱了所有的计划，最后都懒的按计划行事，每天撞钟，老板有事就吩咐，没事就上网，还不让听歌，当然更不让打游戏。甚至还不让看技术书籍，呵斥不干工作。只能上网装作在工作。

老板和员工互相斗智斗勇，在年终奖、报销、出差、平时福利上啊，都明争暗斗。老板卡的紧，员工就在项目和产品上下药，还不知道是谁占了谁便宜，谁给谁打了工。

员工一边在刻苦钻研各种开发工具，阅读源代码，学习做 DEMO 例子，阅读 UML、设计模式、单元测试、敏捷编程等等，一边却懒的修改现在公司的产品，有问题就打补丁，客户不嚷嚷就懒的修改，代码不优化，界面不友好，架构没架构，代码不封装但是，在讨论中，我时时都强烈感觉到，大家是想把产品开发好，把开发过程管理的井井有条，但是都心有余而力不足。阅读了 N 多软件工程的书籍，从重型方法到轻型方法都阅读了，但都无法把现在的开发状态一点点扭转好。

许多人想闹革命，把现在这些产品和团队都砸塌，然后重新来过，但这只是梦想，说说而已。只能希冀下一次跳槽，能找到一个好的公司，把自己平生所学全部发挥出来，但这好像也只是梦想，因为交流了一下，大家彼此的境况基本相同。

一些极端主义者自己开了公司，才发现不持家不知道油盐贵，现在自己和手下变成了老板和员工的关系，走了过去的老路。

更有一些极端主义者辞职，自己做软件，最后由于生活拮据或做做发现这个软件没什么意义，就丢弃了自己的梦想，随便找一家公司开始沉默撞钟。

一些聪明的家伙，有的入了外企，有的进了大的网游公司，有的进了外包公司，有的进了大网站公司，都是讲究大规模开发的公司，希望能找到一条中国式团队开发产品保证之路作为小软件公司，我们真的无能为力了么？我们真的成为炮灰了么？

但是，中国软件行业大部分都是这样的公司。从每年的 CSDN 的程序员调查都可以看到，中国软件公司大部分都保持在这种开发团队规模，开发人员大部分都在毕业 1-3 年。

我们是在等待时间让人变得成熟么？我们是在等待时间让人变得技术综合实力增强么？

依笔者看，作为中国软件群体最大的小软件公司，需要的不是 UML/RUP/CMM 这些重型方法，不是前几年大家关注的小组开发方法，也不是敏捷编程这样的结对方法，我们都无

法有这样的资源实现这样的方法。

但是，想想，星星之火可以燎原。红军能从爬雪山过草地起家，最后解放全中国。我们就没有方法？

那我们就需要想，就我们目前能拥有的权力和资源，我们如何一点点改进。我们需要的是从游击队到兄弟连，从兄弟连到正规军的方法。我们现在还处于游击队，一个队长领了一帮游兵散勇，有的人甚至没有枪还背着大刀，有的人还没杀过鬼子。

首先，要把我们自己变成兄弟连。

我常常观看国际著名的 CS 战队的比赛录像，他们配合的多好啊。如果他们都单兵作战，那么早就死翘翘了。这和咱们的软件开发多么相像。我们多么神往这种默契的配合，打的多么流畅。我们要的就是这个。他们也不几个人么。

那让我们来分析分析吧。

我们想好好专职的开发软件，但我们的时间都被实施安装、培训、技术支持占去了。为什么我们要做这些？是因为我们软件没有操作说明，其他部门人都不会用。而且我们也没有培训机制，其他部门人更不会用。而且我们的软件不稳定，其他部门人都拒绝实施。由于我们软件不稳定，老出问题，出了问题其他部门人也帮不上忙，只能我们自己去做好技术支持。

从以上来看，主要矛盾就是在：操作说明、培训机制、稳定性。如何保证这三点。而且从以上来分析，稳定性是最重要的。不稳定，你即使有操作说明和培训机制，其他部门人都躲着实施，谁想去客户那里尴尬丢脸挨骂呀。所以，其他部门人会找各种理由向老板告开发部的状，以躲避实施，说软件太烂，根本无法拿出去。这也就是开发部往往和其他部门关系都不好，开发人员老抱怨自己就闷头辛苦开发解决问题，没有人说好，却被奸人陷害。天长日久，积怨颇深。其实说起来，根源还在开发部自己这里。

如何保证稳定性？

大家第一想到的就是招测试人员。当然，一些公司的老板是拒绝养测试人员的。另外，如果你只想到招测试人员，其他方法不配合测试人员，即使有了测试人员，软件稳定性仍然不会有提高。所以，有一些工作，是不管有没有测试人员，都必须是我们开发人员要做的：

每个人的技术水平都参差不齐的，每个人对自己代码的负责认真性也都是不一样的，所以要想提高稳定性，必须专门从队伍中找一个人，他作为公共代码开发员。每个产品或项目的修改需求，必须首先经过他的思考，能做成公共代码，能封装成函数，就他来做。其他的程序员只管调用函数，实现客户 UI 操作和辅助功能。这个公共代码开发员必须具备以下能力：

参与过几个主要项目的开发、实施、支持。这样，他对客户需求有综合的把握。如果队伍中没有这样的人，只有开发经理一个人有这样的经理，那么接到客户需求，分析客户需求，分解析辨是公共代码员来做还是其他开发人员来做。

公共代码开发员具有负责认真的工作态度，代码细心严谨考虑周详异常保护做的到位内存创建释放有头有尾，代码优美，代码可阅读，代码重构，代码性能和稳定都高公共代码开发员的技术能力高，知道封装成什么样的函数接口，在灵活性，以后的修改变化性上最好应该说，找一个技术能力好的，工作认真负责的人，应该是不难找到的。而且专门做这件事，不让他参与各种杂事，他是应该能干好这件事的，而且会越做越好，这就是术有专攻。

刚才还讲到一件事，那就是开发经理要熟悉客户需求，而且是深刻理解客户需求。

客户需求，客户需求。这个让开发部最头疼的字眼。每当想起客户需求，就想起了以下这些话：

程序员说：这是你们家个性的需求，太邪门，我们不做。客户说：不做我们找你们老板去，我们是花钱买了你们的产品的。

客户说：我不会用鼠标，你给我做一个语音输入吧。我们还想要一个类似 QQ 的东西供

我们内部沟通，你们给我们做一个吧。程序员：我晕。

程序员说：等你们内部斗争完，你们协调完了，我再调研需求。

似乎，我们在需求上无能为力，我们永远在追赶客户的需求，满足他们的现状，把 N 多家的客户需求都加进软件中，只要能实现的，我们尽量咬牙实现了。

最后，我们发现，我们的软件无比复杂，谁也不会用了，连开发部门都不会用了，谁也不知道这个需求当时为什么是这样的。因为无比复杂，所以实施、培训、技术支持都成了问题，稳定性更成了问题。代码互相交叉，根本无法理清有多少交叉影响点。维护的程序员都快崩溃了，天天在祈求，千万别接到客户电话，千万别接到客户电话。

这个问题终归是问题，而且是软件开发最大的问题。虽然我们也动用了这样的技巧：

客户业务部门不能随便提需求。必须集中汇总到客户 IT 部门，由客户 IT 部门汇总过滤完，再集中报给软件公司客户 IT 部门的需求，必须客户方负责 IT 项目的老板签字才能生效，才能报给软件公司不能随时报，每 3 个月集中报一次不能口头报（即使在现场实施支持也不行），不能电话报，只能 MAIL 或传真来报必须按我们规定的格式报，要严格写清楚需要实现的功能的界面，输入数据或输出数据，输入输出数据的格式要求，谁操作，多长时间操作一次。

软件上线后只能免费修改 3 次。以后再有需求，就必须另签合同另收费，否则不予修改。

经过这么几招，客户也累了。需求是不提了，开发部欢呼雀跃。但我们真的做好了么？难道客户真的满意了么？客户为什么要用我们的软件？难道仅仅是为了把他们现在手工做的，然后转到计算机去做。让计算机的查询统计计算速度代替人工？

客户为什么要提这样的需求？客户要根本解决什么问题？这些问题谁来想，谁来想解决办法？

，My God!我们无能为力，因为我们是技术人员，我们不懂业务。

那这个问题谁来解决？

程序员苦笑了：没有人解决，也没有人能解决。客户就要，你不做他就要给老板打电话。

噢，那就让程序员的噩梦继续吧。谁也救不了你，能救你的只有你自己。

要救我们自己，必须我们自己走出我们自己。谁让我们就处在这样的处境呢？我们都想过的好，只能我们自己救我们自己。

那我们就鼓足勇气，走出来，从我们的设计模式、OO、软件工程、虚拟接口、反射、持久化、框架中走出来。开发经理来承担起客户行业研究来：

客户行业这个群体有多大？大中小规模各有多少家，各分布在什么省？我们面对的最佳客户是什么规模什么信息化程度的？我们的次佳客户是什么规模什么信息化程度的？

我们的上层竞争对手、本层的竞争对手、下层竞争对手目前的产品怎么样？他们各自的优点是什么？他们各自的缺点是什么？我们应该突出的优点是什么？我们的缺点是什么？

客户行业的过去 5 年，现在 2 年，未来 3 年的发展历史和趋势是什么？他们面临哪些挑战和机遇？

我们现在所做的典型客户，他们的组织结构，人员规模，每个岗位每日业务流程、每个岗位每日每周每月每季每年的异常处理业务流程，每个岗位每日每周每月季每年的输入表格，每个岗位每日每周每月季每年的常用数据查询，每个岗位每日每周每月季每年的统计报表针对以上的了解，客户面对未来挑战和机遇，未来应该如何变更他们的岗位和职责和流程，尽量流程少，效率高，运转快？

其实，开发经理就相当于业务架构师（因为我们还是游击队，不可能有专职的业务架构师），公共代码开发员就相当于技术架构师。

柳传志说的非常好：搭班子，定战略，带队伍。你班子不行，上什么需求管理软件、版本管理软件、项目进度管理软件、自动测试、自动集成软件，都是无法落地执行的。

有了夯实的业务+技术，功能实用、功能符合客户操作、功能稳定。这是软件最基本的要求，就都能满足了。这时候再招测试人员，就能把质量再夯实了。

而且，测试人员由于熟知产品，他们还能做技术支持呢，这样可以有更多的开发人员来专职开发，开发的专业性就能越来越提高了。

好的产品，还需要有好的文档和培训，否则其他部门还是不会接开发部的产品的。

那就招一个文案人员，写帮助说明，制作操作视频，制作学习版数据库，参与辅助测试（这个很重要，否则文案人员不熟知产品，无法写出有质量的文案）。有了这些文案的基础，最熟悉产品的非开发人员就有了两个岗位：测试兼技术支持，那么文案就兼起培训工作（由于他自己写文案自己用自己的文案做培训，在培训中会有各种提问，会更加增进他对文案和产品的理解，能写出更好的文案。而且他不是开发人员，他能站在使用者的角度上来写来讲，而且他属于开发部门，他会给产品开发带来更多更好的产品易用性建议）。

好了，开发部的四套马车终于起来了，这就是我要讲的开发模式：从游击队转变为兄弟连，从软件作坊走向记住：业务架构、技术架构、测试兼技术支持、文案兼培训，四套马车。

我们一直用它，效果很好，搭建团队容易，循序渐进不革命。

有了这么好的团队，就能比过去产出更好的软件，软件的质量，软件的进度，软件的竞争力就都上来了，再上各种管理软件：如项目管理软件、版本管理软件、BUG 管理软件、自动测试软件，就水到渠成了。

其他部门也愿意接软件了，软件的实施和培训和技术支持都被其他部门接过去了。开发部门也终于专职专业起来了，整个公司都很协调了，部门间也不互相陷害抱怨了。公司发展速度蹭蹭的。

老板看着形式这么好，也不抠门了。奖金福利随之而来。老板看着公司产品销售这么好，也不用再为公司生存发愁了，不用随处找单子养活了，给开发部门更带来了专业理顺的计划发展。老板也开始重视研发部门了，研发部门在公司的地位高多了，给与研发部门的资源和支持也更多了。

2、三五个人十来条枪 如何成为开发正规军（二）

上一次，写了一篇文章《三五个人十来条枪 如何走出软件作坊成为开发正规军》，反响异常激烈。

我的一个朋友也看到了我的博文，他是做某个行业企业管理软件的。他说：你这个方法，在我从事的行业不适用。

我对他从事的那个信息化的行业还是有一定了解的。

他们的实施模式是：

一个实施项目，大约 50 万的签单额，做完验收后给最后的 20%-30% 的尾款。

他们是一家小公司，为了多做项目多赚钱（企业都希望利润保持的很高，如果毛利低，做软件就不合适了，受的苦和压力和不规律性比其他行业多的多），所以一个项目只派一个人去，而这个人需要培训、辅助导入旧系统数据、清洗合并数据、规范化数据、报表制作、需求协调、推动切换上线、现场运行监控、个性化定制修改代码。

如果不能推动客户上线，就无法验收结项。不结项，就无法去追尾款。

一个项目这个人，身兼项目经理、开发员、需求调研、软件设计、功能测试、实施培训、定制化开发，还有时候写培训文档。因为公司里都是这样的人，根本没有分工出专门的文档人员，所以产品根本没有培训手册和帮助手册。除非客户必须要，这个项目的这个人才写一

份草稿应付。而公司又没有人来做文档管理工作，所以各个项目各个人写，也没有人合并，也没有人来统一收集。每个文档都在项目每个人的移动硬盘里。

由于项目就老哥一个人全活儿，所以自己答应了客户修改什么需求就自己修改，根本没有啥需求调研方法和版本管理方法，就看这个老哥和客户之间的博弈了。每个项目一套源代码，而且都在各个项目的各个人手里。返回公司后，往公司的服务器上一扔做个备份。以后谁的项目出了问题或需求，就谁负责继续修改。但是，很有可能这个人已经在做其他项目了，还需要修改前几个项目的需求或 BUG，还需要接听前几个项目的支持电话。如果这个老哥是在顶不住压力和焦虑而跑路了，只能把这些所有的活交给现存活的人的手里，啥也没有。无法交接也得交接，反正人要跳槽。

由于每个人都是这样一人挡一摊或数摊项目，而且项目周期长，每个项目都需要 2-3 个月的时间。老板也想把公司做大，但是每个项目能去实施的人，要求都非常的高，新人来了了一年也上不了前线干不了活。所以，对招新人也是不愿意招，干花钱没见起作用，小公司培养不起人。而对项目游刃有余的人，都是跑单帮跑惯了，带着个新人，还干不了活，还浪费出差费用，真是气死人了，还不如自己亲自动手三下五除二搞定爽。

于是，公司五六年了也就那么大规模，老板员工都干的很辛苦，当然老板得到的钱要多一些，赚个 500 多万没啥问题，自己后半辈子算是有靠了。所以，老板也得过且过，反正现在赚钱速度已经比较满足了，这样也熟练习惯了，经验路径依赖，就这样顺坡下驴做吧。

我的朋友是个理想和现实总是不断冲突的人。一方面，他确实想把项目做的很是顺畅，另一方面，他却觉得一切都像是被各种因素牵扯，根本无法转变模式，于是只能认命继续现在。

我说，你这种情况其实在中国很普遍。中国大部分软件公司都是从事行业信息化，因为这块技术难度最低，而且只要有人脉关系就可以做销售开干。而很多软件公司的成立，就是由于老板有一个关系，接到了某个项目，于是拉住了某个客户，小活不断，于是成立了公司。这是很多老板成立公司的原因。既然这类公司成立就没有目标，其目的就是认识几个人多拉一些项目多赚一些钱，所以如何复制模式，他们其实关注性也不大。原因很明白，就是自己不认识的客户，要想打入这个单子，很难，每个客户面前都有 N 多关系户。对于自己有关关系的客户，也就那么多个，有多大关系就能做多大的摊子，那就尽量从现有客户中持续做项目。维护好客户关系是最重要的。这类模式非常常见，并不是你这个行业特殊。

老板的生活已经趋向于小康稳定，而你呢？你还在挣工资。你也在一线客户那里天天呆着，要么你把老板的客户抢过来你做，要么为了你自己工作能轻快些，你必须自己给自己找方法。

我的朋友说，抢过来不可能。自己虽然天天在第一线和客户天天在一起，关系也处的不错。但现在人先认的是钱，后认的是感情。而老板给他们这帮人都持续吃喝玩乐送东西分回扣，自己只是一个干苦力的。自己只能找方法。但你说的方法是针对一个公司的变革，不是针对我个人而言的，所以不适用。我想有一个方法能帮助我自己的方法，你帮我想想。

我想了想我过去写过的文章，确实是，自己一直从事职业经理人操盘产品研发管理，也统管咨询、实施、培训、支持，但都是在公司管理的层面上看问题分析问题解决问题，而没有从一个个体上去思考。而中国，大量像我这样的朋友，他们需要帮助，而我写的却是公司层面的，无法帮助他们，所以他们老说我的文章空洞、理想。

我说，咱们俩一起分析解决。也是给大量像我朋友这样辛苦的人带个福音。

咱们首先先说一下你想达到什么效果。

我朋友说：我现在在这里待的很烦，出差时间太长了，我就想早点回家。

那你什么地方费时间了，需要 2-3 个月在客户现场？

我朋友说： 嗯，我看完你的那篇文章，我也做了一下反思和总结。我感觉有三个方面

特别费时间：客户需求，数据准备，报表制作。

一去客户那里，你是见不到客户老板的，也是看不到用户的，你主要面对的是客户信息科的人。他们一开始要求你先做演示，看看是否符合他们本企业使用。在这个演示过程中，就不断提出需求让你修改。而且，你不修改完，他们没法接受你以下的演示，说想象不出后来的样子，对着你画的界面图想象以后的功能变化，有点纸上谈兵的感觉。而且，往往演示的时候必须信息科科长在，否则底下的科员都做不了主，演示了也是白演示。而信息科科长却老不在。而他们上班时也极为规律，该下班时立马下班，根本不加班。所以边演示变修改再边演示。好容易修改完了，也演示完了，时间一俩个星期就过去了。

信息科算是通过了，就需要录入基础数据了。问题又来了。现在大部门企业都已经上过一套软件了，可能是 Foxpro 的，也可能是 PB 的。人家要求你把数据倒进新系统中，但是一看过去的的数据，都乱七八糟的，过去上线都是没经验，后来也用的乱了，积腋成疾了。现在要导入，真是要把垃圾输入，得出来的也是垃圾。你苦口婆心的说服让他们重新录入，但是他们一看都好几千条，不想录入，让你能导多少导多少，然后在基础上再维护。这一松口不要紧，你不仅忙活了一个多星期写各种 SQL 导数据，而且往往旧系统也没有文档，数据结构需要你自己理解，理解有误也是你的事。好容易导完了，再维护，发现数据是通过 SQL 导入的，在界面上却不能维护，因为很多校验都是写死在程序里的，而不是约束在数据库。磕磕碰碰，自己边后台修改数据，边让他们信息科维护。他们信息科首先检验导进去的数据对不对，没有填写齐的字段填写齐。然后把没有导进去的数据录入进去。然后再打印出来，统一对一遍，看看哪些数据录入的有错误。这样折腾，一个月，22 天工作日就过去了，用户还没培训呢。

第二个月开始用户培训了，但一培训就发现了问题。用户的需求和信息科所的需求，根本不是一码事。原来一个企业，信息科也和业务科室是两张皮，就和在软件公司一样，开发部和销售部是两张皮。于是，用户和信息科开始吵架，各说各的道理，谁都在维护自己的利益。而且用户部门有业务在身，也不可能天天大部分时间泡在 IT 讨论上面，开会不来人，或者要来人也来了个小兵充数，根本起不了决定，还提自己的意见，过几天开会，用户部门的主任来开会，又把需求再推翻。业务部门主任是站在主任的层次上看 IT 管理，而业务部门科员是站在自己轻松使用的角度上提需求，而信息科是为了自己以后维护着想。不断的讨论不断的推翻不断的扯皮。

讨论扯皮推翻再讨论再修改。终于消停了。开始培训了。但问题来了，用户上机一操作，发现基础数据很多不是平常现实那样的。计算机数据过去就和现实数据脱离了，现在想借新系统上线再回到计算机管理上。于是，一边培训一边修改数据，有人报告数据错误就修改。而培训没有文档，培训也没有课程，培训也没有专业训练。培训如何层层开展，培训如何组织，都不知道。反正就老哥一个被订在这里了，只能这么上手了。人没有来齐，也得开始。等来了再重新讲一次。一次不会，再讲一次。有人年龄大打字不熟练，有人看不清屏幕，需要调整大字。一调整，界面发生错位了。有人不会用鼠标双击和单击，有人不会控制打印机。过去是 UC DOS 系统，还没用过 WINDOWS 的，用不惯。从基础培训起吧。否则怎么办呢？人家不上线用起来，人家不给验收结项啊，尾款回不来啊。

用户也培训完了，该上线了，就需要初始化库存了。先得现实盘点，然后再录入计算机，还必须一边得继续营业。于是，真实库存和计算机库存肯定对不上去。由于品种太多，所以只能一批批盘点一批批录入。

由于操作不熟练，特殊业务不知道如何处理，只能瞎处理。处理完后发现不对，想冲抵回去。没有冲抵功能，只能修改数据库中的数据。

由于前期修改，根本没有测试，就是老哥自己一个人改。改完了有时候烦了连自己都不想测试。于是上线用着用着就不能运行了，需要当时就立即修改，中午晚上的连续作战紧急

解决，否则第二天一早还需要开门迎客。

好不容易业务录入了，但是报表不对。一检查，原来是前段时间录入的非法业务数据造成，功能没想到没拦住。怎么办？偷偷自己修改数据，然后使报表平账。过段时间，发现报表又不平了，发现还是非法数据进入造成。怎么进来的呢？想不明白。只好蹲点现场，直到客户都运行正常了才能走人，算是上线成功。

这个累呀，两三个月都是紧着过的。好不容易回来休息会，另一个项目就要启动了，就这么几头能干活的蒜，老板笑着脸让你去。于是，遭遇再次上演，日子就是这么过来了，一月又一月，一年又一年。顶不住的就跑路。

我听完了我的朋友们的诉苦。我说咱们一件件事情的排查。

第一件事，边演示边修改，还得信息科长在，还得他拍板。这段时间的浪费如入缩短。我过去也作过灯塔客户的实施，我过去一去了客户那里，并没有一开始就这么做。我先是调研此次项目组的人员构成、能力、职责、上线时间、用户计算机能力、用户部门对上一套系统的最突出的抱怨，信息科对上一套系统的最突出的抱怨，现在还有哪些系统在持续运行，上一套系统用户部门和信息科觉得哪里好，上一套系统的功能结构和操作流程。这样，我就确定了我该如何开展项目实施。这就是项目调研阶段。人往往很眷恋自己已经习惯的事情。而且人的想法，人的能力，各个部门的利益冲突，人和人的私人关系和恩怨，都有助于项目的推进。亚洲人做事，需要面上的和面下的都得下功夫。纯粹都是正式的或者都是不正规的，都无法做好一个项目。我会在项目调研后，重新建议项目组人员构成、职责、流程、项目阶段时间、各方面负责人、本项目的最突出要解决的前 5 个目标问题。

人常说，上下同欲者胜，庙算者胜。你一开始必须界定好项目的边界和目标和执行标准和责任人，否则大家谁都想管或者谁都不管，大家没有目标，或者大家各有各的目标，肯定无法项目很好的推进。

有了目标，责任人、标准、时间计划，就要按照这个目标来分解做。基础数据的校验，需要用户参与先来校验原有系统的数据。不要认为用户现有这套系统就没有问题。如果没有问题，企业也不会用你这套来代替他现有这套。所以校验现有基础数据很有必要。没有的数据，先让他们做准备，但你要书面把要准备的规范写好交给要参与的各个用户，而且要做好培训工作，不能讲讲就认为他们理解了。有了的数据，需要校验。地基打好，才能上面很快盖房子。而且，信息科和用户对老系统很熟悉，校验数据比你快的多，而且准确的多。只有经过他们的确认，你可以导数据，也可以不负责导数据。其实，基础数据，虽然多，但只要 有 5-10 个人，2-3 天就能录入完毕。比你导更快更准确。

在用户嚷嚷需求的时候，一定要以系统目标为约束。因为每个人看法不一样，站的利益角度不一样，每个人的计算机应用水平也不一样，所以每个人都有看法。你让百家争鸣，而且什么需求都可以提，没有目标没有边界，就让你一个人修改，那么你结果不会好而且你会心身疲惫，你会很快就厌烦了项目。吃力不讨好，就是方法不对。需求，一定要围绕时间阶段和目标为约束，大家要一个目标。

还有你刚才讲到的没有培训方法、培训文档、培训素质，说明必须要有专人来做培训。这块是项目实施非常重要而且工作量大的一块。这才是真正的项目实施。项目实施不是让你来修改需求来了。培训做不好，上线会出一堆麻烦，软件再约束不强，报表就是平不了。而培养一个培训的人员还是容易的。如果想培养一个会协调推进来事的、会修改软件的、懂得业务需求的、会 SQL 语句导数的、会培训的，这样的专业神人确实很难。而且这样的神人一定不专业。所以，要带人，先要让他搞培训，而且让他编写针对不同用户的培训手册，有培训时间课程、培训规范、考试考核、模拟练习环境、模拟数据。这是这个培训专员可以做到的。

软件修改，尤其项目型软件，不修改是不太可能的。我不赞成在客户处修改软件。因为

不仅仅你只会以事论事的修改，容易陷入到这家客户具体的需求中，而不会考虑其他客户的需求兼容，所以你修改的软件有很大局限性，最后形成只能一家维护一套代码，最后客户越多越累成本越高越不赚钱，被客户多而拖累死。而且你在现场那么多事情，那么多人打扰，你根本无心踏实下来修改软件，只想着赶快改完上线回家，你急躁，潦草，应付，软件质量就没法保证了。你想改变这种现状，你必须把需求整理好，交给在家里专门编写代码的程序员。你在现场，你也很懂业务，你和你本公司的程序员沟通肯定比客户沟通要顺畅的多。

这样，你在现场，你的任务就是当好一个项目经理，专门协调，控制，理顺，制定流程、规范、目标、时间，保证执行到位。现场还有培训专员分担你的培训工作，可以帮助你校验数据，测试功能。公司里还有专门 coding 的程序员分担你的开发测试工作，而且人家写的代码更加多家客户兼容使用，而且质量稳定性比你高。

只有专业分工合作，才能转成正规军。否则，你只能把自己熬倒了，心力交瘁，最后心灰意冷，跳槽而走。

从民兵，到武工队，到土八路，到正规军。这条路有好几个阶段。不能想着一步到位。现实情况也不容许我们一步到位。我们只能是能改进什么就改进什么，天天进步一点，我们就会大变样。

如果从心里就认为不可更改，直到心冷不想改进，那么我们永远不会进步。

为了我们自己心身愉快，我们也要进步。

记住，你是项目经理。你是这个项目的领头人。你决定这个项目的成败。

如果你连这个定位都没有，那么你什么都决定不了，你这个项目的成败只能随波逐流，那样你真的很失败了，你什么作用都没有，要你干吗。

3、三五个人十来条枪 如何成为开发正规军（三）

自从写了关于《三五个人十来条枪 如何走出软件作坊成为开发正规军》走出软件作坊：三五个人十来条枪 如何成为开发正规军（二），系列文章后，收到了很多网友的评论，也收到了很多网友的疑问请教。而大部分人都已经当上了项目经理，手下有个 2-3 个人或 5-6 个人。少部分人还在上学或者才毕业出来 1-2 年，询问的还是学什么语言和什么才是核心技术的之类问题。

从接到的请教来看，许多中国国内软件公司都是以项目为主，有单做单，没单就干靠，靠的时间长了老板心毛了就裁人，来活了就招人，就这样反反复复。所以，大量的公司没有开发部（因为除了销售，开发部从开发到实施到支持都全做），当然也没有开发部经理，只有项目经理。更不用提技术总监和 CTO。即使有个技术总监的头衔，也是为了给客户的名片，而手下也就 5-6 个人，项目一来，技术总监也需要编码和实施，其实就是一个项目经理。

在国内，项目经理这个词如此常见。均为实施项目经理和开发项目经理混为一身，统称项目经理。虽然，开发和实施是软件产品的不同阶段，不同阶段关注的重点也有不同。但既然都为项目经理，那么其关注点也有共性之处。

项目经理，主要职责是：

项目范围定义项目计划制定、分解、分配、协调、汇报项目质量控制项目需求变更控制国内项目经理一般没有人事权和财务费用权。老板给分配什么人就带什么人，自己只是一个最能干的工人加工头而已，当然更没有财务费用权，要想请客户吃顿饭，当然需要和老板打报告（自己团队想休息娱乐会，只能联机打把游戏，想团队吃顿饭，不可能给费用的）。

不过，从现状来看，国内现在的项目经理，连项目范围和项目需求都无法控制。老板说

什么就是什么，客户说什么就是什么，用户说什么就是什么，只要自己和自己的团队能做，并且不累死或者不跑路，能做的都照单全收。当然，做什么，什么时候做完，都不属于自己管理和控制，当然，项目计划的制定由项目经理制定，就是虚设了。唯一剩下的，就是项目质量控制：开发有代码的质量，实施有实施的质量。

接到网友很多询问，都问我工具的使用情况（对组织结构和流程问的极少，可能觉得都是自己改变不了，根本没有机会实现，道理能不能行的通也就不去想了，因为想了也白想）。问我现在的团队使用什么 UML 工具、什么压力测试工具、什么数据库设计工具、什么版本管理工具、什么需求管理工具、什么进度管理工具、什么 BUG 管理工具。

在他们眼里可能觉得，一个团队，只要用上先进的工具就会成为一支装备了机关枪的军队。就跟我们的客户一个想法，只要上了这套 ERP 软件，我们的管理就上了一个台阶，我们的盈利就会提升。这个想法，真是奇怪，就如同一个人拿了一把屠龙刀，人没砍到，倒是把自己砍伤了。一把好厨子的刀，到了不会做菜的人的手里，仍然做不出好菜，就这么浅显的道理，但大家还在幻想。

许多人想得到答案，觉得一个正规的开发团队应该使用是 Rose、Together、LoadRunner、PowerDesigner、VSS、CVS、SVN、ClearRequest 等等。

但其实，我们也并没有使用这些工具。

我一直在商业软件公司工作，也深深的明白自己的责任就是帮助公司最大限度的利润最大化。而利润最大化的实现手段就是最小的成本、最少的人、最少的时间、最简单的方法达到老板的目的，拿出合适质量和功能的产品，包装好，卖上尽可能高的价格。只要能赚到老板想赚到的钱，达到老板的目的，只要不影响这个目标，不影响大目标，小磕磕碰碰自然难免，有问题解决问题，没问题继续前进。哪个企业没个矛盾没个利益集团，哪个企业没个问题没个埋怨，有人爱自然有人恨。就是这样，这样是常态，不是异常。所以，我使用工具，一般都是在各种手段我都使用的差不多的情况下自然使用的，而非为了正规而正规，而是为了解决问题，而且是很有效的解决问题，而且是最简单的解决方法。我从来不为面子工程付出成本。

我们最先遇到的问题当然也是软件质量的问题。软件的质量问题，引起了实施培训、实施推动上线的困难、客户使用效果的困难、支持费用的增高、支持难度的加大。最后实施部不愿意实施、销售部不愿意销售、支持部直接把电话转开发部。所有人对把自己工作的不顺利和不顺心归罪到开发部。当然，这样的开发部，不被老板开掉才怪。

于是我空降入主了。

我采取的第二个策略就是：专门划出一个辅助开发人员（因为他对客户需求也不了解，讲了 3 遍也不懂，写的代码也考虑不周全，所以代码漏洞百出。不过这个小伙耐心还不错，就是有些懒。看来懒人一般都耐心不错。不过还是有些得过且过，做一天和尚撞一天钟。就这么个才。），让他做技术支持兼测试。

过去是实施部有不少人，每个人都直接打开发员的电话。支持部也是。客户也是。老板呢，不懂软件也不深入操作研究软件，却从使用者角度老提意见，看到哪里想到哪里就直接给开发员打电话让开发员修改，从最皮毛的字的字号到最深入的商业智能问题，都提，而且让立即改掉，其他所有人包括客户提的都靠后。这样，一个开发被干扰的无法工作，最后离职。

我划出开发部专人支持后，规定流程。所有的需求，不管是哪个部门或哪个客户，都归口到他这个人手上。即使还有人直接打给开发员，包括老板打给开发员的，开发员必须把需求或问题再并口到这个技术支持手里，我来统一安排调度开发。

开发人员是消停了，可以安心按我的安排的进度和优先级修改了。而支持小伙子呢，电话开始被打爆。幸好我给小伙子的指示是，都先接上记录好，能不能解决，能不能快速解决，

看自己能力，不着急，谁跟你急，你跟我说。于是，小伙子被吃了一颗定心丸。

小伙子一开始使用的是一个 EXCEL。别人提的问题都自己记录在里面。但是弄到最后，我的手里、小伙子手里、开发人员手里、支持人员手里，都出现了不同版本的 EXCEL。互相都说这个已经修改了，那个说没有修改。这个说有多少 BUG，那个说不可能。

于是，我上了第一个工具，BUG 管理系统。不管是 BUG 还是需求还是建议还是疑问，谁想提，都提到这里来，随时记录。不管你是出差还是在支持部坐班，都记录到这里来。凡不记录者，一律不解决。

于是，天下太平。经过技术支持和开发人员努力，一个大风浪过去。利益冲突处于一个平衡或者可能随时崩塌引来下一次冲突。

我于是给支持小伙子分配了另一项重要工作。测试。为了不让你以后继续享受折磨，那么你必须卡好关。你自己卡不好，那么以后的技术支持仍然很痛苦。小伙子为了自己以后能过上幸福的上班生活，于是测试做的不错。所有测试出来的 BUG 也记入到 BUG 管理系统。现在，开发人员工作量和 work 质量有了量化，支持人员的工作量和 work 质量也有了量化，给我安排计划和考核人员和申请资源做了大量的支持工作。

所以，一个 BUG 管理工具，能把计划、进度、质量、需求、BUG 都能管理起来，而且能追溯，能考核，能统计工作量和 work 质量。真是必备。

但是，接下来发现了一个问题。就是在修改的时候，老误会客户的需求。程序员一天在家里面开发，不了解外面的客户和在第一线战斗的实施人员到底想表达什么。于是修改完，程序员觉得自己费了很大的劲，而实施人员和客户却非常恼火，一点不领情还发怒。最后，搞的开发人员和实施人员冲突不断。

需求如何描述清楚，成了必须提上日程的事情。许多没有经验的项目经理尤其会在这一步犯晕。UML 工具、数据库设计工具，需求管理工具，能上的都上，最后也没解决问题，把自己和自己的团队累的半死。

我使用了 PPT+WORD+脑图+EXCEL 的描述方法。

因为很多需求都是这个支那个叉出来的。程序员往往想的了这头想不了那头。这就是人的思考的周密性差异。

想让人能从千万丝缕中理出头绪，于是脑图软件上场。把各个分支来龙去脉表现清楚。

到了描述某个节点的时候，PPT 上手。一页 PPT 相当于一个界面窗口。每页 PPT 的图形模仿了菜单、输入框、按钮。按钮按下，还可以跳转到其他的 PPT 页上，和软件操作流程非常相似。

让程序员很直观的看到未来软件作出来是什么样子。关于 PPT 的详细描述，如字段，流程，特殊注意，特殊控制，都用 WORD 说明好。

遇到有报表功能的时候，用 EXCEL 把报表画出来，让程序员喜闻乐见。

这样，从表及里，从概要到详细，从分支到关联，都表述 OK。客户也能明白，程序员也能明白，实施人员也能明白，老板也能明白（这点非常重要。虽然老板不懂软件，但他要干涉软件，他如果不明白，他就不知道这帮家伙到底在干吗，是在真正干活还是在偷懒，到底工作量是大是小，软件功能是复杂还是简单。老板如果不明白，老板在给与资源和时间上就会很谨慎，处处提防。这是许多项目经理都忽略了大事。还拿 UML 做秀，谁也看不懂，谁也用不了，白花费时间画那些好看的图。这就是中国的现状，我们站哪个山头就唱哪个山头的歌，有效解决问题提高销售收入才是我们的根本任务，我们不抱怨不幻想踏实推进解决问题）。

于是，老板的天平开始向开发部倾斜了。资源，当然就容易申请了。

画这些 EXCEL+PPT+脑图+WORD，当然很费时间（我直到引进了日本外包开发过程管理才发现，我们的解决方法和强调质量的日本人的做法非常相似）。于是，我申请一个人，

把过去实施的一个项目经理（还居然会写点 SQL，从数据库查数据，调整个报表。实在太强了）调入开发部，专门编写这些文件。

开发部开始蒸蒸日上。项目经理、开发人员、测试兼技术支持已经到位。工具也已用的不亦乐乎，深入到了公司的每个部门。每个部门都按照标准描述方法和标准流程走。现在，连实施人员都会画 EXCEL 报表格式、PPT 界面。

软件到位，就需要包装，否则软件就卖不上好价格。这是很自然的事情。干啥都要个品相。漂亮的姑娘谁都喜欢。

软件包装，第一步就需要帮助文件、视频操作、解决方案、产品介绍、演示系统。当然，文案人员很快到位。美工美化也自然到位。能多赚钱干吗不做，老板也不是傻子。谁喜欢卖一个土灰土脸的产品。

有了好的产品，出不去开发部也是个问题。只有自己内部人知道功能怎么用，怎么满足客户的需求，其他部门都不知道。许多人都不知道新功能和旧功能的改变。文档中都写了，更新说明也有，就是没有人看。还是打电话找技术支持，技术支持只能不断解释。问题又来了。

文案出马。每次版本发布，功能更新，文案反复举办集中培训，办班，一批次一批次的培训，百其不厌。

四套马车，于是真正的天下太平了。

从此，开发人员和实施人员过上了幸福的生活。

后续记：

接到很多网友的评论，都说老板不可能给资源的。说我写的太理想。

嗯，如果你看完我的文章就直接找老板要资源，当然是会被赶回来的。因为，你什么都没有做就开始要资源。

有人还说，公司就这几条枪，能干活的更是那几头蒜。根本不可能给你派人。

嗯，如果你思考的目标不是为老板赚取更多的钱，那么老板不可能给你一丁点的，甚至还会把你干掉。如果你觉得，这样的老板我还不伺候呢，那么中国大部分都是这样的公司，除非你转行不干这行了。要干，就别混日子。想得过得且过让老板公司倒闭，这个基本不可能。再说老板倒闭了对你一点好处都没有。

迈出你的第一步吧。不迈出第一步，你都会觉得这是不可能完成的任务。

想过幸福的生活，从现在开始脚踏实地的动手吧。

4、人，是人，真的是人

有网友评论我之前的几篇博文：分析的不错，方案似乎也很能解决问题！不过必须满足一个潜条件：一定要找到非常合适人。现实中，就连最基本的程序员，找个合格的也不容易（聪明伶俐的养不住、经验丰富的养不起、迟钝呆傻的没法要、碰上心术不正的还够你喝一水壶的）还有网友评论：楼主所说的很多方法，都是假设了客户还不错、对项目的重视程度、习惯于正规化的程度都还过得去，而楼上有些朋友的质疑则是指出这些资源不一定满足的情况；但是跟贴最多的评论就是：现实问题描述的很精确，但解决方案不现实，太理想化，老板根本不可能给你人。如果真的发慈悲心，也是给你一个新人让你哭死。你想主导项目，省省吧，死了你的心吧，一切都是老板说了算。而且，你敢和客户说个不字，看来你是不想要你的饭碗了。还是乖乖敲好你的代码，多干活，多跟客户搞好关系。高手做啥都是高手，低能再培养再有方法管理他也是低能。你这样研究，只能吃饱饭瞎想瞎扯蛋，有你这工夫，早

就把项目做好了。

种种评论来看，一切的根源都是人，是人。大家都觉得我的方法要想实施，必须老板支持，员工也是高素质的，客户也是高素质的。而三者要想凑到一起具备，根本不可能。所以我的方法算是理想的痴人说梦。

能支持的老板从哪里来？高素质的员工从哪里来？高素质的客户从哪里来？好像一切都是运气而来。好像我们就有高薪能聘得起高素质的员工。好像我们的产品面对的就是高素质的客户。

但我回顾了自己 10 多年的从业经验和管理经验，我并没有发现这个规律。我并非供职国际巨头公司，也不是国内知名企业，只是信息化行业内略有名气而已。手下很少出现名牌大学的员工，也很少能达到所谓的高薪（我自认自己还没有在马云、史玉柱、牛根生、柳传志这样大胸怀大眼光的企业家手下任过职，我们所从事的行业信息化也不是暴利的行业，大家也都知道管理软件没啥暴利，定制化修改、实施、咨询、培训、支持占去了很多成本。），我们的客户也是各种各样的人都有，从挖煤暴发户的私营老板到死气沉沉勾心斗角的国企，我们的客户千奇百怪。

在这样的环境下，我能把方法用起来，我和许多网友也交流过，最重要的是我认可了以下观点，这就是一个职业经理人和老板的关系：

老板都是疑人也用用人也疑。用人不疑，疑人不用，我不奢望。

再劳苦功高，我也只是职业经理人，我不拥有这个企业的哪怕 1% 所有权，所以我做好职业经理人本分，老板的归老板，职业经理人的归职业经理人。职业经理人的职责范围的，老板权力范围的，不要超越，也不要动歪脑筋。即使公司大部分的收入都是你研发的产品带来的。

计划不计划一件事情，执行不执行一件事情。一定要以老板利益为目的。老板不赚钱，一切好事一切好想法都会被老板推翻，老板就是老板。老板赚钱赚的眉开眼笑，其他的事情就好办的多。这是很多职业经理人居然都认识不到的，他们总抱怨老板限制太死，什么资源也不给，干活还贼累。根源就出在这里了。想实现你的想法，必须在实现了老板想法和目的的前提下才能做。所以我的方法能实现，多靠此。

而且我的方法不是为了我自己有什么好处，我的每一个方法也都不是为了正规化装修门面图好看。我的方法都是为了解决实际问题，为了老板赚钱更快更省成本更容易，员工更省力，客户更满意，而且每个方法都是在本企业能力和成本范围内能执行落地的解决方案。这样的解决方案，哪个老板会不支持呢。但很奇怪的是，很多研发部主管都忽略这个重点，老板在想利益，他在想技术。两人说不到一个目的去，互相不理解互相不支持互相埋怨，久而久之互相猜疑互相提防互相留一手。其实技术就是个手段，赚钱是目的，双方一起绑定去赚钱，怎么合法的赚更多钱怎么来。如果研发主管能脚踏实地的从本企业的能力和困境和现状去思考改进方法执行落地，而不是抱怨这样的环境没法实现想法消极怠工或心想跳槽，我想很多心结就都打开了。

只有和老板具备了这样的距离和关系，我的方法才好实施下去。所以，很多人觉得太理想化，就是和老板没有找到自己的位置。

但是，即使有这样的基础，要实现我所述的方法，也需要其他的环境支持。

我个人是这么看的：

好的氛围，才会引入、留住好的人（乱世强盗多就是这个理）。

好的人，才会有好的制度，并且保持好这个制度（制度是人定的）。

好的人和好的制度，才会遇到好的客户（有句老话，夜路走多了总会遇见鬼。有些人老想着邪门事，最后也被邪人玩。近朱者赤近墨者黑，什么人总遇到什么人，就这个道理）。好客户就会产生好的结果。

所以，好的人才，好的客户都不是运气来的，而是来自你自己。你就是控制源头的人。如何制造好的氛围，我讲讲我的职业经理人管理人的一些心得：

师傅制。这里没有总监，没有经理，只有师傅，老师。总监，经理，会让员工产生隔阂，距离，权力争斗。每一个人总有一个师傅。每一个新人进来，都要指定一位合适的师傅。尤其是新人，更要短期内注意看时候合适，不合适就要更换合适的师傅。什么问题都可以问师傅，从技术到公司制度到公司新闻公司历史到职业发展规划到个人生活问题。团队的凝聚力，配合性，归属感，责任感，很多问题都被人的感情消化了。

朝九晚五，禁止加班。其实大部分程序员也是不喜欢加班的（不过有些程序员是光棍，也是漂在北京，反正也是一个人，于是就喜欢呆在公司上网玩游戏看小说看电影吹空调，美其名曰加班。还有一类老板喜欢看表面功夫，谁加班就喜欢谁，于是大家都装做很忙都要加班）。因为加班不给钱。不给钱，还加班，天长日久就觉得自己很亏，心里不平衡，各种心思就都有了。其实也没有多大的事。我的老板一开始对我的不加班也是心存戒心，但是每次交给他的结果比加班的部门做的都好都快，他也就默许了。

良好的办公环境和良好的个人形象。我们看到美女就兴奋的口吐莲花，我们看到阳光溪水草地我们就心情舒畅。当然，我们看自己，别人看我们，都是一个道理。心情好，工作才能好。一个满桌狼藉充满烟味饭味脚丫子味有人在冥思苦想解决问题有人在打游戏有人在放朋克音乐有人在骂有人在打闹嬉笑有人把脚放到桌子上的办公环境，我看谁都会逃离。

以更快更省成本更容易完成任务为目标，以赚更多钱为目标，以提高产品质量产品价值产品售价为目标，鼓励员工进行自我岗位上的改进创新，我经常给与交流和指导，一旦有效，进行精神或物质的奖励或职位提拔或工资晋级。

好的氛围有了，就需要有好的人才。以下是我引入好的人才的几个心得：

人的年龄和工作经验拉开距离。年年招，时时招。不断看人，试人，滤人，培养人，形成层次感有阶梯有接力的员工组织，绵绵不断前赴后继，不会出现人才地震、集体疲劳、小团伙争斗。避免不同高低职位上全是 80-84 年的人。下属还在窝里斗互相不服（很多员工不看对方能力，就看对方的工资和年龄。凭啥你就是我师傅？），那么客户逼你，老板压你，其他部门利益冲突你，下属还闹你，你这个孤独人算是失道者寡助也。

人的技术能力高低先放一边。首先要过 EQ 关。有些中小型企业没有 HR 经理，一般考察 EQ，都是老板把关。如果你现在招人没有老板把关，那么必须先考察人的 EQ，再考察他的技术能力。我最怕有些羡慕科学管理的管理者照搬什么 EQ 测试问卷或什么团队游戏来评测。我的评测方法仍然是不讲道理，要讲经历。没有工作经历，至少有学习经历和生活经历吧。一个人的情感、压力、正义感、真诚感、领悟力、心细观察力、思路整理总结能力、关注全面平衡能力、执着力，都能看的出来。

招聘程序员也得看这些。我曾遇到一个程序员，思维混乱所以代码也混乱，思考也不全面，程序到处都是漏洞，思路也不自我整理总结，无法举一反三，给他讲了多遍的需求他都无法自己重述，一有了问题很急躁说搞不定了，一看还是很简单的问题，把错误提示原模原样输入到百度中查百度就能搜到好多，你说这样的程序员算技术合格吗？

其实，试用期的三个月就是主要看他的 EQ 和他的技术能力、理解学习成长能力，而不是片面只看他的现状技术能力。一个不愿意学习钻研，没有方法钻研快速学习理解，推一下动一下，或者怎么说都理解不了的，都需要统统辞掉。另外，对于心术不正有仇必报不服管教之类，早就扫出门外。一个讲究吃穿用享受或者满口脏话习惯毛病一堆或者不孝顺父母或者满口介词的人坚决不能要。

专业发展，流程协作。如果不专业化，老板有什么活就分配什么活，时间短了还认为自己是学习更多知识在锻炼。时间长了就会觉得自己就像个混子，干什么干过，但什么都拿不起来。出去应聘啥职位，是应聘开发呢，项目经理呢，实施呢，支持呢，销售呢。啥都

做过，但啥都没做专，都了解个皮毛，真要让上手还真给人家拿不下来。心就慌了，觉得自己是个被老板困在手心的小鸟，无法飞出本企业的樊笼，一旦飞出就要饿死没有能力存活。好可怕。难道只能在这家公司耗死了？赶快能逃逃吧，逃到一个正规的专业的公司去。

下一阶段目标交流制定。交流，我想每个 CTO 或技术总监或研发经理都会做。交流可以了解员工的困难和心中的疑惑、个人期望、个人专业兴趣的变化、人生观世界观技术观管理观生活观（以调整自己以后和该员工如何交流、如何讲解工作、如何鼓励、如何布置任务、如何考察等等）。交流也可以让员工多了解自己是怎么想的。双方在日常很多事情的分歧和误解就会消除，心会往一处想，劲会往一处使。但是，交流也不仅仅实现这些目标。更重要的是，交流，主要为了能给该员工制定一个切实可行的、某段时间段内可达到的、他也喜欢也愿意努力的、也会他未来职业发展很有好处的职业目标。没有目标的工作，虽然他很努力，但是他容易迷失方向。如果他又是一个不能很有悟性很有规划的人，他的工作就会形成做一天和尚撞一天钟。撞钟撞的不错，但没什么更高层次的提高。天长日久，就会木然，倦怠，不思进取，思想守旧，遇到新问题无法突破。所以，我会根据双方的交流，和员工一些协商一个下一阶段的职业发展目标，并且时常指导调整他的做事方法和思考方法，给他讲解一些我过去的工作经验和我的感受，鼓励指导他们有计划有目标的走的更高更专业。这是很多研发部门主管没有做的一点。

最后有几句话和大家分享一下：

毛主席说：社会主义就是打土豪分田地（不是资本论这样的天人天书），要天天讲，时时讲，到处讲，要团部建设到连队。所以，借用毛主席的方针，咱们的团队建设也得这样。天长日久，就形成了文化精神，就形成了习惯。

5、习惯决定性格，性格决定命运，细节决定成败--实施经理的工具箱

前段时间，项目经理的工具箱---走出软件作坊：三五个人十来条枪 如何成为开发正规军（三） 写完后，发现写的有些偏，偏向了开发经理，而没有顾及项目实施经理。在项目实施的时候，有些独特的地方，需要有独特的工具来帮助。

前天晚上，和一位做了多年实施项目带领的朋友吃饭。

我笑着跟他说：实施，能不能不实施？！不去人，也不搞实施，把软件卖了就 OK，你们做好 IT 咨询就可以，把什么数据准备、培训、协调业务部门和信息科需求、推动上线、报表制作都让客户做。咱也不赚他的实施费用。因为你们是个合伙成立的小公司，你们如果也是从开发到定制化到实施到支持，你们根本没有那么多人，项目周期又这么长，销售价格竞争又如此激烈，你们赚不了几个钱。实施尤其是最耗成本的，你们好不容易拿到的单，实施完剩不了多少，所以你们这么多年公司也没有大发展，不断在年年求生存。

他说：你纯粹是白日作梦。我一直在想怎么能缩短点或干活轻快点，你还在做梦不实施。不实施，人家买你的啊。企业那帮人，连数据准备都不想录入，你让他们自己实施？

我说，我虽然没有你实施的客户多，但我也做过灯塔标杆客户。再说，我多年统管开发、实施、服务三大部门，没个方法能搞定么？我给你介绍一下我的一些心得。可能不会真的让你不去实施，那样确实可能带来客户连单都不签的危险。咱们一起交流一下怎么能让实施尽可能的短。能短一点也是一点。我这个人就有个习惯，能改进我就不再原地踏步。这个改进方法不行，我就继续想其他改进方法，不断尝试不断推进，哪怕一点改进我都要去实现它。

量多了就会引起质变。许多人就等着大机会大改变，对小改进懒的动，我不赞同。

我说说我在项目实施和项目管理上的一些好的方法和心得。

做实施，最怕的不是人家使用中出现各种麻烦。而是业务部门抵制用，不想用，出各种各样理由，项目进展很慢（真不知道过去是怎么签单了）。究其原因就是：你们用软件能做到的，我用 EXCEL 也能做到。我现在就用的挺好，你们的软件还挺难用，还不根据我的需求我的习惯修改。修改了我就用。

完了。原来我们辛辛苦苦研发出来的管理软件，跟一个电子表格没啥区别。人家 EXCEL 还可以盗版免费用，网上随便下载。我们这个还花钱，还有时候有 BUG，还要服务器，还要专职系统维护。

实施人员没招了。都是些刚出身社会一两年，学生气和学生思维习惯都还没有摆脱，就让我来实施管理软件，并且给人家管理人员讲软件中的管理理念。这有点勉为其难。

其实有些管理软件不仅仅是减轻工作量，用电脑代替人工这么简单。它还蕴藏着丰富的管理经验。但说到管理经验，就很玄妙了。管理这个东西，是个公说公有理婆说婆有理的东西。很多人经常一说，他们管理落后。啥叫管理落后？你具体说说。说不出来了，指东打西的说的不到点子上。

如果先进的管理理念说不出来，有些软件就跟做手工一样了。

这是很多实施管理软件人的难题。先进的管理理念说不出来。因为自己就是个实施人员，又没有管理过企业，又没有多少管理经验，也就管过 1-2 个人，怎么能说服人家天天待在企业处理业务的管理人员呢。无法说服人家，人家就觉得这个管理软件就跟 EXCEL 一样，还不如 EXCEL 好使，人家肯定不用，没法上人家用起来呀。怎么办呢？

针对这个现象，我专门从软件附着的管理理念中抽取出了管理模型 KPI、管理模型计算公式、管理模型流程。把管理模型 KPI 一亮，都是管理人员很喜欢的和利润费用成本相关的东西。他们就来了劲了。然后就给他们展示这些 KPI 是怎么得到的。计算公式上场。计算公式中的值是怎么得到的呢。管理模型流程上场，是这么走流程就可以得到。那这么流程怎么能保证让各个部门一致的顺利的走下来呢。好啦。管理软件，水到渠成，客户老板立马拍板，谁拖延了上线就问谁的责任。

好的开端有了，就需要做实施的第一步，数据初始化。

做实施，在实施的前期，最大的时间消耗就在数据准备。这是一座信息化大楼的基础啊。基础出了问题，就会引起输入的问题，更会引起输出的问题。越发现的晚，以后调整的难度越大。我曾经遇见一个实施人员的案例，就是数据准备这块没做好，上线十来天才发现。最后他发现是他的问题，就自己偷偷改后台数据，没想到还改乱了，系统更是问题百出，客户急了，他也慌了。最后紧急救场。但仍然有一部分数据已经错误很难对齐了，给企业带来了当月财务处理核算很大的问题，我们不仅开除了员工，还给客户赔了钱，真是损失惨重。

痛定思痛。

首先做的第一项决定就是严厉测试数据字典准备功能。每一个基础数据录入窗口界面，都各种边界测试，非法字符测试，乱点乱按测试，删除默认值测试，机器反应速度慢测试，突然断网测试，突然断电测试，严把数据口。有些基础数据是互相关联的，就要严格测试数据关联性，保证前置数据没有准备好，后续数据字典不准进入维护窗口。

第二步就是封锁数据库。把数据库访问权限严格控制。所有视图、存储过程加密。所有更新删除插入语句留下日志审计，修改前修改后的字段信息日志。每条记录的最后更新时间 and 更改人留下日志。停用标志的停用时间和停用人的日志。这样增加了数据的安全性。

第三步，错误日志。一旦发现了没有程序预先想到的处理的错误，就立即终止软件，把软件的错误界面快闪自动保存成 GIF 文件，真正内部错误，都保存下来，可以点击按钮通过网络发送到公司报告。

我还专门组织编写了数据准备手册。详细的数据准备操作流程，输入输入规范约束默认值不可为空不可重复等等都说明的很清楚。而且还给了一份清单，每通过一步，就打一个勾。如果这个清单上的列表，每项勾都打好了，说明数据准备阶段就完成了。很清晰，很了解自己目前的工作进展。

其实，实施，做数据准备是非常耗费时间的。他们过去的数据用了那么多年，有很多重复，废旧，编码或命名不规范的数据。而且没有人愿意做数据准备。因为基础数据往往挺多，录入就是个人工活，还要校对规格和价格，否则以后业务处理就有问题。

所以，实施人员一般去了才去整理过去的的数据。说整理吧，人家过去的系统还不了解，又不是自己公司开发的。而且居然大多没有文档。数据结构根本不明白。就是根据数据瞎猜。

到了数据录入阶段吧，人都溜的贼快。反正你在现场，反正你着急回家，反正你的老板正焦急的催你上线节省费用早日上线早日催尾款。于是，只要自己一个人录入一个人校对，其他人都在偷着乐。这种实施，真不是人干的活。怎么他们上线用软件，他们自己不忙，实施人员倒是成了长工。唉，谁让人家是出钱的呢。有钱就是娘啊。

现在，我力求实施人员能不去现场做数据准备就不去，给他们按照数据准备手册按照流程给他们信息科培训几次，模拟操作几遍，就回到公司，不会在那里继续干耗。他们拖时间是他们自己拖。他们自己不想上线，他们的老板会找他们算帐。而且现在已经做的这么简单安全稳定了，客户的信息科他们都会自己根据安装手册做了。如果他们还懒还说不会，就说不过去了。

我的朋友开始限于思考状态中。

做实施，在实施中期，最耗费时间的就是培训。需要把人聚集起来，需要培训教室，还需要定点，还需要组织人，还需要模拟练习的机器。这就很难办了，业务部门是用户使用者，但他们都在工作中，让他们扔开工作来培训，谁来接替他们的工作，大家都很忙。另外，培训教室也是个事，那么多人需要培训，即使按拨来，也好几拨，哪有那么空闲的地方搞培训教室。人还有时到不齐，还需要重复培训。培训一次，不会，还需要再次培训。累啊。

针对这个问题，我们也想了招。这都是被迫了，老板要讲究成本和时间 and 人力。你搞不定，你就下课。

我首先，让培训专员制作了培训课程、培训教材、培训考试卷、模拟练习学习版软件、视频教学软件。在没有实施的时候就发下去光盘，让他们自己看视频看帮助看教材做练习。懒的看懒的学？可以，我还有培训。

到了真正的培训期，网络教室管理软件又派上了用场。他们每个机器都配个随身听那种耳机，随便一个耳机就可以，街上批发有许多，花不了多少钱。我在信息科电脑跟前坐，他们在他们的电脑跟前坐，根本不用培训教室，也不用他们离岗。他们的电脑一律显示的是我的电脑的演示。我边操作边讲。他们边听边看。

在讲的过程中，我也启动了我机器上的录像软件。讲完后，谁忘了听或者有事或者没听懂，都可以重复看。

谁想浪费我的培训苦心，随便听听，把培训当玩。我这里还有考试卷，考试打分。然后报给他们领导。而且还从中选出优秀者做业务标兵。这就很尴尬了。谁也不想当科室里的落后者。爱怎样就怎样的科员我还比较少见，因为现在的企业都不养闲人。

我的朋友眼睛开始闪光，兴奋中。

做实施，后期最大的时间就花在了上线后的监控运行上。那个客户端出现了问题，或者功能不会操作了，就需要立即赶去处理。由于上线后第一个星期，你正跑到 18 楼解决问题，4 楼的用户就给你打电话了，让你去解决。刚解决好 4 楼，15 楼的用户又给你打电话了。你的手机不断，挥汗如雨的奔忙在楼层之间，电梯人还多，每层都停，让你累的半死一个上午也解决不了几个问题。

现在网上很多免费的或收费很少的软件，如网络教室管理软件，如网吧管理软件，如远程监控软件很多。你给每个客户端在装 PC 的时候就都标配装上。这样，你以后就可以在信息科就可以掌控所有的计算机。那个计算机出了问题，你连接过去就看到了解决了，电话一交流，甚至内部 IM 系统一交流就全搞定了。

我的朋友不断点头称好。

在项目的维护期，就涉及到版本更新的问题。尤其是有些行业客户，需要你在实施过程中就修改需求定制化软件，否则不修改完不让上线，非要按照他们的习惯做才肯用，自然更新版本不断。

而客户端非常多，更新一次非常累。而且哪个更新了哪个忘了更新，更新的版本一致不一致，都会引起数据异常的问题，以后报表不平，查问题就很困难了。所以，为了更新，网上有很多局域网内文件同步软件，可以设置定时监测更新，如中午吃饭的时候正好自动更新，也可以设置每次启动计算机自动监测更新。你也可以用用。

我的朋友脸有点窘说：嗯，确实是个点子。我现在更新仍然需要一台台的安装更新覆盖。更新一次确实挺累。

我说：我现在已经改进的更好了。直接在软件中集成同步功能了。客户端软件一启动的时候，先自动监测服务器上的版本一致不一致？如果不一致，就自动更新同步服务器上的软件文件。但是客户的局域网由于这权限那权限，网络安全设置极为怪异，所以有时服务器数据库能访问，但就是无法访问文件夹。这样的情形我们的同步功能也考虑了，一旦检测无法同步，会自动提示版本不一致，需要手工版本同步。就不允许他继续登陆软件继续使用，否则他机器上的软件还是旧的，BUG 仍然没有修复，他输入进去的数据还可能是错误的，给后续的技术服务会带来很多的困难。

我过去经常遇到这样的情形：网络管理员打来电话说版本更新了仍然软件功能不好使。最后双方争论的很厉害，客户支持部呢说他没更新，网管说更新了。客户支持部说再更新一次，可能更新时候有异常，网管说已经再次更新了。客户支持部说：那我远程支持连接看看。他说无法上网。只好出现场。如果这个客户在海南岛就惨了，成本居贵。去了一看，是有的更新了有的没有更新。更新一次，OK，全搞定了。惨，3 分钟搞定的问题，却花了飞机出差，也花费了大量客服支持人员找问题的时间，客户满意度还不行。

自从软件有了同步和版本监测功能，客服支持电话少多了。而且由于一次机缘，客户的服务器必须定时在线数据上传，我们又利用这次机会，做了在线更新探测。我们一旦发现问题更新了软件，就放到了我们的支持服务器上。客户的服务器有驻留软件定时探测，一旦发现有新更新，自动下载更新，可以更新数据库，也可以更新文件。服务器更新完了，客户端就会自动按照服务器的版本变化自动更新了。从此，客户满意度提高了不少。因为有的客户还没有发现那个 BUG 的时候，就已经被我们更新了。客服的工作更轻松了。

上线还有一个小窍门，这个也能帮助你缩短时间。这也是我的一个心得。

我记得我做灯塔客户的时候，两家客户在不同的两座城市，但是两座城市比较近，2 个多小时的路程。我实施完了 A 客户，去了 B 客户那里继续实施。但是 A 客户打了电话，说需要有些工作需要支持支持。我就去了。因为我已经实施他这家了，所以他也不好意思继续用我。我来支持他们，也是一是人情二是近。于是我一去了他就问我这次能在他这里待多长时间。我说大概 1 天。于是，他会立即召集他的下属，把平时积累的问题都拿了出来，非常配合也工作节奏非常快工作效率也非常高的完成了。如果我说大概能待 3 天，估计他的人影在第三天才能出现。这就是人的惰性，时间不催赶着他，他总觉得还有明天。

所以，如果你去上线实施，如果一开始不明确告诉所有人，你必须 1 月后离开，而且必须实施完毕，那么他们半年都上不了线，即使上了线也是用的松松垮垮。如果限定项目时间，努力奔着这个时间，而且限定好项目此阶段着重解决的三个问题，他们就会工作节奏快的多。

注意，不限定项目边界，项目时间目标都是假的，很容易就超过项目时间，再想遵守项目时间就很难了。

我过去还实施过一家客户，没有实施前就是个松松垮垮的企业。小城市，人们中午 11 点下班后还回家买菜做饭，不像北京大城市中午回不去必须吃工作餐。他们还有午休时间。所以，小城市的生活是安逸的。但是我想快速实施。我早就准备好了很多项目过程管理表格和项目进度汇报流程。一去了，各种表格方法一拿出来，他们一看，来的人非常专业，混是不好混的，于是心情揣揣不安看我会如何。我每天邮件报告给我的老板、他的老板、项目涉及到的每个人，通报今天的工作内容和明天的工作计划。本来大家都觉得很难啃的一个客户，被我按计划时间完成。大家都一开始笑称我需要在那里买套房安家才能实施完，没想到我这么快。

这个案例就说明，你自己得过且过不正规，别人更就不把你当回事。你举止文雅谈吐内涵，别人也不好意思在你面前大放厥词。

我的朋友很尴尬的说：我服了你了。我实施多年，也没有想出你这么多招。总觉得什么都动不了。这些方法我们现在一个都没有用。如果用了，我相信能缩短现在一半实施周期。缩短了周期，就能减少成本。成本低了，利润就高了。

我说：我也是没有办法，老板逼的，老板向我要效益啊。人在压力中，自然就能想出办法。你如果觉得无法突破，那么你真的就无法突破了。我就是由于不相信这个老规矩就破除不了，所以就大胆思考大胆尝试，最后还真管用。这些方法不仅仅能降低成本。你实施周期短了，你可以实施更多的客户，这是一个开源节流的好方法。企业利润，不外乎多赚钱，少花钱。我全办到了。

6、客服顾问的工具箱

这段时间，写完了项目经理的工具箱---走出软件作坊：三五个人十来条枪 如何成为开发正规军（三）、实施经理的工具箱--走出软件作坊：三五个人十来条枪 如何成为开发正规军（五）。于是想一气呵成，干脆把客服支持的工具箱也一并写完得了。从此，开发、实施、支持三大部门，都有各自的七种武器。

我们一开始客服人员的武器只有电话。但是电话却有以下几个问题：

一般客户打来的电话疑问，都不是一句话能说清与搞定的。所以，客服人员需要问客户不少问题，以确诊客户到底问题出在哪里了。而客户也不能马上回答客服人员提出的诊断：如，你看看某个目录下的文件是什么时间的，你看看数据库某个表的主键在不在？这些问题都需要客户不断来回看，有时候技术水平低的客户都找不到客服人员说的那个配置项，有时候抽查，一个电话打了 1 个多小时。客户也累，问题也说不清，客服人员也累。虽然是戴着耳麦（如果是普通电话，我们能想象的出这个客服人员的形象：脖子歪着，夹着电话，一手敲键盘，一手拿鼠标），但是累啊。打完了电话，客服人员大呼耳膜都疼。

一旦一个客户接通了某个客服人员，那么他不仅仅是占住了这条线路，而且他也独占住了这个客服人员。因为这个客服人员只能听他的电话，而不能同时干其他的事情。而且这个电话其他客户再也打不进来了。这就太浪费资源了。CPU 都讲多线程，在我们这里串联排队了。

电话这个东西，各地方言口音都有。我说 D，他听成了 B。由于计算机的技术支持，很多都是技术名词。而且这些技术名词往往是英文词。中国人说英文就邪门了，标准的英文对方听不懂，更别说不标准的英文发音对方更听不懂。有一次，我让某个客户找个数据库字段，

我一个字母一个字母说，他一个字母一个字母在纸上写好，还要互相校对一下是不是这些字母。这就效率太低了。

电话费巨高，不能这么干了。

于是，QQ 出场。

，在企业是个偷偷摸摸玩的东西。企业的老板见 QQ 如见玩物一般，立即赶尽杀绝。IM 软件不能用，光在上面聊天了，不干活。要用也只能用 MSN。

但是客服没办法啊，QQ 好使：

可以截屏。客服支持很需要这个功能，往往跟客户打字说不清楚，把窗口界面贴上去，对照界面截图给客户讲，就明白的很快了。

传文件比 MSN 快。这是大家经常传文件的人才能体会的到。做客服，经常需要某些数据来测试，经常需要把更新补丁发给客户，或者把一些支持 FAQ 文档发给客户。QQ 快的多，而且还能断点续传。MSN 不行。尤其 MSN 还老掉线。QQ 在各种复杂的网络环境下都能登陆，MSN 就脆弱的多，不知因为什么就上不去了。

有个很厉害的功能就是远程协助。一开始我们使用 PcAnywhere，但是配置太麻烦，而且需要 IP 地址。但是企业只有 ADSL 上网，都是动态的 IP。而 QQ 远程协助，只要有 QQ 就能连通。而且 QQ 能变远程操作对方的机器，还边能和客户不断交互。不过，QQ 的远程查看速度，由于中国南北网络的原因，所以北方的客服中心连接南方的客户，速度很慢，支持起来就很浪费时间。

还有个功能叫远程演示。你不会，我演示我机器怎么操作，你看。边看，我边文字聊天教你注意事项。搞远程培训，绝对好。

的聊天记录查找起来不方便。经常跟客户上个星期说完，客户这个星期就又忘了，又不想跟他重新说一次，叫他看聊天记录。QQ 可以翻页，可以看某天。而且还原过去的聊天记录非常好。当时的截屏都在。一回顾，明白了。

类工具，都可以同时和好几个人交流，这就突破了电话独占客服人员的问题。而且敲文字和英文，大家理解统一，消除了电话口音的误解。而且 IM 类工具，是免费的。反正公司已经开通 ADSL 上网，网费包月，再多了 IM 应用也不多交钱。越多利用互联网资源越好就这样，QQ 就在地下发展开来。从此，用起来就一发不可收拾。

做的客户多了，客服人员明显不够用了。电话+QQ，忙的客服人员手耳并用。增加人？老板在考虑成本。人来了，就需要有工位，但是现在租的房子不够用了。而且还有各种工资福利办公杂费，还有人员管理的成本。

自己的问题看来只能自己解决。

客服人员压力中自己创新，成立了 QQ 群。

客服人员发现，客户打来不少电话，但是问的问题比较集中，异常个性的问题相对比较少。同样的问题同样的回答每天这样重复，自己都觉得腻。于是把客户的信息管理人员都召集到 QQ 群里。有啥问题，群里回答。一回答，大家都能看见，就不需要我自己一个个的重复回答。

果然奏效。客服支持电话减少了不少。甜头尝到，看来创新和解决方法都是逼出来的。

群的成立，又带来了更多的好处：

客户的信息管理人员之间居然互相讨论问题，交流经验，互相学习。哈哈，从理念理解到操作到支持疑问，什么都讨论。不需要客服人员的支持，成了大家帮助大家。客服工作再次轻松不少。

有什么更新版本，不要一家家通知。QQ 群里面一发公告，没几天，都看见公告自己就到指定 FTP 下载了新版本做了更新。及时把问题用新版本解决掉，免得留下以后技术支持的难度。

但是，有了 QQ 群，又发现了新问题。

客户有时候问起一个问题。客服人员想了想后在群里说：我记得在上个月哪天已经详细讨论过这个问题了，我找找当时大家的讨论。但是 QQ 群的历史记录太多，尤其 QQ 群人多后，讨论的不光是技术产品问题，还有不少灌水娱乐，找自己想要的东西很难找到。

怎么办呢？

看来必须有个地方把这些经典的讨论归档整理出来，放到一个论坛上，这样就好找了。

于是，BBS 搭建起来。

有理念讨论区、常见问题 FAQ 区、功能操作问答区、技术问答区、补丁下载区、实施方法区。

有了这个 BBS，好像很多事情都发生了变化：

群消停了许多，省得我们电脑上频频弹出消息框，烦的不行，禁止接收信息不行，暂存消息不行，弹出也不行。现在安静了许多，灌水的也少多了。

上挂上了所有客服人员的 web QQ。因为有的客户的信息管理人员还没有加入到 QQ 群中。有了问题，直接在 BBS 上点击某个客服人员的 QQ 就可以直接支持交流。而且，谁在线，谁不在线都可以看见。而且，找自己最熟悉的客服人员。

突变突然产生了。一个实施经理无意中在 BBS 上写了一篇实施当天的心得。老板眼睛一亮，这可是珍贵的实施案例呀。活生生的实施案例，颇具有说服力和成功实施经验的实施案例呀。

老板支持执行，谁还不赶快跟进。于是每个实施团队，每个实施案例一个帖子，流水写下来每天的工作心得。这个案例库现在真是宝贵的要紧。老板到哪里打单，证明实施实力，都拿这个炫耀。都成了打单利器，无往不利。

虽说这么多工具都已经加快不少客服的工作效率，也减轻了他们不少的工作压力，但是真正重量级的工具现在才商场（重要人物一般都是压台的，最后露面）。

这个重量级的工具就是客服工单系统。没有这个系统，客服部就不能称之为客服部。

所谓客服工单系统，就是客户的每个支持，都记录进这个系统。哪个客户的哪个部门的哪个人提的问题，提的是哪类的问题，这个客户是谁实施的，谁过去做过他的现场支持服务，问题的严重级别多高，解决的流程是什么，提出时间是多少，解决时间是多长，回访确认解决了没。谁解决的，谁回访的，客户满意度如何，哪个工单还没有结束，哪类工单最多，哪类工单耗时最多，哪些工单拖的时间最长也没有结束，哪类工单未结束的最多，哪类工单耗时最短，哪些客户工单最多，哪些客户工单最少。

我们每个月，每个季度，每年都要统计对比这些指标。根据指标，确定下一阶段重点支持的问题，重点支持的客户。耗时多的工单，集中各部门把解决方案提出。拖的时间最长的就尽快结束掉。耗时短的都总结出 FAQ，发布到 BBS 中。问题最多的那类，就和研发部一起合作，确定下一版本的修改事宜。

谁接待的工单谁负责工单的关闭，现在解决到哪个流程，停留在哪个人的手上，就要追查。一个工单多次解决都未能关闭，每次解决的过程都要记录下来。高级别的工单要优先解决。高级别的客户要优先解决。而且每次解决的电话录音都和这个工单关联在一起。想听录音直接播放即可。

根据客户的购买产品种类、客户购买金额、客户影响力、客户联系次数、客户技术能力、客户应用能力、客户人员配合程度等等不断打分，决定客户的优先级。客户优先级不仅决定客户的支持力度，也决定了客户以后购买产品的打折优惠程度。

工单系统还和呼叫中心绑定在一起。

客户的每个人的档案都记录在内。客户的姓名，部门，职务，电话，客户等级。我们不断跟踪更新，保持这里的客户档案为最新，谁离职了，离职到哪里了，我们都有记录。因

为客户行业是一个行业，客户的员工离职跳槽，一般都是跳槽到这个行业的某个企业，这会给我们带来新的潜在销售机会。因为我们有这个熟悉的联系人。

每次客户电话一来，立即来电弹出在客服人员的电脑上。这是个什么等级的客户，这个打来客户的人是什么部门什么职务。而且立即定位到这个客户最近一次的工单上。这个客户的以往的工单历史也都能看。往往客户的问题是关联的，而且一个客户有一个客户的习惯和认知，所以出现问题，解决思路，每次工单支持很相似。看看过往的工单，就知道当前的这个问题估计出在哪里。

而且，工单系统和呼叫中心绑定在一起后，还能实现工单的智能分配。

所谓的工单智能分配，一般的呼叫中心都是哪个电话线在空闲就哪个电话会自动响铃。但如果好几个电话都空闲，哪到底哪个该响？按照一般的呼叫中心处理就是找第一个空闲的机位。这样，第一个空闲的机位人就特别累。

我们做了改进。根据谁今天接待客户的时间最短，谁现在电话空闲，就谁的电话响。这样利于大家劳动强度一致。

但又出现了一个新问题，就是往往客户为了一个问题，今天会打好几个电话进来。如果按照这样的分配，就老需要把电话转到刚才给这个客户解决问题的客服人员。

老转也是个麻烦。于是我们又做了改进。根据上一个工单是谁接的，这次如果他空闲，就响他的铃。

在使用过程中，发现，客户和客服人员的关系越来越紧密了。因为常常接待他的是同一个客服人员，互相的讲话思路，解决问题的思路都彼此了解，对这家的客户也比较了解，解决问题快了许多。

工单系统有各种统计工作量和 capabilities 的报表。每天每个客服人员的接待客户量，接待时间量，解决工单数量，解决不同级别的工单数量，未解决的工单数据，拖延超期未解决的工单数量，自己能力不行解决不了转给其他人的工单数量，收集的销售机会数量，收集的销售机会金额，更新客户档案的数量有了量化的工作考核指标，客服人员就有了高级客服顾问，中级客服顾问，客服顾问三个等级。工资不同。而且不同等级的客户，不同严重级别的工单，也开始分配给不同级别的客服顾问。

现在，我们还把工单系统开放给了客户。客户可以不用打电话进来，自己登陆上来，把问题的描述，图片都传上来，就直接进入了我们的工单支持流程，和他打电话进来是一个效果。而且，客户也免去了电话说不清楚的问题，客服人员也减少了电话支持的难度。客服人员更有工作效果和工作安排了。

这就是计算机自动化的好处。谁说管理软件公司不用管理软件。

我们自己就尝到了管理软件的好处。

你的企业还在手工操作（容易丢失损坏不易统计不易共享不易同步修改）或是单机 EXCEL 复制来复制去（容易各个人手上的 EXCEL 内容版本不一致）的裸奔吗？

7、你这该死的销售

上个星期，我的一个朋友给我出了一道难题：

一个问题：销售在客户面前总是夸大公司的能力，在项目谈判时，总是这也可以实现，那也可以实现，但实际项目执行时，却发现根本就很难执行。但销售对实际的执行并不了解，而且也许如果不这么做的话，可能根本就无法签单，如何解决这个矛盾？

我说你这个问题很普遍。大部分的前期跟单、签单都是销售在做。一般小公司，老板就

是最大的销售，所有的大单子都是老板在跟。有的老板认为管理软件是管理的事情，管理软件有没有效果，和管理方法有关系，而和软件没多大关系，软件开发只是把管理方法实现编码而已。而且老板自己也平时挺注意 IT 界这些新产品的东西，什么 Blog、工作流、BI、SAAS、地图、搜索引擎、大众点评也都知道。所以自觉自己技术感觉和商业感觉和管理感觉都不错，没什么名字，就造个名词，什么 LAN 方法，什么 ANL 方法，造呗。所以签单的时候，往往领着一个销售助理就上场了，PPT 熬夜做的很漂亮，从麦肯锡、罗兰贝格中 COPY 了不少好看的图。关系再加上临场发挥，单子就算搞定了。

单子签好后，程序员根本不知道合同都签了些什么，合同都放在了财务和老板的手里。到了项目调研、项目开发或项目实施阶段才遇到客户的狙击：哟，我们要的 XX 功能你们怎么没有？

，我，我不知道呀。谁跟你说的？

你们合同里答应我们的呀。

我没见过合同。项目经理浑身在颤抖：你这该死的销售。

无独有偶，上个星期也有一个网友跟我诉他的苦：

一次，销售人员和老板一起在客户那里开会，客户的一个人说如果你们的软件中能和地图结合在一起，实时跟踪状态，那就非常妙了。老板一听，眼睛一亮。这可是我们软件的亮点啊。地图很好做，现在网上好多地图网站，都提供 API，我们能很快实现。（我 KAO，老板还知道的真不少，佩服，佩服）。

老板把他（开发经理，手下 3 人。来项目干项目那种，人手缺了还得自己组装机器掐网线，捎带给老板调打印或公司上网的路由器之类）叫到老板的座位旁，说要在软件的哪里哪里加一个地图，能怎样怎样的实时跟踪。

项目经理对地图不熟悉，但也听说过地图 API 这类事情。于是自己熬夜研究地图网站提供的 API。但地图的 API 是地图网站公司做的。地图网站公司估计是做产品做的脑袋木了，根本没考虑这些想集成他们地图的开发商。可能提供出来 API，只不过是为了秀一秀现在流行的 Open API，表示自己的技术先进或开放。实际做事，没戏。于是告知老板，做不了。

老板急了，这可是咱们的软件亮点啊，有了这个亮点....。人家都提供 API 了，你们只是调用一下，又没有让你们做地图网站，我看很多网站都集成了人家的地图 API。人家能做到，你为什么做不到？

项目经理开始解释地图 API 提供的技术缺失。但老板无法听懂。

于是，项目经理被老板认为技术不行。他现在很苦恼，老板不当自己回事，客户那里又被客户指挥来指挥去，下属干活还很烂，说明白了的功能被程序员实现的极不顺手，真想操刀把程序员的代码都大块删除了自己三两句代码给他改了。自己这个项目经理当的还真郁闷。看着人家销售跟着老板吃香喝辣，和客户一起吃饭唱歌，还能报打车费，和老板说说笑笑，老板有什么事都第一个人家知道，而自己就是个被老板指使干活的打工仔，郁闷的不行，想跳槽。

我问这位网友：人家为什么能跟着老板吃香喝辣？

网友说：还不是因为他会拍老板马屁呗。

我说：不。每个人都喜欢被别人捧，老板也是人，也不例外。但老板是开公司的，开公司是要赢利的，老板要赚钱的，光会拍马屁，如果不能给公司带来收入，老板也不养这样的马屁精。老板比谁都精。之所以老板很喜欢人家销售，就是因为在老板看来，销售是把钱能拿回来的人。假如没有单子，全体人就得喝西北风。

网友很不服：销售他再牛，他吹的天花乱坠能把客户忽悠晕了，也得我们这帮人去给他收拾屁股去。你再能说说出个花儿来，客户看不到真货也不会给钱的。

我说：人家为什么能跟老板一起到客户那里去开会交流，和客户沟通，而你不行呢？直

到人家签下单子来告诉你让你做的时候，你才知道有个项目来了。

网友说：他不就是人长的精神，嘴会说，会做个 PPT，会写个方案呗。要我，我也能写。满篇都是虚词，什么这架构那架构都是扯淡，客户都是不是傻子没个明白人？

我说：那我看看你的 PPT。

他说：我没写过正式的 PPT。没有机会啊。

我说：那你给客户实施，给客户培训时没有 PPT？

他说：我们从来没有那玩意。谁来写。我写？我就是个干活主力，我加班开发都忙不过来，我还写文档？

我说：那你给我讲讲你们的产品的好处。

他说：我们根本没有产品。老板签什么单子我们就做什么单子。项目来了，客户要什么我们就开发什么，我们就是个编程机器。

我说：那你说说你在这个项目对于客户的意义。

他说：都是老板的关系，整点事而已。其实一点没有意义。

我无语了。这样的项目经理，我看他再做 3 年也会是现在这样。光有抱怨，遇到问题抱怨问题，而不是去解决问题。问题不解决，仍然是问题。而这个问题对他的影响又很大，而他认为是别人的问题。其实，不管是谁的问题，只要是阻碍我的问题，我都要解决，千方百计去解决它。因为我知道，我受了影响，我如果连自己都不去救自己，就没有人救我自己了。

我面对这位郁闷的网友，我给了他一个建议：

你要先学会做 PPT，不断对比销售的 PPT，改进自己，超过销售的 PPT。先给客户展示，让客户认同，然后要找时机展示给老板，让老板认同。我建议你作 PPT，并不是让你炫 PPT 技巧。而是为了让你梳理自己的思路，将彼此关联的事物能条理清晰的结构化出来，并且能有重点的表达出来。想、做、写、说，对于一个成功的人来说都是必不可少的。

另外，你身上散发的程序员气质太浓。你需要把自己定位成一个项目经理，给自己加入经理的气质。否则，你给客户讲你自己做的 PPT，客户觉得很变扭。因为他认为你就是个写程序的，你讲 PPT 就不是你应该干的事。这样，会阻止你的 PPT 得到客户的认同。你所做的事，一定要和你的气质匹配。你把自己定位成编码机器，那么老板当然把你定位成编码机器了，那么老板干嘛要带一个编码机器去客户那里呢？

网友沉默了。

我也沉默了。我也回想起了我艰难起步的阶段。那个时候公司所处状况和这位网友说的情形很相似。我是职业经理人，而非老板。我面临的突如其来的项目（其实销售和老板已经酝酿已久，但自己一点风声都没有，老板和销售，谁也没有和我讲过）也是很郁闷（一点过程不让你参与，直到项目定型才让你参与）。直到有一天，老板内部开会，我说：我说说我的一些个人观点。于是，我打开了自己的 PPT，给大家讲了起来。

老板说，这个 PPT 是你做的？

我说：对。

于是，我的机会越来越多（谁说开发人员只会闷头写代码，而不会讲和写？）。

后来，公司软件业务在我的推动下蒸蒸日上，老板给与研发部的资源也越来越多，对研发也越来越重视，我的手下逐步在我的带领下发展出来专职的测试、文案、公共代码开发人员。然后我的职位和职责也越来越高，我管理的人越来越多，部门也统管了好几个。我没有太多的精力放在做销售用产品和技术文案上面了。但是，老板没有发现这个变化。每次打单，还要叫我参加。看来，我需要一个人来接替我的这个角色。

我需要物色一个人：

他首先必须懂得客户业务，否则和客户交流一看就是个外行。

他最好能干过开发。否则他也会和销售一样，不知道能不能技术实现，一律放炮答应下

来。

他制作 PPT 或写个什么文字，也不要太惨不忍睹。这个是可以训练的。

他需要有个项目经理的气质，别一坐那里就像个程序员。到客户那里打单也是塌着背，衬衣也不换，领带也不系，连讲话思路都不清晰，满口技术思路，这就不能要。至少要思路清晰，讲话考虑客户思维。

有那么点灵气劲。别不识眼色。该自己接话要接，别眼看老板要开始“发挥”了，还仍然继续由着老板放任。其实，老板这时候心里很没底，非常需要有个比他懂的人解围，否则老板也不会现场发挥了，不发挥就没法把自己的话说圆了。所以，需要这个人识眼色。但也别一出口就放炮，说不行，这个做不了。自己拿不准的，要说的委婉和中性一些，不要把话说死了，办法总比困难多。

不知道网友看到这 5 点会怎么想。我想这样的人还应该比较好找。现在不少人，啥都干过，啥都不精，实施、测试、技术支持都做过，报表写个 SQL 也做过，自己写个小软件也玩过，但就是跳槽什么职位都应聘不准，正茫然不知道该如何职业发展。

我就是从实施部门找到了这样的一个人。

这个人有个特质：很努力，很吃苦耐劳任劳任怨。对公司非常忠诚，没个花花心思看着这山望着那山。自己也没什么想法，只想着把领导交待的事情兢兢业业办好。

他实施过多家客户。对客户业务细节熟悉，对客户实施过程中发生的困难也很熟悉，但客户行业格局、客户行业未来发展、客户行业现状困境和机遇和挑战肯定是没有这样的大局观了。

因为我们的实施是要写实施总结和实施每日报告的，所以他写作还算有点基础。虽然写的口头语多，但总算能把事情说清楚，还能有个 123。

我们的实施也一直要求见客户必须衬衣领带，所以“程序员气质”（看来程序员形象太差了）是没有的。

他也写过点小程序，SQL 语句用 DELPHI 拖控件保存进数据库。

实施多了，讲话、开会讨论、客户联系就比较自如。不像一些程序员，让给客户打个电话联系联系问问具体情况，就是不知道怎么和客户开口。

就是他了。我让他做了项目经理。

他当然不会入了老板的法眼。

所幸，我们有个大项目，需要和 SAP、IBM、西门子合作。长达 5 个月。IBM 来的是一个团队，飞的来飞的去。住的是五星酒店，拿的是上万工资。做事的文档流程职责极为清晰。要求我们也上报各种文档，而且人家也早就规定了格式。每个星期还要有项目团队协调会。需要每个星期聚一次。

我选定了这位老哥就充当了这次项目的实际项目经理，我挂名。但实际的文档、开会都是这位老哥。

老哥压力巨大，面对豪华的 IBM、SAP 阵容，和人家一个圆桌上开会，有点气短。屡屡打电话希望我过去支援。

但我就不去。我说有问题咱们解决问题，你有什么解决不了的问题我给你出方法。你需要什么样的人我给你，我全力支持你。

老哥要开发，我就派开发人员。需要测试，我就派测试人员。老哥说自己文档需要人帮助修饰和理顺，我就派文案人员配合。

直到如今回顾那次项目，老哥都非常感谢我给他的那次和大佬们合作的机会。没有那次机会，他说他自己可能都无法脱胎换骨成为真正的项目经理。

以后，老板让我做一些方案，我就把工作安排给了这位项目经理，并且指明：你需要什么样的人，都可以受你指挥。你需要项目组由哪些人组成，你自己挑。我全力支持你。

于是，一篇篇越来越有质量的 PPT、文案、报价、工作量报价，都从这位项目经理之手而出（文案人员可真是他的好帮手，很多细节细致的收集资料、反复校对修改都是文案在做）。刚开始的时候，他做完就传给我，我们俩一起修改。我记得，刚开始的时候，一个方案需要修订多达 32 次的版本。我们总是在每个草稿后面加上版本号。以后工作就变得越来越顺，直到我后来都非常放心，把思路跟他说明白，他自己就会按照思路制作方案，我只需要审核一下符合我的大框架思路即可。现在不仅仅是项目文档和项目管理归他管，现在的产品设计文档都由他来编写。否则，谁来编写设计文档留下可确定的变更记录，谁来弥合客户和编码人员对功能需求的理解差异？

现在，他的 PPT 功力连老板都觉得专业。他越来越多的和老板一起去打单演讲。过去老板和销售煞费苦心的熬夜制作方案的日子也不见了，现在手下人都能干了，自己也放心。老板现在也轻松了许多。

这位老哥由于一直在研发和实施工作，所以对产品对客户，比销售人员了解的更深刻。自己做的 PPT 和 DOC 也逊于任何一个销售，讲产品讨论需求也是把好手。销售人员再也不是老板眼里的关键人物了。过去，公司的实施、开发、咨询、服务都对销售部蛮有微词，觉得老板捧他们太高，觉得谁跟老板关系好谁就能拿高工资，实在干活的部门都是死干活的，干的再累再好，也不如人家销售一张甜嘴。现在，公司的氛围挺和谐的，大家都觉得现在公平了许多，干活也有积极性了。（其实，在外企，一般都是一个商务经理加一个产品经理来共同打单。商务经理负责客户跟踪和价格谈判和商务合同，产品经理给客户结合客户现状和需求来讲解产品，负责技术方案。两者都有各自配合的分工。如今，我也是这样的格局：销售和售前两个人搭配打单，而不是销售一人独大。各有各的优点和专业）我对这位老哥说：你就是咱们公司的售前经理，就是咱们部门的宣传官。没有你，咱们研发再好的产品也传达不出去；没有你，咱们受了再多的压力受了再多的苦，老板也不知道。你这个项目经理的职位，对于连接研发和其他部门、连接客户非常重要。你是咱们研发必不可少不能代替的人。

8、水清则无鱼

我的朋友开了家屁小公司，纯粹的三五个人十来条枪。每年还不死，但活的也很辛苦。平时做的也就是两三万的单子，运气好能做 8-10 万的单子。那天，突然给我打了电话，说要请我吃饭。

饭肯定是不能白吃的。朋友告诉我：唉，烦心啊。客户不成熟，是麻烦事。客户太成熟，也是个麻烦事。

我说，此话怎讲？

我朋友说：你看，我过去跟单，客户对软件不懂，但他却知道有个华军软件园，里面有可以免费下载的管理软件。我报个两万的价格客户直晃脑袋说：我以为你们的软件 600 块钱就能买到，怎么你们要杀人啊？

我朋友一脸苦难相：两万块您都觉得贵啊。我们可是带技术支持，还有培训，还要给您做定制化开发呢。我给您这两万的报价，也已经是我插草标卖人的价儿了。

客户还是直晃脑袋：谁不知道你们软件开发，就一台电脑一个人，啥费用也没有，就卖我们两万。你看我们，进原料买地建厂房招人买流水线搞运输招商做广告打价格战，我们可是水深火热啊。我们全是成本，毛利极低，赚的都是血汗钱。你们呢，啥原材料也不进，流水线也不买，也不建厂房，居然卖两万，你说你们的利润多高？

我朋友无语。

我朋友就盼着能有个了解软件行业艰难的企业 CIO 能互表衷肠。

这回他总算遇见了。

大活儿。这次可是大活儿。我的朋友这样评价这个项目。

原来是一家企业，计划要给它的全国经销商渠道开发一套管理软件，以能管理全国进销存的情况。但是这次就是由于是个大活儿，所以客户方要求把价格报清楚了，不能开发费+实施费+...=总费用这么简单。他需要我给他说出个道理。为什么这么多。而且这个企业 CIO 最难搞的就是，他明告诉我，别跟我说你们一个开发人员的每天费用是 3000 块。你开发人员月工资 9 万啊？

这回我是折了。我过去都是做小单子，和客户反正也有关系（现在哪个上三万元的单子还是打陌生电话打出来的？），于是就囫圇吞枣就把单子签了。水清则无鱼，看似软件利润高，但软件这个东西看不见摸不着不好销售啊，报的价格高了谁都觉得你利润高，不吃你吃谁。所以到处都是爷，哪个门都得把香烧到了，最后落手里其实我也剩不下多少个子儿。还得给员工发工资，还得请客送礼，还得交税，还要交办公室租金，每月这费那费都是要钱的。苦啊。

我故意逗他：那你干脆关了公司，回去继续当程序员。

我的朋友诡秘一笑：呵呵，还是比给别人打工赚的多。别跟我打岔了，快给我想个方法，怎么报价让他觉得合理。

我说：看到了吧。耍嘴皮的哑了口，混天黑的犯了愁。以后，拍脑门的时代就算一去不复返了。客户也不是冤大头，你说个 10 万就给你 10 万。你以后不给说清楚了，谁也不会跟你签单。

我朋友说：那你快教教我啊。

我说：唉。我遇到的问题和你不一样，但我的解决方法也能解决你的问题。

我过去遇到的问题是：我一直在琢磨如何提高销售收入。否则，这么老路子做软件，费死劲也挣不了几个钱。所以，我在研发过程管理上引入了一些方法，引入专职的项目经理、公共代码开发、测试员，来收集需求、控制需求、安排进度、把控质量。在产品制造上我引入了美工来美化界面，引入了文案编写了精美的帮助文档、产品白皮书、演示版和视频教学。让销售拿出去给客户打单，一看 PPT 一看产品演示就感觉这是个专业的产品。另外，我在实施和咨询和技术支持也下了大功夫，让这种专业性一直连贯的保持下来，否则就会让客户感觉签单前很专业，一签了单子就糊弄人，这样生意就没得做了。

有时候你会遇到这样的场景：客户说你的东西太贵了，而你的回答往往是一点不贵啊，我才给您报 3 万块，您的企业一年的流水就一个亿呀，您坐的宝马车一次保养就 3000 多块。您看，软件能全体员工用，而且能提高您企业的运营效率，才 3 万块，多值啊。

但客户还是觉得贵。为什么呢？不是他掏不起这三万块钱，他可能请他的客户吃饭，一顿饭就能吃 3 万块。他为什么觉得他吃饭 3 万块不贵，而 3 万块买你的软件就贵呢？原因就在于他觉得你产品不值 3 万块。但到底值多少钱，他也不了解软件成本构成，所以他按心理估计，给你的报价上打个 5-7 折应该没错。

而软件这个东西，尤其是管理软件这个东西，就非常类似古董字画一样。在识货的人看来，他非常值。在不识货的人看来，一文不值。而管理软件的销售人员，尤其是销售这个行当，正规的管理方法很少，大部分都是术和案例，喜欢出新招出奇招，而对按部就班的流程运作颇为不感冒，所以对固化流程的管理软件也是觉得没什么大作用，比 EXCEL 贵还比 EXCEL 难用，所以也讲不出来这个管理软件该值多少钱。客户也不知道值多少钱，销售也不知道值多少钱。这样，双方的价格磋商，就成了互相猜心。客户觉得打他个 5 折，他估计就实在了。而销售呢，估计客户能承受 xx 万的价格，所以我就报个 xx 万。价格和软件成本没了关系。到底这个项目做到最后是赚是赔，都不清楚。反正单子我是签了，我的销售任务

完成了。

我就是为了应对这种销售和客户乱猜拳互博弈谈价格，才出此方法。

我先给你说说软件开发费怎么报。下次再请我吃饭，我再跟你说如何给实施报价。

产品开发由以下阶段构成：

客户调研、客户调研报告编写功能设计、功能设计说明书编写开发测试帮助文档、培训课程、培训演示版本编写其实，产品开发完毕后，还必须由开发部的文档组对实施部门的培训专员和项目经理进行内训，否则产品知识无法通过培训专员和项目经理传递给客户。这个内部培训也都是有成本的，而客户和你，往往都会遗漏的。

由于功能说明书为内部产品开发使用，所以功能说明书不会销售给客户。如果客户非要连功能说明书都认为是他的产品的一部份，那么他也需要花钱买走。这也是文案人员和项目经理辛苦劳动的成果，又不是自动产生的。有付出就必然要求回报。

产品开发团队由如下人员构成客户调研团队产品开发团队产品测试团队产品文档教育团队客户调研团队，由项目经理和培训专员构成，亲自到客户现场去调研。每个调研团一般由 2 名人员构成，1 名项目经理，1 名培训专员。由于不同经验的人当然工资待遇不同，而工作的质量也就不同。所以我们的调研项目经理，也有高级调研项目经理，中级调研项目经理，调研项目经理三个等级。你如果作为客户，你敢把企业的需求和流程梳理交给一个刚毕业出来的毛头小子么？他对企业的感觉连企业的一个看门人都不如。当然，不同等级的调研项目经理，调研一天的工作费用当然也是不同的。与之匹配的调研助理（培训专员）也是有不同的经验等级的。

调研费用，按调研团队人数和工作天数和他们的工作费用和调研家数计算。出差过程所花费的交通、住宿、餐费不包含在他们的工作费用中。调研是调研费，这是调研的工作，但出差产生的费用是出差的费用。如果你把这些都整到一块，你的调研费用肯定看起来很高，客户又质问你怎么一天 3000 块了，还不如报的时候就清楚的报。哪里都需要钱呀。

一般，调研周期为两周。先是调研收集客户现有状况细节，然后进行现有状况梳理。梳理解后，总结疑问，并与调研方开会，双方就疑问和目标和需求进行讨论。最后是根据多次讨论进行调整，得出最后的调研报告。

我为啥说需要两周呢。因为一个企业有许多部门许多岗位。你要把客户的整个组织结构和岗位职责和流程，和他们的现状问题，和他们的需求目标，都需要调查清楚，而且还要和他们协商统一认可好如何改进，否则你怎么设计适合他们的系统。没有两周的调研和讨论和反复修改磨合，你出来的软件更是痛苦，调研 3 天，开发 1 月，实施和不断修改 2 月，维护修改和稳定和技术支持 3 月。最后客户还抱怨你的软件太不稳定，N 多需求都没有做，当然不能付尾款了。你不答应他们的需求，你就拿不到尾款。于是，需求修改更新，发现又引出了其他的 BUG 和需求，噩梦循环啊。

咱们说说调研。

例如，调研 10 家客户。客户分布在东北、华北、华东、华南、华中、西南、西北。调研需要 2 周的时间。如果遇到星期六日还需要出差进行工作，那么休息日加班费就按国家规定累计。

这样来算一下费用。高级调研项目团队，中级调研项目团队，标准调研项目团队，（每个相应等级的调研项目经理每天的调研工作费用+他的每天对应的餐费交通费通讯费住宿费+每个相应等级的调研助理每天的调研工作费用+他的每天对应的餐费交通费通讯费住宿费）x10 天工作日 x10 个城市，你看看多少调研费用吧。你用一个 EXCEL 做个自动计算就 OK 了。你把这个公式内置到 EXCEL 中，客户只需要选择自己想要等级的调研人员，填写要调研一家的天数，再填写要调研的总家数，调研费用自然就出来了。如果客户觉得不能接受，就让客户自己玩这个计算公式，直到他自己都觉得确实没什么水分了作罢（他也不是变

态狂，以榨干软件公司逼死软件公司为乐。所以，他认可的是一个合理的报价，而不是离谱的报价。他认可的合理的利润还是手下留情的）。

想想吧。没有跟客户要一点浑水摸鱼的费用。都是大家都能认可和理解的费用。假设，一个在本行业信息化工作了 5 年的调研经理，他的月薪水达到 6000 块应该没有人质疑吧。那么他这个人每天最低的成本就是 200 元。再说了，这个员工消耗的其他办公费用与福利与税金也是有均摊成本的（企业总不能无原因生钱吧。钱肯定还是要从客户这个源头来）。

另外，我还没有计算调研团队在这 2 周的调研时间内，还有 4 天的休息日，如果工作的话，这个加班费用还没算呢。你可以算一下。

调研完了，就要开发了。你以为雇佣几个编码人员就能搞定了。那样的搞定也是活稀泥的，最后不断修改的成本能让你把赚的钱都吐出来。所以为了以后不秋后算账，我们就得从一开始好好做（如果你想赚更多的钱，可以把我给你讲的配置都缩减了。当然，质量和效果也就缩减了。一文价钱一文货，谁也不白给。国内软件如此现状，就是为了多赚钱，最后把客户给的合适的费用都缩减了，但是给客户报的却是达到 100% 效果的人员数量和人员素质和人员天数。这个中间差就是个小九九了）。

产品开发团队，一般由以下角色构成开发总监 1 名架构师 1 名，公共代码开发人员 2 名业务开发组长 1 名，主要代码开发 1 名，辅助开发 1 名。每个子系统由 3 名人员构成。假设有 4 个子系统，就需要有 12 名开发人员产品测试团队，一般由以下角色构成测试员 1 名。假设有 4 个子系统，就需要有 $(4+1)=5$ 名开发人员，其中+1 是公共代码测试。

产品文档团队，一般由以下角色构成每个子系统一个文档编写与内部培训人员。假设有 4 个子系统，就需要有 $4 \times 1=4$ 名文档人员。

所以，一个产品开发过程，以 4 个子系统为例，共需要参与开发人员客户调研团队 2 名产品开发团队 16 名产品测试团队 5 名产品文档团队 4 名共计 27 名参与者（估计许多人会说我三五杆枪能有这么多人么？楼主真是搞笑痴心妄想白日做梦。我报这个人数，是为了计算一个能达到客户效果的一个合理人员配置上。国内软件公司往往无法满足，所以效果就不断衰减。不过，我一般都建议三五条枪的企业最好能从实施部门抽调过来调研、文档、测试人员，这样人数可以缩减，但要做的事情不能缩减，这样效果衰减的就不会很厉害。很多三五条枪的企业现状都是做一单骗一单砸一单，就是什么人也没有，两个开发人员从调研到设计到开发到测试到实施到支持都这两位老哥。甚至干脆一个人全挑。这样的现状如果不去想办法改观，软件质量和软件竞争力仍然无从提高。）。

客户调研周期 14 天。产品开发周期 60 天，产品测试周期 10 天，产品文档周期 10 天，产品内部培训周期 10 天。由于产品测试周期和开发同步测试，自己独立出来的 10 天是综合测试。产品文档周期 10 天也是如此。而产品内部培训周期 10 天，是必须在产品文档发布后才能培训。

所以总的项目周期会是 $14+60+10+10=94$ 天，27 名参与者。才能保证一个产品（4 个子系统+1 个系统管理系统）开发的质量。

平均每人 6000 元月工资算（因为有低工资的培训专员和辅助编程，也有中等工资的各职责经理组长，也有高工资的总监，所以拉平 6000 元，应该也算比较合理。）。一个产品的开发大约是 6000×3 月（94 天） $\times 27$ 人=486,000 费用（这可都是成本，诸位客官都看到了，里面没有加任何企业的利润点）。再加上 30% 的毛利，共 $486,000+145,800=631,800$ 。大约 63 万开发费用和 3 个月的时间才能保证 4 个业务管理系统的软件质量和进度。

这些人的成本，你都可以做一个计算公式，明明白白得算。在客户要求的质量、时间限制内，使用多少人，使用多高工资的人，一算，成本就出来了。成本出来了，再加上你希望的合理毛利，就得出了你应该报出来的价格了。否则，你拍脑门报价，做到项目结束，你还真有可能亏损了。那些被项目拖死的公司还少吗？

的毛利，剔除销售费用和税金和管理费用，纯利在 15%以下。也就是说，年销售 1000 万，纯利才 150 万，其他都产生了费用支出。根本无法支撑下一年的生产再发展。

这样来看，软件公司确实活的挺惨。

这样来深入计算软件公司一个软件的成本和利润，我想那个说 600 元软件的客户该理解了吧（当然，那些总觉得老板是黄世仁的程序员，心里或许也能舒坦舒坦）。

9、实施费用也能DIY

咱们书接上回。水清则无鱼--走出软件作坊：三五个人十来条枪 如何成为开发正规军（八）上次咱们讲完了开发费用的计算，很多人在后面跟帖在那里算费用。

有人说：你把程序员都不当人，94 天，一天都不休息啊。

我想答曰：94 天，是工作时间。不算双休日在里面。也就是说，实际的开发周期长度是 94+3 个月之中的所有双休日。我只所以按照 94 天算，是算 94 天工作日，并没有把双休日都算进去。你试着想想，你都休息了，你怎么跟客户算费用的时候把双休日也好意思算进去。所以说，算费用，按 94 天算合理。

另外有人说了：根本不是全体人都干 94 个工作日。你按全体人干 94 个工作日算，这也太黑客户了吧。

我想答曰：这个详细开发过程我会以后一篇篇文章详细道来。我在这里先简短说一下：你以为调研完就把调研文档一转给下一阶段的人就算 OK 了？你以为直到代码编写完后才测试？你以为直到测试完后才写文档？这样的串行，根本无法保证产品开发质量，也无法适应国内这种“说过的话可以不算合同可以一撕两半”的需求朝令夕改的状况。我会以后讲到全程研发团队配合的方法。所以说，确实需要整个团队研发 94 个工作日。

由有人说：楼主在这里搭建自己的梦想空中楼阁呢，都是在遐想或瞎想。现在谁这么报价啊。水清则无鱼，你也讲了。谁也不是冤大头。吃饭 KTV 澡堂一条龙，外加回扣搞定他。

我想答曰：现在的客户都是荤素全收。吃饭 KTV 洗澡加回扣，这些都是正常流程。但桌下的归桌下，桌上的还要看你桌上的。现在需要的是面上好看面下也好看，既要捞了实惠还要项目做的好。现在企业打单的竞争都已经不是一两层的竞争了。先开始比关系，发现竞争对手关系也不弱。比价格，发现还是差不多。再往上加竞争层，比谁的功能强，比谁的售前方案和演示做的好，比谁的实施承诺和服务承诺好，比谁能够客户化修改，比谁能在现场驻扎时间长，比谁派的实力干将厉害，比谁客户化修改次数无限改动幅度无限。等等等等，比到大家觉得这单子签了也是个擦屁股事，坚持到最后的就胜出。如果洗洗澡吃吃饭给点回扣就能搞定，那销售就简单多了。过去可以，但现在不行了。未来更不行了，喝酒吃饭玩，那是工作之外的；谈到工作，还要怎么正规怎么来。想混水摸鱼报个价过关，以后出了时间问题和质量问题，谁的位子都可能丢，这个险没人敢冒。

当然，今天也不是对上一篇文章做 FAQ 问答。今天咱们仍然续接上回，给大家分享一下实施费用计算。

一个项目实施团队，由项目经理和培训专员两类角色构成。往往一个项目实施团队一名项目经理负责，若干培训专员组成。若干培训专员根据客户所购买系统模块数来定人数，一般一个子系统派一名培训专员。

项目经理的职责为：

项目范围界定制定项目计划组织项目成员协调相关人沟通保证项目质量推进项目进度而项目经理，也分为高级项目经理、中级项目经理、项目经理。分别对应 5 年、3 年、2 年

项目管理经验的实施项目经理。当然，不同级别的项目经理和培训专员，薪水都是不一样的，能力也是有高低差异的。

实施阶段分为项目调研期、项目数据准备期、项目培训期、项目上线运行期、项目验收期。

项目调研期，项目经理负责项目范围界定、制定项目计划、组织项目成员。培训专员收集项目详细资料，以利于项目经理界定项目范围和制定项目计划。

项目数据准备期，培训专员负责培训基础数据准备负责人员需要准备什么数据，数据如何编码，什么样的数据算是合格数据，培训数据准备操作员如何进行数据准备操作录入，帮助校验录入的数据是否正确。项目经理负责客户设计数据准备过程中出现的特殊数据处理方案。

项目培训期，由项目经理负责培训计划和培训协调组织，培训专员负责培训。

项目上线运行期，项目经理负责推进项目保证项目平稳全面的上线，培训专员负责现场指导手把手深度培训。

在项目验收期，会对前期所做的工作进行总结，指出不足，报告以后系统维护的重点和注意事项，对项目上线效果给与评估。

我遇见过这样的客户：不用我们培训。我们提供了详细的数据准备规范文档，系统初始化手册、业务功能使用帮助文档。他们也不需要我们定制化修改，他们自己培训自己上线。他们甚至不需要我们做项目效果咨询评估。

我们有不同级别不同费用的项目经理和培训专员可以让客户选择。当然，项目实施周期阶段，客户也可以选择。

有的客户需要定制化修改，我们就签项目调研期。他们不需要我们做培训，那么就把培训期的费用去掉。如果不需要我们辅助指导数据准备，那就把数据准备期的费用去掉。如果上线不需要我们监控切换过渡异常，也可以把上线运行期费用省掉。

客户可以根据自己家的 IT 建设复杂情况、历史遗留问题情况、人员素质情况、人员规模情况来决定每个阶段签多少天。但是，例外的是：我们的培训是单算费用的。

这样，实施费用=（所选级别的项目经理每天费用+N 套子系统 x 所选级别的培训专员每天费用）x 实施天数。

项目经理和培训专员级别不同，就有不同的每天实施费用。而实施，是由于出差到客户现场的，所以软件公司要支付员工出差福利补助，这部分钱当然需要从客户实施费用中拿了。实施过程中花费的电话费、餐费、住宿费、交通费，样样都需要钱，这都要考虑到实施费用中。

对于培训。有正规的培训专员，也有正规的培训课程、培训课本、培训练习、培训考试。靠嘴皮子培训，靠一份使用帮助文档对着每个界面操作来培训，这种土包子方法已经无法在正规性和专业性上让客户满意买单了。

针对培训，每个子系统都有自己的培训课程。而且分的很细，成为多门课程。有针对最基层操作人员的，有针对小组长管理的，有针对科长管理的，有针对该系统的信息科系统维护人员的。

而且，由于客户的 IT 水平不一。所以即使针对某一角色，如最基层的操作人员，也有初级课程和高级课程。

每门课程都有课时长、参加人员需要具备的技能条件要求、合理的培训人数限制（培训人数太多会影响教学质量）。

而且，培训都有次数。

这样的话，培训讲师费用=（某一级别的培训专员某门课程的培训费用）x 培训次数。

我们有个培训课程列表让客户选择打勾，客户调整自己需要的培训次数，就会自动计算

出来培训费用了。

培训，除了培训讲师费用，还有培训课本可以选择。因为培训课本都是印刷的，所以培训课本也是收费的。每本书价格都有不同，明码标价。买多少本，客户自己计算。

所以，整个实施费用，都是可计算的。给客户一套实施费用计算表，客户根据自己的财力和想达到的效果来调节参数，就可以选择适合自己的项目经理、培训专员、实施阶段、培训课程、培训课本。

要么继续混水摸鱼，打包了整个铁板一块的实施方案（现在软件都讲究积木化，为什么实施不能量贩式）给客户一个总数字，等待未来被淘汰掉。要么把选择和决定的权力交给客户，去主动拥抱未来。（我想起过去的商店和现在的超市）如果你觉得未来还是浑然一片水而不是 DIY 的时代，那么，请继续

10、将服务费用DIY到底

前一段时间，讲了一系列开发经理、实施经理、服务经理的工具箱：开发经理的工具箱---走出软件作坊：三五个人十来条枪 如何成为开发正规军（三），实施经理的工具箱--走出软件作坊：三五个人十来条枪 如何成为开发正规军（五），客服顾问的工具箱--走出软件作坊：三五个人十来条枪 如何成为开发正规军（六）。

这次，也就顺坡下驴给大家分享一下开发经理、实施经理、服务经理的小算盘账本。

前面有了水清则无鱼--走出软件作坊：三五个人十来条枪 如何成为开发正规军（八），实施费用也能 DIY--走出软件作坊：三五个人十来条枪 如何成为开发正规军（九）。那么就图个痛快，把这个费用 DIY 三部曲给它一个大团圆。

我想起了我在南方曾经见到的水果摊（现在北方也在学了），水果老板把进来的水果做分拣。大的堆一堆儿，而且擦的光光亮，小的堆一堆儿，中不留的堆一堆儿。各有各的价儿，各有各的卖法。而且早晨价和晚上价还不一样。货剩得差不多了，还有全包圆儿价。

他为什么这么做？

大家也都看到了。比如超市，把一只鸡都大卸八块。你爱吃鸡腿，你就专门买鸡腿，你爱吃鸡翅，你就专门买鸡翅。如果你想吃整鸡，你也可以买一只鸡回去。不同客户需求都有针对性产品。而且如果你是个爱算账的人你会发现，你把这些鸡的各个部件都零散买，比买整只鸡都贵。这就是个性需求的代价。

我们做 IT，技术支持和实施培训都是我们的产品，是我们的服务型产品。所以我们也顺应这种策略，分解组合不同的服务项目来满足不同需求不同价格要求的客户。这样不仅满足了不同层次的客户而且也发现了可以使我们整体的服务利润比过去整个打包服务要高的多。

我们的开发有主程、辅程，我们的实施也有高级项目经理和中级培训专员，所以我们的支持服务顾问也有高级支持、中级支持、和普通的服务支持顾问。他们的能力不同，工资不同，当然，门当户对，他们支持的客户级别也不一样。你让一个高级支持做一个烂泥扶不上墙的客户，那是浪费资源。你让一个普通支持去做一个高级客户，让高级客户觉得我们怎么这么低能。

所以，不管从客户需求，还是从我们培养人、考核人，还是从我们支持策略和收费策略来说，都是要求分个层次的。

人有层次，服务也有层次。

论坛和 QQ 群，就是免费的支持。你可以提问题，但不保证能回答你的问题，更不保证

及时回答你的问题，更更不能保证解决你的问题。因为，Free。当然，也不是说 Free 就啥价值也没有。论坛里有许多过去经典的案例和问题解决，你可以自行学习自行解决，你也可以把问题抛出来，可能其他客户看到了，也回答了你的问题。我们也有专门负责论坛和 QQ 群支持的人，每日来针对一些突出问题和难点问题进行回复。

收费的服务支持，就有远程支持和现场支持两种。

一般常用远程支持。实在是突发紧急事情，才会去现场出差服务。当然，出现场就比较贵了一些了。如果赶上国家法定休假，费用更贵。

远程支持，你可以打电话，也可以用 Skype，也可以用 QQ 语音或 QQ 视频，也可以配合 IM 工具如 QQ 之类，也可以配合邮件和 PcAnywhere 之类，也可以配合截屏和软件操作录像软件。所能使用的互联网交互通信工具，均可以一用。

如果你不想一次性支付服务支持费（年付），你可以按半年或季度来交。当然，你觉得这样也不合算（毕竟不经常有问题），当然也可以临时按支持时间来交。每个级别的客服顾问，为了解决你的问题，花费了多少时间，就有不同的收费标准。（不能以次数来计算时间。因为曾发生过一个客户的一个问题就解决了 2 天。还有的客户，一次性积攒了 N 个问题，当一次性提出来支持）。

支持，有 24X7 的支持，也有工作日工作时间支持（下班立马走人倒不会这么绝，客户还是上帝）。费用当然也有所不同。

出现场工程师所花费的交通费、住宿费、餐费，均由客户方支付。

客户支持，实施后第一年支持较多，一次性交清支持年费，可以打折。第二年一次性交清，打的折会更多。但肯定的是，一次性交一年的服务费的折扣，肯定会比一次性交半年或一个季度的服务费要折扣好的多。

咱们可以列一个 DIY 表格来让客户自己计算，也可以写一个 WEB 页面自动计算。

服务顾问级别支持方式年、半年、季度临时小时第一年一年以上第一年一年以上高级服务顾问远程工作日工作时工作日现场工作日工作时工作日中级服务顾问远程工作日工作时工作日现场工作日工作时工作日服务顾问远程工作日工作时工作日现场工作日工作时工作日组合还是真多。没个软件或 Excel 宏，自动计算还真是费劲。

看来，一个服务收费系统，还真是必须。

11、物以类聚，人以群分

上个星期和一群刚认识的朋友吃饭。很多朋友都看过了我的博客，对我写的《走出软件作坊：三五个人十来条枪》非常感兴趣，纷纷询问我怎么了解这么多。而你为什么会这样想，你又是如何做到的？

我说：其实我特别局限性。

一则我只工作了 10 年，但我一直在商业软件公司工作，赚钱为目的，而且每件事都是缺这少那，从来没有人、时间、能力和要做的事情匹配过。而且我也没有在日企、台企、港企、美企、法企、德企干过，不了解这些不同类型的管理风格如何，这些跨国公司资源匹配如何，内部斗争如何。我从始到终面临的老板都是典型的中国人，员工也是，客户更是，从国企背景到暴发投机的客户都有。

二则，我 10 年只服务了两家公司。这是很多人惊讶的一个事。在浮躁、跳槽、2.0 风投融资如家常便饭的 IT 业，尤其一直工作生活在 IT 业最发达的北京，尤其我服务的公司也不是名企，也不是外企，十年两家公司就令很多人惊讶了。这可能来自于我的家庭出身和我的

家庭影响。我的父母从 19 岁进入国企，就在大国企工作了一辈子，直到退休，也是国企发的退休金。国企这个大环境，人很多，人也很复杂，有时候你很难进，有时候想出去的人也很难出去。工资不多不少，撑不死饿不死，过年可以发一大堆年货，从植物油到春联，什么都发。一年复一年，在早些时候，父亲退休，儿子可以接班。一般厂里的职工，也都是厂里的子弟，想升上去可能需要等到你的领导退休了。想必许多人都有这种经历。这种情形让我常常想起张恨水的一部小说的名字：死水微澜。这种氛围也决定了我的性格：不紧不慢不急不躁。我的父亲出身财务。财务这个工作，上有领导，下有员工。再上面还有国家和税务局。钱的事，每个人各有各的心思各有各的算盘。平衡不好就有经济错误。所以父亲也常常教导我要谨慎低调，要注意平衡，要注意观察，要如何处理自己和领导的关系，要如何保全自己，要如何保护下属，要如何处理部门间斗争矛盾。我的领导管理风格气质很多都得益于我的父亲。而且，我也喜欢看历史书。小时候看《三国演义》、《三言二拍》之类，长大了就喜欢读历史和历史小说，尤其喜欢读《明史》、《清史》。历史小说二月河的作品《雍正皇帝》爱不释手（其他几部如《康熙皇帝》、《乾隆皇帝》不太喜欢。雍正是处于最内外交困斗争最激烈的时代，雍正处理需要非常高的手段才可。）。有个外国人写的《毛泽东传》我也非常喜欢。国内文字评价不客观，看看外国人写的，再看看国人写的，可以比较客观的阅读历史。这段历史离我们最近，而且也深刻的影响了现代人的精神和人性。（我的一个朋友老问我，我是有什么方法很快就能理出事务的头绪和模式，和事务的关键点？我想，这和我的父亲耳濡目染有关。如果你经常经历斗争，那么你会很快找到自己最合适的生存点）现在看书，我也一般喜欢看《财富》、《IT 经理世界》、《计算机世界》、《程序员》这些杂志。我对理论性的不太感兴趣。总觉得升华了的东西就脱离了具体产生的背景，容易无法落地。而通过业界动态兴衰的发展，可以客观的看出发展的历史脉络。（很多网友问我：你不看技术书籍，你不怕被技术革命淘汰了么。我是这样的：我总是在观察业界和跟踪业界，但我所处的职业和行业需要，大部分技术是用不到的。能用到的，我们会去安排深入跟踪和技术原型应用开发。而且我总是在梳理技术分支脉络，所以每一项新技术的出现，都会和现有技术有个关联，很容易知道一项新技术的发展）三则，我一直在管理软件公司工作。从事的一直是企业信息化管理软件。我没有从事过互联网网站，也没有从事过网游开发、短信 SP 开发、嵌入式开发、对日外包开发。可见，近 10 年流行的热门的细分 IT，我都比较落后。所以，我拥有的经验教训，只能是所工作过的行业信息化的一些工作经历。而且还很局限于所对应的行业信息化的发展程度。所以，所给大家博客中分享的很多都太片面。所以也有不少网友反映不适合他所处的行业。

四则，我所写内容大部分是我这 10 年间的一些工作心得总结。曾经历高级开发、灯塔客户实施经理、开发经理、首席架构师、技术总监、CTO。所以关注的层面越来越脱离一线。所以目前一线的开发人员、服务人员、实施人员、项目经理的真实感受和突出困难，我已经无法亲自经历了。由于目前已经处于公司运营操盘层面，所以有时候写出来的文字就有点站着不腰疼的虚。所以众多网友的批评我也接受。我也在尽力能去换位思考，所以我在博客上公告了自己的各种联系方式，希望能和网友直接对话，为他们排忧解难，让博文真正能解决大家的问题。

这四个因素，就决定了我的框框。我只能真诚的把我想所做真实的呈现给大家，大家读了我的文字，可以有所思考或启发，或在我的方法上有所改进以解决自己的问题，我的文字就达到目标意义了。

吃饭席间，很多人问我如何带好团队的问题。

我说，一个团队的好与坏，就在于团队的创建者和第一任运营者。就好像我们老听的一句话：什么是企业文化？企业文化就是老板的性格。一个公司尚且如此，一个团队也是如此。

如果你的团队的创建人就是小农经济、投机、疑心猜忌、不择手段，当然能在团队中待

住的人也会是这样的人。物以类聚，人以群分。那么团队的继任者也会是这样的人，因为上一任要提拔人，也是提拔和自己思路一样的人。处处和领导冲突的人，也不会被领导提拔。能跳出这个圈子的几乎看不到，所以我们总是凡夫俗子，而无法成为史玉柱或陈天桥或马云之类。

我不是一个喜欢控制的人，我喜欢抓大放小。往往大部分事务的成败，也就有那么两三个关键点。你抓住了关键点，事务就不可能偏的太远。如果一个管理者处处事必躬身，那么只有像诸葛亮一样呕血而死，死后还蜀中无大将。这就不是一个管理者了。有了问题，我就在远处静静看着，也不干涉，我看看我的团队会如何。这样很锻炼一个团队，我会在合适的时机插手，否则老板就对我要开火了。

有问题解决问题，没有问题你茫然什么，你压力什么，你慌什么？我总是这样对我的下属说。

经我这么一说，我的下属也回一回味，是啊，我到底在瞎忙什么呀？唉，都让这帮家伙搞晕了。

我说：不要晕。啥事也没有。

我判别平衡和矛盾冲突，也一般从人性和利益这两个方面去思考。企业员工间的矛盾，不外乎就是一口气、一个位子、一个面子或者是一点小利。其实最大的控制者还是老板，所有人都是棋子而已。解决了气、位、面、利，就很好搞定团队和谐了。中国人性，阅读历史长河，变化甚少。

许多管理者喜欢把下属揉捏有余，雷厉风行，显得自己很有手腕。我更喜欢给大家提供舞台，引导目标，提供经验指导，让他们互相搭配表演一场好戏。我类似导演而不是管理者也不是教练的角色。我喜欢这个群体中能制造出明星，也有踏实的群众演员，也有幕后的剧务。我希望这场好戏能够卖一个好价格，于是路演和推广是必不可少。我需要拍一部好戏的资金，于是我要去给老板讲故事获得资源支持。也许，这就是商业和软件艺术和软件工程的结合。

所以如何描述一个我，可能是商业+产品+管理+技术。我总是在这四者间不断揉和。所以大家看到我的文章也是各方面的内容都有。

有网友曾经给我跟帖：你要找的人，没有 30 万根本不可能找到这样的人。

但现实中，我的手下没有一个人能达到如此高的薪水。但是我们通过方法，通过团队精神，做到了我们的目标。因为我是职业经理人，我要做的事就是率领团队，不管用什么方法，有这个目标，就这群人，就要达到。达不到，你走人。

许多人是择良木而栖。我干吗在这个不毛之地辛苦达到目标呢。我干吗不找个更能实现想法的地方去实现呢？

我是这样想的：问题依然是问题。你跳槽了，问题仍然你没有解决，你也不知道如何解决，你只知道这不可能实现。但是，企业经营过程中，没有一件事是条件成熟的资源具备的。要人有人要钱有钱要时间有时间，我不知道会成为什么。可能结果会成为中国现在的科研现状（博士+国家拨款，就成就了现在中国的科研水平）。所以，现实来看来做，从手头的资源从手头的问题做起，行进中开火，解决问题中长大。不管你是创业还是做职业经理人，你只要想成事，你永远会面临问题。

我看人，引导人，锻炼人，提拔人。首先就是责任心。这个责任心，好像现在都成了中国 IT 业非常稀缺的资源。有企业投机滥做的因素，也有社会投机滥做的因素，也有中国“小皇帝”一代的特征。结婚有老妈出钱，买房有老妈出钱，生了孩子有老妈带老妈养。工作的意义是什么？个人的事业目标是什么？是生存糊口么？我想很多现在 80 后没有糊口危机。而有事业目标的甚少。所以工作，就是个毕业了顺坡下驴的事情，没有啥目标性。

在责任心之后，才能说到才能。否则这个才能就根本体现不出价值。有才，但做起来掉

里浪当，最后的结果是很平庸。这样的高才自己很自傲，觉得自己应该有更高的薪水，但是他不看他最后的产出，就看自己满身的才。这有什么用，我们看结果。

对于下属，我主要是给机会给项目锻炼人。我会经常和他聊天交流，什么都聊。在聊天中，大家就互相知道互相的性格了。做事就懂得谅解和理解了。我不喜欢新人一进来，就为期一个星期的培训，从公司文化到公司制度到公司产品，一个星期把人给洗脑了。这种方式不好。我喜欢那种润物细无声的方法。我会给每一个进来的新人根据他过去的工作经历和能力，给他指派一位合适性格的师傅。公司的文化不是口头说的，那是假的。公司的文化是你在日常细节工作中潜规则感受到的。我经常会和新人说：有什么问题，都问师傅。从怎么休假到公司怎么给出差补助到中午怎么吃饭到公司发展历史都可以问。师傅解答不了，可以直接MSN问我。

我经常用高于他能力一倍的项目锻炼人。我会给他分配任务卡定任务时间。我会去追究他的任务时间，从试用期我就一直这么用。我不会给他看书学习的时间。我相信实战锻炼人。现代企业又不是战争，他又死不了。我也不会把项目成败的关键点任务给他。如果他顶不住压力，自觉放弃跳槽，那么他就不适合我的团队。我会鼓励他，我会指导他，我会问他遇到什么问题了，我会问他进度提醒他任务时限。许多人刚一进来，觉得工作很忙压力很大。但留下来的，最后回顾自己的第一个任务，都觉得成长太快了，是以往都没有那么快成长的。现在的工作能力，感觉很充实。

很多网友抱怨自己的手下要一壶没一壶，啥也不会，现在的人素质能力太差。

嗯，这句话我很耳熟。

我刚出道的时候，我的部门经理也是这么说我们的。我的部门经理是60后70初，我们是70后。真是黄鼠狼下仔，一代不如一代。

但是，我们也成长了起来。我们也推出了优秀的产品，我们也承担起了公司的运营重任。并且在产品创新、运营创新上比他们更胜一筹。

、90后，他们有他们自己的优点，也有他们的缺点。很多老板的思维已经固定，他们习惯管理60后和70后的员工，对于80后、90后怎么都想不通，就跟想不通他们的孩子为什么留长发钻耳钉跳街舞喜欢周杰伦一样。但，80后、90后必然要成为公司的主流员工，你强迫他们成为60、70年代人的样子，这是违背整个社会主流人群。整个社会都没有扭转“小皇帝”现象，我们一个公司就要扭转？我们更应该去研究80、90后，改变我们的管理方式，学会运用他们的优点，避免他们的缺点。或许，我们的产品思路 and 经营策略也要改变。因为，创造产品并且把产品输送给客户培训给客户的这帮人，就是他们这些员工，而非我们这些管理者。

所以，我招人的时候，也只要能力差不多，说话理解能力还可以，能抓住重点，能重复表述，能细心反复的改进工作任务质量的，勤勤恳恳做好本职工作的人就可以。最怕没多少经验就乱谈老板乱谈公司乱讲战略，有一点工作成长就要求长薪水的。

很多人问我：你当职业经理人，你和老板意见不一致的时候多吗？不一致你怎么处理？

我说：我一般会找合适的时机向老板说明我的思路。而且我的思路的出发点和目的点一定为老板利益最大化考虑的（如果你的思路从一开始不是为老板着想的，那估计什么时候都得不到老板的认可）。所以，我有这个基点，所以我会找时机进言（时机很重要。老板正固执的时候你即使对的也不能接受。老板也是人，老板也有面子。）。如果老板仍然不接受，我会职业化公事公办的执行，安排人，推动，检查进度，监督质量，消除异常，调度资源，指导方法。我的意见，一般老板会听30%。而且，我当多年后回顾老板当年的想法的时候，我发现，老板大部分是对的。我想的确实不够更远更利益平衡。老板是比我掌握信息和资源多的人，老板也知道内外矛盾和困境，他就像在下围棋统揽全局，他想的步数可能会影响大

局或影响未来的第三步第四步第五步。而我只能看到前三步。如果我也能看到比老板更多的步数，可能我就是他的老板的。所以老板自有当老板的资本。经过这么多年职业经理人生涯，我还自认自己有一点能力，但老板仍然是老板，我仍然服务这么多年。可见，老板不是混饭吃的。所以，我总是告诫自己，低调低调再低调，再考虑远一些，不要乱讲。我尚且如此，我手下的员工更是如此。

我一直处于行业管理信息化 IT 业。这个细分 IT 业有个特征：

行业关系。每个能做某行业信息化的，都不是看着有前途突然杀进来的。（这个建议也适合那些创业者）行业知识。这是每个做行业管理软件都深刻能体会的。

技术不是首位关键，大部分都是增删改数据库应用。倒是大批量定制、大批量实施、大批量服务是很大的挑战。所以需要先进的运营和管理方法。简单说来，就是工程方法。需要团队大批量作业，而非精英研究攻克型。

这类型的企业就是人多。因为项目多。而人的能力也要求不太高，只要理解了行业细节知识，做起来开发、实施、服务都比较顺利。所以，我们这类企业也一般薪水不高，对人的能力要求也不高，但管理严格性却挺高。所以很多网友抨击我的管理说我把人当工人使。

嗯，话虽不好听，但本质确实如此。这是行业信息化这个 IT 行业特点所决定。我还没有在这个层次上创新模式。所以只能在项目进度、质量、成本上多做创新管理方法。所以网友说的个性激情程序员晚上开发白天睡觉可能不适合在这样的行业工作，可能互联网行业更适合。因为互联网行业大致是面对个人消费群体，而行业信息化大致都面对企业。企业都要一分投入一分回报，精打细算手中的钱，而不是娱乐而已。所以我们的成本和赢利意识挺强。

而且，这个行业也经常项目与产品互相交替，没有明确的界限。你暂时不要希望出一款产品然后满中国铺广告招商渠道。所以，做这行，往往行业内听说过，跳出这个行业就没有人知道了。

有什么样的人，就有什么样的产品，就有什么样的客户。

物以类聚，人以群分。

如果你老遇到让你很倒霉的客户，那么你先检查一下你自己。

此博文是为了呼应人，是人，真的是人---走出软件作坊：三五个人十来条枪 如何成为开发正规军（四）为什么

12、DIY报价

前段时间，写了一个开发、实施、服务费用计算三部曲。

水清则无鱼--走出软件作坊：三五个人十来条枪 如何成为开发正规军（八）实施费用也能 DIY--走出软件作坊：三五个人十来条枪 如何成为开发正规军（九）将服务费用 DIY 到底---走出软件作坊：三五个人十来条枪 如何成为开发正规军（十）引起了网友的大讨论。

软件如何报价如何定价一直是软件业讨论的热点。这算捅了马蜂窝了。

有个网友给我一个评论，很值得深思和大家讨论：

刚开始写得不错，越写越觉得离谱，不是三五个人了，和教材接近了。现在做项目的价钱是我们能够左右的吗，你算着 80 万，人家招标价 60 万，别人报 30 万，你干不干。不干有人干。既要吃饭回扣项目又要做好，这是真的。我想问一下，中国的软件项目需求到底多大。我觉得你在算你自己的帐，算得不错。可是人家客户人不认账。中国不光你这一个公司。

我经常说：我们是商业软件公司开发。我们的编写代码工作是为了更少的工作，但是能赚更多钱。所以，不能让我们减轻工作，不能让我们多赚钱的工具或方法或技术或管理制度，

我们一概不用。

所以，我们这个开发费用、实施费用、服务费用的计算表也不是为了什么好看或什么正规性，我们都是为了解决我们自己的问题。我们很现实。我们设置售前，就是为了怕销售乱说乱答应客户，最后项目实施周期长难度大需求变更多。我们做这个表格，也是为了怕销售乱报价，最后糊弄签了单，开发部实施部和客户一见面去执行，才发现那点钱根本不够项目成本。最后擦屁股挨老板骂的还是开发部实施部。而销售部由于和老板关系好，板子是打不到的。

大家都知道，现在这种做方案投标讲标签合同都在走浮面工作。知道项目签了合同，真正开发软件和实施软件的人才知道有这么个客户单子，前期都是销售在跟。而销售，对开发、实施、服务这些细节过程和成本都不了解。而销售跟单的人也往往是企业的决策拍板人，对软件功能细节也不看。而且，现在做行业管理软件，纯粹听到招标杀进来的非常少。都是这关系那关系过来的，都是关系认识。所以方案呀，讲标啊，都没有细节疑问，做方案也没有细节调研。签了这么个结果的合同额，真正项目执行起来，需求到底会变化多少，项目周期真正会多长，真正项目结束后是赚是亏，都是一个未知数。

反正已经签单，亏不亏是老板的事。而且是项目结束的时候才能知道。况且，项目的执行都是开发部和实施部门，他们耗费了项目资金，以后项目亏了，也是他们的问题。我把单子签回来了，我是从客户口袋掏出钱拿回公司的人，没有我，公司那些程序员他们一点用也没有，就知道天天等发工资等我把项目拿出来才能工作。这就是销售的想法。

所以，在软件公司，销售地位很高，开发人员居然地位很低。被老板骂怎么还不完工，被客户骂怎么我们的需求还没有做。

代码是老板看不懂的。老板就看功能做完没做完（而且做完没做完，也只是看一下开发人员的操作演示，然后再问问项目经理具体实际进度，真正客户要求的功能做完没做完，只有项目经理和程序员自己知道）。尤其是软件，越表面简单，其内部其实越复杂。除非这个功能本身就很简单。一般都是，为了把复杂的事情屏蔽了让计算机自动处理了，要写很复杂的代码，而呈现给用户的是简单的操作，只有这样，才能提高用户的工作效率，这就是软件的好处。但老板看不见也看不懂内部代码。老板就看见这么简单的操作功能，你怎么两个星期都没有做完？

所以，我们过往很多工作，不仅仅为了我们自己工作需要，我们也是为了让老板看到我们的劳动成果。所以我们编写了设计文档、测试案例、测试报告、帮助文档、演示版、需求管理库、BUG 管理库、每一次版本的归档源代码和文档，并且也用了专门的开发部服务器，表明里面装的都是公司最重要的财富：软件源代码。老板一看公司最重要的产品源代码都在上面，文档也在上面，各个版本都在上面，就放心许多。（老板越疑心，他就会派自己的心腹亲信来监督来约束，并且给与资源越约束越谨慎越拖延，怕这帮不知道整天在忙什么的程序员家伙把自己的钱给乱用了。所以，开发部一定要把老板能看懂的东西主动的完整的呈现给老板，让老板减轻疑心。这是很多开发部主管都没有做的事情，所以开发主管往往和老板关系很僵硬，最后越发资源少干事受阻碍，最后老板也看不顺眼他，他也看不顺眼老板，从此分道扬镳）。

我们出这个开发费用、实施费用、服务费用计算表的初衷就是为了让老板明白我们确实干的很辛苦，让他明白一个管理软件不是他经常画单据表格和报表统计用的 EXCEL。这个软件，确实需要这么多步骤，这么多人，这么多天的配合才能完成。

但是，我们不能这么和老板说。老板对员工吃多少苦不感兴趣。老板感兴趣的是赚多少钱。所以，向老板进言，就要从多赚钱这个角度去讲。

我们就讲了将鸡翅鸡腿脖分开卖，比卖整鸡要合算的多。而且，咱们这样报价有根有据，客户就不会心虚的拦腰砍五折了。因为他觉得每一笔帐都很实在，实在没法讨价还价。

我们可以再深入思考一个问题：客户是怎么决定自己招标价是 60 万？客户是企业，它又不是软件公司它肯定不了解软件公司的成本构成和项目人员配置。它怎么知道解决他的问题的信息化软件，60 万就可以搞定。

原因可能有两个：

一、看自己企业这几年赚不赚钱，自己的老板一向重视不重视信息化，这个项目重要不重要，自己企业能掏多少投资。

二、同类型软件，询问了一下自己认识的朋友，也根据自己过去的信息化的费用经验，大致在市面上的价格也就这个数。

于是，管信息化的 CIO，60 万拍脑门决定了。企业老板一看，嗯，能出的起。就这么定了。

就这样，一个 60 万就定了下来。

但是，这个 60 万决定的过程漏洞百出：

一、确实是，企业有多少钱就做多少事。但是需要信息化来解决的问题，到底需要多少钱才能真正做好？谁知道怎么计算到底需要多少钱？如果企业的 CIO 不知道怎么计算得来，那么他选定的最后软件公司，只能是报价最低或演讲最精彩的，或者就是他的熟人，确信这个熟人给他好处，而且不会把项目做砸了连他都受了牵连。

二、中国的信息化一直在不断规范化，成熟化，专业化。所以企业 CIO 询问的周边朋友，自己过去的信息化费用经验能适合如今的价格变化吗？（我母亲老提 2000 年的菜价和房价，对现在东西的价格觉得太离谱了）而软件公司呢，不去调研客户产生问题的现状，也不去思考如何解决问题，也不去计算解决这些问题的费用。也跟客户一样拍脑门定报价。

为什么软件公司也要这么做呢？

你如果真正去正规的做事，可能解决问题计算出来是 100 万。你怎么办？你能不要单子么？你要么往下砍功能，使一些客户问题无法得到解决或无法很好得到满足。要么，你就说服客户这个报价很实在，解决你的问题确实需要这么多钱。

客户会说什么？

一种结果：因为客户的 60 万报价本来就是拍脑门的，对 60 万能解决问题本来就不确信。所以他会去听为什么软件公司报 100 万。软件公司对问题的分析理解和解决方法是在往大了讲呢，还是在讲实在话？

一种结果：不好意思，我们只能掏 60 万。

对于，第一种结果，走势很好。因为他愿意去真正踏实下来去听去分析而不是拍脑门。你报的有根有据，他会去调整自己的底价。如果他确实预算只能这么多只能掏 60 万，他会去平衡缩减一下项目范围。我们前面也都讲了，开发、实施、服务都有高级、中级、标准三个层次的人员，费用是不一样的。而且很多项目都是可选择的。可以企业自己内部自己做，无须强制购买。客户会去调整自己的选择项目和选择的层次的开发实施服务人员。

如果竞争对手报 30 万。他是怎么做到 30 万的？是他的开发实施服务方法先进，所以成本低？是他的开发实施服务员工资低出差费用低所以成本低？如果竞争对手确实方法先进于你、人员费用低于你，那么你报 30 万肯定是亏死，你是在这单子上成功不了，自认心服口服。如果都不是，那这个竞争对手也会有两种可能结果：

结果一、报价低于实际必要成本，亏本关门。

结果二、为了不亏本，那就降低项目质量，糊弄完事。

对于这两种结果，第一种结果可能性小，因为谁也不想越做越关门。那只有第二种结果：糊弄了。所以，国内现在管理软件价格越来越低，关门转行的软件公司也不在少数。糊弄人的项目比比皆是，惹的企业都不敢上软件都已经不相信软件能解决问题。

根源在哪里？

根源就在企业不知道软件成本构成，乱设底价。而软件公司，为了得到单子，报价更低于客户的底价。就这样，一轮轮的循环，价格越来越低，软件公司为了生存，不断糊弄事保本。最后直到糊弄的客户都知道这家软件公司是个骗子了事。

要么糊弄到最后关门，要么创新解决问题突破恶性循环的价格樊笼。

我们现在就是这样报价的。你也可以试试。但前提是：你一定要有售前人员和销售人员一起配合打单，售前人员调研收集分析客户现状和问题，并且提出解决方案，然后再和销售一起完成销售报价。销售人员会在理性价格和感性价格之间做一个很好的平衡，既照顾公司项目完结盈利，也照顾客户价格心理承受线。

13、敢问路在何方

由于写了这个《三五个人十来条枪》系列，受到了许多网友的欢迎，所以也每天接到了很多网友们的问题请教。

我整理了一下，大部分网友有以下四类：

正在上大一或大二。问最多的问题就是学什么语言好。

正在着急找工作，但不知道如何才能找到工作的应届毕业生已经做了 3-4 年的开发，但感觉自己已经没有上升出路了的仍然原地踏步的程序员做了 1-2 年的项目经理，大小也算个头儿。但整天没完没了和客户和手下和老板沟通推进，每天很忙，每天很累，但总觉得自己很空，没有什么真本事，就觉得自己到处窜腾，客户逼着赶快出功能，老板逼着怎么还不结束到底问题在哪儿，手下素质太低，好几天搞不定问题还带着耳机边开发边听歌。

开发语言大战，论坛中一堆堆的口水帖，每次都极为壮观。虽然大家都说开发语言并不重要，整天盯着开发语言层次太低，但每次这样的争论帖子发出，都跟帖无数。

我也曾有过选择开发语言的经历。

我过去学的是 C。但是我在校期间出去打工的时候，发现社会需要的是 dbase、Foxbase、Foxpro。于是我就改学了开发语言。但是现在，会这些开发语言，想去找工作，势必登天难。

大家争论各种开发语言，其根源就在于此。尤其准备两年后毕业工作的大学生。如果现在选择了一门开发语言，自己在学校努力学习了两年，一毕业发现这门语言根本社会很少有用，那么找工作就困难了。所以很多学生朋友问我该学什么语言。

我在我的另一篇帖子中也写过流行技术我到底该学哪一样。我大致给大家在这里总结一下：

现在社会，主要的开发应用是互联网网站。主要是 asp、asp.net、JSP、PHP、Python、Ruby、Perl。

网络游戏。主要是嵌入开发、硬件开发、通信与网络开发，主要是 C/C++。中国大量的家电、数码、手机、电信设备都属于这类。

外包。主要是 JAVA 和.NET。

企业管理类软件。WEB 开发，主要是 JAVA 和.NET。C/S 开发，主要是 DELPHI、VB、VB.NET、C#、PB、VFP。

所以，你选择了什么开发语言，那么你应聘的公司就有了区别。所幸，我上述所说的五类开发应用，现在都有许多公司。所以，选择其中的开发语言，学扎实，有实际案例经验，人品端正，做人踏实努力积极主动，应聘应该是没有问题的。

不过，工资是有高有低。互联网网站公司，大公司薪资福利好，就看你的毕业学校和你的聪明劲了。如果你感觉自己一般，能选择的就是无数的互联网创业小公司。这类公司倒闭

风险大，薪资福利和工作条件可能艰苦，要的人也可能是熟练手，而不是新手。还有一些中不溜的互联网公司，比较偏向伪互联网。主要做广告推广或网站制作或电子商务线下买卖，做了 5-6 年了，可能需要一些刚毕业的学生做维护开发工作。

现在热门的网络游戏和嵌入开发，工资高、未来发展潜力大，但技术门槛也高。如果你学技术中不溜没有快速成长天分，也不愿意深钻，总想着机会主义，这个流行就学这个那个有兴起了赶快转移学习目光。这种思路，别说这些热门行业，就是那些传统行业也难找到工作。

对于外包，外语是第一位置，而开发技术反而是其次。因为外包都是大规模作战，分工很细，每个程序员能做的都是熟练工种，人海战术。尤其一些对日外包的项目，人家日本人连伪代码，函数名，参数名，参数类型都给起好了。

对于企业管理类软件，和外包很类似。技术普遍要求不高，常见都是增删改数据库的应用。也是人海战术。不过工资就比外包要低了，因为外包是老外掏钱，而面向国内销售的企业管理软件售价就低了。而且国内很多公司都是从事企业管理类软件。因为只要有客户关系，就可以做，没有多少技术难度。找工作是好找，但打一枪换一炮，反复需求修改，一个人捣鼓一个项目身揽数职，让人感觉没多少发展。

你觉得依你的毕业学校和你的人品和你的技术学习能力，你觉得你能达到哪个你喜欢做哪个，你就选择定不断努力，不要还在晃来晃去，最后什么都不精什么都看了点，这类人什么工作也找不到。

我过去上学的时候，网游、嵌入、外包都还不流行，很难找到工作。互联网刚大家知道，新浪 SOHU 刚出来，外国互联网发展成啥样都还不知道。所以主要热门的就是企业管理类软件开发。用的最多的就是 VB、PB、DELPHI、VFP。DOS 下就是 dbase、Foxpro 之类。当时 DOS 应用还非常多，街面上还有许多培训打字和 WPS 的培训班，WINDOWS 刚开始普及，Foxpro 和 VB 的书还卖的非常好。我一边学了 foxpro 打工赚钱，一边学了 DELPHI。大部分同学什么都不学，跟着老师听课做作业，准备毕业了回家乡让家长找个好工作，进个电厂或银行或公安局。我那时候已经有了不少打工工作经验，而且我订阅的《计算机世界》给了我许多看业界前沿技术和业界最新消息的启示。那时候好多同学都不看报，少数的订阅《电脑报》，整天在琢磨那些小技巧。我就是得益于《计算机世界》，让我在省城看到了中国的 IT 发展，世界的 IT 发展。因为当时热门的主要就是企业管理软件，所以我选择了组件技术和数据库技术作为主攻学习的方向，这都是开发企业管理软件的核心。当时由于感觉 VB、PB 在语言严谨性、技术先进性、代码开放性、控件多样性、底层控制性上都不如 DELPHI，所以我选择了 DELPHI，放弃了我心爱的 VC++4.0（由于从高中就自学 C 和汇编，所以对 C 很有感情，虽然当时没有什么 C 的应用让我很茫然到底学习 C 有什么用，而且 VC++4 当时的版本向导和可视化弱，都靠手敲代码，敲个 400 多行代码，才能运行一个什么都没有的普通窗口。而 DELPHI 能很快就出一个普通窗口，让我惊喜万分。但是，如今 JAVA 和 .NET 的雄起，DELPHI 的陨落，让现在学习 DELPHI 的大学生不知道如何出来找工作，只能赶快换开发语言）。

对于正在着急找工作的应届毕业生。和他们交流过程中发现了一些共同的特点。按说他们现在有互联网，有 BLOG，有论坛，有电子书，有搜索引擎，大量开源代码，而且学校里电脑几乎普及。但是他们的学习状态，和我 10 多年前上学的时候还是一样。像我的同学一样除了毕业证什么也不会。就连毕业设计，还是图书馆管理系统之类的毕业设计。我过去在上学的时候，互联网极其资源匮乏，而且上网牛慢费用巨贵，而且没有搜索引擎。我是到处买书，到处找源代码进行阅读。我当时阅读了 DELPHI 的源代码，从学校老师那里找来的 UNIX 的源代码，严援朝的 CCDOS 源代码，WINDOWS API 库 SDK 帮助说明。我做的本科毕业论文就是《从单机到 C/S 到 B/S》。我收集了大量的资料来写来论证。记得前几天，

我指导一个网友去下载一些源代码阅读。几天后，给我又发求助，说找不到啊。让我帮他找一个给他。我无语了。看来，这不是搜索能力不行，这类员工我是不会要的。居然让我帮他找一个。亏他能想的出来。

我也面试过许多应届毕业生。他们老给我展示他们在学校的干部职位，拿了多少优秀学生 and 奖学金，参加了多少社会活动。这不是我所关注的。这是 HR 关注的。他们会在收到你的简历筛选第一轮的时候就看这些。到我这里，我只关注技术问题。

一个应届的毕业生，当然实践工作经验有限，技术也有限。当熟练手来问问题是显然招不到一个合格的毕业生的。我一般会考察他的技术理解思路和技术理解速度和他的表达思路是否清晰有重点。我还会问他看过哪些源代码，平时看什么技术类的书籍，参加过哪些打工开发工作。一个不主动努力，不勤于思考钻研的人，工作中也会如此。一个说话思路都不清晰没有重点的人，写出程序也是一片混乱。他看什么样层次的书籍和报纸杂志，就能知道他的眼界有多宽发展有多少发展后劲。如果他做的毕业设计很独特，很有思考力，我就会比较赞许。因为他是真心思考和努力，而不是混毕业设计。

我一般建议应届毕业生，先不要着急找工作。很多人跟我说：怎么找工作啊。再找不到工作就饿死了。我看到不少手下的 80 后员工，现在自己赚钱了还和老妈要钱花，也没饿死，反而每月工资打车、吃饭、买 ipod、买 PSP。所以，饿不死。你既然在学校什么都没学到，现在要找工作，就拿点东西出来看。否则，你什么优点也没有，没有一壶可以提起的，怎么能让人家要你呢。到网上下载一个源代码，进行修改。其实修改并不是目的，也不是让你去跟招聘者去说这个系统是我做的。我让大家修改源代码，是为了让大家动手去分析源代码，学习人家的模块分割，架构，编码规范，编码方法。你在修改的过程中，你就会遇到问题，你就会被迫去寻找如何解决技术问题。这是一种有明确目的的学习，所以学习非常快，而且学到的东西都是非常实用的。在学校为什么无法做呢？就是由于你没有压力，到了临毕业才有压力。有压力才会去主动思考和主动解决。没有主动性的人在这个世界上还是占大多数。所以到了企业才需要管理。

对于已经做了 3-4 年的开发人员，仍然原地踏步。我非常关注这类程序员。因为作为一个毕业了 3-4 年的人，毕业前两年是拼命工作和学习的两年，第三年是发挥和做事的一年。第四年，因为第三年做事和发挥，发现遇到了不少阻碍，却搞不清楚问题到底出在哪里了，就很怀疑是不是过去三年的学习和努力到底对不对，哪里不对了。但眼界又决定了他们不能想清这个问题的答案。于是他们对未来该怎么发展都觉得迷茫。想跳槽，又不知道自己干什么，正处于灰心期。想学习，又不知道学习什么有前途，于是什么都学，新技术层出不穷，反而弄的更心慌了。有的同学自己创业了有了自己的小摊子，在国企和公务员的同学也升高了，有的同学也升做了项目经理，自己还是个程序员。想开发个什么网站，尝试后发现自己都是瞎捣鼓，想开发个什么软件，却发现现在什么软件都不好卖，自己又没有客户关系。唉，怎么混的这么惨。有些程序员，就是在干了 5 年程序员后，不是转行了，就是抛弃了企业管理软件开发，从头做起，改做互联网网站了。发现思路格格不入，技术也是新学，比不上人家一开始就做互联网的。尴尬自知。

我自己也经历过这个阶段。我深入研究了许多技术，发现并不能很好解决软件开发中遇到的问题。该如何解决，我也不知道。

大家看我的经历，就会发现，我研究技术，是为了解决软件制造和实施和服务中的问题，而不是纯粹因为感兴趣而学习技术，为了显示自己是公司技术最厉害的人而学习技术的。这在商业软件公司根本吃不开。商业软件公司，赚钱为主。如果你的技术无法给公司赚钱带来帮助，就根本没有用。

有些做了 3-4 年的程序员，做到这个阶段，新技术看了一大堆也不明就里，仍然在学习 hibernate 怎么配置怎么用，struts 怎么配置怎么安装怎么调试。说明这类程序员缺乏开发天

分，无法在技术上成长为优秀的程序员或技术专家了。

不过有些技术很牛的人现在也很困惑，工资就是不涨。我建议他从帮助产品提高销售额的角度去把自己的技术应用到产品中。我过去有个手下，做行业信息化管理软件，却不愿意深入了解这个行业。自认自己要成为技术专家，要做最好的软件架构，于是拼命学习了 N 多框架，对比分析，做源代码阅读，做实验尝试新技术，整天熬夜。做出来的架构却是并不能减轻业务功能开发人员的工作量。老需要注意 N 个地方，配置多个选项。配置错误就运行错误。这类架构还不如没有。我们是在开发行业信息化管理软件，不是在做变型金刚。我们不希望一个能制造汽车，也能制造轮船的东西。我们就需要制造小轿车的平台架构，连制造卡车的平台架构都不需要。但你制造的却是一个个的螺丝和钢管。

如果有的技术牛人，技术也能很有效的帮助产品提升。但工资还是不涨。可能跟公司抠门有关。可以建议去发表一些博客来提高江湖的知名度，这样请你去做技术咨询方案的人也有可能找到你。

在企业管理软件开发公司，一般有以下这些职位可供发展：

实施人员、实施经理、咨询经理、售前、市场、销售服务人员、服务经理开发人员、高级开发、客户化定制开发、技术专家、开发经理、技术总监、如果你善于组织和调度人，善于推动项目和控制项目，善于和客户沟通理解客户，那么你可以往项目经理职位转变。实施经理、服务经理、开发项目经理，都可以选择。开发经理，未必是技术最好的那个人。

如果你不善于和人打交道，技术也不行。那么做一个踏实稳定勤恳的客户化定制人员或技术服务支持人员。并且在工作中不断小改进，让自己的工作更有效率更有效果。

如果你不善于和人打交道，技术也不行，但对客户业务比较熟悉，那么建议你踏实工作，做好实施（做好实施的人未必会与人打交道。我发现很多性格内向的人，提升自己的职业化工作细节，公事公办也达到了很好的实施效果）。从实施，可以转向咨询、售前。但咨询、售前都是很需要结构性思考和细致观察的工作。

如果你技术无望、不善于和人打交道，也不善于组织控制，也不善于细致观察思考，也不想踏实勤恳，却想到处跳槽涨工资。我想你恰恰什么都得不到。你是最容易被裁掉的那个人。选一样，你必须选一样。即使你一无是处踏实努力干活保证质量和进度也好啊，现在，踏实努力干活的员工在每个 IT 公司都是宝。

我有个朋友，过去是做开发的，最后做了实施项目经理。老觉得自己的工作很空，混了几年代码也忘了，就会跟人扯皮了。自己也不会结构化思考，当不了咨询顾问。也不想做市场和销售。问我该怎么职业发展。

我给他讲了一个故事，我问他：你觉得，西游记师徒四人，你要开除，首先开除谁。

我的朋友说：当然开除猪八戒。他又自私又贪心又好色，诚恳不如沙和尚，武功不如孙悟空。

我又问他：哪第二个应该开除谁？

我的朋友回答：当然是唐僧。他没啥本事，还老误解人，什么本事都没有老拖后腿，每次得解救他。

我说：那好。如果就让孙悟空和沙和尚两个人去取经，他们能取到吗？不过他们不能一个跟头驾云去，那就没什么讨论了。

我的朋友说：他们俩怎么能取来经呢？在公司里，如果把一个牛人和一个踏实老实的员工，让他们俩去完成任务，多半会半路闹崩了。

我说：那如何不让他们闹崩了呢？

我的朋友说：需要有一个项目经理领导他们俩。

我说：OK。这就是项目经理。公司里已经有一个唐僧了，他就是你的老板。唐僧既然已经有了，牛人也有，踏实的员工也有，但还是完不成目标，就是需要有项目经理。你就是

那个项目经理。这种职位永远需要，但总是不那么突出，但老板明白谁才是最重要的。你看看历史：刘邦封功，韩信张良萧何。萧何就是那个项目经理到处串线搭桥。明朝，徐达刘伯温李善长。李善长就是那个项目经理。项目经理就是主板上的 CPU，用来协调各个其他计算部件的。所以，你很有价值。

我的朋友现在已经是很好的项目经理，老板也放心将历时一年价格 500 万需要牵扯多个部门几十号人的大单项目交给他来负责。

我问他：过去你怎么当不了一个好的项目经理呢？

他说：观念转变不过来。是工作强奸你还是你享受工作，就看你怎么看。

敢问路在何方，路就在脚下。

14、懈寄生

他渐渐合上流露挂念的双眼时，我意识到自己是一株懈寄生，当他枯萎时，猛然发觉，我失去的，不只是他给的养分很多人问我，我是怎么知道这么多的，别人怎么成为我？

我突然想起了痞子蔡的一篇小说：《懈寄生》。里面有上面的一篇诗句。

我回顾了一下我从大学到如今，哪些书影响了我，哪些人影响了，哪些关键事件影响了我？希望能给大家以启发，大家可能在阅读的时候突然有所通灵，你可能也看过同一本书，遇见有人跟你说过同样类似的话，可能遇到过同样类似的情景，但可能就是转眼的一瞬间，一瞬间虽然已逾 10 年，但大学期间最影响我的是以下这五本书：

严援朝的《CCDOS 源代码剖析》

严援朝前辈的这本书，让我完整的，系统的理解了一个操作系统的工作原理，不仅指出了操作系统的各个模块结构，而且还详细的描述了如何实现。在描述的过程中还指出了面临的当时的硬件限制和 DOS 限制，更指出如何去巧妙的解决。从大学一开始，我学习的内容就比其他同学学的深入。同学们还在跟着老师学习课本，我就已经在阅读剖析真正的业界产品源代码。

我有两个启示：

想去深入了解一门技术，阅读源代码是最好最快的方法，虽然有些艰难，但不断阅读不断研究思考不断做笔记，突破后就能发生质变。如果你从入门到精通，这个时间将非常长，可能你在前进的过程中已经失去兴趣再也不想到达精通了。

想去深入理解什么产品，就去找制造这个产品的人写的书。这样的书没有歧义，能看出创造者的思路和创造的来龙去脉和他的眼光。不是亲自做的产品，不是亲自写的书。很多就有理解歧义，容易误导人，而且还不深刻，无法一步到位。

的《DELPHI 高级开发指南》

这位老哥真是惊世。这本书在 98 年就敢卖 80 块钱。而且是只发行了一次印刷，再未重印过，可见读者之少。虽然写的内容是以 DELPHI3 为蓝本，但该书的内容直到如今，大部分 DELPHI 开发人员都无法阅读懂。老哥和 DELPHI 开发组人员一起工作良久，书中对 DELPHI 在 WINDOWS 编程、WINDOWS 线程与内存控制、RTTI 元数据与反射、COM 编程、控件编写、DELPHI 开发 Internet 功能都做了深入的描写。我真正第一步理解 WINDOWS 编程，恰恰是通过这本书。虽然过去学习 VC++，但一直浑浑噩噩，学了许多但没有一下醍醐灌顶的感觉。这本书让我一下子把过去在 VC++ 学习时代没有本质理解的 WINDOWS 编程突然全联系在了一起，功力大增。而且，我在大学期间就不断编写代码，从函数编程到 OO 编程，在这里我遇到了面向组件编程，一下子就迷上了属性方法事件这种结构。我疯狂

的学习 COM 组件和 DELPHI 控件的编程。为了编写控件，又深入学习了 RTTI 元数据与反射。现在我对 SOA、WebService、.netJAVA、组件、OO、函数、WINDOWS 的通贯理解，全得益于它。现在这些技术，我都能从 DELPHI 控件和 COM 组件中延伸出来理解。虽然 10 年过去，但技术的变化并不大。

李维先生有一本书，叫《Inside VCL(深入核心——VCL 架构剖析》。和这位老哥的书的思路挺像。但李维先生的书是在 2005 年才出版。但李维先生的通俗易懂深入浅出幽默诙谐的写作风格还是非常值得大家一读。

就是这一本本很难的书，我一一攻克（我回想大学，老想起一天睡 3-4 个小时，不断编程打工看书，精神状态不好，有点疯子痴呆状。我的宿舍兄弟怕我出事，老拉着我去和他一起挖蚯蚓钓鱼，说你一握到鱼竿你就什么都不会想了，你就会全神贯注在那根鱼线是否有颤动）。就是这一次次的攻克，让我不断质变。我还没有毕业，就感觉省城这个池子太浅，萌生了要去独自北京闯荡的念头。

《WINDOWS 程序设计》

这又是一位神人。如果开发基于 WINDOWS 的软件而没有阅读过这位神人的书，真可谓可惜，而且会感觉你入门不正。这位神人不知道是否出身微软，我也没有百度过。好像我记起一个故事，不知道是不是关于这位神人的。说的是此爷曾经写过一本《WINDOWS 未公开 API》还是什么书，惹的微软要告他泄露微软的技术秘密。但该爷并没有阅读过 WINDOWS 的源代码。现在，微软开发操作系统，都要请这位爷做顾问。我过去看《DELPHI 高级开发指南》理解 WINDOWS，只能算是旁门左道。而这本书才是学习 WINDOWS 开发的正宗之道。该书对内存、线程、文件、窗口、消息、GDI、SOCKET 都有非常深入的描写。当初打单身的时候每次阅读都要在星期六日只泡方便面不出被窝一气呵成从头读完。即使阅读多遍，每次阅读都还读的酣畅淋漓；每次阅读，都能对 WINDOWS 开发有一层的提高。

当时还遇到一本好书《TCP/IP 原理》。让我对网络编程，网络通信有了很透彻的理解。可惜自己一直从事企业管理软件开发，所以搁浅网络研究。如果有谁从事网络开发，此书必读。

《微软的秘密》

这是一本我在 97 年买的书。这 11 年来，我还一直读它。这是一本跟了我 longest 的书。有关微软的书多不胜数，但能本质的看微软，看一个研发帝国如何研发产品当上软件霸主成为业界事实标准，唯有这本书。这本书从微软灵活的组织结构，专业的专家小组，既懂开发又懂商业的人才，项目管理开发测试三套马车，里程碑的开发阶段、不断推出改进树立标准，不断自我反思自我总结学习改进真的真是有结构有条理。我许多的开发管理思路都从此得到启发和借鉴。我的开发管理体系模型就是从这里一步步从点到线到面到层的不断完善起来。而且，我现在的咨询思想、流程梳理思想、需求调研思想，皆出于此。是我迈进开发管理的导师。

迈克 波特《竞争战略》

上学的时候，不仅仅痴迷计算机，对公司管理也非常感兴趣。所以自修学习财务管理的学位。这本《竞争战略》是一本 MBA 课程。当时此书给我真是打开了另一片世界：原来企业还可以这样做。企业要思考客户，思考竞争对手，思考在产业链上的竞争位置，思考产品差异化。这些思想，都对我以后产品研发和产品运作带来了很大的影响。

《计算机世界》

这不是一本书，这是报纸。我在大学校园，一个北方省城，一个普通的大学，而不是在北京，更不是在什么北大清华。能够了解业界，了解最新的技术，了解世界的各大公司动态，从他们的动态总结他们到底想干什么，想构建怎样的产品战略，想如何和竞争对手竞争，我每个星期盼着报纸的到来，每次阅读都做了大量的笔记和分析。在大学里，我的心中就不仅

仅只有程序，还有业界竞争和未来趋势的思考。

在校园阶段，最影响我的是我的两个老师，一个是我的 Pascal 老师，一个是我的操作系统老师。

我如今都深深记得我的 Pascal 老师对我的发火（他其实一直很看好我，因此爱之深恨之切）：你不要老提钱、钱。他嫌我太商业化，而污染了深入研究技术的心灵。

他这句话我仍然现在记得，并且在我商业运营产品和公司的过程中不断想起，不断反思。让我在商业和人道之间做着调整 and 平衡。这句话，让我对师一直肃穆崇敬。

我的操作系统老师，是他把我引入了一个正式的操作系统的世界。给全体同学讲完课之余，还给我讲 UNIX 的操作系统，从结构到源代码到操作系统的发展历史到未来操作系统的演变。让我对操作系统从技术到架构到操作系统的本质意义都有很高的视野。我们俩经常在一起交流操作系统、编译器、开发语言之间的互动影响关系。

我出道了，来到了北京。由于我对 COM 三层架构的深刻理解和实际开发经验，还有对 DELPHI 的深厚编程功力，我很顺利的就找到了一家公司担任了高级开发人员。在北京，更大的世界展露在了我的世界。我在海淀图书城战栗世界之大技术之广阔，于是一个猛子就扎了进去。一本本好书，让我如坐火箭，一年之中就成为了公司最顶尖的技术人员。

以下是我刚出道最影响我的火箭之书：

的《Microsoft SQL Server 7.0 技术内幕》

假如没有这本书，我的技术世界就会缺一半。可见这本书给我的影响之大。我一直从事企业管理软件信息化开发。企业管理软件的开发，一直以来就有两个很重要的技术，一个是数据库，一个是组件技术。我过去用 DLL，然后用 COM，然后用 EJB，然后用 WebService，直到如今的 SOA，都是组件技术的发展。而数据库，我却一直固守在 SQLSERVER 的天下。很多人学习 SQLSERVER，其实是在学习 T-SQL，知道很多 SQLSERVER 的 SQL 函数而已，会写复杂的取数 SQL 和 SP 而已。而我一开始进入的数据库世界就是数据库查询引擎、数据库存储引擎、数据库的数据内部存储格式、数据库事务、数据库日志、数据库锁。这本书都是从原理和实现的角度上来讲。其实这本书是以 SQLSERVER 为蓝本，真正讲的是一个商业数据库产品的架构和实现。如果你阅读完此书，然后你阅读 SQLITE 源代码，再阅读 MySQL 的源代码，相信你能创造一个数据库产品。

本书作者是 SQLSERVER 的开发组组长。相当于 SQLSERVER 之父，一直把 SQLSERVER 从无到有到跻身世界三大商业数据库之列（ORACLE、DB2、SQLSERVER。过去辉煌的 Infomix、Sybase 如今已经风采不在）。本书的序也是神人级别，Jim Gray。吉姆格雷，但愿我没有拼错他的名字。此爷为计算机界最高科学奖项“图灵奖”的获得者。因为此爷提出了一个概念：数据库事务。

《COM 本质论》

我的技术世界的另一半。我是幸运的，我居然能遇到我技术世界的两个支柱中最重要的两本书。此爷大家估计都知道，此爷对 COM 的理解，比微软自己还深刻。此爷还制定了 SOAP，是 WebService 的通信基础。没有此书，我仍然停留在 COM 编程应用的层次。有了此书，我的组件技术世界才算有了组件体系。我才彻底理解了面向对象、组件，以及如今的 WebService、SOA、WCF、SCA、SDO。即使以后出了比 SOA 和 WebService 更更新的技术，我都能很快理解它的规范和它为什么要这样做。

《设计模式》我也买了一本，我也做架构，但它对于我的架构影响并不大。可能我是个伪架构师，只为了解决企业软件具体问题，而非搭建一个钢筋水泥的框架。所以此书我和《基业常青》放在一起，有时候拿出来看看，把他们看作前进的目标。但现在的路，还无法到达，我还需要继续走。

李维《DELPHI 高级开发》[台湾版名字我忘了，实在抱歉，网上也找不到。当年我们

为了用 DELPHI 开发 COM+（当年 DELPHI5 开发 COM+有内存泄露和线程并发问题，我们甚至联系了 DELPHI 开发组），但是业界却没有这方面深度的书。我们最后到台湾把李维先生的书买回大陆。此书对 DELPHI 开发 COM 和性能调整做了深度的描写。估计大部分 DELPHI 开发者都没有阅读过本书。本书也没有在大陆销售过，我们看的也是繁体文字。没有这本书，我们就会割裂，一边在深入研究 DELPHI，一边在深入研究微软 COM 技术，但两者的结合就是有问题。李维先生给我们做了完美的结合，给我们一代产品的研发带来了扎实的技术指导。

王玉荣《流程管理》

在公司工作的时候，我已经很快成为最顶尖的开发人员。我能解决大家遇到了各种性能问题，内存问题，线程问题，接口变动抽象问题、数据库设计问题等等。但是，一个产品如何畅销，如何更能正规化，如何更能提升产品价格和形象，如何让产品更能帮助客户业务竞争力，我当时在苦苦思索解决方案。王玉荣女士的《流程管理》来到了我的眼前。这本书为我打开了咨询之路。原来软件实施还能这样做。

我们过去开发软件，都是开发完去实施做数据准备，做培训，做推动上线，做服务技术支持。但从未有正规的方法体系，也未有咨询师的实施方法，所以实施显得很土，产品形象提不起来。

《谁动了我的奶酪》

当年我还是个快乐的程序员，不断思考着解决产品开发实施服务全过程的各种问题。从高级开发到实施经理到开发经理做了个遍。直到遇到了这本书。这本书让我惊觉，我的职业规划是什么？我以后要如何发展？

过去，我一直有个梦想就是成为中国一流的开发人员，但是我曾经接近这个目标的时候，让我遇到了此书。此书给我敲了一个大脑壳。

我的奶酪是什么？我的奶酪什么时候会变质？我的下一块奶酪如何得到？我究竟想要什么样的奶酪？

于是，我的职业规划开始的思索，我的职业发展开始走上了计划性，而不是随老板随公司逐流。

《CORBA 企业解决方案》

没有这本书，我的开发我的产品可能还叫一群程序员写出来的软件。我们虽然号称做着企业级的软件，也在实践着企业级的解决方案，与 EMC、IBM、HP、飞利浦、微软成为合作伙伴一同工作，但我们的软件仍然还像是程序员的软件。

没有这本书，我不会对 COM，对 EJB，对中间件，对 ESB，对 SOAP、对如今的.NET 战略有如此透彻及长远的思考。此书对对象注册、定位、消息、事务、安全、容器、异构、持久化、池化、用户会话、可伸缩性、负载平衡、容错都有深刻而体系的描写。此书，从气质到内容，无一不透出企业级解决方案的大气与广度。

你做的是企业级应用吗？看看这本书。

陈宏刚 林斌 凌小宁 张益肇 熊明华 张亚勤 的《软件开发的科学与艺术》

《微软的秘密》有些站在产业的角度上看微软，而且主要看微软这个商业公司如何运作。而这本书更深入地描述了微软的软件开发是如何做的。张亚勤的提纲挈领，指出.NET 战略，服务化、全球化、互联网化。张益肇的软件生命周期、凌小宁的软件设计场景，熊明华的软件项目开发管理都是我多次汲取养分的章节。其中很多做法都被我引入我自己的日常软件设计开发管理当中。很有实用性。

在我出道和升级的过程中，我仍然要感谢两个人，他们深刻影响了我职业发展：

我的师傅。我的公司有个企业文化或企业制度，就是每个人都需要指定一个师傅。有什么问题都可以问他。他也会监督你的工作，带领你快速融入团队融入公司融入当前任务。他

不仅教会了我在职业化公司如何工作如何生存，还教了我很多看人看事的道理。

我当年也禁不住互联网泡沫的诱惑，看人家 263 住在嘉里中心，而且工资给的巨高，我想跳槽。于是请教我的师傅。

他给我的答案是：如果你想挣钱，那么你就去吧。如果你想发展，那么你就留下来。

于是，我想了想，就留了下来。2001 年，国内互联网泡沫破裂。N 多互联网公司搬出嘉里中心。

还有一个小例子，我现在居然也记得很清楚。有一次，我和师傅闲聊起买车，我说我想买个黑颜色车。他问我为什么？我说黑颜色大气，很有企业感。我问他你想买什么颜色的车？他说：红颜色。因为红颜色在黑夜安全，别人都能很容易看见。

对，这就是我师傅。可见他的性格与情操。

李维先生。李维最影响我的不是他写的繁体书，而是两件事情。

第一件，李维先生在 2002 年，在 CSDN 的撮合下，在机械工业出版社的支持下，来大陆进行了一次演讲。他不仅讲到了技术的演变，业界的竞争背景，还讲到了未来的技术发展，程序员的职业规划。短短一个下午，当时北京沙尘暴，但当时的场面比沙尘暴更热烈。李维先生的这次演讲让我更深刻体会了业界的发展与软件人的发展。我的一个多年好友评价我说：你就是从那一年一下变了。变得成熟了，有广度有深度有眼光有战略了，再也不是过去那个莽撞的意气风发的年轻人。

第二件，李维先生推荐我去 borland 中国公司。尽管我婉言谢绝了李维先生的好意，我在此仍然要好好谢谢他。我感觉我自己的全部知识和实战，都是来源于我时时刻刻接触的企业和客户，他们是我所有灵感的源泉。没有他们的问题，我的技术可能没有用武之地，我也不会知道一项技术的客户价值到底有多大，我也不会知道这项技术的缺陷和应用点。我一旦脱离这个环境，让我去介绍纯技术的产品，可能我很快就会被架空，变成一个职业的产品经理，以产品讲产品了。很多外企的产品经理都有这个问题。

再次我还要感谢 5 个网站，在上面我认识了很多朋友，也吸收了很多养分让我认识了很多 DELPHI 界的好友，也提升了我很多的 DELPHI 技术（左轻侯、张晓龙、蒋涛、曹晓刚、千中元、cakk、温柔一刀、王寒松、周爱民、soul、宋兴烈等等，好友太多了）；让我看到最新的软件业动态和最新的技术（上面 N 多牛人）；让我看到了 IT 业界和互联网业界（林兴陆、谭智、刘韧、麦田、何田、老邢就是在那里认识的）；让我和业界第一线的企业管理软件实施人员近距离交流探讨（白鱼谭、白菜、王甲佳都是很好的朋友）；让我打开了咨询之门（一叶知秋和萧秋水是很欣赏的两位）我的大学基础，我的出道成长，书和人不断影响着我，让我变得成熟和深厚，但是仍然有一些书让我变的更加张弛有度。没有这些书，我只能是个技术人员，或是个项目经理，我可能无法把控一艘企业大船的航向。我通过学习这些书和这些人，让我有了出海远航的能力。

它们是：

二月河的《雍正皇帝》

企业由各种各样的人组成。每个人或者每个利益团伙都有各自的目的和利益，而企业也有自己的每年每月的盈利目标和任务。这是一条大船，在大海上，会遇到各种各样的情况，所以每个人的位置，每个人的际遇也在不断变化。可能你会成为船长，也可能突然被大副扔下海。我常常感觉管理像是小时候玩的挑木棍的游戏。每个动作都必须轻，都必须看细节看仔细，不能动其他的任何一个小木棍，你要小心观察哪个小木棍可以挑起来。你要小心翼翼的挑起来。

雍正皇帝给我的感觉就是这样。所以我常常阅读，常常思考企业现状，平衡每个人，包括各个部门的经理，包括客户，包括老板，包括手下。

程东升的《华为真相》

《雍正皇帝》说的是皇帝的事情。皇帝有生杀大权，而且也有古代忠君思想。所以还是有一定局限性的。而现代企业，老板不能决定员工的生死，而且员工也不必忠诚于老板。双方都是互相依存的关系。所以《华为真相》更有借鉴性。华为处于快速发展的通信产业，面对的是国际化的竞争，还面临国内政策的变化，华为的新人管理、员工管理、股权管理、接班人之争、领导人思想，每次我心浮气燥的时候，我都看一看，都能让我静下心，不喜功，不消沉。国内为企业立传的很多，包括《联想风云》《蓝色海尔》《蒙牛内幕》等等，很多，但这本是最客观的。

李维先生写的《Borland 传奇》也非常好。过去风传 Borland 要卖给 Bea、要卖给 Oracle。没想到如今 BEA 也被 Oracle 收购了，Borland 把 IDE 独立成了 CodeGear，最近又被一家不知名的公司收购。唉，有时候时也运也命也。花落谁家，人生就是这样不可测，你只有能掌握的时候尽量掌握，但命运来袭时也要保持好自己，如果真的失败，那也只能是命运的安排。一声叹息。

老外的《毛泽东》

老外名字我忘了。但毛泽东能带来一无所有的中国旧社会民众，从无到有的直到建立新中国，其中面临多少斗争，沉沉浮浮多少，最终成就一代伟人。读古鉴今，互相思考共性和差异，更让我客观的理解如今的中国社会和中国人行，在操盘企业和客户管理和销售突破上，都有很好的心态。

《程序员》杂志每期都买《程序员》杂志。每年还要买合订本。这本杂志让我不断能跟踪业界最前沿的技术和动态发展，给我在具体产品规划发展中起了很大的参考帮助。

《财富》杂志由于我处的公司也是一个典型的中小型企业。整天处理部门间、部门内、客户的各种异常，天天日复一日。就感觉企业只能做这么大，再往大做就不知道怎么走了。所以，经常阅读《财富》杂志。上面的文章并不能解决我做大的问题，也不能给我做大的启发，因为上面大部分的文章都是跨国公司，很多都是成立上百年的顶级跨国公司的案例。之所以阅读，就是希望能把自己的胸怀与眼光放长远了，不要拘泥于现实，要往远看，不要在这个小盘子中斗争这点蝇头小利。心有多大，舞台就有多大。

要感谢的人，目前只有一个，我的老板。

我的老板，他在平衡利益、平衡团队、拿捏权力、客户突破、销售契机、战术迂回、战略创新、管理创新都比我高很多。我从他那里学习了很多，虽然他也把我多次上上下下，有时核心有时踢出，有时捧起有时打压，有时多疑有时亲信。但这么多年来，是他对我的“揉练”让我沉稳成熟许多，把握这艘航船更加坚定更加胸有成竹处惊不乱。

这些书，这些人，在我 10 年的工作生涯中影响着我，让我不断提升自己的技术层次、管理层次、运营层次、眼光与胸怀。

很多人评论说：你的运气真好，总能遇到关键的人和关键的书，你一点弯路都没走啊。我怎么就找不到关键人和关键书呢，问我有什么方法没有？

我想给大家讲一个故事，可能对大家有启发：

说的是两个朋友，一个是品赏蟋蟀的专家，可以凭声音就知道是什么蟋蟀，而且厉害不厉害。而另一个是个商人。他们俩在田野里散步。蟋蟀专家掉了一个铜板，但是蟋蟀专家并没有意识到，他的商人朋友为他捡了起来。蟋蟀专家奇怪的问：你怎么知道我掉了铜板？他的商人朋友说：因为你喜欢蟋蟀，所以你老关注蟋蟀，所以你只能听见蟋蟀的声音。而我喜欢钱，我老关注钱，所以我只能听见钱的声音，而听不到蟋蟀的声音。

就是这个道理。你关注什么，你就会得到什么。想想你自己持续关注过一个东西或事物 10 年么？

还有些网友问我我是怎么读书的，怎么读了这么多的书？这个问题我的朋友也问过他，他很想学习我。他看一本书，往往 3 个月也看不完，看完也讲不清结构，而我和他一块去西单

图书大厦，都是逛同样的时间，我总是能找到很好的书，也能很快找到书的重点，并且能给大家回来后复述并讲清自己的观点。

关于读书技巧，有一点心得：

第一，我经常梳理自己的知识体系，技术的，管理的，业务的。不断梳理，所以心中的主线、重点、未来趋势、空白弱点，都一清二楚。所以我读书都是由针对性的，我从不走马观花。我知道自己的所需，我也知道自己的拥有。而且每看自己需要的内容，就很容易和自己现有的知识体系对接在一起。

第二，我读书就和猕猴吃食一样，先看书的目录，然后发现自己感兴趣的章节，直接找自己关注的问题答案。如果发现这本书有不少自己想看的，就会买回去，细看，并且做笔记。先把自己想看的看完，做完笔记。然后再思考目录，看看自己遗漏什么没有。然后再根据筛选再读一遍遗漏的地方。然后就把书从头到尾都看个遍。对于自己已经明白的就很快翻过去。

第三，从小到大，我都保持着读书的爱好，一本好书在手，经常饭不吃觉不睡。在大学如此，上班后单身也是如此，现在结了婚也是如此。所以家人也理解我这个爱好，给我提供了很多的条件帮助，让我能安心读书。在大学的时候，也是每天睡 3-4 个小时，打工编程赚钱（有时编程太累就把键盘推到一边，头上盖好西服就趴那里睡着了），并且还把财务管理和计算机两个学位拿下来，还阅读深度技术书籍。即使工作后单身，也常常周六日通宵达旦的看书。现在做了高层经理人，负责的事情多了许多，出差也多了不少。但每次都是上洗手间的时候也拿本书，飞机上也带本书。每当《财富》和《程序员》来了的时候，晚上拿回家，吃晚饭收拾完家把孩子打发睡觉后，自己起来偷偷阅读直到全部读完才睡觉。这么多年，睡眠时间短都这么多年过来了，好像我一直是这样，也就习惯了。

15、那根胡萝卜

昨天，有个网友给我写了一个 MAIL，里面诉说了他现在的矛盾和困境，他是一个项目经理，但是他现在很尴尬，根源就是项目奖金。

具体情况是这样的：

一个项目经理，带了三个人。项目经理主要管详细功能设计与测试，其他人开发。但很关键的一点是：需求是老板定的。老板说我希望这个产品具有什么什么样的功能，然后项目经理根据老板与他的交流，他理解后进行设计，然后再分配下去实现编码。

但问题就在于：老板并不是开发人员出身。老板想要的东西，今天是这个样子，明天是那个样子。每次说的都不一样。老板想要一个他心中想要的东西，但是他只是凭感觉说，也没有很成熟的思考。希望让项目经理把细节思考出来，再来验证到底能不能行的通。行不通就再想其他的方法。

就这样，拖拖拉拉一直做着，没有个项目终结，大家都撑的非常累。而且有怠工情况和厌倦的感觉出现。着急什么呀，等老板再变了再说。咱们辛苦开发完，他一觉睡醒想法变了动动嘴皮，咱们的前期工作就被全部推翻，干嘛要浪费自己的精力和感情呢。

已经在这种状况了。这个倒霉的项目经理又做出了一个举动：和老板申请项目奖金。

项目经理是这么想的：以前没有什么加班费(当然现在也没有因为基本不加班)，也没有项目奖金，只有每个月的死工资。福利也没有，所以正在向公司申请项目奖金。得到的结果就是以前公司没有这方面的制度，所以一切得我们先拿出相关方案出来。再因为由于办公室内部开发氛围死板，没有什么激情，我想有项目奖金也有助于提高大家的工作激情。

但我立刻给他指出。这真是一个昏招，尤其在这个阶段。（估计不少人会骂我了，这时

候不要，啥时候要）我想说的是，大部分人创业开公司成为老板，一般两个主要原因：

过去给别人打工，自己出了大力，钱却被老板收走了。所以自己要当老板，把钱自己收了。

过去给别人打工，自己出了大力，但却被老板牢牢控制着，做的很憋气，不如自己当老板好好把自己的想法大展宏图。

从以上两个原因看，我们就很容易理解了老板的两个核心：为钱，为权。所以我们经常看到老板会出现这样的情形（这是现实）：向员工要利益（一般能不出钱就不出钱，从逃税避税到三险一金到出差补助，能不交就不交，能少交就少交，能少给就少给），向下属要权力（经常不想管了就让项目经理管，想管的时候就把项目经理踢到一边，完全无视他人感受）。

大部分老板开公司的出发点和目的点都是钱和权（有其他梦想的除外，一般以实现梦想为主要目标的，大多企业要么大成，要么就是失败。不过还是失败的多。企业存活，还是很现实的），所以我们也不难理解老板的行为。

不过，面对这种常见企业现象，中国内地员工会有两种表现：一种是农民工级别的，他们还处于生存线上。所以一旦失业他们就面临饿死的情况。所以他们忍了。对于中国 IT 业，大部分员工都是大学本科毕业，学历和家庭都还过的去，所以对于老板这种不仁行为，就会产生你不仁我就不义的行为，和老板做斗争，玩老鼠和猫的游戏，有时候老板是猫，有时候老板是耗子，此消彼长。与人斗，其乐无穷。

但是，企业不是黄世仁和杨白劳的关系。员工还要靠老板发工资（你可能想此处不留爷自有留爷处。但大部分企业都差不多，你在此处不愉快，其他地方也愉快不到哪里），老板还要靠员工赚更多钱（虽然说每个员工都是一颗棋子，离了谁都行，但真的突然集体辞职了，还真是麻烦事）。

我们经常看电视《麻辣婆媳》。双方越斗争，裂痕越大，最后只能决裂，然后再找一家，继续斗争继续决裂。人生如此反复。要么双方互相忍让互相融合互相理解。

老板也是人，老板也是从员工走过来的。他当员工的时候和你现在当员工是一样的想法，一样痛恨老板。所以，两好放在一起就是一个大好。

而老板，项目拖拖拉拉这么多月都还没有成型的结果，还每个月发工资，本来心里就有气（可能根源就在于他，但灯下黑这种事情谁都知道）。你还这时跟我提项目奖金？我给不给，我怎么给，给多少是个合适？给多了给少了都是麻烦，把问题踢回去，看看你们这帮家伙怎么想要多少。要的少了，为了项目，就当破财免灾。要的多了，再说。为了防止谈崩了这帮家伙起坏心眼，我得想好退路，不能让这帮程序员把我给耍了。我要首先行动，把他们先控制起来，防止意外。

而员工呢，一看被老板给拐弯抹角的踢回来了。KAO，要么明说不给，还让我们提方案。不想给就明说，来暗的。我们辛辛苦苦这么长时间，你朝令夕改还怨我们。看来这样的老板成不了什么气候，跟着他也是受窝囊气，吃苦不来钱。该跳槽了。

而项目经理呢。这时被夹在了中间，该怎么办呢？

我给大家先摘录一段很著名的管理小故事（摘自成君忆先生的《水煮三国》）：

兔王遇到的难题南山坡住着一群兔子。在蓝眼睛兔王的精心管理下，兔子们过得丰衣足食，其乐融融。可是最近一段时间，外出寻找食物的兔子带回来的食物越来越少。为什么呢？兔王发现，原来是一部分兔子在偷懒。

2.奖励的必要性兔王发现，那些偷懒的兔子不仅自己怠工，对其他的兔子也造成了消极的影响。那些不偷懒的兔子也认为，既然干多干少一个样，那还干个什么劲呢？也一个一个跟着偷起懒来。于是，兔王决心要改变这种状况，宣布谁表现好谁就可以得到他特别奖励的胡萝卜。

3.随意奖励，激起不满一只小灰兔得到了兔王奖励的第一根胡萝卜，这件事在整个兔群

中激起了轩然大波。兔王没想到反响如此强烈，而且居然是效果适得其反的反响。

有几只老兔子前来找他谈话，数落小灰兔的种种不是，质问兔王凭什么奖励小灰兔？兔王说：“我认为小灰兔的工作表现不错。如果你们也能积极表现，自然也会得到奖励。”

4.兔子们学会了变脸于是，兔子们发现了获取奖励的秘诀。几乎所有的兔子都认为，只要善于在兔王面前表现自己，就能得到奖励的胡萝卜。那些老实的兔子因为不善于表现，总是吃闷亏。于是，日久天长，在兔群中竟然盛行起一种变脸式(当面一套背后一套)的工作作风。许多兔子都在想方设法地讨兔王的欢心，甚至不惜弄虚作假。兔子们勤劳朴实的优良传统遭到了严重打击。

5.有规矩才能成方圆为了改革兔子们弄虚作假的弊端，兔王在老兔子们的帮助下，制定了一套有据可依的奖励办法。这个办法规定，兔子们采集回来的食物必须经过验收，然后可以按照完成的数量得到奖励。

一时之间，兔子们的工作效率为之一变，食物的库存量大为提高。

6.注意奖励制度的改革兔王没有得意多久，兔子们的工作效率在盛极一时之后，很快就陷入了每况愈下的困境。兔王感到奇怪，仔细一调查，原来在兔群附近的食物源早已被过度开采，却没有谁愿意主动去寻找新的食物源。

有一只长耳朵的大白兔指责他惟数量论，助长了一种短期行为的功利主义思想，不利于培养那些真正有益于兔群长期发展的行为动机。

7.当规矩被破坏之后兔王觉得长耳兔说得很有道理，他开始若有所思。有一天，小灰兔素素没能完成当天的任务，他的好朋友都主动把自己采集的蘑菇送给他。兔王听说了这件事，对都都助人为乐的品德非常赞赏。

过了两天，兔王在仓库门口刚好碰到了都都，一高兴就给了都都双倍的奖励。此例一开，变脸游戏又重新风行起来。大家都变着法子讨好兔王，不会讨好的就找着兔王吵闹，弄得兔王坐卧不宁、烦躁不安。有的说：“凭什么我干得多，得到的奖励却比都都少？”有的说：“我这一次干得多，得到的却比上一次少，这也太不公平了吧？”

8.胡萝卜也会失去激励作用时间一长，情况愈演愈烈，如果没有高额的奖励，谁也不愿意去劳动。可是，如果没有人工作，大家的食物从哪里来呢？兔王万般无奈，宣布凡是愿意为兔群做贡献的志愿者，可以立即领到一大筐胡萝卜。布告一出，报名应征者好不踊跃。兔王心想，重赏之下，果然有勇夫。

谁也没有料到，那些报名的兔子之中居然没有一个如期完成任务。兔王气急败坏，跑去责备他们。他们异口同声地说：“这不能怨我呀，兔王。既然胡萝卜已经到手，谁还有心思去干活呢？”

老板和员工就如兔王和兔子。管理的激励大多如此。

也就是说：在管理技巧上，项目奖金不是唯一的激励手法。而且未必是最有效的方法。每个人每个阶段的诉求是不一样的在我讲的这位项目经理的案例上，他们目前遇到的最致命的困难是：

主要最大的就是需求分析难确定，改动比较大，累； 2 朝令夕改，也改腻了，不想改了； 3 周期太长，没个结束，什么是个头儿啊？

这个现象和我们做客户定制化开发实施很相似。

我们一般采取的解决方法是：

每次的和老板的交流更改，都先写文字性的东西，给老板看理解的对不对。当然，老板认为他已经说清楚了没必要再详细看你理解的对不对，所以看也不看，最后你认为他看了就开始做，做出来老板一看根本不是自己想要的。所以，要逮住他跟他对照着文档讲。有机会有时间就逮住他。看不到他就给他打电话。老板一般都声称很忙。但你不逼他，他一看你这么长时间也没有动静，也没有开发，到底在干吗。所以与其让他产生误会，不会你直接明了

找他，否则你就会被他 K 了。你想想后果。

每次和老板的交流更改，都把他过去的几种思路都在文档中列出来，并且说明每一种的优点，每一种的缺点。并且你要综合出一个综合所有优点避免缺点的方案也写在文档中。让他感觉，这就对了，没有缺憾了。

很多网友估计又要暴跳了。扯淡，老板会听你的？人家是老板啊，人家想怎么做就怎么做，你就是一个干活的小兵，你把自己当个人物，什么狗屁项目经理，人家都不把你当人看。

嗯。这种方法我就用过，而且奏效。我在空降的时候公司也是草莽摊子，老板直接指挥。但每次我想的方案比他想的还全还远，于是他就放心了，他干涉的就越来越少了。

有人又讲了，你比老板还周全还远，那你干吗不当老板。

嗯。老板这个工作不是你想的全想的远就能做的。职业经理人和老板有本质的区别。有的人适合做老板，有的人就适合做职业经理人，一当老板就干不好了。大家职业经历久了就明白了，现在多解释也理解不了。多经历，会明白许多。

我给这个项目经理又支了几招：

大家找一天下午不开会。务个虚。就是大家发泄一下。随便说。

你把大家的发泄主题都记录下来。

把大家对于产品对于需求对于客户的抱怨记录下来。大家集体把拖时间最长、改动最频繁的、改动最累人的都一起解决了。把改动频繁的，改动难的，做好公共代码或灵活的接口技术。

几个人一起解决这几个问题。不要分散解决。分散解决一是思考不全面，会出现新的问题。另外，项目团队现在很涣散，通过大家共同解决一个问题，会感觉大家在一个战壕里战斗，大家的心又凝聚在了一起，目标也是一样的。

攻克了，就感觉轻松多了。

就这样，问题解决，老板员工齐欢喜。

但项目奖金问题，没了下文。

有网友肯定暴跳。这不是扯淡么，我辛苦工作，不就为了钱么。现在是老板的问题解决了，我钱没挣到啊。你这种方法是亲老板派。

嗯。老板是不见兔子不撒鹰。而且见了兔子也能不撒就不撒。（胸怀宽广任贤用人和谐共处的老板除外）在项目前期，中期，老板一点好处都没看到，未来能不能卖个好价也不知道。一点收入没有，就看见每天的费用哗哗哗的流，还跟要项目奖金？

如今，项目成功，大家总有点好处吧。但现在项目已经结束，老板已经无后顾之忧，你提也没有用，爱谁走谁走。卸磨杀驴的老板比比皆是。像史玉柱这种隔天就发两三千块钱的老板少见，所以像史玉柱这样成功的老板也很少见。

我当年做开发经理的时候（我也是从基本员工一步步上来的，劳苦大众经历的事情我也经历过），我也没奢望要到个多少钱，很明确，封闭开发 1 个月，封闭期间双薪。

为什么封闭开发。并不是日赶夜赶。原因有三：

为了有个明确的限制。这样，项目周期就不会漫漫无期。

要项目奖金有理由。封闭开发，压力大，加班。

这个双薪很好量化衡量。就是多一个月的工资。老板很容易做出决定。

所以，不要让老板想问题答案。而应该把老板能想到的答案你先想到，然后给老板一个选择题，这样你就不会出意外了。

我想起了赵本山的一个小品：

宋丹丹说：你在家也不和我说话，老一个字一个字往外蹦。

赵本山：我说蒜，就是咱家紫皮大蒜。我说醋，就是咱家的陈醋。这么多年了，我说啥想啥你都知道，我费那口舌干啥啊。

嗯。跟随老板久了，也就这个味儿。

16、七里香

窗外的麻雀在电线杆上多嘴你说这一句很有夏天的感觉手中的铅笔在纸上来来回回我用几行字形容你是我的谁又是凤凰花开，新人毕业，老人跳槽的季节了。这几天的电台老是这样的主题。有人要找工作，有人刚刚入职，无措的看着这个和学校 and 课桌截然不同的公司和工位。有老人要跳槽，突然由一个老油条变成了一个新员工，一个新环境新人际关系需要适应。对于项目经理来说，有新的员工要管了。他可能是刚刚毕业什么都不会的耷拉着头发的新新人类，他也可能是在各个公司混迹跳槽的老油子。

我想起了我毕业，以白纸新人的面貌进入了公司。这份工作对于只身一人闯荡北京的应届毕业生来说非常重要，因为这份工作要养活我在北京生存下去。在北京上学的外地孩子可能有一个惯性思维，就是反正在北京上的学，毕业了同学大部分都留在北京了，那我也留在北京，熟悉的城市熟悉的人，就这样落了下来。但是对于一个和北京一点关系都没有的人来说，到北京，就是为了闯荡，为了闯荡出一个自己的梦想。和美国梦一样。这就和北京人、在北京上学后留在北京的外地人有截然不同的区别。因而对自己要求分外严格，我为什么来北京？虽然我已来北京 10 年，在这里买房买车结婚生子生活工作，但我仍然会时常想起这个问题。

当时住的地方，离公司大约坐公交车一个半小时。公司九点上班，还不是门卡制所以有人掌管钥匙。我往往在八点二十就到了公司，坐在公司的楼梯上看英语书。

公司为集中进来的新人做了一次为期一个星期的培训，从公司制度到公司产品，从出差报销到员工制度，从行政领取到福利介绍，全的很。但是，我都忘记了。到了真正正常工作的时候，到了报销的时候，还得问师傅，还得问行政助理。产品知识，还得日常工作中研发部内部培训。

我也曾经做过几次新人培训，但一工作开，发现也是全白费。所以我现在就不培训新人了。采取行动中开火，实战中成长，师傅带领指导监督管理。

当然，我也被指派了一位师傅带我。我的师傅，并不是我的领导。我的领导是研发部经理。我的任务是研发部经理分配的，而师傅是指导监督我的新人期工作的。师傅只是一个非正式传帮带，不是职位或职务。每个新人进来公司，都有师傅。即使一位在其他公司工作多年的老员工跳槽进来，也会由研发部经理适当的安排一位师傅。

每个企业和组织都有其内部的制度和规则。这些规则和方法可能不是被白纸黑字的写下来的，但由于长期部门内协作和经理的个人管理风格和部门间的利益力量博弈，所以形成了部门内说不清道不明的规则。它存在，它不是明面上写的，它存在于人们日常的工作流程和人际关系中。

而一位新人，不管你是应届毕业生还是老油子，你既然来到了一个新的公司新的环境，就必须很快速融入这个独特的部门文化中。你不会从任何文字性的东西中阅读理解到，但它就是影响着你的日常工作开展。所以师傅是很有必要的。

我的经理为我分配了一个活，做报表。

做过企业管理软件信息化的人都知道。最难的不是编程开发录入数据的窗口，而是做报表，并且报表数据做平了，几张关联的报表的数据还能对平了，而且钩稽关系都正确。

这是很不容易的。因为表结构已经定死了，设计的时候，可能设计者多考虑了数据录入，却轻视了数据统计，所以很可能这个表结构出报表非常难，甚至你不拿存储过程+临时表来

做就想不出更好的办法。更糟糕的是，数据库不知道是谁调试程序时发生了问题，侵入了异常的数据，你怎么都想不明白怎么数据平不了。更为严峻的是，我面临的是 22 个子系统，1000 多张表的一个大数据库结构。更让我着急的是，我的经理还给我设置了时间期限，不是让我学习试验的，而且我做的东西要给客户用，这是要卖钱的，做不好，客户就不给钱。

我对业务一无所知，对内部怎么写的代码也一无所知，对这 1000 多张表的关系也一无所知。如果放在如今，肯定新入职的员工会直接跟我说：我不会。

我很是珍惜这份工作，当然全单收到。幸亏公司的数据表管理用的是 PowerDesigner，里面有详细的表结构说明。

业务不明白，我找到我的师傅。我的师傅给了我一个市场部的 PPT。里面有关于产品的总体介绍。我很快就找到了重点的系统模块，以及各个系统的关联关系。

管理软件这个东西，我在学校期间打工做的就是它，自然对这类东西有一种规律性的认识。先看各种字典，然后看各个业务窗口。各个业务窗口用到哪些字典。各个业务录入窗口都被我看作是一个业务单据。什么入库单、出库单、采购单之类。然后我再看报表。对照着报表再去找这些数据大概是来自哪个数据录入窗口产生的。

学完了业务系统，看经理给我安排的报表任务。我已经心里有数，基本知道了这些报表都涉及到哪些数据录入窗口。

尝试录入数据，修改数据，用 SQL 跟踪器跟踪 SQL 输出，得到 SQL 语句，一下就明白了能用到哪些现成的视图，以及我要用到的表和表之间的关系。

我并没有去研究这 1000 多张表。我也没有把 22 个子系统全部研究完毕。我只研究了我要研究的表结构，我只研究我要研究的业务模块。

我从入职工作到把报表出来，并且测试出乎，只用了两个星期。很快，我就成为了高级开发人员。我对业务系统的了解，我对内部结构的了解，超过许多老员工（很多新人都是一年后才能上手）。我能清晰的画出业务流程图，并且指出现有系统还可以再优化业务流程的地方，还有处理不合理比较绕弯的地方，也能指出哪些地方应该留下历史追溯，却现在没有留下，有审计漏洞。

所以，我现在引导新员工，也是通过做报表。我不会安排任务时间让他们专门学习业务系统，让他们阅读系统设计说明书或帮助文档。这没有用，因为这样做没有目的性，他们不知道该看什么才有用。直到真正遇到任务，带着问题去找相关的答案，他们才会去真正用心了解这些。

所以，我一向的管理风格都是以问题找答案。没有问题，就不乱动。即使有问题，问题不影响最终的大局和目标，也能不动就不动，能不做就不做，不急不躁一切都在预料之中。我不喜欢大革命大运动大学习大整风（可能是我家在 60-70 年代历史原因影响）。平时不润物细无声的管理、指导、引导，到了发怒的时候才去大动手，对于一个组织来说太伤筋动骨。

在研发部，还有免费的咖啡和茶可以喝，当然茶也不是那种茶叶沫子，而是还算不错的茶。看得出公司的巧思安排人文关怀。喝没了，不用开发人员自己申请领取，研发部有行政助理管，她会照顾大家的一切工作以外的事项，什么报销整理阿，纸呀笔呀，乱七八糟影响大家开发工作的，都是行政助理全包。我一直喜欢这种氛围，可以安心去做自己最擅长的事情，把身外之事都交给细心忍耐机灵的女孩助理去做。

在研发部，还有每周五不强制的下午学习。大家如果觉得有必要，就早就计划好让研发部内某人讲一次业务系统难点，或设计难点，或难点技术最新技术共享。这种学习型的组织一直让我很推崇。看似大家在浪费时间闲聊，但大家的团队凝聚力，团队学习力，团队配合力，真是空前的高。

这都是很特别的研发部文化。在三五个人十来条枪的企业，可能做不到，也可能能做到。做不到，大部分都是老板连茶叶都不舍得给员工买，甚至企业文化弥漫着争抢夺的气息，刚

买回来一包茶第二天就不翼而飞，最后老板都下令让某个人管茶，平时茶放到这个人的抽屉里，谁喝谁来要。没想到，很少有人来亲自要，茶叶最后也被看管的这个人带回家。之所以形成这种尴尬的文化，就是由于企业根源与老板的根源，可能就是争抢夺，所以从上到下都是如此。对于利用工作时间团队学习，很多中小企业都做不到。因为每个人都顶着一个到几个项目的开发或实施，每个人都超负荷工作，大家根本没有心思去挤出时间团队学习。而且经常每个人都出差到各地去现场开发现场实施，根本聚不到一起，所谓的团队，只是一个虚词，只有个人，只有单枪，只是乌合之众，根本没有CS角色配合这样的战队。

我最后也成为了一个老员工了，我也由于个人发展原因跳了槽。天下没有不散的筵席。我又作为一个新人来到了一个新的环境中，要面临一个我未知的办公室潜规则，要置身于一个我还没有了解的利益集团中。

这次我是娴熟的老员工。

公司所有人的联系方式要到每个部门的头都短期内认识了。认识方法就是吃饭现有产品深刻使用操作了一遍，发现了问题，也知道了现状，也知道了未来如何改进现有客户，现有正在跟单的项目情况离老板最亲近的几个人，和他们玩，聊天，吃饭，深刻观察了解他们的做事方法，观察他们为什么能离老板这么近。近朱者赤近墨者黑。这样也能看出老板是个什么样的人每个部门的事实领导人，有些人不是头，但是很有影响，是事实上的内部头目，和他们玩，请他们吃饭，把部门的利益团伙和公司历史，公司的真正收入来源和真实收入数额和盈利模式都了解清楚瞅机会，创造一切机会，和每个老板聊，了解每个老板的想法改进现有产品最大的几个难点问题，寻求短期内出彩的活让公司所有人立刻刮目相看很快，我又成了优秀的管理者，开展工作巧妙平衡执行恰到好处。

我想起了李开复曾经讲过的一个经验，他也是要联系方式，然后主动联系吃饭聊天（聊什么就很多啦）。如出一辙。

希望自己的新人经历和跳槽适应新公司的经历能给大家带来启发。不管是还没有走上社会或刚刚毕业或刚刚入职的学生，还是不知道跳好还是不跳好正在前后犹豫的老人（估计还没跳过槽），还是跳过去但遇到排挤或冷遇孤独的老人，希望能帮助你们走好自己的第一步。

附录给大家摘录一段华为任正非写的《致新员工书》，笔者个人认为是很好的新人入职心态指导。

作者:任正非您有幸进入了华为公司。我们也有幸获得了与您的合作。我们将在共同信任的基础上，度过您在公司工作的岁月。这种理解和信任是愉快奋斗的桥梁与纽带。

华为公司是一个以高技术为起点，着眼于大市场、大系统、大结构的高科技企业。以它的历史使命，它需要所有的员工必需坚持合作，走集体奋斗的道路。没有这一种平台，你的聪明才智是很难发挥，并有所成就的。因此，没有责任心，不善于合作，不能集体奋斗的人，等于丧失了华为进步的机会。那样您会空耗了宝贵的光阴，还不如试用期中，重新决定您的选择。进入华为并不意味着高待遇，因为公司是以贡献定报酬的，凭责任定待遇。对新来员工，因为没有记录，晋升较慢，为此十分歉意。如果您是一个开放系统，善于吸取别人的经验，善于与人合作，借助别人提供的基础，可能进步就会很快。如果封闭自己，怕工分不好算，就需要较长时间，也许到那时，你的工作成果已没有什么意义了。实践是您水平提高的基础，它充分的检验了您的不足，只有暴露出来，您才会有进步。实践再实践，尤其对青年学生十分重要。唯有实践后善于用理论去归纳总结，才会有飞跃的提高。有一句名言，没有记录的公司，迟早要跨掉的，多么尖锐。一个不善于总结的公司会有什么前途，个人也不是如此吗？

实践改造了人，也造就了一代华为。您想做专家吗？一律从工人做起，已经在公司深入人心。进入公司一周以后，博士、硕士、学士，以及在内地取得的地位均消失，一切凭实际才干定位，已为公司绝大多数人接受。希望您接受命运的挑战，不屈不挠地前进，不惜碰

得头破血流。不经磨难，何以成才。公司要求每一个员工，要热爱自己的祖国，热爱我们这个多灾多难、刚刚开始振兴的民族。只有背负着他们的希望，才可有进行艰苦的搏击，而无怨言。我们总有一天，会在世界通信的舞台上，占据一席位子。任何时候、任何地点都不要做对不起祖国、对不起民族的事情。要严格遵守公司的各项制度与管理。对不合理的制度，只有修改以后才可以不遵守。不贪污、不盗窃、不腐化。严于律己，宽于待人。坚持真理，善于利用批评和自我批评的方法，提高自己，帮助别人。

您有时会感到公司没有真正的公平与公正。真正绝对的公平是没有的，您不能对这方面的期望值太高。但在努力者面前，机会总是均等的，只要您努力，您的主管会了解您的。要承受得起做好事反受委屈。没有一定的承受能力，今后如何能做大梁。其实一个人的命运，就掌握在自己手上。生活的评价，是会有误差的，但决不至于黑白颠倒，差之千里。您可能不理解公司而暂时的离开，我们欢迎您回来。您更要增加心理的承受能力，连续工龄没有了，与同期的伙伴的位置拉大了。我们相信，您会加步赶上，但时间对任何人都是一样长的。

希望丢掉速成的幻想，学习日本人的踏踏实实、德国人的一丝不苟的敬业精神。真正生活中能把某一项技术精通就是十分难的。您想提高效益、待遇，只有把精力集中在一个有限的工作面上，不然就很难熟能生巧。您什么都想会、什么都想做，就意味着什么都不精通，任何一件事对您都是做初工。努力钻进去，兴趣自然在。我们要造就一批业精于勤，行成于思，有真正动手能力、管理能力的干部。机遇偏多于踏踏实实工作者。公司建立了以各部门总经理为首的首长负责制，它隶属于各个以民主集中制建立起来的专业协调委员会。各专业委员会委员来自相关的部门，组成少数服从多数的民主管理。议事、不管事。有了决议后由各部门总经理去执行。这种民主原则，防止在一长制中的片面性，在重大问题上，发挥了集体智慧。这是公司六年来没有摔大跟头的因素之一。民主管理还会进一步扩展，权威作用也会进一步加强，这种大民主、大集中的管理，还需长期探索，希望您成为其中一员。

公司永远不会提拔一个没有基层经验的人做高级领导工作。遵循循序渐进的原则，每一个环节对您的人生都有巨大的意义。您要十分认真地去对待现在手中的任何一件工作，积累您的记录。要尊重您的现行领导，尽管您也有能力，甚至更强。否则将来您的部下也不会尊重您。要有系统、有分析地提出您的建议，您是一个有文化者，草率的提议，对您是不负责任，也浪费了别人的时间。特别是新来，不要下车伊始，哇啦哇啦。要深入地分析，找出一个环节的问题，找到解决的办法，踏踏实实地一点一点地去做。不要哗众取宠。

在公司的进步主要取决您的工作成绩，一个高科技产业，没有文化是不行的。业余时间可安排一些休闲，但还是要有计划地读些书。不要搞不正当的娱乐活动，绝对禁止打麻将之类的消磨意志的活动。为了您成为一个高尚的人，受人尊重的人，望您自律。谁为谁服务的问题一定要解决。公司总的是为用户服务，但具体来讲，下一道工序就是用户，就是您的“上帝”。您必须认真对待每一道用户。

要关心时事，关心国家与民族的前途命运。提高自己的觉悟。但不要卷入任何政治漩涡，指点江山。公司不支持您，也不会保护您。公司坚持员工必须跟着社会潮流走。当前，要承认只有共产党才能领导中国，否则就会陷入无政府主义。一个高速发展的经济社会，没有稳定，没有强有力的领导，陷入无政府主义状态是不可想象的。共产党的缺点，通过整党和教育来解决。我们可以帮助它，但必须是善意。公司在飞速的发展，迫切地需要干部，希望您加速磨炼，与我们一起去担起明天的太阳。

17、走钢索的人

架构师是个很神圣的词。盖茨，世界首富。微软，世界最大最富有的软件公司。盖茨是微软的首席架构师。

好多程序员流口水，一听某人是架构师，就两眼发亮，比技术总监的头衔还要厉害。

一想起架构师，大家就想起那些 UML 设计工具、类图、时序图，想起那些水泥大楼的框架和地基，想起了那些如百变金刚的开发平台，想起了那些让人眩目的反射、元数据、FrameWork、设计模式、面向对象、重构。

很多人想当架构师，感觉架构师是技术职业发展的最高境界，在往上走就有管理职能了，如技术总监和 CTO 或研发总裁之类的头衔。

李维先生曾经有过一次演讲，讲到了一个架构师应该具备的特性：

核心软件技术。要攻克数据库设计问题，必须深入了解数据库的工作原理，而不是会写复杂的 SQL 会管理个备份会设计个表结构就算精通数据库。有人甚至把会用 hibernate\struts\spring 当作自己会核心软件技术。

产品特性。你学了那么多核心技术，到底要干吗？我一直在商业软件公司工作，没有在研究所工作过。我各种技术要做到的就是帮助企业软件生产，如何更快更省力气质量更好市场竞争力更强。我总是以这个原则来验证一项技术是否对于我的工作来说而实用。现在技术多如牛毛，在各个层次各个领域解决着各个环节的问题。如果不以解决自己工作中的问题为圆心，很容易陷于到大量学习却越来越茫然找不到出路的境地。

软件趋势。在企业管理软件开发领域，往往会见到这样的现象：不少开发人员精通客户业务需求，深入第一线做客户实施。他们学习技术也是为了解决现有手头的问题。尤其企业管理软件开发领域，技术要求并不高，而如果不了解客户需求，开发的软件实用性就不强，即使你的功能开发的又性能好又安全性好也没实用意义。所以，不少在企业管理软件开发领域工作多年的开发人员，形成了技术轻视观，甚至有种核心技术学习无用论的思想。但企业管理软件开发领域，经过十多年的发展，已经面临了不少挑战。但是很多人觉得那是大环境的事情，大环境不是一个人一个公司能改变能影响的。大环境变，咱们就跟着变。大环境不变，咱也照旧。但是，我已经经历过了很多时代，见证了很多遗憾，大环境发生改变了，自己却跟不上了。

时代、单机\局域网时代、互联网时代、移动增值时代。每一个时代都出了黑马，赚取的金钱突然高出传统模式数倍，而传统模式者还是在继续走传统模式，辛苦的赚钱，而且随着价格战的加剧，越来越辛苦，但还不思改变者并且还认为不可改变者大有人在。

创新技巧。我们往往会遇到这样的情况：要解决手头的问题，摆在面前的有 N 种技术方案。选择哪个都有缺点，综合来用又感觉牛刀杀鸡了。有时候，我们还会遇到另一种技术选择，未来的软件趋势一定是那样那样的，但现在还没有达到，现在的技术方案都是过渡期的，所以我们还要等。否则利用现在的过渡期技术，开发出来就被淘汰了。如果是这种以现状看技术的思路，不管技术发展到什么阶段，都有遗憾，都在向未来的未来过渡。所以，作为一个架构师，比别人厉害就厉害在，总是能把手里这些技术巧妙的利用，以解决自己的问题。当然，你想把你手中的技术能用活，你必然是理解这项技术的来龙去脉和这项技术的适用领域，还要深入理解这项技术的工作原理，还要清楚的认识到你解决的问题领域，否则，你无法把你的技术和你要解决的问题结合在一起。

我对李维先生的这四点讲述颇为赞许。架构师总是游走在技术和业务之间，既要兼容软件历史不能割裂又要面向未来发展，所以我老把架构师称为走钢索的人。

很多人也想成为具有这样特性的架构师，但就是不知道该怎么走这条路。我就讲讲我的经历。

我刚出道的时候，很快成为公司技术出众的程序员。有人跟踪调试了一下午也找不到错误的，找我；有人不知道这个错误是怎么引起的，找我；有人不知道某个需求怎么实现代码方便，找我；有人要优化数据库性能，但怎么都速度提不上去，找我；有人要修改一段超复杂的代码，怎么也理不出来，N 多判断和函数嵌套，脑袋思考不过来了，对代码的复杂度掌控不了了，找我；我就跟一个活雷锋一样，大家也好像觉得我就是个活字典，有技术问题找我总没有错。就这样，我在研发部有了很好的技术信任，也有了很好的人缘。

而架构师要做的工作，是许多人工作的基础。如果没有很好的技术信任，大家怎么敢把他们的工作搭建在你的基础之上。如果没有很好的人缘，大家怎么愿意把他们的工作搭建在你的基础之上。

就是由于我解决了很多业务开发的问题，我了解了很多业务开发的现实状况，并且还能提出简洁有效的解决方法，而且解决方法不死板不铁板一块能保持独立灵活通用性，给其他人的工作带来了很好的工作效率，所以领导才信任我能做好这一块，并且适合做这一职位。不是随随便便一个人深刻学习了核心技术，然后申请领导要当架构师。

其实，我开始做的也仅仅是公共代码员。但是，很快面临了一个尴尬。

简单的，虽然可能每个开发组都重复写同样类似的代码，但是由于简单，所以每个业务开发组都自己写了。

复杂的，往往业务开发组组长都认为这个功能是自己这个组的个性功能，所以还是自己写。

所以，只有人们解决不了来找我时，我才能上场。

这干坐着不是回事。我得自己想辙。

于是，我在忙“公益事务”做活雷锋之余，看到他们在扎堆开会我就主动去旁听。每次我都能提出很独到的见解。并且能帮助他们写公共抽象代码，能帮助他们提高不少工作效率。所以他们非常愿意让我旁听，并且听取我的意见。我也能很快写完让他们用。他们一用，发现果然好用，而且不用他们自己写代码了，功能实现的还非常巧妙公用，性能也好稳定性也好扩展性也好。到后来，每次开会都主动叫我。这样，我的工作就越来越多了。

随着各个业务组不同差异的需求都希望我来帮他们抽象出公共的，我就在思考我手里的这部分代码。我不能今天他们提一个我写一个。他们倒是轻松了，但我手里就好似一盘散沙一样。于是，我不断重构我的公共代码，架构体系框架就这样慢慢成型了。各种各样公共工具，调试工具、优化工具、动态设计工具，凡是能帮助业务开发组人员加快效率的，我都做了工具或写了公共函数 DLL。尽量简单易用，不让他们觉得麻烦或不顺手。

过去，各个业务开发组过去是开发人员素质不齐，有人责任心强，有人随意；有人细心，有人粗心；有人理解客户业务深刻，有人理解不深刻。所以开发出来的质量良莠不齐。自从这样做了以后，各个组写的代码少，很多都是我写的公共代码。我的技术好，写的代码质量高，而且是公共的，有错误，一改，大家都没有问题了。所以我们整体的软件产品的产品质量、生产速度都提高了不少。

我发现，大家越找我，各种需求交织在一起，越复杂，我就越需要更深入的学习技术，深刻理解各种技术的差异性和适用领域，去思考各项技术的发展历史、未来趋势，并且自己做 DEMO，看能不能更好的解决大家的问题。因此，我的技术能力也越来越高。如果我不去解决这些不去第一线想也想不到的客户需求，我根本就想像不出我某项技术还能这样用。

这就是我的螺旋上升之路。

我那天重翻上个月的《程序员》杂志，看到了我朋友周爱民写的一篇《做人、做事、做架构师-架构师能力模型解析》，他也提到四点，技术能力、业务能力、人际关系、个人内在

素质。和我的情况很类似。

有一部分所谓的架构师，技术超深厚，框架堪比 Spring 之类，但自己一个人闷头写框架不断优化，力竭使用最先进的技术思想，希望把最豪华的设计模式融进去，希望把 OSGi 融进去，希望把 AOP 融进去，全无视那些想利用框架减轻自己工作量提高工作效率的应用功能开发同事。这是在用公司工资玩技术呢，还是在满足个人技术幻想呢，还是在实验呢？到底在干吗？价值在哪里？

还有的人不会推广自己的框架。不善言辞，就幻想着技术总监能够通过行政命令让大家必须用框架，能不自己写代码就不自己写代码，能交给框架做的就交给框架做。但技术总监号召完了，大家仍然我行我素，各自开发为政，让框架开发者很孤单。

还有的人也不会推广自己的框架，沉迷在自己的理想世界。好不容易技术总监召集大家让大家来听听框架如何应用，但自说自话，满口自己最得意的词汇，听得业务功能开发人员云山雾罩。大家问些问题，如这样的业务开发难题，框架怎么解决？于是，框架开发员就和业务开发员争论了起来。框架开发员觉得这根本就不能答应客户这种变态的需求，而业务开发员说这就是现状。框架开发员说你可以这样这样，业务开发员说这样太麻烦，框架开发员立刻还口这还麻烦？于是双方各执一词，框架也没推广成功。

我手底下有个框架开发员。他的技术渴望很强烈，为了技术难题攻克，可以不吃不睡。并且技术敏感度很强，学习也快。所以当时我感觉他是个程序员的料，就把他拉到我的手下。

但是有个问题，他写出的框架代码，在平时开发业务功能的时候挺麻烦。大家可能需要的是一把铁锹，但是他却给大家 N 根不同长度不同粗细不同材质的木棍，N 个不同形状不同用途的铁锹头。大家会有 N 种组合。不仅导致他写代码老超任务期，而且也让使用人感觉没多大帮助。使用起来复杂，而且还得配置这个配置哪个，需要注意的地方太多。业务开发组的同事就不愿意用，还不如把代码自己直接写死了得了。

超期还会影响业务功能开发组的使用。本来人家是为了想加快自己的工作效率。你答应好这个星期给业务开发组提供一个功能，但你没有拿出来。就耽误人家进度。你多次拿不出来，人家业务开发组还不如自己开发一个呢，求人不如求己。

我最后警告他：如果你认为自己技术够牛，那么你必须证明你能很快做出来。如果你认为自己技术够牛，最好能牛到，只提供一个函数就解决了他们的问题。别这个代理类，那个聚合类，这个唯一实例类。最好连参数也没有，大家调用一下写一句代码就 OK。甚至你做得好，大家都不用调用你的代码，你可以包含在基础框架中，你自己去判断什么时候什么应用需要执行这个动作。如果你认为自己技术够牛，那么在业务功能需求发生变化的时候，你能够保证接口不变的情况下还能适合变化，这才你够牛。别让业务开发组的人跟着你也得改他们自己的代码，那样的设计就很烂了。

小伙听了我的话。进度保证，代码接口简洁。

他说，你真高。我感觉现在我的技术比过去进展飞快。看来人不逼，是不会自己创新更好更快的方法的，老认为自己现有的方法已经不能优化了。我现在发现，很多我过去写的东西还可以做的更好，我准备在开发任务之余优化代码，但肯定保证不影响大家，接口还跟过去一样，我要重构一下。

我对小伙的成长感到欣慰。

但是，小伙还有一个没有逾越的鸿沟。这个问题不解决，我知道，他不会成为一个真正的独立的架构师。

我复查过他的代码，由于他对业务没有深刻理解，所以考虑了 N 多种情况，给自己以后的修改留下了后路。但也因此代码量大，开发周期长无法适应越来越短的客户需求响应时间，可阅读性不强，功能复杂，稳定性困难。但我从客户行业出发，很多情况他其实都是自己假想的，而且想错了。

我指出了他的问题。他问我该怎么学习业务，他又没有机会到客户一线去实施，也不接听客户电话，客户需求都是业务开发组的人跟他说的。

最了解客户业务的，是在一线做客户咨询、做客户实施的人员。其次是做客户定制化、客户服务支持的人员。最不了解客户的，就是架构组的人员。但恰恰要命的是，架构组的人员做的功能是大家的工作基础，如果基础设计错误，那传递的“牛鞭效应”破坏力就很大。所以，架构必须了解业务。

我了解业务的思路，和我了解技术是一个思路，都是来龙去脉法，研究一项事情的过去、现在、未来，以及和这件事情关联的其他事情，研究方法也如法炮制。

你要制造的是卡车还是轿车，你得明确好。你是要造 100 万的轿车，还是 5 万块钱的轿车，也得定好。你是要制造一辆可以自由改装的轿车呢，还是一辆只可以大致改装一些的轿车的，也得定好。这些疑问，都是和咱们面临的客户有关。而我们能面临什么层次的客户，和咱们公司的实力、品牌、组织规模、盈利要求有关。

你如果是一个小公司，想做百万大单只能做的一塌糊涂。你如果是个大公司，你老去竞争那些 5 万块的小单，做一个赔一个。所以一个公司的出身就决定了它的竞争地位和它的目标群。我们要为这个目标群服务，所以我们就不要做一个百变金刚的架构。公司有公司的目标，公司招了你给你付工资，就是为了让目标客户群服务。如何更快更好更有质量的服务，就是公司的目标。我们就是为了帮助公司实现这个目标。

我一般都是把我们这个产业的竞争格局现状了解清楚，我们的过去现在，竞争对手的过去现在都了解清楚。然后我去研究我们的客户行业的竞争格局、层次现状。看看这个客户行业盘子，三教九流到底多大多复杂，我们现在是占了多大，我们还能占领哪些客户群。

然后我就研究客户行业目前的挑战、机遇、困境。能解决其中一两个问题，就是咱们的竞争亮点。如果作为软件一点都解决不了这些业务困境，我就思考如何让产品做的更易用。微软不就靠着易用发家的么？

最后我会去研究我们公司现有的研发优势和弱势、实施服务销售的优势和弱势，找到适合我们突破的地方，具体归研发能承担能起作用的事情，我就会去动员做。脱离现实资源现实矛盾现实包袱的改良，是无法做到改良的。

我还关心各种新的技术应用。可能这项技术很久了，但大家都没有想过还能这样用。所以，我常常在媒体上关注这些、思考这些、在论坛上和业界交流这些。对于新的技术，要研究原理，要尝试，但不要冲动引入到商品生产中。我们不是自己在创业在玩在实现自己的梦想。我们承担的是公司所有人都要吃饭的产品。如果有闪失，这么多人以及他们的家庭都要受到影响。这不是闹着玩。

当我研究完业务领域的这些大的框框以后，每逢有业务同事跟我交流客户需求，我总能把这个需求和我的业务框架联系在一起，把这个需求归好类。并且能判断出这个需求是个反趋势的需求，还是个短期眼光的需求，还是个长远发展的需求。

很多人都在抱怨说需求老变化。其实，不是客户需求在变，而是你对客户的需求老是不用思路去理解。我心中有业务框架，有过去，现在，未来，所以能识别出一个需求是稳定的还是临时拍脑门想出来的。有时候，有人向我提一个需求，我会眼睛一亮，对，这个需求符合未来发展，我就会很快加入。所以，我曾经在做实施经理的时候，老是能很容易说服客户，让客户听从我的意见，就是由于我想的比他们还要远还要周全。好多程序员说客户非要某个功能不做不行，就说明这个程序员并没有理解客户。客户并不是那个非要和你作对的人，他只想解决他的问题。可能你不理解他的真正根源问题而且你又提不出更好的方案，所以他要跟你急，要让你必须实现某个功能。

只有你不断游走在业务过去现状未来与技术过去现状未来，你做的架构才是真正的实用、弹性、易用，而且最小成本，不走弯路，不多花开发精力。

我们需要架构，不就是为了达到这个目的么。

18、焦油坑

我有一个以前的同事。过去他总认为能成事的人什么时候都能成事，不能成事的人你再扶他也成不了事。所以他带领人的方法一般是他以身作则，你如果有悟性，你就照着他做，如果你看不出来，那么你就自己一个人玩着去，能玩成什么样玩成什么样。

我主张的是：普通人通过使用一定的方法和规则，做事情虽然无法做到优秀，但也至少能保持一定水准，不会把事情做烂。如果任由普通人自己去想自己去做，这要管理者何用？作为管理软件开发公司，其管理思想，竞争不过管理咨询公司，其技术实力，又没有技术门槛，属于那种规模化生产实施服务的类型。所以，管理软件开发公司要想成长，必须走规模化路线。而规模化路线就需要依靠大量的普通人才，而非个别的英雄。英雄是很难找到大量人才的，而且优秀的人才其成本也高，更约束的无法规模扩张。这和规模化路线有悖。

所以，他带出来的兵都是单兵作战能力很高，都属于那种能救火队员型，有问题冲上去喊哩喀喳就搞定。领导是很喜欢这样的人才的，因为这样的人是多面手，是特种战士，把一个人随便往哪一扔都能把项目完成的很好，很省成本。但是这样的人才有个特点，没有一年半载，这样的人培养不出来。而且这样的人往往都游兵散勇似的，一遇到必须合作的事情就变扭，老觉得其他人办事都不放心，都觉得别人做的没他预想的那么好，还不如自己一个人包办了快速且省事。

而我带出来的兵都是团队作战型，每次做事，都需要好几个各自发展专业的人配合，一个人还搞不定。就好像开发的时候用开源的框架，本来只想使用 A 框架，没想到 A 框架使用了 B 框架，B 框架使用了 C 框架，最后软件没多大，倒是框架很大。虽然我都是极力推行四套马车 4 人一个小组，而且可以看项目进展安排不断调度组合，但终究不如一个特种战士那样自由。但有一个好处就是做事专业，发挥稳定，培养成长快，好招聘好留人团队稳定，薪资成本也低。

隔了这么多年，我的电话也换了，他的电话我也没记住。我们就这样断了联系。

突然，上个星期五，他给我发了封邮件。说他偶尔看到了我的博客，写的不错（不知道是不是真的不错，反正他以前一直反对我的这种思路），希望我能写些关于需求管理的文章。

我赶快根据他邮件中留给我的联系方式联系到了他。

我问他：你怎么找到我的博客的？

他说：这几年越来越不好做。客户对开发对实施对服务的质量要求越来越高，一个人去现场边修改边实施，客户觉得付那么多钱不合算，怎么着也得派一个团队来实施。但是，团队实施没有人手啊。培养一个人一年都上不了手，对于我们来说团队实施就不合算了。但客户就要团队实施，不团队实施就无法接单。所以，我现在也在找一些如何团队开发实施的文章，无意就找到了你的博客。

我们现在也是文档化管理。从需求调研到需求确认到需求设计到需求开发，每一个环节我们都是和客户文档确认，大家都觉得没有问题了双方看法一致才开发。一开始使用就发现需要修改的东西很多，而不修改又导致客户无法使用下去，最后不断修改，导致实施周期长，结果还跟过去一样，质量也没提高，周期也没有缩短。所以，比较迷茫，是不是哪里做的有问题？你一直挺注重过程管理，看你写了那么多文章，肯定这么多年你一直在研究这方面，所以我估计你有一些好的方法。

我说：我这几天写博客，接到不少网友的评论，他们也强烈希望我能写一篇关于需求管

理的文章。我过去也有一些只言片语的积累，所以这次我就准备写一篇以了大家心愿。

当我开始动笔写这篇博客的时候，我脑海里直接的就是《人月神话》中最著名的一段话（不好意思，其实我没有看过人月神话，不知道作者提供了什么解决方法解决需求难题）：

人类史前史中，没有别的场景比巨兽在焦油坑中垂死挣扎的场面更令人震撼。上帝见证着恐龙、猛犸象、剑齿虎在焦油中挣扎。它们挣扎得越猛烈，焦油纠缠得越紧，没有任何猛兽足够强壮或具有足够的技巧去挣脱束缚，到最后它们都沉到了坑底。

这段话描写的栩栩如生，和我们日常遇到的需求困境如出一辙。

中国人的需求很特别，好像事情都特别赶人都特别忙似的，都寥寥数语就以为他的需求已经说清楚而且你也肯定明白了，到底能不能实现，实现后能不能可操作可执行，都是匆匆一个见面一个电话几句 MSN 几句 MAIL 然后就等着软件开发出来用。

我看到日本人花费一年的时间反复和客户讨论需求，深入到生产一线天天蹲点，还派人拿专业的测量工具来记录，如拿秒表记录操作的时间，拿摄像机拍摄操作流程，有大量的过程检测指标需要 15 分钟确认一次，很是认真，然后才设计解决问题的软件功能。对于每个操作的软件界面也是反复和客户确认，如何更容易理解更方便操作。

我见到许多国内项目经理调研没个方法，东一榔头西一棒槌，需求调研像是开座谈会，一屋子人，有干系的人都请来开会。有最终操作用户，有部门管理者，有大老板，有二老板，有计算机室，好几个部门。每个人站的角度，层次都不同，关注的问题重点也不同，眼光长远也不同，有人悲观有人乐观，有人不懂装懂，有人不懂瞎嚷嚷，有的部门影响力大气粗的不行，有的部门比较势力小唯唯诺诺，有的部门不愿意多干就推来推去反正不是他部门的事情，到底是哪个部门的事情需要大领导来定，但可惜大领导没来，有的部门怕自己那点内部发小财被破坏了就故意找各种各样的困难还说的头头是道，于是一帮人叽叽喳喳，没个结论。有时候开会开多了，实在说不过去了，必须要有一个结论，于是每个人的意见都上了需求本，回来一整理，没法弄。

我曾经参与过一个项目就遇到了这样一个情形，最后拖的时间太长，项目主导方认为没什么利益可赚，而客户冲突利益太多，就放弃这个项目了。

“表面上看起来好像没有任何一个单独的问题会导致困难，每个问题都能获得解决，但是当它们相互纠缠和积累在一起的时候，团队的行动就会变得越来越慢。”对于问题的我们很难看到本质，不过，如果我们想解决问题，就必须试图先去了解问题。

这是《人月神话》中的一段话，我从网上找到的，估计那个项目经理很遗憾没有看到这句话。

风水轮流转呀。真不小心，新一轮领导换届，新领导上台。这个项目又被作为领导新政提了出来。（注意，不是政府换届，而是企业领导换届。我从来没有做过政府的单子）这回，项目主导方变成了我们。我成了项目经理。但是，除了这位新领导，过去的那帮人还是那帮人，我仍然要解决这帮人。（我的一位助手笑称这帮人是成事不足败事有余。上一次我是挂名，他是真正参加了上一次的项目每一次开会。这次，我找他做助手，希望他的上一次经验能给我提供帮助）我这次没有像我的助手上次项目一样，让计算机室帮忙到业务部门要这信息那表格。

我首先向计算机室要了一份全企业的部门组织结构图。我先看看这个盘子到底多大。我曾经刚出道的时候就遇到一次滑铁卢，半路地杀出来一个部门领导竟然严重影响了项目走向。

这次，我把全体部门都纳入思考范围内，了解我这个项目和各个部门的关系。最后我按项目关系紧密程度把客户各个部门排了一张表，每个部门的负责人的名字，联系电话我都要到，每个负责人的年龄，每个负责人的性格，我通过晚上请计算机室没有结婚的小伙吃羊肉串了解了个一清二楚，谁说话算话，谁比较和事佬，谁比较见风倒，谁和谁不对付。想问问

他们大老板是怎么看这个项目的，想达到什么目标，他说他也不清楚。

我先去亲自找最容易配合我的那个部门（我并没有采用聚众开座谈会的形式，而是单个击破法）。但他不是和我这个项目关联最紧密的部门。但他最容易突破。

他和我发了一下午的闷气，有对企业现状的灰心与不满，有对理想做法的想象，希望我能在这次项目中把他的想法全部实现了。

不过这次聊天我也受益匪浅，对这个企业的各种来龙去脉了解了许多，使我在以后的小心翼翼项目推进中避免了很多人为的障碍。

但这不是我这次找他谈的真正目的。他絮叨了一下午，我临走时才说出了我的目的：我需要把他这个部门的报表和单据收集回去。

他很配合，把他的得力干将叫了过来准备安排，我说别了，我跟着他去自己收集，这样我对您的需求也理解的更深更直观一些。

于是，我跟着他的得力干将，一位 36-37 岁的中年女士一一到每个重要的人的桌边，我和每个人都讲明了我的来意，他们将他们手头的报表和单据全都拿给了我。有 EXCEL 的，有 WORD 的，有每月 PPT 述职报告，有纸张的，有电脑软件上的报表都打印出来，有电脑软件上的数据输入，我就帮他们 COPY 屏幕打印放到我的 U 盘上。

我一个人一个人的边收集边问，通过他们给我的表格，我就大致知道了他们的工作岗位内容。

然后，我问：哪些表格是你最常用的。

于是，那个人就帮我挑出了好几张。

然后，我依次把最常用的，次常用的，偶尔用的报表都分了类。

我又提出了问题：哪些报表影响你的考核和奖金工资福利之类？

他又帮我挑出来一些。

我又对着他挑出来的影响他的考核的报表，拿起每一张问他：在这张报表中，你最关注哪几个指标？

他一一指出来。我顺着他的指，边和他交流这些指标的关联关系。

然后我拿着这些指标问他，这些指标是怎么的数据是怎么得来的。

他就帮我从这一堆的电子的、纸张的单据中找了出来，并且解释怎么输入的。

然后我对着每一个单据都问他这个单据的使用频率，是每天、每周、每月、每季还是每半年、每年。是每天（周、月、季、半年、年）的期初做、期末做、还是平时做？哪个频率高？高到什么程度？

这样，我就明白了每个人主要真正做些什么事，怎么做，最后怎么考核，哪些事最重要，哪些事每天做，哪些事频率最高。

常用的功能，重要的功能，性能压力大的功能，稳定性要求高的功能、数据精确要求高功能，易操作性要求高的功能，就是从这些提问回答中自然浮现出来的。

我的那个助手用笔在笔记本上不断记录，手累的最后回来都说手写的都抽筋，告诉我下次不要这么快，让他好记录全一些。还给我看，为了记得快，字都写的有些现在不认识了，还得靠回忆，整理起来特费劲。我说，行，行，我一定注意。

我们回到宾馆后，先制作了一份草稿，粗略的列出来这个部门的组织结构、人员岗位角色说明、业务流程图、考核报表。第二天，然后针对这次项目，把这次项目相关的详细描述出来，并且把核心业务流程和非核心业务流程分开。

第三天上午，我们又去了那个部门，针对每个报表间的钩稽关系，每张单据录入的每一项录入要求、默认值、必填项、唯一约束、录入校验、单据状态、可选值都做了详细调研。

在交谈中，我们又发现了一些流程，这些流程都是些特殊处理流程。我们也发现了一些异常的操作，是发生特殊业务时候的土处理，我们都如实记录了下来。有时候，你专门问他

异常流程的时候，他反而回答不出来。大部分人没有系统性思维，都是以事论事，讲到哪里想到哪里。所以发现异常流程，发现新流程，全靠调研人自己细心发现和甄别。可能，他无意的一句话，你直追着下去就会发现他日常处理的空白和漏洞和矛盾的地方。

第三天下午，我们继续工作，单个人访谈，把每个人工作中认为最想解决的问题都提出来，但只能说 5 个，能想到哪些就说哪些，我们一一记录。

第四天，我们把过去画好的组织结构、人员岗位角色说明、业务流程图，经过昨天的调研，又修改补充了一些，在整理的时候，我们用红圈标好了业务处理漏洞和矛盾的地方，并且对这些地方都提出了改进建议。把他们每个人认为最想解决的问题都考虑进流程和业务单据报表中，建议增加什么流程、建议增加什么单据、建议增加什么报表，谁来做，怎么做，谁来监督，怎么考核。

一份优化好的流程展现出来了。

第五天，我们又去了客户那里。这次，我们组织了部门座谈会。我们给他们整个部门都讲解了我们梳理过的流程现状，给他们说明漏洞和矛盾，给他们说明我们提出的方案。

座谈会非常顺利，全在我们的意料掌控之中。他们非常惊讶我们能在短期内画出这么专业的流程图，其理解透彻度比他们自己还要清楚。而且对他们问题的把握准确，对他们问题的解决思路之巧妙，都不禁赞叹我们。每个人的疑问和建议都融入了我们的流程改进思考之中。客户部门给与了我们很高的评价。

接下来，我并没有把这种方法扩展到其他客户业务部门的调研。而是我把这份调查报告又给了他们大老板演示了一次。他们的大老板从来没见过这样专业的调研，赶快召集各部门头头都来开会，乐的喜笑颜开，大赞这钱花的值，他们只想到上套软件，没想到上软件讲究这么大，他们自己都没有如此明晰专业的流程图。他们的大老板赶快让我给他也 COPY 一份，如获珍宝。

有了大老板的肯定，我做一个部门的调研，就给他们的大老板发一个报告邮件。邮件抄送给调研的业务部门负责人。

我所有的调研一帆风顺，各个部门配合极好。上一次项目的扯皮推搡都不见了。

我的助手说：你这个人有点邪门。

我一笑。

19、一个人的战斗

今天早上，有个网友给我发了一条消息：他是一个老产品版本维护开发人员。他应聘到这家公司的时候，这个产品已经卖了 4 年了。最初的开发者已经都在这 4 年中不断流失走掉了。他来了，任务就是维护这套软件，而且就他这一个人维护这套代码，有 BUG 改 BUG，有需求就改需求。

虽说这套软件卖了 4 年，但真不知道是怎么坚持了 4 年。他接手的时候仍然是 BUG 百出。代码没有文档，没有注释，连表结构说明都没有。代码莫名其妙，经常横插一句代码，显然是客户报告了某个错误，为了临时解决这个错误而做的针对性处理，但到底是为了修补什么错误，代码也没有说明。所以他也不敢乱动，但还要修改需求，只能硬着头皮来。他也不知道自己修改的代码是否还会引起其他的问题，只能凭空企求千万不要出问题。老板还在到处吹产品很成熟，而他每天都在心惊胆战，害怕这套代码不知道哪天突然崩溃，出了错误自己都收拾不了，那只能自己被 K 掉。他能想像的出老板发怒的情景：这么稳定的产品你都搞不定。

他希望我能帮助他出出点子。

我想了想，能有机会开发新产品或新项目的程序员是很幸运的，因为没有历史包袱，白纸画画。而现在大部分的软件公司都是拿一套已经有了型的代码到处修改客户化做新项目。真正做一个新项目从头编写这个项目的第一行代码，这样的机会比较少。

对于修改现有代码适应新客户新项目，这种情况非常多，也是大部分没有文档，修改定制没有注释。客户打电话说了一个需求，能技术达到就答应下来修改，修改完就给客户覆盖，根本没有需求管理、版本管理。而这样的代码，还不是一个特定客户一套特定定制化代码，是要给其他客户也更新的。很可能这个客户好使，那个客户使用其他功能的时候就出了错。按下葫芦起了瓢，是很常见的现象。

我问他：现在你改的代码有注释吗？

他回答：没有。自己修改的自己都记得，即使忘了，看看自己写的代码也能回忆起来。所以也没有写。

我问他：那以后他走了，其他人怎么办？

他回答：反正也是一个烂产品，其他人怎么办，他就管不了了，就应该让这套烂代码尽快死亡，省得祸害别人。

我问他：那你找我帮助的目的是什么？

他回答：在我工作的这段期间内，它不要崩溃就可以了。

我无语了。

关于老系统维护这个话题，我和许多开发人员都有过深入的交流探讨。

许多从事开发的网友认为，一个老系统要维护好，必须具备以下关键因素：有责任心、有文档、设计前做好详细的需求分析、要有需求管理、要 OO 编程、要有专门的测试人员。如果没有这些，干脆推倒重来，如果不让推倒重来，那就赶快跑路，否则就容易当了冤枉替死鬼。

但现实中，往往维护老系统的就一个人。这是很矛盾的事情。一个软件的开发，往往 1-2 月就完成，而它的销售、实施、升级周期却长达 4-8 年。但每个老板好像都认为软件已经开发完毕，修修补补都是小功能，所以一个老系统维护人员就 OK 了。殊不知白纸好画画，而要在别人的画儿再能点睛成龙就难上加难了。

我在管理运营企业的时候，发现遇到的难题也和维护老系统面临的很类似。都是缺这缺那，部门之间利益冲突，人的素质怎么也提不上来，员工和老板互相做猫和老鼠的游戏，不断博弈薪水和付出劳动力的平衡。总有些公司的历史留下的人留下的势力格局留下的客户印象留下的做事方法不能改变，也无法推翻重来，但公司还要发展还要提高，就必须以目标为中心，不断象骆驼一样挺着风沙干渴饥饿领队前进，有各种困难阻碍都要不断清除，无法根除就想办法平衡与缓解，时而让步时而迂回时而强势时而突然决策突然执行，公司就这样不断持续经营下去。

所以，维护老系统，也要象经营企业一样，不断继承包袱，不断细心剖析问题，剥茧抽丝理清思路，不断改进，才能渐渐从恶性循环走向良性循环，才能把一套烂代码扭转成可持续维护的代码。

我给了他写代码的八个建议：

一、重点把控输入数据的校验。你看见很多横插进来的代码，就是由于输入的漏洞进入，最后引起后续数据处理出错，所以以前的程序员他不截源头，他在最后爆发的地方堵漏洞。现在 WINDOWS 程序都是消息事件触发式的，还说不准这个流程会走到哪里，他堵得了这个口，其他根本想不到的触发，他能堵住吗？所以，把输入数据的校验，在保存按钮第一步代码写好集中的详细的校验。而且，这块代码要写成函数，不要大流水，省得代码复杂性会让程序加速崩溃。

二、以后的需求再往上加，都写成函数。遇到比较大的 IF.ELSE 判断，就把其中的代码段再分出一个函数。

三、以后再加功能，尽量不要做成联动触发的。也就是说：保存，最好是单表保存。即使是主从结构的单据，如果客户不强烈反对，也做成先保存主表后再让录入明细表。而且录入明细表要单独的窗口，这样功能和代码都简化了。如查询一张单据，也不要上边是主摘要，下面就是明细联动。这样影响性能。更因为速度可能慢，用户会连续点击多次，触发事件就会乱，莫名其妙的错误就都产生了。最好是双击主摘要，弹出独立的窗口显示明细。

四、你以后写代码，把特殊处理业务和正常处理业务的功能代码分离。就好像你走路，老有人给你下绊子，你就感觉不爽。

五、现有的功能，把不常用的功能做一些隐藏处理，放到一个不起眼的位置。使用的人就会越来越少。到时候就适机真正藏掉，不让它触发了。

六、其实很多时候，你觉得程序很烂，索性破罐子破摔，是由于以前程序员的代码排版可能和你不一样。你喜欢两个空格，人家喜欢三个空格，你就觉得不爽。人家喜欢把{放在最后，你喜欢新开一行。你可以使用代码格式化工具重新排一次版。我看到很多关于老代码维护人员，抱怨变量都是 M、N、S、Button1 之类，但其实你阅读理解代码，这些并不会使你理解有歧义或读不懂，只不过你不爽而已。理解了这个不爽，你就会心平气和一些，修改代码会更加顺利一些，你越和旧有代码生气，你的工作越乱。（看到这里，相信很多程序员都会会心一笑。真正的根源在于此，老系统无法维护只是借口而已，可能希望老板认为自己的工作很辛苦很复杂而加薪）。真正危害大的是全局变量和大流水代码。所以写代码，要严格避免这两个坏因素。

七、修改需求或 BUG 的时候，要按照模块来集中修改，而不要挑好改的先改了，不好改的就最后改。按照模块来集中修改，你会通盘考虑所有这些需求和 BUG，而不是糊窗戶式的补窟窿。

八、我曾经和很多做维护的开发人员都做过交流。他们都觉得一个软件没有文档，没有注释，简直就没法维护。但确实是很多软件没有任何设计文档，连帮助说明都没有，代码也没有注释。而这些软件又出自他们自己之手。也就是说他们一边抱怨没有文档没有注释，一边自己也不做文档不写代码注释，不知道在等谁来专门做。我问他们到底需要什么文档才可以将一个软件维护的越来越好，从一套烂代码扭转是一套良好渐进的代码？他们说要表结构说明、要详细功能设计书。表结构还好说，可以整理出来，详细设计说明书就不容易出了。

我曾经也维护过别人的代码，也是什么文档都没有，连操作使用帮助都没有，更别提详细设计说明书和表结构，代码当然没有什么注释。我并没有去整理表结构说明。幸亏这个人喜欢数据库上倒弄，写了大量的视图和存储过程。视图中有各个表之间的关系连接，也有各个表中重要字段的中文名。这样我就不需要表结构说明了。因为表结构说明不仅需要需要描述每个表中字段的中文含义，也得描述表之间的关系，这和视图能表达的效果是一样的。所以，我现在也建议开发人员写代码，多写视图，多写存储过程。有的老代码，SQL 语句都生写在代码中执行，没有视图。对于这样的老系统维护，就是把这些 SQL COPY 出来，做成视图，这样就好维护了。

对于详细功能设计书，其实对于程序员来说，其目的是想弄清楚业务流程的来龙去脉细节。光直接看代码是很难弄明白意思的，又没有什么其他文档可以参考，所以只能猜测代码的意思。尤其很多维护人员，很多功能细节都是为了处理某些特殊需求和异常业务的，都是以前的程序员写的，但是以前的程序员已经走了，现在的维护人员连软件中具体的这些细节功能都不知道。当新的实施人员或支持人员反馈回疑问，想问问程序员某个细节功能是怎么回事，程序员都发蒙，嗯，还有这个功能？我也不知道呀。

要解决这个问题，我曾经做过的事情就是组织实施人员写功能操作说明帮助。因为实施

人员要给客户去培训讲解，没有帮助说明，只能一张嘴叭叭叭的干说，实施人员是最需要功能操作说明帮助的。但是实施人员认为这个帮助是软件的一部份，而且是开发部开发的软件，开发部最了解功能，所以帮助文档应该开发部写。而开发部认为开发部的职责就是编写代码，你自己培训你连个操作说明都没有，你怎么培训，所以帮助文档应该实施部门自己编写。于是帮助文档谁也没有人写。

归根到底，帮助说明是终究要写的，主要是谁写的问题。谁最有动力写呢？实施人员最有动力，因为这和他们的工作息息相关。而程序员明显没有动力理由。而且实施人员熟悉第一线客户的素质，理解客户的具体操作思路和理解思路，写出来的帮助客户都能理解，帮助文件才能真正为客户服务。很多帮助文档的写作都是从来没有见过客户没有实施培训过没有客户支持服务过，连软件测试都没有做过的纯粹文档人员编写的，可想而知帮助文档到底能对客户有多大的帮助性。

在写帮助说明的时候，我要求实施人员把每个按钮都要点到，每个 Grid 中的每一个字段的数据来源和数据含义都要说明到，每一张报表中的字段的数据来源和数据含义，每一个明细录入中的字段的数据来源、数据录入要求和数据含义。这一写不要紧，发现了很多隐藏的特殊处理功能。很多功能很多人不了解，因为很多细节功能，都是为某个客户定制的，只有负责实施该家客户的实施人员才知道。于是，实施人员之间互相通气，才算补足了不少功能细节的帮助说明。实在有些功能，都不知道是哪家客户提出来需求，也不知道为什么要这样处理，就留下空白，转给开发人员，让开发人员看看代码是怎么处理的。就这样，一份详细的帮助说明在压力艰难中终于出来了。从此，开发人员理解需求快了许多，当然也就明白了那些过去自认为乱七八糟的代码的含义，心情好了很多，修改代码也轻松了许多。原来，一切都是自己跟自己作怪。不盼望软件工程，不抱怨一穷二白，不幻想增加人手，从现实入手解决自己的问题，发现很多解决方法既简单又有效，根本无须动辄就是团队就是 UML 就是 OO。

另外，我还给了他一些关于需求控制的建议。

需求，是很多方面的。有关于功能的（尤其是每家客户特殊的业务需求），有关于异常错误的、有关于性能的、有关于兼容性的、有关于易用性的、有关于特殊权限的、有关于美观性的。

而需求的来源也是很多方面的，有的是客户计算机室直接打电话，有的是客户业务部门直接打电话，有的是实施人员，有的是支持人员，有的是市场人员，有的是销售人员，有的是老板和客户打单或开会突然想到谈到就直接给开发人员打电话。

而需求的优先级也不一样。有的客户态度强硬，你必须尽快满足他，否则他就给你老板打电话。

而正是这来自四面八方的各种层次各种看法的人的各个方面的需求电话，把程序员就烦的要命，还要去开发。而且很多都是一个电话就认为程序员能开发了。但往往程序员开发完后，客户一看不是自己最想要的，于是再修改。

所以，需求多，其实是一个幻觉。

第一、把需求分类，做个 EXCEL 表格，量化解决。这个需求管理表格会有下列这些项：客户名称、需求提出人、提出日期、需求关闭时间、功能模块名、客户现在版本号、需求描述、需求分类（需求、BUG）。我在最初没有需求管理系统的时候就使用过这种方法。过去没有使用的时候，我的手下老叫忙死了烦死了。我就让他把现在手头的事情都整理一下给我报个邮件。但一整理，肯定不超过 10 件事。有些事情是等待客户给资料，有些事情是调试跟踪不出来错误，有些事情是需求模棱两可。我给他一分析，他现在正在进行的事情就两件，而且都是他自己能独立做的，根本不需要别人配合参与的。他忙吗？他瞎忙，或者故意说忙。没有工作效果，就是这样。帐不算不清，话不说不明，就这个道理。

有了这个表格，要定期（可能是一周，可能是一月）给老板一份。这表明你的工作量，让他看看你确实一直很辛苦的在工作，而且干了这么多活。而且，这也能看出你工作的认真负责态度。

有些程序员不做这个表格，也不给老板报告。很多时候是程序员并没有干那么多活，能推则推，能混则混，能拖就拖。怕自己有一天混不下去，所以心理压力很大，每天不干活却总觉得很累。这种累就是自找的。想必一些程序员看到此会想起自己。

第二、需求描述不清晰是反复修改的罪魁祸首。对于 BUG，要有错误报错整个的屏幕截图，千万不要就截那个错误消息框那么一小块。对于需求，是报表需求，要给出表格格式，还有每一项数据的来源，有公式关系的要给出明确的计算公式。对于输入单据需求，要给出单据格式，每一个输入项的要求：可选值、默认值、不可为空、唯一性、约束输入，数字要有小数点后精确度、日期要分辨精度是到日期还是时还是到分。

对于这位网友的现状，我还建议他开始版本管理。

、VSS 之类就不必了。因为这是一个人的战斗。连版本都没有概念，一上来就是特正规的工具，大半感觉太麻烦，又退回到最初原始状态，还对版本管理产生了不好的印象，觉得繁琐还没什么用。所以我们有一句话：一管就死，一放就乱。其原因就是缺乏一个中间过渡解决方法。

所以，我建议先把版本意识提上来，按照版本管理的方法走，走顺了就自然接受了正规的版本管理工具。版本管理工具可以分支，也可以合并，可以针对 Bug 进行补丁发布，而不发布还未完成的新功能，可以发布为某个客户专门定制的版本，也可以回溯历史版本，对比历史差异，源代码安全性也高。

有几个过渡性建议特别实用：

一、有大的修改或没有把握的修改之前，先把代码备份到其他的机器上。备份目录要跟上一日期。

二、在大的修改前，先定一个稳定的版本发布出去。很多程序员没有版本这一概念，每天都在持续修改。结果，给客户的，每个都是半成品，有半拉子没有修改完，还自己没有做屏蔽处理。客户不小心用了，产生了错误，再告知千万不能用这个功能，还没有完善。但晚了，错误数据进入了，以后报表平帐就是问题了，又得特殊数据特殊处理了。自己的孽障自己还得解决。

三、即使是琐碎的修改，也要每天或隔天备份一份源代码，别怕代码多，现在的硬盘大的很，而且备份复制一下也就是 5 分钟的事情。别怕每天备份太烦。我们经常会遇到这个客户让改了，另外客户不让改。一个功能改了又改回去，但过去的源代码没备份，忘了怎么写了，这时候你就想起代码备份的好处了。尤其现在有不少免费的文件同步或文件自动备份的软件，都能定时做。功能还强大，有些还有些差异备份的功能四、现在有不少文本对比软件，如 WinMerge 之类。可以对比两个文件的差异，这个功能和版本管理工具中的源代码差异对比一样的效果。

五、如果每次发布新版本，就把从上一版本发布之日之后的关闭的需求列表都单独摘成一个文件，附带到这次新发布的版本之后。这样即使没有人写更新说明文档，根据追溯也能明白这次版本解决了哪些问题和需求。很多程序员没有需求管理表格，版本发布要求写更新说明文档，这才从脑海记忆中想，想的就有些遗漏，甚至错误。好多程序员有过这些的情景：我记得改了呀。真正一翻代码，一点没动。大叫：我的代码怎么没了，我记得我改了呀。

我这些建议，从需求描述、工作量管理、遗留系统代码重构技巧、备份管理、版本管理、更新说明文档一整套说明了一个人如何维护老系统的工作方法，但希望能分享给大家，给大家以帮助。

有方法，你就不是一个人在战斗。

一切皆有可能。
相信自己。

20、累斗累

有时候，我感觉事情就好像大螃蟹，总是一串一串的。

我刚聊过新项目如何收集需求，就有人跟我提老产品升级需求的管理。

有人说：老师，我看了许多 IT 项目管理的书籍，也讲到需求管理。但他们是需求调研、需求分析、需求确认，好像都是针对新项目的，我们是老产品维护，老板随便打一个电话就让我们添加一个需求功能，我们哪里去做需求调研、需求分析、需求确认这些环节啊。老板说我们一天坐到家里面编程序，根本不了解客户需求。最了解客户的是每天和客户待在一起的实施人员，所以要让实施人员来给我们软件提需求加功能。但是，实施人员那叫什么需求啊，比如说 XXX 功能不好用，比如说建议更易用一些。老板不相信我们，怕我们把实施人员反映的需求给屏蔽了，专门派了一个人每天收集各个渠道来的需求，每天还要上报给老板，而且还每天向老板要报告，今天修改了多少需求，还有多少需求没有改，没有改的需求什么时候能改完，都要我们开发部给答复。而且，隔几个月，老板就把全公司人员都召集在一起，为软件提需求。一群人坐在一个会议室，每个人都可以提，一个模块一个模块的提。当场打开软件功能，演示，说明，阐述软件应该做成什么样。我们开发人员就跟开批斗会一样，这个标题不要这样叫，那个对齐不对，这里不够明显应该加粗字体或者给一段红色的提示。唉，活得真窝囊，天天修改这些乱七八糟的所谓需求，还要被人追着赶着问进度，还要忍受被所有人都骂开发的什么烂软件。老板也觉得我们水平不行，需求也怠工不修改，两天才修改了 3 个需求。而且越修改需求越多，软件越不稳定。我们哪还敢提涨工资啊。

我说：你这种情况很普遍。现在很多软件公司都是老板开店，三五个人十来条枪，他有客户关系，卖个软件做个实施赚点钱。公司也不大，如果软件做不好，实施做不好，多年积累的客户关系就以后不好再销售了，这就等于把老板的命根断了。所以老板肯定会盯死软件开发和软件实施。客户使用效果好才能以后有更多的单子做。而且你们作为开发人员，又不接触客户，怎么能理解某个需求的真正意图呢？当然老板的思维逻辑对，你们这样，怎么能理解客户需求呢？当然实施人员最了解，提的需求最有权威。换你做老板都这么想。

我也经历过这段日子。人和人的信任，都是在做事中不断看人试人品人，才能取得信任，才能放权。我的老板也如此。

如何比实施人员更了解客户需求？如何让老板相信我比实施人员更了解客户需求？这也是摆在我面前的一个问题。

我是把产品放在一个产业中看。看竞争对手，看业界标杆，看业内管理专家，所以我对产品的升级完善，是基于产品战略的，是基于盈利模式和竞争力构建的。如何引出更多的盈利增值产品，如何保证产品更有竞争力，是我思考的重点。

而实施人员提的需求，都是和他实施的客户具体业务流程有关。客户就是这样，咱们软件不满足客户这样做，就要改。不改，客户就不满意，实施就推进不下去，实施项目就延期，自己的实施就不力。所以，实施人员为了保护自己的利益，也要开发部必须改。而且一开口就是你从来没有去过现场，你根本不了解客户。

老板信谁？

我不是业界知名管理专家，我也不是业界明星 CTO。老板怎么能信我对业界产品竞争的判断呢？

而实施人员，天天真实的和客户待在一起，他们反映的肯定是真实的。

第一轮回合，以开发部门老老实实修改需求为结束。

后来，老板又亲自主导组织了几次实施人员需求会议，什么都可以提，都记下来，让开发人员改，让实施人员监督开发人员修改，确认修改的符合实施人员的要求，并且实施人员负责测试。每次大约都能记个上百条，从要求简化 SQLSERVER 数据库安装（因为实施顾问都非技术出身，没接触过 SQLSERVER 之类的产品，用的最多的是 EXCEL 和 WORD，因此用报表设计器给报表格式挪挪位置大部分人都不会）到把界面都换成绿颜色背景（说符合公司形象）。这种局面直到我在各个部门各个环节各个层次应用了多种策略才改变。

过了一段时间，老板看着实施人员都没有事情可干，很奇怪，就问为什么不测试软件？实施人员回答说：开发人员没有改多少需求，我们正在等待他们开发呢。

然后老板就找我，为什么不修改？

我说：程序员一直在修改，但另一方面，我们已经修改了多次，满足需求 300 多条，但是我们的产品竞争力增强了么？我们的特点在哪里？我们的亮点在哪里？我们的竞争力在哪里？

老板说：不管怎么说，这是客户的需求，客户买了我们的产品，我们就有义务为他们满足需求。

我说：看这条需求，客户要求我们做一套视频会议在我们的管理软件中。

老板说：那你找找网上有没有免费开源的。

我不再争辩。因为我知道，在他心中，做什么产品，叫什么企业管理软件都无所谓，客户需要在 ERP 中加入游戏，如果理由充分也是可以的。重要的是客户不能得罪。客户关系是他的生存之本，而非产品。

我开始在网上写对行业的观点。因此也有幸结交了许多知名的行业内管理专家。他们称赞我的观点独到，思考有远见。文章也受到不少网友的追捧。

我会把一些我写的文章链接适时转给老板，转给喜欢思考提升的实施顾问。我也会经常把和我思想观点相似的文章链接转发给老板。

老板继续组织全体人员需求讨论大会，但这次他有了转机。虽然仍然是让人统计未完成需求数，老追问什么时候才能完成，但显然他不像过去那样主导与控制整个过程了。他说：你也要跑跑客户，不能老待在家里。

于是，我跟着实施顾问也跑了一些客户，有时候，还带上主力开发一起走访。

过去，开发和实施冲突很大。开发觉得没必要改，实施觉得必须改，最后实在不行就拿出老板来压。而且，客户有自己特殊的业务地方，有能人者还自己想解决方法。而实施人员呢，在现场实施，陷于此境此客户，也觉得客户的解决方法有道理。于是非要开发人员按照那样的方法修改。但开发人员知道，按照那样的方法修改软件，那软件就死住了，这个功能只能给这家客户用了，其他客户没法用。于是开发人员骂实施人员胳膊往外拐，站在客户那方对付自己公司的同事。

但是一走访，开发人员也才认识到客户原来面临这样的问题，客户那样提需求其实是为了解决这样的问题。但很可惜，客户的想法太局限，只为自己这个问题想解决方法。如果开发人员当时在现场，就能明白客户面临的问题根源，就能设计出更好的软件功能来满足。唉，都怪实施人员不分析问题根源，只看表面问题现象，还自以为是提最佳解决方法，还让开发部实现。应该是，你提出你的需求问题，怎么解决是我们开发部自己的事，我们会综合考虑平衡。

我走访客户的目的，主要目的有三：

改善一下老板和实施人员对开发部的认知。他们都认为开发部只是一群不懂客户需求整天做在电脑跟前不用出差说话神叨胡子拉碴的敲代码者。“你们不懂需求”，让这个理由去一

边去。

改善一下开发人员和实施人员的冲突。让开发人员也理解理解客户的现状。有些事情是不可改变的，有些事情是人为故意的，我们作为开发，不应该去把软件假设在一个理想的工作环境下，那样的软件是不适合现实使用的。该委曲求全还得做。骂客户管理水平太次，骂客户都是白痴都毫无意义。我对开发内部老说：人家是白痴吗？难道人家买你的软件也是白痴。那岂不是反过来说你的软件谁买谁就是白痴？那岂不是说明你的软件很烂毫无价值么？咱们每个月的工资从哪里来？是老板发给咱们的吗？老板又不是白痴，发钱给咱们？老板的钱也是从客户口袋里掏来的。

了解客户现状，想办法如何去引导与影响客户。客户面临最急需的问题是什么？客户对软件的认知程度如何？客户把软件价值认为成什么样？客户认为这套软件应该是干什么的？（有的客户认为我们的软件应该带上 QQ）。实施人员是怎么给客户讲解软件中的管理思想的？实施人员是如何培训客户的？

走访回来之后，我做了两件事情：

发表了一篇我的走访感受。发到网上，也发给老板，也发给实施人员。老板看了以后说了一句话：“看来走访走访很有必要，以后要多做”。老板、实施、开发，之间的关系改善了许多。

把走访过程中确实应该改进的需求安排给开发人员。由于开发人员也是切身体会，很快就改了出来。实施人员说这个功能做的不错，很实用。

第二回合，开发、实施，平手。

然后，我又把业界知名专家的一篇文章中提到的评价模型内嵌到软件中。过去，我们虽然在产品 PPT 中老是宣称自己的管理思想，但在软件中很难看到落实。所以，实施人员在实施过程中只能讲操作。而操作，客户觉得还不如一个 EXCEL 好输入好修改好查询好统计，所以客户希望软件修改成这样修改成那样。反正，客户现有系统解决不了的问题就都提出来。而现在，终于有了一个可以讲可以量化的管理模型。这套评价模型成了这个软件的核心亮点。客户为了应用这套评价模型，也乐意录入数据，维护数据。

第三回合，开发胜出。

然后我提出了需求管理系统，WEB 型。不管谁在外面实施，或者是老板突然提出，或者是销售提出，都录入到需求管理系统中。把需求分类，分优先级，量化安排开发计划，有的放矢吸收需求改进软件。

我还提出，每年召开用户需求会议。邀请有思路有积极改进想法的客户参会。大家面对面交流共同问题，共同需求，共同探讨未来行业机遇挑战，共同寻找解决方法。

引领客户，引领老板和实施人员对产品，对业界，对未来，对竞争的认知提高。

在我的影响下，公司的 IT 咨询业务、IT 教育业务、IT 服务业务、IT 整合业务、IT 产品发展战略、集成合作伙伴，都逐渐专业化的发展了起来，给公司提供了越来越的盈利模式与收入渠道。

、设计模式、接口？！和客户一点无关？那客户为什么要给我们钱？那老板为什么要给开发部钱呢？那开发部存在的意义在哪里？翻译成代码？就这点意义？还要月薪上万？凭什么？

你的价值在哪里？

21、我要飞的更高

又到了半年，公司这几天要开半年会。

老板让我做一下总结报告，对上半年的研发成果、下半年的研发计划、明年要做什么新产品的规划，希望我都能谈到。

对上半年做了哪些工作，这些都有工作记录，也有项目管理系统，也有 Bug 管理系统，也有版本升级发布，所以很容易总结出来。

下半年做什么，有需求管理系统，客户的需求都排着。共性的、希望做的更专业更深入的模块也在详细设计之中。对于扩展支持更多的客户组织模式、远程管理模式、更低层次和更高层次的客户，分离更清晰的高级版、标准版、简化版也已经有规划。并且根据定位不同，已经规划了不同的产品形象策略、宣传策略、销售策略、定价策略、实施策略、服务策略。

对现有产品线进行深化、扩展，这都是没有问题的。但明年要做什么新产品，是最难的。

因为新产品的研发，有很多影响因素。

我们在行业信息化深耕了多年，已经有一定的知名度。因此，我们有时候想接到某一个项目，但该项目的客户负责人（都是老关系熟人了）就笑着跟我们说：你们又不是做硬件的，啥钱都想赚啊。

但是，我们确实想赚取更多的钱，扩展更多的可销售的产品和服务项目，而不仅仅就把我们定位在软件开发和 IT 服务、IT 咨询。因为行业内的客户数是一定的、大中小的客户数量是很明确的，行业内的竞争对手都按照自己的实力大小，已经占领了属于自己开拓的那层客户，并且在产品层次、销售模式、实施模式、服务模式都做了匹配。我们当然想扩展我们的客户层次，逐步上中下客户都通吃，并且在产品线上，各种 IT 产品和 IT 服务都涉及。只有这样，才能保证我们每年都能业绩提升。否则今年是这个销售额，明年还是这样的销售额，没个发展老原地踏步，这样干公司就没劲了。

所以，这个已经定了型的公司形象就容易阻碍公司新业务开展。公司没有鲜明形象吧，客户不清楚你到底是个干嘛的公司，到底能干嘛，到底什么做的专业。真正通过多年做产品做客户，让客户有了定位，反而客户认为你就是干这个擅长，其他你就不行。就如同百度要进军电子商务，业界很多人都说百度不可能成功，因为他是做搜索引擎的，它又不了解电子商务。

我不同意这种观点。很多人的思维是守阵地型，觉得自己有多大能量就考虑多大的事。但是做企业，守阵地是守不住的，你不攻打别人，别人就要主动来吃掉你的份额。我一贯的思维都是，先放大眼光，放眼整个 IT 行业，整个客户行业，甚至是和客户行业相关联的上下游行业。能不能做到还没论证呢就否认说不能，这种人无法引领公司产品成长。先研究，说不定还能做到。一旦成功，就可以获利颇丰。当然，我也会考虑风险，不会让公司要么赔死要么赚死，我是职业经理人，我要保持平衡发展。老板可以大刀阔斧进行革命，但我只能曲线救国。

公司现有产品体系，公司现在的客户群，都是需要照顾的地方。新的发展，不能与过去完全隔离，否则就容易把过去积累的优势都浪费掉。当然，有句话说：最大的优点就是最大的缺点。优势就成了前进所无法抛弃的包袱。

公司现在的人的技术素质是否能达到，如果想做的事情确实挺好，公司愿不愿去提供更高的薪水来招聘人。是自己独立研究开发，还是通过合作先代理别人的产品进行销售维护在别人的基础上进行扩展，还是请外脑顾问进行指导。合作者是否理解深刻意图，能否在利益上达成一致，未来分道扬镳的时候是否能好聚好散，都有很多沟坎，都需要拿捏到位。

公司现在的资金如何，公司现在的业务销售是否具有可持续性。这也是需要考虑的事情。一项新产品的研发，是需要花费很多时间来跟踪收集研究，做尝试，做推广，逐渐在客户群众树立影响力逐渐扎根逐渐认可，是相当长的时间。在新产品还无法代替现有产品的盈利地位之前，必须能保证现有产品销售稳定性。否则大家都要喝西北风了。

竞争对手现状如何？它的产品、它的研发、它的销售。不能造成自己在这里抽调资源研发，竞争对手在那里攻击，还没出来，就先死掉了。我常常想到一个玩《红色警戒》游戏的场景：你的电力系统被敌人偷袭，你赶快造一个新电厂，还没有成型，就被敌人摧毁。

在研发当中，更要验证对未来综合竞争力的提升。不能是单独的产品，而无法成为一条产品线或产品体系，就不容易利用现有的销售、客户、实施网络进行落地。必须与现有产品形成互补。就好像我们下跳棋：既要给自己留下路，而且同时这颗棋子还能成为对手的绊脚石。如果我们既不是行业中的第一或第二，还要防止行业领头羊打击新兴模式，在你还没有成气候的时候就灭掉你这个新模式，让你失去新产品发展的土壤和环境。

这都是很宏观的一些环境考虑。决定我们的方向。

很多网友可能觉得不需要考虑这些，屁股决定脑袋，这是老板的事情。很多所谓的技术总监其实在老板眼中就是个带项目的头。也有的技术总监，老板认为他是技术总监，但他自己却把自己定位成一个项目经理。

但我们处于行业管理软件这个领域。这个领域，客户数量是一定的，客户行业发展也就那么快。你必须追随客户行业的发展。如果客户行业好多年都没有变革，机遇与挑战变动不是太大，整个行业环境趋于稳定，那么信息化也就是夯实。你想引领先进模式，客户还不走，你也就走不动。

行业管理软件领域，不如现在火热的互联网行业、网游行业。QQ、百度、盛大、征途、阿里巴巴、51.com 争奇斗艳，风投不断，成长迅速，日夜吸金，快速上市。几乎 5 年，就能把一个创业公司膨胀到一个上市公司。而行业管理软件领域，往往随着客户行业发展而发展，销售额一直不高不低，撑不死饿不死。所以我们也一直在寻找和互联网、虚拟产品、网上营销相结合。我们也在跟踪现在火热的技术，如 SAAS、Open API、SOA、BPEL、DSL。我们也在跟踪行业内先进的管理专家发表的观点和思想。只有盈利模式+先进技术+先进管理，才可能成就新一代的产品。所以，我一直在这三个方面进行跟踪研究交流。

我们更要考虑的是我们传统模式和流行模式结合在一起的新产品新模式，到底面对什么样的客户，市场容量多大，可能会有什么样的竞争对手进入，我们会有什么危险，我们如何应对，会吃掉多少市场容量，未来我们最大可能占多大容量，这样的容量值不值得我们做。我们做需要多少资金投入，需要多少钱来推广，需要多少时间达到我们的盈利点，在达到我们盈利点之前我们如何防御性发展。

摸清了我们自己的优势、困境、竞争地位、未来目标、未来业界位置，摸清了行业业界现状、未来影响，摸清了竞争对手，摸清了现在流行的盈利模式，摸清了未来更加敏捷的技术，我们才能给我们的产品定一个位置。新产品必须与公司的实力和战略相结合。老板的想法你不了解，你自己一个人瞎琢磨新产品的研发，很可能想法很好也未来很可能成功，但不符合老板的观点，没走出第一步就壮士未捷身先死了。

知道了做什么，能不能做，做完有多大好处，这个饼画完，并且画的能把老板和销售总监说动心了，就要落实怎么做了。

我仍然采取的是标杆法。

我承认自己是个保守的创新派，所以只能做职业经理人。开创性的，在我的位置，我的角色，也决定我不能做开创新的事情。这是个鸡生蛋蛋生鸡的问题。是由于做职业经理人而不能有开创的自由度和支持度，还是由于自己做了多年职业经理人而无法有开创性的胆略和思考眼光？

我想起张树新在瀛海威艰难的时候的一段话：天上下着雪，雪很厚，前面什么也看不见，很黑，我们开着车摸索前行，不知道路走的对不对。

我想这种创新的模式和产品我是不可能做的。所以我要做的其实就是金庸+古龙，形成温瑞安这样的模式。这就是创新。

我先剖析行业标杆老大的产品。所以，我常常跟踪分析 SAP、用友、金蝶这些标杆企业的产品，学习他们的架构、产品气质、实施模式、咨询模式、服务模式、销售模式、市场模式。

虽然他们都是做通用企业管理软件的，但是每个细分的行业管理软件领域发展有快慢不同。所以借鉴他们，就能在自己这个所定位的行业领域取得先机。有句话：我不期望比刘翔跑的快，但我只要比你跑的快就 OK。

只比客户前进半步。这是做产品的很重要的指导原则。你可以想的远，很好的前景和盈利模式使你心情荡漾，但真正做起事一定要只比客户前进半步。你可以想的功能策划的功能很多，但一开始推出第一版的时候，一定要突出特色功能，而不要把你的想法全盘托出。一个新出现的东西，本来大家就抱着试探怀疑检测的态度，一打开产品密密麻麻的功能，客户就会感觉深陷森林之中，到处走都走不出去，就很压抑了。而且这样复杂的深入的开发，周期长成本高控制管理难度高，很容易没有竞争对手攻击，自己就内乱先倒了。而且客户还不买账。

经过把行业标杆的功能进行遍历列清后，细化到它们能支持的国际化、网络化、多组织模式化、权限控制程度、功能菜单、控制参数、基础字典、输入窗口、输出报表、统计、组合查询、特殊业务处理模式等等，然后对比优缺点。整合出一份吸收优点的特性列表。

我们还会遍历我们的需求管理系统，把客户的反馈加入进来。研究竞争对手产品，把优秀功能加入。这样就结合了既站的高，也能落地现状。

然后在这份优点之上，再加入我们看好的新的赢利模式、新的行业管理模型、新的技术模式，形成我们自己的未来产品框架。这个产品框架不是散点，这个软件中拉一个亮点，那一个软件中拉一个亮点，而是要具有新的高度、新的视角，对行业企业组织、业务流程进行审视、形成有组织、有流程、有层次的管理信息化体系。

只有自己能讲通了，并且自己能说服自己，才能给研发内部扩散你的产品规划体系，从行业客户发展过去、现在、未来变化，现在竞争对手、咱们的竞争地位，讲到现在的新兴赢利模式、新兴敏捷技术、咱们的战略发展思路，说的头头是道，才能让你自己的研发手下相信你的判断正确，思路深远，才能形成目标一致。

获得公司内部最多人支持是一件非常重要的事情。尤其一件新的产品。到底行不行，大家首先就有疑问。抱着怀疑的态度看事情，没有问题也有了问题。

需要认同的人上下左右有四类：上有老板，下有手下，左有客户，右有营销部门和实施部门和支持服务部门。缺了谁的配合都干不成。

首先要获得自己下属的认同，这是最容易认同的人。这是你的群众基础，在你势单力薄的时候，他们就是你的拥护者和同声者。

然后是获得老板的赞同。没有老板的赞同，你和你的手下想做都不能做。老板不会给你任何时间和资源做的。想让老板赞同，你必须所有的思考都要按照老板的思考方法去给他讲，不能你按你自己的思路讲，那样行不通。另外，你的想法一定是可以帮助老板赚取更多的钱为唯一目标，而不是其他和老板切身利益无关的目标。那样的事情，老板不感兴趣。很多人说我遇到慧眼识人的老板，其实没有，我只是能帮助他赚取更多的钱而已。这样的人，谁不想要啊。（有人想拿着老板的工资，敷衍老板的做事，干自己的事或者积累自己的客户、技术、朋友等等，这样的人，老板自然不喜欢）然后是获得客户的支持。我走访客户时候，和客户老板经常会谈到现在的行业发展、行业挑战、新兴行业机会、互联网新型模式。和客

户中层经理谈的就是管理监督、流程、评价模型。

客户是给老板钱的人，老板是给我们钱的人，而营销部门在乎的是好不好卖，能不能大卖，能不能赚更多的提成，而且卖完后没有后遗症（小心自己多年维护的客户关系被毁了）。而实施部门在乎的是东西好不好让客户接受，能不能很快培训完上线走人。而支持服务部门只在乎以后别出问题就行，工作越少越好，但也不能没有 BUG，否则他们就没有多大用处了。

得到者多助，失道者寡助。上下左右逢源，你才能成功。

否则，你就是个项目经理---干活人的头儿。

22、波、波、波

这几天，去了一趟罗布泊。

为什么去罗布泊？

因为罗布泊没有办公室、没有客户、没有员工、没有邮件、没有电话。大沙漠中一望无际还是沙漠，地表温度接近 60 度，人很渺小，人的生命也很脆弱，唯一能做的就是找个凉快地方躺下保存体力、保存好水和食物。就这样，在路上，可以有很空的心去思考。（让我想到《商道》中的戒盈杯）帮助一个公司做大。过去是求生存，有什么项目就逮什么项目，不管能不能做了，人员素质能不能达到，有单就收，生存比任何诚信、质量、企业文化都重要。

终于几年来业务模式、收入、团队都稳定了，但未来如何走就摆在了面前。33%的税、办公写字楼租金、员工工资三险一金、销售费用、出差费用、办公费用，不管你有没有单子，这些就要支付。即使今天下班公司关门，钱还在哗哗的流，尤其员工多了，每月的流水费用就在上百万。而每月能够拿回超过 100 万的销售单子给员工发工资，并且还能处理每个月的租金、销售费用、出差费用、水电互联网服务器复印机打印机传真机等等等等，都是一个未知数。而下个月的单子是否能成功，更是未知数。而下个月的费用却要照常发生。

于是，我把这几年所做过的所有项目都整理了一番，目前的产品也整理了一番，现在流行的可被风投的业务模式也梳理了一番，希望得到能继承过去还能结合未来发展的良好盈利模式。

我手头也有需求管理系统和 BUG 管理工具，关于产品的增强，有 N 多的方面。但增强了会如何？我要的是能大规模销售（对等员工的薪水和素质，理解能力和销售能力有限，不能销售带有高深管理理念的产品，员工无法理解软件中含盖的管理理念，当然也无法传递给客户）；我也需要能大规模实施培训，大规模服务支持（如果需要一个员工花费半年的学习才能胜任岗位，这样的时间成本和人力成本就非常高了，不利于大规模扩张，因为可用人的数量跟不上规模）。而且产品能具有普遍适用性，涵盖尽可能多的客户群。

我们都知道，做新软件研发，是很消耗资金的。所以，我们都尽量使每一个产品都能生命周期尽可能的长。

在过去的几年中，我把软件分离为简化版、标准版、高级版，分别对应那些 IT 信息化水平和信息化理念层次不同的客户。

对于低级用户，就塑造产品成为一个电子化功能工具，你可以用纸张手工代替，你也可以用 EXCEL 代替，但这个电子化的管理软件（可以称它为 MIS）可能比纸张手工和 EXCEL 更好一些，所以价格也便宜。在销售队伍的配备、产品宣传、产品包装、定价、实施方法、实施团队配备、服务方法、服务团队配备上都有很大差异。对于中端客户，就会嵌入可量化

的管理模型。管理理念是一个很缥缈的东西，你说你这个管理软件管理思想先进，你凭什么，你拿出来真凭实据看。所以，给从未做过管理也不懂管理的普通员工讲通管理方法，并且让他们传递给客户，这种思路是不靠谱的。所以需要可量化的管理模型。输入数据，经过模型计划，就得到有价值的输出。这个模型是创新的，有价值的。很多管理软件遭到最终用户的使用，就是由于最终使用者并未感觉到软件给他自己本人带来了什么明显效果，反而只是一个数据录入者，供管理者分析决策用，而他又不清楚管理者有什么用，就觉得天天很累的输入数据没有意义。所以，需要有可量化的管理模型，是很好给员工传递的，员工给客户传递的时候也不会走样，客户也感觉录入数据有了意义，皆大欢喜。

对于高端客户，就需要给软件注入管理方法。如何更有效的管理生产，如何更有效的管理销售，如何更有效的管理服务。而这种管理方法，是普通员工所无法理解并传递的。所以这类客户，就需要咨询式销售，与之跟随的产品市场宣传、实施、服务都不一样。

我们很容易从这里例子联想到中国移动。就一个打电话，很简单的功能，还需要把客户区分为全球通、动感地带、神州行。全球通选择了王石作为代言人，“我能”的宣传口号寓意了企业领导者掌控企业的信心和霸气。而动感地带选择了周杰伦，我的地盘我做主，很嘻哈。而神州行选择了葛优，我看行，整个一个精打细算的老百姓。大家如果解除过中国移动的客户杂志，你会发现全球通、动感地带的杂志内容截然不同。这就是客户细分。

我们的软件产品也需要客户细分。否则低端客户觉得产品复杂，高端客户觉得小儿科，我们两头客户都得罪了。

这几天的罗布泊之行，公司管理团队做了很多交流，对未来产品发展和盈利模式和销售模式和现有员工素质都做了综合讨论。要不要开发新产品，什么时间开发，市场目标客户群多大，潜在竞争对手是哪些，我们如何防御，我们的盈利模式怎样，他们会占有多大的市场份额，我们最终可能会占多大，值不值得我们做，我们现有的资金和人才是否匹配我们去开发新产品，我们的困境和发展是否是新产品能解决的，在我们投入人力和资金研发的时候，哪些业务和人员来支撑发展，竞争对手如果发动进攻我们如何防御。我们一定不能一边忍着挨打一边研发，那样很可能我们没有尝到新产品的益处就自己先死了。

新的产品而且要与现有产品形成增强的核心竞争力。就像微软，虽然是世界软件帝国霸主，虽然微软的产品线有上百条，但微软主要营收也仅仅是在于四个产品：Windows、SQLSERVER、Office、VisualStudio。四个产品有机结合，互相增强，使微软整个产品战略整体推进。别的公司也生产微软类似的产品，但整合性不好，或者只是生产某一类产品，就无法形成整合竞争力。我们经常会说一句话：用了微软的东西，就得什么技术都得用微软的，否则就配合不力。可见微软的产品整合竞争力多强。我们做产品也需要这种产品体系。像金山词霸、毒霸、影霸，虽然都是消费类软件，通用性强，受众广泛，但几个产品并没有形成整合的竞争力，全是单点，在翻译、杀毒、多媒体播放各个领域都面临着强大的竞争对手，无法形成自己的整合竞争力，所以一直前进的很辛苦。我们做企业管理软件，就是不断产生主打产品，几个主打产品互相结合，并且这几个主打产品还能扩展形成相关的小的子系统，这样每个主打产品其实都是一个产品群，几个产品群互相结合，这种解决方案就很威力强大了。所以我们做产品规划的时候，都要思考新产品与老产品的关系，最好形成整合互补关系，而不是代替的关系，更不能形成互不相关的关系。

我们不赞成跳跃式产品规划，看什么有前途就去做。因为一个公司在行业中有固有的形象和影响力，客户认为你就是IBM而非DELL而非迪斯尼，当然苹果转变不成通用。这就是企业气质，和人的性格一样，有些人永远无法成为管理者，有人永远无法成为软件开发高手。有网友提到蓝海战略。第一，蓝海不是那么容易寻找的，但公司需要每天开张销售。第二，寻找到的蓝海往往与现在无法结合，看着好却无法去做，做企业不能乱做，有可为有不可为，今天看搜索不错就去做搜索，明天看地图不错就去做地图，后天看网游不错就做网游，

这类企业活不长。你如果在这样的企业工作，你能适应不断转变的要求么，你是一个搞搜索开发的，你能很快转去做网游吗？所以，产品战略规划，最难的就是继承过去，结合过去，还要面向未来。很多程序员有个毛病就是老喜欢推翻别人的代码自己重写一次，这种程序员能力是很低的，高的程序员都是如庖丁解牛一般，在看似复杂关联的环境下还能游刃有余。

产品规划，我们尽量促使产品成为消费类软件或基础类软件。所谓消费类软件，就是每个人都需要，每个人都得装一套。所谓基础类软件，如中间件、数据库、阿里巴巴平台等等都是基础类软件，上面可以插入或扩展许多合作伙伴的产品，形成了强大的产业链。这种产业链就类似整合产品核心竞争力，互相结合整体推进，就强大的很。大家研发产品的时候一定要记得这两个关键点：整合产品竞争力、基础类软件与合作伙伴。

其实做产品，都有一定的产品周期。我这10年来，经历了两代产品的研发。第一年开始提上日程，解决要不要研发的事情。要不要研发就涉及到我上述提到的那么多问题，不商量定，新研发或许刚开个头就被下马了。而且要不要研发，需要全体人的支持和老板的坚决决定，否则有些利益团体给老板吹风，本来研发就是需要大量的人和资金，需要长时间研发、推广、销售才能看到效果，如果目前一面临点小困难，就很容易让老板动摇，就把新研发砍掉了，大家都去忙于应付目前的困难。所以，很多公司没有新研发，都是随着客户的项目来完成产品的实现。这样必然留下这个项目的缺陷和局限。所以，大家也总是忙于应付困难，而且困难总是一个接一个，怎么也解决不完，而无暇顾及未来发展研发。

解决了要不要研发的事情，就需要把人和资金和时间都抽出来。需要的人，现有的员工无法达到，就看是招聘还是与其它公司合作，还是做代理先下水探探路？另外，抽取出来现有员工，他过去的职能就需要相应能力的人顶上。这个交接能否顺利交接，都需要时间。而时间就会引起变动，新的异常又会出现，到底去救火还是抽刀断水？

把人布局好，才能规划产品。所以一个产品的研发，往往是立项、筹备就花费一年，设计、开发、测试、文案，为新产品而匹配的市场、销售、咨询、实施、支持团队的打造这些都需要花费1-2年的时间。

而产品一出来，需要花费大量的资金来做宣传。宣传了还不一定能起个水花。这个阶段的销售收入，根本无法承担公司主要收入来源，还处于投入大于产出期。一小部分客户才知道，很多客户还不知道。即使这一小部分知道了，许多人还在观望，希望看看别人应用的效果。好不容易有先锋客户愿意尝试，但由于为了推广，销售单子就必须低价推行，否则人家觉得风险太高。等做出来几个典型的成功案例，可以带其他潜在客户进行现场参观的时候，一年已经不知不觉过去。

一年多的销售攻势和市场攻势，还有实施成功案例的支撑，终于有数量的客户签单了，他们看到了应用的成功案例，开始应用，但远还未到大斗收金的阶段，这才是开始投入产出刚刚能持平的一年。不要乐观，这个阶段很危险。因为灯塔客户的实施，只是个别。如果失败了，影响也就那么大。再挑选一些客户，还是能东山再起的。到了这个阶段，可能是销售硬顶着劲花费了大量手段强挤出单子，最后却被做砸了，这时候影响就大了。因为这类客户不是灯塔客户。灯塔客户很先锋，知道尝试是有可能失败的，失败了也在情理之中。但这一阶段的客户，都是想尝鲜但又怕烫手的主儿，一旦失败，很容易传播其他更谨慎的客户，说他们的产品不行，不能尝啊。这就，没有后来了，半路地夭折了。

在第五个年头，终于迎来了收割年。大规模的销售、实施都展开了，团队也磨合成熟了，做事也有固定流程和模式了。如何低成本、有质量、大规模的实施成了关键思考点。这个时候，看着销售势头挺好，往往年底一算账，成本也高，最后没捞到几个子儿。

在第六个年头，多年的媳妇熬成了婆，销售收入节节高，开始资金补仓。第七、第八年都在延续、持平，不断升级维护、做相关周边功能的开发。那些怀疑者、守旧者也都随大流购买了软件。但危机已经暗浮。一个市场，往往目标客户数是一定的，软件这个东西也不是

像矿泉水一样天天喝完还得继续买。鱼总有打尽的一天，做维护服务和升级服务，也只是把鱼养起来慢慢吃，但总是吃的少，现在的服务费用还无法支撑公司主流收入。

我们运作产品，就是希望推广期尽量的短，销售期越长越好。通过细分客户群，通过新业务功能的增加，通过关联产品的开发，我们不断延续。可能，换了一个新的界面风格，改进的性能，改进了稳定性，就可以发布一个重要的版本，做个市场秀。市场这个东西不能冷了，一冷了再想炒热就比较难。不延续做，客户就容易遗忘。就如同可口可乐，如果一年不做广告，就在渠道默默的铺货，很多消费者都奇怪可口可乐是否要倒闭了，种种猜测就让消费者购买产生迟疑影响了销售。

在销售好的时候，大家都觉得产品肯定还能火好几年，反对下一代产品研发的人很多。

而真正的下一代研发，其实最好在产品第七年就开始启动研究立项。这样，老的产品在推出 8-10 后，慢慢进入衰落期，下一代产品就能成为新的现金牛，继续带领公司营收创新高。

研发的早了，推出来市场接受不了，不仅花费大量市场费用而无起色，就连员工和公司高层都觉得没有希望。本来好好的一个产品，就这样被停掉了。研发的晚了呢，人家竞争对手已经推出，我们才匆忙跟随，很容易新产品不优秀，老产品在衰老，几个回合就全军覆没。所以做公司，真是每日如履薄冰，随时都有覆没的危险。

关于新产品的研发，一定要有这个产品生命周期的视野。要从产业、竞争环境、国家环境、客户行业、技术发展多个角度去思考产品规划。在产品 8-10 年的生命周期内，每一年的产品完善重点，每一年的市场、销售、定价、实施、服务策略都需要不断调整，而且要针对不同产品层次不同客户层次调整所有策略。

很多人问我是怎么研发一套可扩展的架构。我说，你不了解这个架构最后的演化未来，不了解这个架构的生命周期，那么你就无法研发可扩展的架构。我们是做企业管理软件，不是做一个百变金刚的企业业务开发平台。架构所应用的产品，它的生命周期是 3 年，5 年，还是 10 年，还是 20 年，决定了架构的要求。而且客户行业未来的 8-10 年的变化会如何。而客户行业往往受中国政策和发展变化影响很大。所以我也喜欢看《经济新闻联播》之类的节目，也阅读《中国经营报》这样的报纸。我一般也就能看到 3 年，5 年都看不到。这样的素质，把握一个 8-10 年生命周期的产品，确实需要不断的升级自己的知识面和眼光层次，以保证自己能不断调整，不断滚动这 3 年的步长眼光。

这就是一个 CTO 需要做的产品生命周期规划。大局规划，细处入手，客户最关键的需求实现。大规划、大研发，既耗资金又不容易掌控这么庞大的团队又不容易控制这么长的研发周期。而技术总监，可能重点关注于这个产品的实现执行组织。而 CTO，就需要在客户、产业、技术、盈利模式方面能有长远并且综合的产品战略。CTO 和技术总监的关系类似于国家主席与国家总理之间的拍档关系，一个负责产品战略，一个负责产品实现执行。

波、波、波，我想起产品生命周期，老想到这个声音。这个声音来自于我曾经痴迷的《街头霸王》。里面的 KEN 一样，会连续发出白色的气功波，几个连环组合就把对手打倒在地，一波接着一波。我希望，产品波也是如此。

23、八部众

这几天在规划新产品，新产品要做什么，两个来源：

看看业界最新的产品，先来个海阔天空的头脑风暴。从 ipod 模式谈到金山与 google 的合作，从 android 谈到百度的电子商务，从孙正义的投资校内网到汽车 GPS、车载充电、车

载 MP3。但这些只是引新思路，真正还要落回到自己所在的行业所在的客户。正规的干，和现在业界的标杆比，我们水平差，和他们用正规的方法交锋，只有输的份儿。所以，历来以少胜多，都是以奇取胜。我们作为中小企业，把金庸+古龙，或者王朔+鲁迅这样来个改良菜，把其他行业或领域的新产品新模式引进来，或许才能突破现有大佬制定的游戏规则。

踏踏实实，还是要去检索我们的需求库。历年来，全国客户提出来的数以几千条的需求都记录在其中。小说，就是来源于生活而高于生活。所以，需求就是现实生活，我们的新产品必须从现实需求中提炼出来。否则就是空谈新产品，而根本无法落实。

我经常看到不少内地程序员拿 Google 的开发和国内的开发比。在 Google，软件是艺术，大家阅读源代码也阅读的赏心悦目，大骂国内软件业毫无创新。我个人以为，我们的积累还是太短，从业者年龄和经验结构偏低，还无法从现状而创新。另外我们面临的资金、客户的限制，我们更是没有多少发挥空地。所以我认同软件工厂做法，大批量高质量低价格快速度的生产。但是，现状是，连能把这种生产模式做的好的软件企业都寥寥无几。大量的软件企业无法实现高质量大批量快速度，低价格也是降低质量克扣员工得来，势必引起客户和员工的反抗而终结低价格。所以，我们做新产品，也不能抱着科学家研究的思路，而更应该是在资金、时间、人的素质、人的数量、竞争压力、客户现状的框架下的量化可控的研发，有阶段，有目的，有重点的研发。

要有控制的研发，就要有需求管理工具来管理需求库，可以分类查询、检索、统计。就需要老老实实去回顾需求库。如果需要调研，就需要有方法的调研，从组织结构、过程、考核、优化这几方面来梳理需求，而不是开讨论会。

有了这么多需求收集回来，大家估计会很晕。因为需求来自全国客户七嘴八舌，有来自豪放的东北，有来自细腻的华东，有的来自客户老板，有的来自部门小经理，有的来自部门小组长，有的来自 IT 人员，有的来自最终操作者。所以，真是林子大了，什么鸟都能飞出来。我记得我在 2000 年实施的一家客户，最终用户年龄大，打字不灵，希望有个语音输入。这就是需求。如果我们面对这样的需求，我们肯定会说不可能完成。但我们往往不仔细想如何去解决问题，反而耻笑客户傻瓜，怎么不把年轻的换上岗位。但我们了解到那位最终用户是一位专业经验很高的人员，岗位无法代替，我们都闭嘴了，我们最终还是解决了问题，但肯定不是语音输入。所以，我们整理分析需求，不能耻笑这条需求肯定是用户拍脑门想出来的，那条需求是客户自己笨。到最后，还是我们自己愚弄了我们自己。就如同遇到 BUG，老是有程序员说不可能，我机器就不报错，或者说肯定不是我的问题，我都检查过了。但最终最终，还是代码问题。这种现象屡见不鲜。

一个软件，对于不同的人来说有不同的要求。如果你把这些需求归类，你往往会得到这些要求特征。

对于销售来说，演示的时候稳定、易用、好看、速度快就是关键。

很多销售并不懂软件，但要买软件，这就是现状（别笑）。我们要就问题解决问题，耻笑问题也解决不了问题。本来就不懂软件，还不易用，销售根本没法给客户讲清楚自己卖的这个东西到底有什么用。

如果还不稳定，突然报出一个英文错误，销售在那么多客户面前更是漏了脸，吹的那么大现实却是如此，让销售的诚信损失了，必然会到老板那里告一状，以挽回自己打单不利的面子，错全是研发部的问题，那时研发部只有吃不了兜着走了。

如果软件做的灰不啦叽的，自己都觉得没什么出彩的，客户也觉得很普通，在价格上就无法提升。而销售，最重要的就是卖一个好价格。软件也实用，也先进，但打单过程中，给客户演示也就那么短的时间，而且来听的人大多是以以后不实际操作软件的人，对软件到底功能做的多细腻，处理各种复杂业务多么灵活，都根本无法看出来，就看到这个软件不好看。就如同我们遇到一个女孩，很漂亮，到底心灵如何，还没有那么多时间和机会去了解，但首

先感觉就不错，愿意去接近。如果遇到一个女孩很普通，从心里一开始就有坎，不是抱着主动去了解的心态，而是怀疑和旁观的心态，就不容易了解一个女孩子。软件，和女孩一样，都同样的道理。

演示的时候，我往往会看到这样的情形，点下一个查询按钮，10 秒钟出不来结果。客户和销售都在等，会议室很静，大家都在盯着屏幕都在等待，但就是不出来结果。销售急了，拼命乱点，更加剧了不稳定和性能约束，最终报错不断，或者干脆销售很尴尬的按下 Ctrl+Alt+Del。

对于实施来说，最重要的就是软件稳定。软件不稳定，客户怕软件使用过程中出现问题，就不敢放实施人员走。而实施人员上面还有实施总监在管，有项目进度和经费的控制，所以实施总监老催实施人员为什么还不回来。实施人员也急呀，今天查报表，明天改数据，总是按下葫芦起了瓢。

软件稳定的基础上，最令实施人员头疼的就是客户需求的事情。如果每遇到一个需求都需要让程序员搞定，而程序员又少，就把实施人员晾到现场了。实施人员本来就想和客户搞好关系，以希望能够把项目顺利进行下去。这下，长时间解决不了客户需求，实施人员就交待不下去，当然要给程序员施加压力。这就是开发部往往是软件公司最受攻击的一个部门，销售、实施、支持都攻击开发部。所以，为了能让实施人员现场满足客户需求，开发人员往往想了很多办法。最常想到的就是开发平台。但我见到许多开发平台，简直就是面向开发人员的（什么 XML、SQL、流程编辑）。实施人员只懂业务，对计算机软件操作并不娴熟。所以，开发人员必须要给实施人员提供实施人员能用的起来的实施工具。很多软件系统没有实施工具，只有一个光杆软件，孤立无援。

对于培训来说，软件易用最关键。你帮助写的再详细，相信看的人都不多（只有我们可爱的程序员才去勤奋的钻研那些详细的 WINDOWS API 帮助）。软件易用，培训的工作就轻。

有很多软件，没有演示版，没有操作视频录像，没有最新版本帮助文件，没有新版本更新说明。就凭一张嘴对着电脑屏幕讲。出了新版本，还不知道哪些功能发生了改变，还照着过去学习的知识讲。客户亲手一操作，发现讲的和看到的不一样就有了疑问。培训人员都脸红，自己都不知道怎么使用，也解释不了。所以培训文档对于培训人员来说也很重要。

对于支持来说，软件稳定是第一位的。否则自己的支持电话非打爆不可，那样，支持的工作又累、钱又少还整天郁闷解决不了，还不如不干支持。软件如果不稳定，那么软件必须可跟踪。最怕软件出了问题，客户打来电话就说：软件不好用。这个不好用怎么查问题啊。到底怎么个不好用，报错的界面截图是什么样的，在哪个模块，怎么操作的，录入了什么数据，报的内部英文错误是什么，版本号是多少，软件打了多少补丁，操作系统是什么版本，操作系统打补丁没，数据库打补丁没，防火墙是怎么设置的，年月日和货币符号设置对不对，打印机设置对不对，自己的 IP 网络设置对不对。这些内容，最好像 WINDOWS 一样，出了错，把所有需要跟踪的信息都自动收集起来，然后报出一个提示框，可以发送报告给微软。所以，我们做了一个日志模块，可以自动截图，自动发送日志，自动记录当时操作的 SQL 语句，自动记录当时的客户输入数据和点击操作流程。给软件跟踪解决问题加快了许多效率。如果一个个去问用户，用户都不知道这些信息到哪里去收集，再一顿反复解释寻找，解决问题就很慢了。有很多时候，用户由于时间太长有其他事在身，就放弃了解决这个问题，久而久之，由于问题越积累越多，就渐渐不用这个软件了。

对于支持来说，软件自动升级也非常重要。我们往往遇到很多问题都是软件没有打某个漏洞补丁造成的。而且还有很多问题是由于客户端版本不统一造成的。如何能自动的、全部客户端一起升级，一发布补丁就自动全国升级，很多问题客户还没来得及发现就被解决了，满意度就上来了。

对于后续版本开发维护人员来说，代码容易看懂，代码好修改才是最关键的。程序员想

了很多方法：业务开发平台，有意义命名，小函数分割，函数接口灵活，面向对象，设计模式、重构等等。但是，代码仍然越来越复杂，越来越不容易看懂，越来越不好修改。其原因就是由于每个客户都提出各种各样的需求，有时不同客户之间的需求还是矛盾的，大量的代码其实是为了处理客户异常业务，还有的是为了堵某个特殊操作发生的 BUG，还有的是为了兼容不同版本之间。这才是代码难阅读难修改的根源。

对于数据量大性能要求高的应用，性能是很关键的持续改进方面。

对于安全性要求强的应用，设计安全的方案，编写安全的代码，安全的测试覆盖是很重要的工作。

对于测试人员来讲，软件必须具有可测试性。软件代码写完了，什么样的操作或结果是正确的，什么样的操作和结果是不正确的，没有人告诉他，也没有文档，这就不具有可测试性。这就要求有设计文档，详细写明什么样的操作和结果是正确的。这样就有了可测试可验证的标准。很多软件不稳定，最后加了专职的测试人员也不稳定，其根源不在于测试人员测试方法不对，测试经验不丰富，而在于根本没有测试依据，测试人员只能自己凭经验乱点乱试，根据经验来判断这个结果是正确还是错误。尤其一些报表，输入条件，数据都出来了，但是数据之间是有关联关系的。但这个关联关系并没有设计文档说明，测试人员并不知道，就认为这个功能是好用的。其实这个报表数据是错误的，虽然能正常显示。

对于文案人员来说，软件必须能让文案人员编写文档。许多软件没有设计文档，代码开发完毕，让文案人员自己边学习操作边辅助测试边编写文档。文案人员不是设计人员，不是代码编写人员，不是测试人员，是对软件做陌生的人。他本身都对软件不了解，可想而知写的帮助文档有多大的可帮助性。软件没有帮助文档，其根源就在于没有设计文档。而没有设计文档的根源，在于根本没有编写设计文档的人。谁来编写设计文档？程序员？程序员再写代码。测试人员？文案人员？实施人员？培训人员？到底谁来写这个设计文档。

我看过许多网友在讨论怎样一个软件才算一个好软件，说了很多方面。但是从现实来说，我们真的需要那么多方面吗？

往往现实一开发，什么好软件的标准都丢了，程序员单枪匹马上手。还有一些开发团队，希望能做一个好软件，于是希望把这些好软件的标准都实现了，最后周期长，在有限的人力和开发时间内无法完成，只好虎头蛇尾，最终还是个烂软件。

业界有个笑话，就是说微软的软件，从第三个版本才能使用。

这说明，一个好软件，应该具备好软件的标准，但一个好软件，不是在一个版本就把这些标准全部实现。而是有步骤有重点的实现，逐步达到一个好软件。

那么，这些好软件的标准就必然需要排个顺序。

编软件，是为了什么？

是为了卖。

怎么才能卖个好价格呢？

嗯，包装成漂亮的就能卖个好价格。巩俐穿上棉袄也就是个秋菊，村姑化完妆也是个靓女。韩国整容大家有目共睹。

就是这个思路。

所以，我并没有把软件实用放成第一位。因为新软件研发，你并没有很深刻理解客户，你假设认定这个需求很实用，到了现实使用发现无法执行下去，这就废了。而且实用不实用，每个客户的理解是不一样的。有人觉得电子邮箱很实用，有人觉得电子邮箱没有用。就这个情况。所以，我设计软件，往往只设计不超过 3 个实用亮点，实用亮点多了，我们开发也周期长成本高难度大客户可能还接受不了，而且过于复杂销售和实施人员无法给客户讲明白。所以有 1-2 个宣传亮点就 OK。在不断销售不断实施过程中，客户会自然提出来需求，软件就会不断实用起来。

然后，我就会把软件包装漂亮。专业的销售方案 PPT，专业的帮助文档，专业的软件界面，专业的图标，统一的操作，统一的字体，统一的专业词条，统一的对齐，专业的提示（很多软件提示居然是：“不好意思，出错啦”。真汗~）。这需要美工、文案、界面开发人员三者配合。

漂亮过后，就是稳定。稳定就需要软件具备可测试性。

这样，软件就可以出去销售第一版了。

到了软件第二版，客户签单量就上来了。实施工具就要提上开发日程，否则多个项目并行提出来的需求能把程序员的工作排到明年。

到了软件第三版，客户签单量更大，老客户也用了 1-2 年了。服务的压力上来了。所以软件必须可跟踪可自动升级，在持续不断增加实用功能，增强功能稳定性的基础上，这是软件第三版的重点。

到了软件第四版，软件越来越复杂，如何兼容，如何容错，如何容易阅读，如何容易修改就成了首要问题。这个版本，重点就是内部代码重构优化。

到了软件第五版，性能是个问题。过去 3-4 年积累的数据使系统越来越慢。优化性能成为第五版的重点。

到了软件第六版，由于这么多版本的升级，功能很是复杂了，使原本易用的软件现在变的越来越难懂了，到底是特殊处理的业务。把常用功能和非常用功能分开，把正常业务处理和异常业务处理分开，做到组合裁减，一部份不常用的功能就渐渐萎缩掉，一部份常用的功能做增强。在这一版本，重构易用性成为重点。

软件到了第七版，就几乎在打补丁了，不再投入大量人去维护了。软件进入大销售大实施大维护的收割阶段，维护本版本的开发团队在萎缩，下一代产品在酝酿。

这就是一个软件生命周期中，不同时期的不同开发重点。把握好节奏，合适的时机做合适的事情，一点都不浪费投入人力。

但是我们要注意，性能是一个结构性的问题。所以虽然在第五版才调整性能，但对于企业管理软件来说，必须在基础设计的时候就强烈关注数据库设计。因为数据库结构一旦固定就无法更改。从过去的经验来看，只要数据库没有设计缺陷，性能的瓶颈主要在代码上，只要改进代码和功能设计（有些功能设计本身就性能很低，大量的功能集成在一个界面上岂能不慢？），性能是很好解决的。

对于代码的重构和优化，如果从始到终遵循着函数封装，小函数分割（我曾经遇见过一个 3000 行的大流水函数，不敢下手，怕一旦修改不知会发生哪些 BUG），优化和重构也是很容易做到的。

网友们讨论了许多，有实用性、稳定性、容错性、性能、可测试、可理解、可修改、可实施、可支持、灵活性、移植性、兼容性、安全性、易用性.....。

但这么多要求，我们都要有目的分阶段的一步步达到。而且，往往我们不断补齐上一阶段留下的遗憾，我们此阶段的努力又会形成下一阶段的遗憾，总是无法达到一个赏心悦目可以笑看江波的软件。可能，世事轮回皆此规律。

后注：

八部众为佛经故事。八部分别为八种似人非人，似神非神，似鬼非鬼，似善非善，似恶非恶的种类组成，他们散落在佛界三十三重天，或喜或嗔或怒或思或乐，但种类之间总是互相关联互相矛盾又互相共生，层层迷障无法拈花微笑。

一个好软件，也是如此多的标准特性，也是如此共生又如此矛盾，颇为神似。

24、葵花点穴手

我的手下经常会面临这样一个问题：客户必须让咱们按他们的需求改，您看怎么办？

这种情景大家可能很熟悉，一个业务处理，可以这样处理，也可以那样处理。你的软件采用了你的处理方法，客户采用了客户自己的处理方法。两种方法平风秋色，没有优劣。但客户用惯了自己的方法，所以必须让软件改成客户自己的方法。

不改吧。没有理由，因为两种方案都差不多，但客户就是客户，客户占上风，否则就不验收不给尾款。改吧，又有什么意义？这家客户习惯了这种方法，下一家客户又不适应这家客户的方法怎么办？到一家改一家？

这都成了什么事。

我也深被这种问题困扰，至今没有好的解决方法。但我仍然力求找到一些方法去改善。我就是这样一个人，能改善一点就改善一点，量变就会发生质变（当然，做这样的管理者需要时时观察细致分析研发过程中的问题）。我这几天看丰田管理方法，丰田连一个工人的左右手空闲和手臂抬高高度和次数都做了细致分析。因为手臂抬高高度和次数决定了这个工人的有效的工作时间和工作强度，我们工作的时候如果老需要不断扭脖子，我相信很快就会得颈椎病。

一个软件，经过多年的发展，不断的客户实施，加入了很多客户的需求。我们返回头可能会发现这样一个现象：我们走的太多，我们甚至忘了我们为什么要走。当时当景，我们觉得客户说的在理，不修改就说不过去，于是做了修改，但天长日久的修改积累，我们发现软件主要要解决的问题已经被无数的功能左延伸右扩展，已经不是一个能一句话说清楚到底干什么的软件了。所以，做企业管理软件有句笑话：**ERP** 是个筐，什么都敢往里装。

说不清楚一个软件到底是干什么的，是个很大的问题。销售给客户说不明白，老员工给新员工说不明白（甚至新员工不知道这个软件有什么价值，没什么存在的意义），渐渐的，整个员工群体都在糊里糊涂的工作，反正有这个产品在，就打电话销售跟单呗，客户问什么都说软件都有都能满足。研发呢，客户有需求就修改。实施呢，签了单子就去实施，有什么软件功能就教用户怎么使，然后尽量让他使起来，管他需要不需要，管他急需不急需，管他合适不合适。支持呢，有问题就回答。大家都在做一天和尚撞一天钟。

我想到了曾经看过的微软的一种方法：

对于什么目标市场的客户（一定要精确描述好你的目标市场，千万别用什么中小型企业之类的词语）而言，你的什么产品或服务（针对这类目标客户，你提供了相应的什么产品或服务，这个产品或服务一定要与你的目标客户匹配），提供了什么主要价值（要精确，千万不要说提高了管理水平。管理这个东西很模糊，就类似这个企业管理水平低，到底指的是什么），因为你的这个产品或服务提供了什么独到之处（一定要独到，否则人家为什么买你们，而不是买别人的）。

这是微软的一个产品定位方法。

还有一种我自己总结的方法：干什么，用什么。

例如，杀毒，用瑞星。聊天，用 QQ。看新闻，上新浪。写文章，用 WORD。就这么定位清楚。我们知道，QQ 有很多功能，不仅仅是聊天，新浪也不仅仅是新闻。但我们能把我们的企业管理软件也讲的这么清楚就非常好了。记帐，用 XX 软件。记录进销存，用 XX 软件。但偏偏这个世界发明了几个很高的词，还云里雾里解释了很多，越解释让客户听的越摸不着头脑。如：SCM、ERP、CRM、Web2.0、SOA。而做企业管理软件，却又往往要

遇到这几个词。落实下的产品到底是不是 SCM、ERP、CRM、Web2.0、SOA，自己和客户谁都不明白，只能是客户目前有什么需求（这个需求可能根本不属于这些范畴）就做什么需求。

为了防止软件成为大杂烩（想起了猫扑，我一直不知道这个网是干什么的，只好把它定位于搞怪集中地，但好像也不对。年轻人的门户，好像也不对。），我经常用微软的方法和我自己的那个简单方法来校验产品，定期给大家传达，让大家渐渐模糊的产品印象又清晰重点起来。

我们也会经常一起讨论，我们的产品最适合什么样的客户，第二适合的客户是什么样的客户，第三适合的客户是什么样的客户。由此不断校正我们的目标客户群，调整我们的营销策略、销售策略、定价策略、服务策略、需求定制策略。

我们在描述最适合什么样的客户的时候，为了描述精确，用的方法是拟人法。

我们不会泛泛而谈，我们会从现在的客户群中找出最适合的已经购买使用了我们产品的客户。——描述他们的组织结构、企业文化、老板性格、老板管理风格、中间干部的管理流程、考核方法。不过，往往我们无法找出一个完美客户，能把我们的软件各个功能都是用的很好的客户，于是我们就拼凑，把各个功能使用优秀的客户分别挑出来，然后都按照组织结构、企业文化、老板性格...等等这些方面来描述，然后把他们综合在一起，一个“完美”的最佳客户就出来了。

我们在讨论期间，为了深刻的体会，往往都把这些客户的公司名称，城市（中国真是差异很大，东北人，浙江人，广东人，截然不同），用户姓名，部门，年龄，性别，婚姻状况，从业经历，性格，如何思考问题，他讨厌什么，他喜欢什么，他关注什么，家乡，什么学校毕业，计算机水平，工作环境，每天大部分基本在干什么事都描述下来。如，我们老说：XX 公司的 XX，老喜欢反对别人的意见，不管别人是对是错，总要说出自己的意见，显得自己很独特。还老爱用手那样捋一下头发。唉，他也是三十五六岁的人了，但还是个小头儿，老觉得自己满腹经纶老板不赏识。

这样，一个活灵活现的形象就出来了。我们大家的讨论就有了基础。省得老有人提：不可能有这样的事情，谁傻蛋了这么想。但一说到现实具体例子，大家就都明白了，确实是林子大了，什么鸟都能飞出来。（可能也有鸟人）

我们以后去做销售、实施的时候，就把当前这个客户和我们设计的这个最佳客户进行对比，很快就能分析出这个客户最适合使用哪部分，它的优点在哪里，它的缺点在哪里，它的差距多大，从什么方面去提升。所以做销售，总是很快能把握住客户的兴奋点，做实施的时候也很快能根据客户的关注重点来突破和提升。其实这种方法，在业界有个很标准的名字，叫“标杆法”。

我们描述完最佳客户，我们就要去列举这个最佳客户的需求。我们到底要解决这个最佳客户的什么问题？

一个企业的经营，都有一本难念的经。每个阶段都有烦恼，小有小的压力，大有大的臃肿，中不溜还上不上下不下难受。各个方面都会有问题，从老板的管理风格，到企业财务核算，到企业财务报销，到企业考核，到招聘培训，到销售到服务，涉及到各个层次各个角度。真要数落问题，每个公司数完后都觉得这个公司没救了，但事实上这样的公司还活的好好的。所以，看事情要平和，你眼睛老盯着灰暗，你会觉得真个世界都太黑暗了，你活着太痛苦了，这就不对了。这真是自己跟自己过不去（有句笑话：饿你两天，你啥也不黑暗了，能吃个窝头也感动的痛哭流涕）。

所以，如果我们真要去解决企业这么多问题，我们也不是万能公司，也没有能力解决。我们只能解决我们擅长解决的问题。但是我们擅长解决的问题，真的是企业现在急需的吗？这又是一个问题。解决了也不疼不痒，那企业怎会买单？

所以，我们把企业急需，而且我们擅长的，而且我们的解决方法是我们的目标客户能够执行的（别想出一个解决方法，但要人要钱要时间，哪里有这么充分的条件？这样的解决方法说明就不可行），而且解决后能有明显效果（很多东西是长期才能看出效果，这类问题的解决就比较痛苦，我们不希望这么做，这样会使观望的客户的投资购买延迟），而且客户愿意为这样的解决方法买单，而且买单的金额符合我们的盈利目标（我们也不是雷锋）。

经过这样一筛选，确实剩不下几个急需解决的问题点了。有时候大家讨论偏了，还会讨论到什么点都不剩了，于是停下来，重新过，不要讨论的那么极端。然后大家在这些问题点上统一认同。我们做营销，就是针对这些问题点提出我们的解决方案。我们的软件发展导向，也是趋向解决这几个重点问题点。这样，我们的营销和产品就是统一的。就不会出现我们老看到的一些现象：营销说的是一个东西，拿到手是另一个东西。

全公司同一个思路，同一个目标，这是最厉害的。（这让我想起了，有人曾经想 UML 把客户、设计、开发、测试都串在一起）。在软件公司中，把营销、销售、开发、实施、服务想串在一起也是非常困难的，大家对客户的体验是不一样的，关注的重点也是不一样的。所以，用这种方法来描述客户，描述客户问题，全公司的认识就统一了（当然，那些技术至上者除外，他们可能只想借公司资源把自己练成高高手，我直到现在也不明白练成高高手而不能解决客户问题到底是为了什么要练。我过去深入学习组件技术，搭建业务平台架构，学习数据库技术，皆为解决客户手头急需的问题。客户其实只想解决问题而已，什么技术都无所谓，只要能解决了，最好是越简单越好的解决方法）。

25、文档知多少

去年，我们要让软件开发团队管理上台阶。

我们由于处于企业管理软件开发领域，而对日外包大部分接的单子都是管理软件之类的单子，但是人家的项目管理、进度、质量都比我们好，如果他们再配合管理咨询公司作为合作伙伴，再加上大规模的服务呼叫中心，像我们之类岂有出路？

于是我们就想到了引入对日外包的开发过程管理。

大家一想起对日外包，就想到了大量的文档和大量的代码工人，想到了详细设计说明书甚至到函数级、伪代码级。

要不要引入的时候，我们内部也做了争论。觉得对日外包，人家接的单子额比国内客户大，所以也能招聘大规模的员工，而且对日外包，日本人是很理性的看待项目周期的（国内客户要求一个月开发完上线），而且日本人都做了半年到一年的调研和设计分析（国内调研几乎只是一个上午，坐在一起瞎聊，根本不成方法）。所以对日外包不适合咱们。咱们没有钱招聘一定数量的人（即使我们只需要普通员工，而不是人才，我们也没多余的钱），当然我们也无法分离那么多专职的项目管理、开发、测试、文档、配置管理岗位。我们的客户对于软件的认知决定了我们无法在调研、设计上下太多功夫（单子额就那么大，客户认为软件就值那么多钱，当然无须对软件生产各个环节进行重视）。

不过，我还是坚持进行引入。是骡子是马，咱们拉出来遛遛。还没遛，就不行。这不是小时候的小马过河了么？

引了进来，合作伙伴给我们派了一位质量控制部的人员。

一入手，发现有个很关键的问题，方法的源头。

因为对日外包，都是接单生产，主要是编码、测试，保证生产进度和质量要求。但编码之前的所有环节，对日外包公司并不清楚。他们只知道拿人家给的设计书开始 coding，设计书怎么来的，前面环节产生过哪些文档，不清楚。

幸亏我们过去有系统的调研方法，从调研描述现状、然后分析优化的组织结构、工作流程、考核报表、岗位职责四大块来描述需求。客户优化后的流程、工具中包含手工纸张信息、EXCEL 信息、软件信息。

把这些转变为软件中的功能、权限、报表也一气呵成。组织结构和岗位职责决定了功能点和功能权限、工作流程决定了软件流程、工作流程中使用的纸张信息、EXCEL 信息、软件信息就是数据输入，报表就是数据输出。这就是一个企业管理软件的四大块：组织权限、输入、处理流程、输出。

所以说，企业管理软件的开发是有方法和规律的，比较容易，就连最难的调研和需求管理也有方法。所以企业管理软件的开发，现在主要集中在大规模开发团队的组织、任务调度、人员培训（大规模的开发，必然需要的是一般素质的人员，而非高级人才，否则不可能有那么多资金来实现大规模开发）、大规模开发团队的质量和进度（人多了，各个层次各个水平的人都有，理解都不同，如何保证质量和进度是很关键的，否则很容易项目预算失控。一般素质的人多了，对于管理的要求是很高的，很容易成为乌合之众，只消耗不产出）。我特羡慕 KFC，不管我们在大江南北哪里，迟到的 KFC 是一样的口味和品质，享受的服务和环境也差不多，这很难。那么多店，而且都是授权店而非自主经营店，那么多一般员工，而且员工流失和临时员工也非常多，居然能保证一样，管理水平实在了得。所以我经常学习 KFC 和丰田，如何使一般员工大规模配合工作。

对于企业管理软件开发过程中的文档，我们一般有需求分析说明书，其编写格式和思路，和我们的需求调研方法一致，也就是说，我们的需求调研的结果，落实到纸面，就有需求分析说明书，另外还有一份需求调研报告，是偏向于项目过程叙述的。

需求分析说明书回来，研发部内部会进行大家一起学习理解，然后讨论。

讨论主要由：需求调研人、业务组组长、测试组组长来参加。一个个的过流程。因为在需求调研期间，去的只是调研人，可能有想不全的地方。如果这样就直接进行开发，无疑会有很多漏洞。这样给开发、测试，都带来了返工修改，给项目管理也带来了项目进度、任务分配的调整，计划的打乱也间接影响了质量管理。

根据大家讨论补充后的需求分析说明书，就比较容易得到我们下面环节的文档。

首先我们会出功能点文档。

我们会把需求分析说明书中的业务功能都列出来清单，属于组织结构建立、组织角色、权限分配、登陆验证、基础数据维护之类的都归类到系统功能中。系统管理，各个企业管理软件差不多，我们又有公共的系统管理模块，就不需要重新发明轮子了。所以，我们主要重点是分析业务功能。

根据需求分析说明书中的每个流程，都先提出来成为一个功能点。然后针对现在整理出来的功能点，再一个个对照流程，如果这个流程复杂，就拆分，把这个功能点拆成几个复杂性和预估工作量差不多的功能点。经过这样的拆分，就形成了最终的功能点文档。

而功能点之间，根据上述方法的拆分，就形成了功能群。

功能点就成为功能权限控制的最小单位。功能群就成了软件菜单中的一项。几个相关联的功能群就成为了一个业务子系统。

就这样方法，使子系统-功能菜单-功能点（可能是某个功能窗口上的功能按钮）三级分开，与组织结构-员工-角色-用户-权限结合。一个软件，未来会成为什么样子，大框架就出来了。

做功能点清单，就类似于跑马圈地，这个项目到底多大，我先把项目边界框起来，而

不要让这个项目无边无界，那自然也不会有可落实的项目进度和项目管理。知道了项目最大做到多大，就能决定是亏是赚，是做还是不做，能不能做了，有可用的时间和人力来做否。

然后，我们会根据功能点清单，为每一个功能进行优先级的标示。我们通常会把优先级分为三级。这就意味着一个项目，大致分为三个阶段。一级是必须要做的，即使延期也要做，必须调度多加人手多加班也要完成。一级做完后，如果有时间，就把二级完成。如果时间超期，有适度的尽量去完成二级，可以延期，但也要根据预算和时间。如果适当延期也无法完成，我们会给客户去上线实施，变实施边并行开发，使实施团队和开发团队进行并行工作。所以，二级也是重要的功能。三级就是如果时间用完，三级的功能就要舍弃掉不开发。

一般是，按功能的重要性来划分优先级，我们在之前已经讲过，我们调研需求的时候，就把常用业务和异常业务分开，把每天做的业务，和每周、每月、每季、每年做的业务分开。几个结合特别紧密的，互相关联的，我们也会把他们划分在同一个优先级，需要单独开发的基础数据维护界面，我们也会放在同一个优先级。这样，只要我们项目到期，或者我们迫于竞争突变，我们会随时推出一个可以完整使用的系统。虽然这个系统可能功能简陋，但可以完整处理整个常用业务流程，而不会造成中断，无法处理下去。

很多开发团队，开发的方法是先字典功能，然后是业务功能，然后是报表。这种开发方法就导致了业务系统，客户上线使用了，光输入数据，没有输出，业务系统成了一个闷葫芦，用户得不到使用价值，可能只是减轻了工作量，加快了工作效率，提高了部门间的自动配合。更有甚者，业务功能开发了一半，到处都是断路，走不下去，无法成为一个完整的系统，就是个半成品，上不上下不下，无法适应竞争变化。

我们开始针对功能点清单编写我们的功能设计说明书。

我们是按照优先级来编写功能设计说明书的。我们并不会把功能设计说明书都编写完毕后才开始编码。而是，我们先把优先级为一的详细功能设计编写出来，就开始开发。二级的功能编写会在开发人员进行一级功能编码的时候并行。

功能详细设计说明书由业务开发组的组长来编写，属于系统类功能或公共类的功能，都归给公共代码人员来编写，需要复杂的算法，高性能，高安全，高数据量，需求一直未确定最终解决方案的未来未知变化的接口，也都由公共代码人员来编写。所以，一个开发团队，有高技术公共代码人员，有深刻理解业务的业务开发组组长，有普通的 coding 人员。业务开发组的组长一般由熟悉客户业务，编码经验较多但技术能力一般、踏实细心负责适合团队管理的人担任。

功能设计说明书，根据每个功能点详细展开描述。一般一个功能点由一个 EXCEL 文档来详细描述。

EXCEL 文档第一个 sheet 中是版本信息，每次修改都有变更记录。第二个 sheet 是页面布局。我们通常会用 PPT 或开发工具建立个界面草图，然后贴上去。第三个 sheet 是页面上面的每个字段的说明，如默认值、不可为空、输入约束、主键唯一、输入长度、参照录入等等。第四个 sheet，如果有必要，可以用 VISIO 画出业务流程图。第五个 sheet 是关于运行要求，如易用性、安全性、性能、数据量、并发性，这几个特性都分为高中低三个等级。另外，对运行的操作系统、内存都做了最低要求。一个功能，必须考虑它的用户的计算机水平，否则功能很实用，但就操作不习惯，客户非要改成客户习惯的方法，我们经常会面临这样的要求。与其那样让客户赶着我们，还不如我们自己提前在设计的时候就要求我们自己。我们在需求调研的阶段会调研客户的信息化现状，如 IT 维护人员能力，信息化应用能力，客户老板对信息化认知理解，最终用户信息化操作能力。

我经常看见不少客户没有 IT 维护人员，不知道有服务器这一说法，更不知道什么是 SQLSERVER，也不知道备份。对于信息化的理解就是上套软件，装个 XP 还不知道 WINDOWS 要升级（很多上网的机器连 XP SP2 都不装，什么防火墙放木马的都没有），就

知道装个瑞星杀毒，天天抱怨机器超慢。一看机器，也确实是老了，2002年的机器，128M内存。而且操作者打字和鼠标都不灵光，让他双击或鼠标右键，他根本不理解。跟他电话里说打开某个功能菜单，他很莫名其妙电脑中怎么会有菜单？如果你的软件是基于.net的，他连.net运行时都不知道到哪里去下载。所以我们的软件需要在什么硬件上什么基础软件上运行，数据量多大仍然可以运行流畅，我们的软件要达到的易用操作程度，要达到的易用维护程度等等，我们必须在设计期考虑到。

很多做软件开发的，尤其不注重这方面，所以我在这里重点强调啰嗦一下。大家不要耻笑用户，大家的工资都是他们给咱们的，而不是老板。用户不是计算机高手，他们不懂双击、滚动、鼠标右键很正常。我们的衣食父母，我们要好好对待。我们不要和我们的钱作对。

EXCEL功能设计文档到此，其实还缺一块。就是数据操作流程。

我们先不编写数据操作流程。我们会先交给测试组的人员来校验这个功能点的详细设计，是否和需求流程和需求要求吻合，不符合的地方就整改。

整改后的功能设计文档，算是和需求说明文档一致，设计正确。这时候才涉及到具体的实现说明。

我们会让公共代码人员出界面输入输出和业务流程图中整理出数据结构。我们为什么不业务开发组的组长来整理表结构，就是由于业务开发组的组长熟知业务却对技术并不精通。数据结构很影响未来的性能、扩展，所以应该由公共代码人员来设计表结构，并且建立视图和存储过程。

公共代码人员为了考虑性能和扩展，表结构的设计可能就被打散成多个表，而不是业务开发组长自然理解的存储结构。所以公共代码人员建立了视图，保证在视图的层面上让业务代码开发组的人看到的是一个自然的业务实体数据结构，根本无须理解内部的表结构之间的关联关系。而且有存储过程来保证如何无须知道表之间的关联关系就可以增删改数据。

从这种分工配合来看，我们采取的是从后到前的开发方法。先有数据层，有基础表结构、视图、存储过程，然后基于视图和存储过程进行业务流程代码开发，最终由普通的业务开发人员进行界面拖控件绘制界面。所以业务开发人员既不熟悉业务，也不熟悉技术。

公共代码人员把设计完毕的数据结构交给业务开发组组长，组长来编写每个功能的数据增删改查操作文档。增加这部分文档到EXCEL中，成为第六个sheet。

我没有研究过测试驱动。我一直提倡的是全程测试，从需求到设计到代码到文档到打包，都要经过测试。只有每个环节都能保证正确，结果才会正确。如果到了代码编写完后才发现不对，甚至是需求一开始就理解的不对或有矛盾流程，这就糟糕了。许多人喊需求总变，项目进度无法保证，不仅仅是没有配备需求调研人、业务开发组组长、测试人，更是研发流程方法上有问题，没有采取全程测试。

文档编写完毕一个整块（不是全部的一级功能编写完毕后），我们会交付给业务开发组去开发。具体开发人员任务安排，由业务开发组组长来决定。需要由公共开发人员来开发的，业务开发组组长都会根据自己的开发计划和公共开发人员的任务列表（我们用需求管理系统来安排每个人的开发任务），告知公共开发人员预期的实现完毕时间。

这样，公共开发和业务开发在并行，设计编写和功能开发在并行，设计和设计测试在并行，代码和代码测试在并行（我们采取的是版本管理和每日构建）。这样的情形就跟流水线工作一样，颇有点像丰田的流水线生产，一旦发现某个环节出现问题，理解集中人力来解决，而不会让这个问题的这个人延期钻牛角尖，否则，所有的项目进度管理都成了一句空话。没有了实质的进度，也就没有了实质的项目预算管理。那到底能不能赚钱，真成了一个鬼才知道的问题。

很多朋友在开发当中没有写过文档，一旦有想法要正规化研发管理，就动辄要求全程文档，这文档，那文档，把开发人员累的，最后产品质量和产品进度都没有保证。一看试验

失败，就全盘否定编写文档，再回归到一无所有的状况。真是走两个极端。

希望这篇文章能够给大家以平和的心态看待编写文档。我们并不是为了正规才编写文档，我们写的每一个文档都有作用。我们也在力求能少写就少写，根据团队的、客户的磨合理解共识程度，哪个文档或哪个环节不需要写，我们就砍掉。

我们并不在乎正规不正规，我们也并不在乎我们看上去很美，那没有用。我们只是商业开发，我们要的是可控，在有限的人力资源、合同额、合同周期内交付出客户认可的质量产品（不是高质量，是客户能接受的质量，我们从来不为没有利益回报的东西多付出）。

26、狮面人

好多人都说：你这个方法根本就不是三五个人十来条枪的方法，项目经理，公共代码开发员，测试员，文档员，那得多少人的公司才能配得齐这样的团队。

嗯。其实，我们的团队也不怎么大，我们本身也是一个很典型的中小企业。

一般，都是一个产品或一个项目，由一名业务开发组长、一名主程、一名辅程组成。如果项目简单，基本就是一名业务开发组长和一名主程构成。如果业务组长的开发实力也能和主程相当，而且刻苦努力，不仅协调做的好，需求设计做的好，代码开发也做的好，那么比较中型的项目也是这两个人组成。有几个产品，就会有几个这样配对的开发团队。

研发部其他的人就剩下项目经理、公共代码开发、测试员、文档员这些角色了。

一般，研发部会有一到两名项目经理。我们老承接一些大的合作开发集成项目，老需要有人去客户现场去和其他合作伙伴一起开会，讨论，方案提交、工作量报告、工作进度报告。总需要有人去跑这些项目协调会。另外，销售打单的时候，客户总会提出一些技术性的问题或某个需求能不能做的问题，销售也模棱两可，不知道能做还是不能做，于是总会拉上一名项目经理。关于产品的、技术的、开发周期、工作量估算、项目团队组成的 PPT 和方案由研发部的项目经理来做，关于价格和商务条件上的由销售来做。在打单过程中需要讲解产品或回答客户产品疑问，都让这名项目经理来兼任售前支持。对于项目型的，项目经理有时还担任需求调研经理，使用需求调研方法产出需求说明书。不过，需求调研，有时也是业务开发组长来做。主要看业务开发组长在客户面前的沟通力。因为业务开发组长是开发人员出身，但技术一般，业务知识很熟，管理能力差不多够管理个 1-2 人，工作年限长一些工作经验也多一些，但有些人比较内向，不适合和客户调研沟通，就不做需求调研。所以谁来做，看具体人。不过按职责来，项目经理和业务开发组长都要能做需求调研。

然后就是公共代码开发人员，一般就一个人。对于企业管理软件的开发，框架的开发和维护，公共代码的开发，高难度的问题跟踪，需要高性能的设计，需要高扩展性的设计，需要高稳定的代码，需要高安全的代码，需要高并发的操作，需要复杂代码重构。需要性能优化，不知道的技术 API，都可以寻求这位公共代码开发人员的帮助。他还负责新技术跟踪、新技术介绍、新技术试验。但这个新技术必须是为了改进公司现有产品和现有客户。新技术的跟踪必须上报给技术总监，以防不符合公司目标乱跟踪或跟踪方法和思路不对。对于有利于现有开发的新技术，可以筹备好培训课，由研发部经理安排时间，让公共代码人员给研发部全体人员讲解。如果大家认可这种方法，就需要选择适当的时机在产品中引入。

研发部的测试人员，一般也兼任服务部门技术支持人员。如果有服务部门解决不了的技术问题，可以转给他。而且测试人员还兼任配置人员，在产品打包、产品安装测试、产品

发布、版本分支管理、源代码备份、历史版本归档方面都由他来管理。兼职是有好多好处的。如果他不兼任技术支持，他就不了解客户是怎么使用的，他测试也是瞎测试。如果他不管理产品打包发布，程序员就会自己私自发布版本。可能版本还有问题，为了修补问题，就赶快修改完再打包一个，但版本号却不改变，引起了一个版本号代码不同错误不同，让服务支持起来很莫名其妙。由测试人员控制产品版本发布，能不能发布，就是测试员说了算。测试员感觉质量没有达到，就有权不发布。很多软件作坊，程序员权力很大，一个老哥从头到尾负责整个项目，项目质量如何，全看这位老哥自己的素质和责任心了。为了不让项目质量和特定人密切相关，使公司研发保持连贯性水准，必须做到分工专业，互相配合互相牵制。

一般，研发部也就配 1-2 名测试人员，根据同时并行的项目和产品开发和开发的强度来定。我们并不生产向国际上的产品那样的质量。我们做行业企业管理软件开发，是在客户质量要求、客户签单额、竞争对手质量水准这三者平衡上做到一个质量的认可。我们无法做到微软那样一比一的开发测试人员比例。研发部所有的产品和项目，都由这 1-2 名测试人员负责所有的测试工作，包括编写测试案例，编写测试结果，参与项目的需求测试、设计测试。

对于研发部的文档方面，如文档的正规化，都由文案来负责。项目经理经常要提交给客户一些文档，而项目经理往往是技术出身，文档工作水平不行，于是文档的正规化、美化、文字校对、空格段落措辞标点符号，都由文案制作。帮助文档，也由文案负责。帮助方面，有版本更新说明帮助、安全配置帮助、系统维护管理帮助、基础数据配置与维护帮助、业务功能操作帮助、软件操作演示视频、产品简介 PPT、产品演示版，都由文案来做。为了防止文案不懂产品而写产品帮助，在需求说明书、设计说明书这些文档性的工作上，如果有什么文档体力活之类的工作，也由文案人员来做。文案人员还兼任产品辅助测试，主要是作为一个普通的操作者来测试，在制作演示版的过程中模拟客户流程客户数据来进行操作录入，测试出普通使用中的 BUG。一般，一个专业的测试，经常呆在软件的环境中，思维就有一种定势，但实际的用户并不那样操作，但测试人员自身感不到。而文案人员就能充当普通用户来测试。我们招聘文案人员也没有强调会什么软件，文案写的好就 OK。他们确实是最普通的用户，他们的困惑和操作手法代表了大量的普通用户。而一个研发部，文案人员也往往是 1-2 名，随并行的项目数量和规模来定。

所以说，一个研发部，一名研发部经理，1-2 名开发人员，一名项目经理，一名公共代码开发人员，一名测试，一名文案，也就是 5-6 人完全符合一个软件作坊的人员数量。有时候团队小了，研发部经理就是项目经理，公共代码开发人员就是主程，这样，一个开发团队也就是 3-4 人就 OK 了。但方法照样能用起来。因为我所讲的方法也就是适应于这四套马车的组织架构的。每个人都身兼数职，而且都对自身的提高非常有好处，而不是给他身上堆砌毫不关联的工作内容。每一项职责都是能互相互补的，整体提高他的岗位专业性。

作为业务开发组组长，他很大的一个职责就是开发项目的进度和质量。手下也就 1-2 个人，开发人员的素质也就这么高，我们也会经常遇到突然来了个项目的事情或者突然有个过去的项目问题必须开发人员跟踪，所以开发人员也总是会被左抽右调。对于还能保证开发进度正常，业务开发组长确实每天都在监控进度异常，监控每个开发人员目前身上背着的发展任务和开发进展。每天下午 5 点的时候都要询问一下自己手下这两个人的开发进展。因为有的人不喜欢主动说自己遇到的什么问题，总喜欢自己去到处找答案，弄的延误了正常的开发计划。所以，开发组长必须每天下午 5 点主动问遇到了什么问题，是不是很棘手，能不能保证进度。如果不能保证，业务开发组长就会想办法，是全小组一起诊断出谋划策呢，还是寻求公共代码开发员呢，还是寻求研发部经理呢。为什么是下午 5 点？主要因为 5:30-6:00 就下班了。如果快下班了你才去问，大家早就心思不在这里了，谁都想赶快下班回家，问题就被隔了一夜，留了个不清楚的尾巴。如果在 5 点钟询问，有了问题，如果此问题业务开发组长有经历，他会很快决定该怎么解决。如果详细听完了此问题的来龙去脉，业务开发

组长也无法决定，他已经清楚了问题，他会在晚上思考，第二天一上班来就有了决定。这就一叫一点都不耽误。

我们使用的是 BUG 管理系统来管理开发任务。这并不要紧，谁说 BUG 管理工具只能管理 BUG。我们只需要解决我们的问题而已。当然，我们管理真正 BUG 的系统是另一套，只不过我们使用的是同一个工具而已，我们当然不会把 BUG 和任务混在一起。

来自需求管理系统中的需求，来自 BUG 管理系统的 BUG，都会被业务开发组长来评估，根据自己团队当前每个人的工作量来适时添加、定优先级、分配任务、调度任务。也根据这个任务分配，哪些任务超期了，哪些任务完工了，哪些任务还没有开工，哪些任务正在进行当中，来判别开发人员的开发进度和工作量。

业务开发组长也会每日主动向研发部经理报告进度，简要说明一下现有问题和解决思路。进度列表，当然是从工具中导出的，列明今日关闭的任务，还没有关闭的任务。这样，研发部经理一思考：项目已经开始了这么多天，还有这么多任务没有完成，到期能不能完成，他就会思考是不是要做些调整。

对于项目进度，不管客户是不是需要必须在 XX 月 XX 日之前上线，我们都会设立一个项目最终结束的日期。而不能让项目随波逐流。

这个最终的项目开发结束日期，并不是由业务开发组长排脑门想出来的。

我在以前的文章中就有介绍。业务开发组长负责功能点清单整理、功能优先级划分、详细功能说明书编写。而且每编写一块就开始分配任务给开发人员。

在业务开发组长划分完功能优先级以后，因为每个功能的复杂性都差不多，如果某个功能复杂，就会再被拆分。业务开发组长就能预估出一个大概的项目开发周期。根据以往的团队经验，也能预估出给集中测试时间和集中文档测试打包发布的时间。这样，整个软件什么时候能最终出来，业务开发组长是有个预估的。如果一个团队是新组建的，每个人的能力还不清楚，预估就会有偏差，需要磨合才能得到经验值。如果磨合，我也会在以后讲到。

在实际分配开发人员的时候，就是根据这个总目标完成时间来倒推时间的。这个倒推出来的，有一个每个功能的完成时间周期，而项目经理对于某个特定的开发人员的能力预估也有一个时间，而开发人员自己对完成这个功能自己也有一个预估时间。开发人员怕完不成任务被追究，往往会把完成时间往后放 1/3 时间，甚至有人想偷懒干自己的活，会更多出自己预估时间一倍，也就是说自己觉得 3 天能完成，就说 6 天才能搞定。当然，业务开发组长也不是吃素的。业务开发组长也是开发出身，到底难度多大心里有数。而且业务功能就是业务开发组长设计的，如何实现，会遇到什么难点，自己一清二楚。而且天天管这帮开发人员，谁能力高低谁想偷懒，天天在一个办公室，谁不知道谁。所以，每个任务的时间，都会是业务开发组长在开发人员自己预估的时间基础之上进行调整，获得一个开发人员和业务开发组长都能接受的任务时间段。然后根据每天的进度报告来随时调整这个时间，让开发进度尽量能现实，而不是计划前就定好实际工作中就不能改。

对于项目进度的保证，还必须有一个条件，就是：我们不允许开发人员在客户现场开发，我们更不允许开发人员和业务开发组长不在一起。

开发人员在客户现场，往往开发进度和功能需求变更更容易受客户控制，致使开发团队做的计划和设计都被客户视为扯淡的东西。开发人员不满客户的做法，但在现场又没有办法，只好敷衍答应开发权且应付。本来一个理性的设计，被客户自己自以为是的好做法而推翻。软件什么扩展性啊，兼容性啊，都被扔在了了一边。来客户现场，就要听这个特定客户的，你必须口对口服务这个特定客户，你如果和他讲其他客户怎么办，他才不管呢，反正他付了你的钱，在你眼中他必须是你唯一的客户。（客户和女人一样，都希望是男人眼中唯一的女人，但现实的是，世界上都很多女人，而且很多女人都差不多，都要求她所对应的那个男人必须唯一）另外，开发人员在客户现场开发，就无法实现每日构建每日测试。开发，是个团队配合的事

情。一个软件，并不是开发人员就能全部完成的（许多老板都这认为有开发人员就行）。缺少了测试，质量就无法保证；缺少了文档，产品就是光秃秃的软件。而许多老板还认为测试和文档可以在代码编写完后可以做，真是对软件质量如何保证一无所知。

我们也不允许开发人员和业务开发组长分离。因为在开发当中，设计文档又不是代码，机器运行完就一种结果。每个业务组组长的文档水平有高低，每个人的思考思路也不同。我们经常会遇到一个现象，就是用邮件、MSN 沟通老出误会，而且由于不及时调整，误会越来越大，后来干脆气愤的直接打电话。而打电话呢，有时还不行，你问他理解了么，他说理解了，你根本看不到他的表情，你猜测不到他是真理解了还是假理解了。你以为他理解正确了，他也以为自己理解正确了。你问他进度，他说没有问题。开发出来了，测试人员又有自己的理解，到底这三者理解的是不是一个东西，谁都没个准。只有业务开发组长和团队一起工作在一起，每天能看到实际的软件，能面对面和每个人交流反馈，才不至于代码开发完毕才一看不行。有许多刚当上业务开发组长的朋友，往往和手下搞的很僵硬。手下认为他一天三变，频繁推翻自己的代码，很气愤。而业务开发组组长认为手下的理解能力低，多次讲都讲不清楚，还跟自己顶嘴，还不如自己去开发代码省事。完了，又回到程序员了。

我也同样不允许团队出现多种技术。多个技术，会让团队成本升高，每个人都得会多种技术。而我们做企业管理软件，要想上升赚钱，必须大规模一般员工团队低成本开发，这是我和老板都认同的一种思路。所以，我们必须使用最常用最普通的技术。除非没有办法。我曾经有一个手下，怕自己跳槽没有竞争力，于是老学习流行技术。PHP 火的时候，他就学 PHP。

Ruby 火的时候，他就学 Ruby。如今网游和嵌入、通信、无线很火，他就开始学 C。手机开发火的时候就学 J2ME。而且他还想有实际的开发经验，以在应聘中说自己拿这门技术做过什么。于是他想尽办法在项目中要引入这些技术。说：用 .net，我没法保证性能和稳定性，所以我必须使用 VC++。？？唬谁呢？大家都是开发出身，这个借口未免太可笑。

我也不允许团队使用最新技术。我们只使用最合适的技术。我们不要客户为不需要的新技术而买单。客户的水平只能管理了 SQLSERVER 这样的数据库，我们就决不使用 Oracle。如果客户要求要在 unix 上运行，我们就使用 JAVA 开发。我们谨慎的评价和引入框架，都在核心围绕客户能不能简单维护，我们有没有显著好处，我们面临最棘手的问题能不能很好解决。如果只能解决我们不怎么紧急的问题，如果只能解决我们通过人工或管理就能解决的问题，我们就不引入。一切的一切，都围绕速度、成本、质量在寻找解决方法。

我们也采取了每日构建每日测试，来保证软件的进度和质量。不每日测试，哪到什么时候才测试。到那个时候测试，会不会出现其他什么不可控的问题，我们都说不清。这种对未来的恐惧，让我们需要把风险控制到最小，到天，而且到今天。今天关闭的任务，明天一测试，就知道问题大不大。有问题的，必须由测试人员登记 BUG 系统，并且业务开发组长根据目前情况适时把 BUG 修复当作一项任务来添加。

我们也采用了版本管理工具。版本管理工具不仅可以使我们对源代码差异，找回历史版本，随时打包更新版本，分支每个客户的定制需求，还可以是我们的一个工作的体现。大家经常会听到这样的话：不知道开发部这帮家伙在干吗？是不是故意利用我不懂代码在偷懒。

我们到底在干吗？我们能否证明。有的时候确实是，一个问题三两天都解决不了。我们真的不好意思说我们三天就做了一个功能。我们如果解释这般技术难题为什么为什么，老板更是没有兴趣听，他认为你在嘲弄他不懂开发技术故意拿技术问题来骗他。

我们能如何证明呢？能拿一些大家都能理解的方法来证明自己呢？所以，我想到了文档数量和尺寸大小，想到了 BUG 数量，想到了任务数量，想到了需求数量，想到了开发进度报告，想到了版本发布次数，想到源代码归档容量和源代码行数。

一个项目开发结束，任务数 300 多，BUG 数量 400 多，文档尺寸 70 多 M，项目历次讨论开会纪要 30M，项目历次方案提交 20 多 M，开发进度报告 100 多份，帮助文档 100 多 M。这

些都能量化都能看得见的东西，让老板觉得确实做了许多事。

我们曾经有一个客户，嫌 10 万块钱买了一张光盘，觉得贵死了。说地摊上一张才 5 块，你卖我 10 万。我们于是就把所有的帮助文档都打印了出来，600 多页，装订成 3 本书，印刷好封面交给他。然后把软件装在一个很精美的木制盒子里，客户笑的很开心，把盒子和手册都郑重其事的放在了 IT 部门的柜子最上面。我想，软件是由代码组成的，确实不容易让客户理解其中的艰辛和付出，所以客户并不认可我们的付出。能拿客户可以理解的方式去讲明白就是解决方法。我一贯遵守的原则就是：要么解决问题，要么闭嘴。如果你想不出解决方法还抱怨影响团队情绪，那么滚蛋，这种人就是团队的毒药。

过去，我们讲了业务开发组长如何履行功能设计的职责的一些方法，今天，我们又介绍了业务开发组长在任务分配、软件进度、软件质量保证、工作量化的一些心得，这就是一个开发组组长的份内之事。希望能给被提上开发组长的朋友一些启发，不要把自己定为一个写代码的头儿，那样你不符合国内现状，国内的开发组组长就需要做这些事情。外国怎样怎样那是外国的事情，你也享受不到，这是中国。要么去做，要么老老实实继续当个程序员。

27、沙场秋点兵

前几个月，公司旁边的写字间又搬进一家公司，好似也是做软件的，而且好像是刚成立的公司，程序员们天天加班，神情很年轻，头发很油，穿着大背心大裤衩人字拖，男的女的都有，经常站在通道里成群的吸烟，或者干脆席地而坐围成一堆开会，都不下食堂吃饭，都订盒饭，中午晚上都是如此，我经常晚上下班，看见他们不时有人捧着一个盒饭从办公室走出来把吃完的饭盒扔进垃圾筒，不知道他们会工作多晚。

近来发现，他们开始有人正常下班了。他们也很少聚在一起抽烟开会了。通道里有的只是单个的人在打电话，声音很低，神情普通。偶尔看见他们在外边通道里开会，也是个神情紧张或肃穆或很累或木然或沉默。

我想起了一句话：Boy, Slowly, Slowly。

新的产品，新的梦想，大家都很希望能够把多年积累的知识都用上去，做一个业界一飞冲天的产品。但是，这这个新的团队呀。

我想起了我刚出道的时候。我现在仍然很庆幸我能经历一个产品的从萌芽规划到实现到销售到规模销售的整个过程，让我看到了一个产品是如何成长起来的，在成长过程中会遇到哪些问题，是如何克服的。很多程序员没有这样的机会，往往一出道，就加入了一家运行 N 久的公司，团队是现成的，产品是现成的，自己就是做维护开发而已，没有完整经历一个产品生命周期。所以一旦遇到新开发一个产品的时候，就很茫然，不知道如何下手，而且心情很激动，过去一直维护别人的代码了，很多弊病想推翻重做都不行，现在终于自己可以一展身手了，这回要把自己的想法都加进去，不能留有遗憾，不能重蹈覆辙。于是扑入了自己全部的心血，就连睡觉也不踏实，每日心情澎湃。但时间不长，两个多月，各种问题就都来了。新技术、新团队、新产品，一切都是新的，很多交叉出现的问题让解决异常困难，团队也出现了烦躁，看淡，疲惫的情绪，大家想尽快结束，但开发才到中间，正是关键攻坚要命的时候。

我刚出道的时候，加入了一个新成立的研发中心，才成立四个月，我是第六人。产品还在规划当中，团队还在建设当中，技术还在学习摸索当中。

当时的技术总监做了两件我现在也认为很对的事情：一、他让所有人阅读上一代产品的源代码，整理上一代产品的功能，并且得出上一代产品的功能所映射的客户业务流程。并且要求指出上一代产品业务流程中的漏洞和矛盾。二、他让大家通过改造源代码来学习新技术。他不仅仅让大家整理业务，还要画出来详细的流程图，整理出详细的数据结构。然后安排集中会议让大家讲，讲的时候，以流程图为主，走到哪一步，就进入软件界面讲解，并且讲明背后的数据存取到底是处理了哪些字段标志，表之间是怎么关联的，到底从哪些表中取出来，表结构设计的有什么不合理的。

这次没讲明的，这次提问没有确切答案的，下次继续讲，而不仅仅是讲一回。我发现，很多团队缺乏这种不断追根到底的魄力。总是虎头蛇尾，第一次讲的很起劲，讲的期间出现了自己也没有理解透的东西，领导只是安排下去再去研究一下，但就没有了下文。自己下去了看了一下，认为找到了答案，但就没有再次校验的机会了，于是最应该细究的地方被这样放过了。我们知道，编写程序，往往是最细节差异的地方最容易出现问題，而且很少能被测试到，而且维护代码的程序員往往不清楚这块是干什么的，一旦出现问題很难阅读懂。

技术总监还让我们改造源代码。但这个也是有目标的，就是把控件都修改成一个统一的体验风格，没有 DB 的控件，就去寻找。寻找到了，就把这部分代码摘出来，然后形成我们自己的控件包，力求不要为了一个小小的控件，就安装整个的控件事件包。实在寻找不到，就自己开发控件。

学习新技术，我认为这种方法是最好的一种方法。我现在也是这样指导我的下属的。

学习新技术，为了怕误导，一是阅读官方的例子，如 JAVA 的宠物店或.NET 的宠物店。但我后来我明白了，不能这么学。因为本来就对新技术陌生，一开始入主的就是这样的代码，那么以后的编写程序的风格就往往会像这些例子一样。其实，我们编写业务应用程序，并不需要这样的架构风格，也不需要秀那么多代码模式。照猫画虎把我们画的很累，还扭转不了已经固有的思维。我们不需要这种看上去很美的代码。

学习新技术，第二种方法就是找一个网上开源的什么系统，如某些新技术尝鲜者做了论坛系统或什么什么管理系统。但这种方法也有个弊病，就是他自己写的代码有他自己的风格，而且他还处于尝试期，写这个东西可能是他为了学习这个技术，而非目标是开发这个管理系统。本来咱们对于新技术就是一张白纸，这下被他带到沟里去了。

学习新技术，第三种方法就是阅读这个新技术本身的源代码。可惜本来就对新技术不熟悉，而新技术本身的源代码更是复杂，看的云里雾里，吃力看不到进展，欲想放弃。

所以，学习新技术，最好是学习基于新技术的第三方的外国开源源代码。他们对新技术理解快理解深入，他们应用新技术不是为了尝试新技术或者秀新技术，他们是为了完成他们的一个实用产品。我们当时为了摘出某个控件，几乎阅读了那个控件事件包的所有源代码，并且理出了源代码的结构思路，否则我们无法确定我们遗漏了什么，会不会有 BUG。

但是，现在回过头来看，方法是对的，时机却是错的。

我们是新团队、新产品、新技术，很多我们尚未了解清楚的，我们却把我们学习新技术后得到的控件应用到了我们以后的正式产品中，并且作为基础应用。这给以后的发展带来了几乎是灾难性的打击，我最后费了好大的劲才算把基础重构并且稳定住，否则基础崩盘了，上面的应用就不可收拾了。

所以，我现在都是尽量限制使用新技术。也就是说，不会让新产品、新团队、新技术这三个新都同时出现，风险太大了。而且坚决不使用新技术作为基础技术。

我们还犯了一个错误，就是正式开发的时候，我们一上手就开发基础框架。大家都知道一个系统的基础框架的重要性。但是我们却用刚刚学会的新技术开发我们以后业务模块都要深度依赖的基础框架。我想起了某公司一套战略性的大型产品的开发：用的是新技术 JAVA，大家都还是新团队，做的也是新产品，全体程序员都已经封闭开发了，居然有人提出了一套自

己也未经过商业规模应用验证的基础框架，并且自己实现了一个小 DEMO。大家一看这个小 DEMO 非常有思想，就决定让整套产品线都应用这套基础框架。

我想象他们的痛，和我很神似。所以，我现在如果面临新产品、新团队、老技术，我都不会让大家一上手就开发基础框架。

去年，我就面临了一个老团队、老技术，但是是全新一代产品的开发。

具体情况是这样的，经过几年发展，我们的现有产品渐渐老化，所以决定要开发新一代的产品。上一代的产品是 C/S 结构，而且是适合单客户使用的。这次，我们要开发 B/S 结构，而且是适合集团客户使用的。

让上一代产品的开发团队继续维护上一代产品。让新一代开发由新的开发团队去执行。所以，我们就招聘了新的团队（当时公司对开发新产品有不同的利益冲突团体，还没有达成一致，招聘新团队既有为新一代产品开发做准备的目的，也有其他的目的，造成这支新团队的打造过程中往往出现两种极端：要么好几个人都管，要么三不管）。刚开始并没有去动手设计与开发新一代系统，而是为客户定制开发了一两个其他 IT 项目。所以说还算接触了客户行业，大家彼此在一起工作也快八个月了，算是一个老团队。

但是这支团队对客户、对现有的产品并不深刻了解，虽然我给团队多次讲解过业务，也让大家分析学习现有产品，阅读现有产品的说明书，根据新的业务模式也组织大家一起分析业务设计功能、编写功能设计说明书，但理解上确实还不够令人满意，大部分人还是似懂非懂。遗憾的是，在老板的规划下，新一代的系统开发必须启动。

如果这时候开发基础框架，大家根本不理解以后这个基础框架之上要编写什么样的应用代码，那么这个基础框架就会变成形式，以后的应用代码怎么都觉得这个基础框架没什么用，用起来也是格格不入，不像是螺丝对螺母那样合缝。

所以，我先安排团队开发了系统管理工具。这是个没有具体业务的，而且通用的，而且也是基础框架的一部份的开发工作。但是，由于系统管理工具，涉及到了新的组织结构模式，新的权限控制模式，这是大家不太熟悉的。而且编码架构风格和他们以往的开发不太一样。所以开发起来也是疑问不断，需要实时复查代码，实时解问题。

终于开发完毕，我们开了一个总结会。大家都总结了对这次开发的体会，并且讨论出来以后再遇到这样的问题要如何解决，每个人的分工和人员配合流程再次确定，功能和功能的用意再次给大家讲了一次，对于新的编码架构风格再次讲了一次好处和用意。大家都说：如果在开发前就把这些讲清楚了，那么开发就不会遇到这么多问题了。我说：开发前我就是这么讲的，但是大家都不理解。这次经历过来了，就明白了。

接下来该怎么办？

我说：重新开发一次系统管理工具。时间为期十个工作日。

大家都啊了一声。干吗要重新开发，好不容易开发出来就这么废了？

我对大家说了我的经历，我说：系统管理工具是基础框架的一部份，是以后用户很常用的工具，而这个工具却是在不清晰不熟练的阶段开发的，我们如何能把这么重要的基础功能交付给客户呢？尤其以后这个工具要和需要系统结合，这个工具的数据结构目前还无法支撑以后的众多连接，你们也看到了许多遗憾，我们不能一起步就是个瘸子。

大家又问：那过去的代码我们还能用么？

我说：你们自己看需要。我既不赞同你们尽量使用过去代码，也不赞同全部重新编码。如果你们想把一个有残疾的代码上改造成一个优秀的代码，我说不可能，过去的很多缺陷会牵绊着你。你为了保留过去的代码，你就会向过去妥协，而把丝丝缺陷带了进来。每个人每个功能都留了一点点小尾巴，那么所有开发的功能总和出来的缺陷就是一个大问题。所以大家自己看着办，先重新写，需要用的时候自己就 COPY 出来。

果然，理解了清晰的功能和功能用意的程序员，开发起来很快，毕竟都写过一次系统管理工

具了，也是老技术、老团队了。半个月完成工作。

我问：上一次你们编写的系统管理工具代码用了多少？

他们回答说：不到 20%。明白了思路，重写起来很快，反正没有什么高难度技术。本来想 COPY 旧代码，发现老有关联，搞不干净，还不如重新写一个来的快。

我说：好。咱们下一步就实现一个咱们系统中最简单的也是比较边缘的一个业务子系统。在开发中大家重点发现需要什么公共功能，咱们都提炼出来，就会形成咱们的基础框架的一部份。

这个简单的业务子系统开发了出来，我又开了总结会。

大家这次都发言热烈：我现在终于发现了系统管理工具和业务子系统之间的关联关系了，他们有很多代码能够共用。由于这次开发业务简单，而且经过上次系统管理工具的开发，开发方法和代码方法都已经熟悉，对于系统管理工具的认识也深刻了许多，所以这次我开发业务系统的时候，还顺便把过去的系统管理工具的代码进行了重构，发现了不少可以共用的部分。我发现，这些基础代码总结了出来之后，好像系统管理工具也是从这些公共代码基础之上开发出来的一个特殊的业务子系统了，所有子系统都依赖这个基础代码框架。过去本来系统管理工具的风格和业务子系统不一致，这次重构，一下子都统一了。

我笑的很开心，似乎我过去的心结终于可以解开了。

28、代码那些事儿

这个是讲软件开发过程管理的系列，目标人群是那些身处研发管理位置或想成为研发管理的人。所以我不希望这个系列中出现代码，也不希望出现和某种技术密切相关的代码技巧。但是没有办法，软件开发过程管理，有个很重要的一环，就是软件代码编写。不编写代码，说的天花乱坠，管理做的再扭，也成为不了软件。

最近 blog 好友到了上限，所以无法加入好友了。于是就加入了 N 多 QQ 网友。大多数刚出道一两年，还有不少在校学生，希望认识并聊聊，要和我聊设计模式、OO、SOA，还有人建议我去看看 OO 和 UML 的书籍。

我确实没有阅读过 OO 的书籍，我不是一个死钻牛角尖的人。我只是有什么问题，就去找解决问题的方法，能解决我的问题就 OK，而不在乎我用的正不正宗，也不在乎我用的是不是 OO。可能它是 OO 的外壳，但是它实质上可能是伪 OO，我也不想去深究和区分什么是正宗 OO，什么是伪 OO。反正能解决我的问题就行。

我首先遇上的问题是代码大流水的问题。一个代码 3000 多行，数不清的 if..else。我实在无法分析出这个代码到底实现了什么功能，它复杂的就像一把瑞士军刀，如果军刀里面能出来一支小手电也觉得不奇怪。这段代码我想它就是这样。我想写这段代码的人一定是在学校做惯了《图书馆管理系统》，一个程序文件中实现了所有的功能。

所以，经历了痛苦，我就一直要求手下的开发人员，一个函数代码行数不能超过一个屏幕。如果超过一个屏幕，就需要来回滚动。一滚动，就容易把思路滚动的走神，乱了，还得回到上面去重新理逻辑。

这种规定，直接导致了一个后果，函数太多了。但是函数间是有关系的。有时候突然需要增加一个参数，但是此函数已经被其他代码调用了，于是为了省事不改动其他代码（当然，他可能已经忘了再哪些地方都调用到了），他就定义了一个全局变量，在函数之间传。

新的问题有出来了，全局变量的问题。代码在他一个人手里还好说。被维护了好几手，新手

对日益复杂的代码已经不太容易理出头绪，但修改任务是有时间限制的，只能在现有理解水平上工作。这位新手看到这个全局变量的命名，感觉是自己能用的到，就用了，还给赋了值。意向不到的事情发生了，数据库中进了错误的数据，怎么来的，不知道，看代码，处理的都正确，代码没有问题，也不知道怎么出现的。

于是禁止大家使用全局变量，能藏到多低的可见级别就藏的多深。但是仍然问题依旧。

于是，我们就把这个全局变量封装成属性，给它赋予读和写的函数。不管有多少个地方调用了它，有谁给它乱赋值，这里掐源头，都做好日志和异常保护校验。

把所有函数，能变成私有的就变成私有的，需要公共供其他人调用的谨慎的放出来。放出来的函数，一定要做好参数的校验，什么空指针，什么没有初始值，什么非法参数值的，尤其是临界值上下边界，都在门口就挡住，根本不往下执行进行业务处理，否则走的越远引起的问题越大。

而且要求每个函数都要做好自己的内存保护工作，自己函数内创建的就要在函数内释放。每个函数要做好异常处理和日志记录。

这样，一个样子像 OO 的类产生了。它可能只遵守了 OO 所提了封装，它却没有实现 OO 所提的继承和多态。所以它可能是个伪 OO，但它确实解决了我们的问题。

封装了类之后，又一个问题产生了。几个类之间不知怎么的，总是你中有我，我中有你，互相调用了。就好像左右手互搏，弄不好就会把自己给捆起来。

为了堵住这个问题，规定只能单向依赖，如果发现你必须调用另一个类，另一个类也需要调用这个类，就以一个类为主，另一个类开放事件。这也许就是控制反转，AOP 的需求吧（我并没有深入研究 AOP，我只记得控制反转这个词，好像是为了解决相互依赖关系问题）。

“Don't call us, we'll call you”。

但是代码稳定性的问题仍然没有很好解决，测试组也找出了许多 BUG，但是一到客户那里运行，还是出了不少 BUG。怎么自己运行的时候找不到呢？

于是，我们在版本测试的时候、第一个版本小规模放给典型客户的时候，都加了断言。一旦软件出现问题，就立即记录日志，并进行软件中断，而不让错误继续错乱的不按我们预想的代码流程走下去。很多年前，我就惊讶于某公司的软件的质量，怎么折腾操作都没有问题，我常常给我的手下拿这个例子来反衬大家的代码质量。直到我有时随便乱点，居然软件中断退出了，报了一个错误号，我一下子想通了，它用了断言。断言阻止了错误的继续扩散，不让恶果之鞭长袖善舞。于是，我要求开发人员经常性使用断言，很多过去悄然发生的错误，测试员只运行可执行程序无法捕捉到，现在都能明确的捕捉到，在测试阶段就尽可能的消灭了那些过去无法明示的 BUG。

我过去领导过架构组。架构组的人在 2002 年的时候，疯狂迷上了 UML 和设计模式，人手一本《COM 本质论》和《设计模式》。我手下有一个新手，就处处是类，处处是抽象，处处是封装，处处是分离，尽量使代码高内聚低耦合。但是这样的代码太麻烦，他花费了大量的时间，他看自己的代码赏心悦目，别人看他的代码云里雾里，不阅读懂《设计模式》就按照常规理解业务的思路去阅读他的代码根本阅读不懂，不知道他为什么这样写代码，怪异的很。本来，这位想达到可维护性，可阅读性，却真正的失去了可维护性、可阅读性。这和我前几天看我的朋友周爱民写的《大道至简》中写到：有人希望拿 UML 去统一用户和软件设计者。殊不知 UML 有多难理解，而 UML 设计者却认为 UML 可以描述一切。就这个道理，要理解你的代码还要去读懂《设计模式》，这要求太高了吧。

所幸这位新手自己都每次写的累，慢慢的也就懒了，觉得确实需要分离的时候就分离，觉得没什么必要的就懒得做了。用他自嘲的话说就是：被磨平了。其实，依我看，他现在这个代码状态才是刚刚好，即照顾了设计扩展，又照顾了实用。真正的纯 OO，纯设计模式，可能只存在于教学和科学，而不在于我们的商业软件开发。我们作为商业开发，强调的是叫座的

基础上叫好，所以折中方案是必须的，客户和我们自己两相宜就 OK，是否符合正宗，就不在我们的商业开发管理范畴了。

这位新手还写了大量的注释。在每个源代码文件头都写上几月几号，XX 创建的，这个源代码文件主要是干什么的，还画蛇添足的写上版权所有，公司名称。好像这个代码要开源，或者可能会被其他公司窃取了好表明公司版权。甚至每个函数都写了注释，每个参数是什么意思，每个参数可能出现的值代表什么意思，都写的一清二楚。久而久之，也懒的维护了。代码改动了，参数扩展了，参数状态值有了变化，注释说明却没有跟着改动，让后来看代码的人老误解，还不如不写这些注释。

我告诉他：做事不能走极端。要么全写注释，要么不写注释，都是不对的。我只在我认为要小心地方，或者我自己都觉得很难理解的地方我才写注释。否则，我自己都可能会过段时间理解错了。如果某段代码我看看就能看懂，我就不写注释了。咱们做企业管理软件，深入技术又没有，只要代码能把复杂的业务处理描述的逻辑思路清晰就 OK。虽然说理解能力不同，我能快速理解的未必有新手能够理解，但是你看看我的代码你就明白了。

这位新手去看我的代码去了。我的代码几乎没有什么难度技术，但代码也是看上去很舒服。他发现了以下几个关键点：1 代码风格统一，从命名含义，到大小写，到缩进，都一致。每个源代码文件都一致，确实出于一人之手。许多程序员，光自己的代码就有好几种风格，有时心情好，有时心情不好，有时头脑清醒，有时没有休息好，有时敷衍，有时画蛇添足，有时急躁，从代码就能看出来。而我的代码就像稳定运行每天如一日的机器，好似每个源代码都是在同一天敲的。这就叫发挥稳定。这几天要开奥运会了，运动员天天重复练同一个动作，把每个环节都练的精益求精，其目的就是为了在大赛紧张的压力下也能发挥稳定。人在压力下，非常容易发挥失常。如果人老处于这种压力下训练，那么大赛就像平常一样了。

2 我的代码居然能看出业务流程。函数数量均衡，不像他的代码函数太多，跟踪跳转的很累，也乱了头绪。函数长度也正好在可理解阅读范围内。而且有一个流程控制函数，把流程处理环节串了起来。细深入跟踪某一环节，又发现了更细的流程。每个函数都看起来简单，但整体来看，却实现了复杂的功能。他问我是怎么做到的？我说，我的心中只有业务，业务和代码，我认为只是英语和汉语的区别，表达的是同一个思路。而在你心中，业务是 DOC 上的文字，代码是你的技术表现，你老需要把业务和代码映射拧在一起，我则不需要。业务流程如何，我的代码流程就是如何。

3 由于我的程序都是小函数组成的，都有明确报错，所以错误很容易找到，即使出错，也扩散不大，都是小 bug，对系统整体没有大影响。

他还对我的开发方法不理解，问我为什么要让大家从后台往前台开发，他很不习惯这种方法。他过去开发都是先用开发工具拖拽控件画出界面，然后一个按钮一个按钮的处理业务代码，需要什么字段就在数据库里设计什么表加什么字段。

我问他：你为什么要这样做？

他说：我不理解需求，无法凭空想象，只能先画出来界面，然后有了直接感觉后就在开发中理解业务，边开发边理解。

我问他：那你以前有详细设计说明书没有？

他说没有。

我又问他：那你以前有人单独设计数据库和开发业务处理中间件组件么？

他说没有。

我说，问题就在于此。你没有详细设计说明书，所以你看不到一个形象的东西，而咱们现在至少有 PPT 画的业务界面，也有输入要求说明，也有数据增删改查说明，也有业务描述说明。而且数据库，一个中大型应用，性能、稳定性、可扩展性，都在于数据库的设计和中间

件的设计，如果每一个程序员都要从数据库设计到中间件组件开发到前端客户端开发，那么要想保证这个软件的统一整体质量，那有多难。每个程序员需要懂得多少的技术知识才能达到统一的质量要求。所以，让不同技术高度的人做不同难度的事情，把重要的事情掌握在高素质的人的手中，这样质量就不会跑偏到哪里去。企业管理软件，不外乎是数据的增删改查。数据库的视图和存储过程，已经屏蔽了复杂的表之间的关系，提供了统一的业务实体视图和业务实体的增删改操作。这样中间件组件就容易处理业务实体间的流程，到了客户端，就只剩下数据的输入和输出了，真正成了终端。

我还经常进行代码复查工作。发现有人代码出现坏迹象，我就让他整改重构自己的代码。否则，定了规范，光喊口号让大家遵守规范又不检查又不惩罚，谁爱遵守规范？

在代码复查的时候，我经常能发现思路局限、想法绕弯、实现缺乏扩展性可变性、缺乏优化、判断有遗漏有风险的代码。我都一一给他们指出，并且告诉更好的方法应该如何如何。

很多网友都问到如何培养开发人员、培训开发人员、提高开发人员技术。我过去所在的公司是每个周五都不定期召开技术培训会。但是慢慢的也流于形式了。我个人感觉，技术培训会是个好事，但最终达到的效果可能是加强了团队的凝聚力和和谐力，有利于团队建设，但是技术提高，并没有给开发人员在实际工作中有应用。因为大家听了也就听了，会一散实际工作中仍然照旧自己的思维习惯，很难落实下来。

通过代码复查，这就是真真实实的代码，这就是大家最实在的工作成果和天天接触的东西，很有真实性，不脱离，有针对性。通过这种方法，达到了大家对自己日常写代码的技术技能的提高。这就是我对程序员的培训方法。

管理是什么？有网友老跟我讨论企业管理、企业文化、盈利模式、执行力、忠诚度。我不是很喜欢这些空洞抽象的主题，讨论完了也无法解决实际细节问题。而企业的运作，恰恰在于每一步的成功，最终汇合成最终目标的实现，没有脚下的每一步前进，再好的设想也是空想。管理，就在细节当中。

代码高手，也在于细节当中。

质量、进度、成本、目标、折中，这就是核心。写代码，做管理，道理一样。

29、风语者

我们公司开始也是没有测试的。

公司是创业公司，我空降的时候连卖什么东西都不确定。公司的创业仅仅出于老板对原公司的一口气，觉得还不如自己单干快活，公司要成为什么公司，可能想法很多。咨询？培训？认证？通信？数据挖掘？我从老板的书架上找到过这些书。但是真正卖什么？卖这不好卖，卖那不好卖。误打误撞就进了软件这一行当，但软件这行是否可以持续走，是否要持续走，老板还不确定，如果卖的不好就不做软件了，改做别的，现在是生存阶段，就顾不了许多了，有项目就接上。上面有老板关系搞定，下面有老实的干活人努力加班，项目也就过得去。没想到，软件这条路，居然走下去了，还接了一个大活儿，安装的点很多，涉及的用户很多，从海南到新疆，从深圳到山东枣庄。

公司发动了所有实施顾问来测试，只有他们通过，才能去实施。

实施顾问大多来自刚刚毕业的应届毕业生，对企业管理，对软件，对行业领域，都一无所知。对测试更是一窍不通。

测试并没有分工，每个人都测试软件。也没有什么测试方法，也没有什么测试计划，也不知

道该测什么。反正也是对软件不了解，就当是深入学习软件吧。

开始并没有测试报告，大家发现问题，就用电话或 QQ 或邮件，把问题发给开发人员。谁认识那个开发人员，就发给那个开发人员，如果不认识一个开发人员，就发给老板了。报告中尽是不好用，不能用的词汇。但什么功能不好用，是怎么操作导致不好用，不好用的具体表现是什么，都没有。

老板急眼了，怎么这么多问题。

我说有五个原因： 1 很多问题都是每个人都反映了，其实只有一个问题，只不过大家没有分工，都测试，于是都报告 2 不少人见一个问题发一个邮件，所以看起来很多。 3 有的人测试只是随便乱点乱输入，咱们软件还没有做这种破坏性操作兼容防范。 4 不少人不了解功能，不了解行业，不了解业务，本来是对的，按他的理解是错的 5 有些人偷懒，今天发的是这些问题反馈，后天又是同样

我说，基于现状，我给大家一个测试方法： 1 分工测，几个人测试一块功能 2 不全部测，只测试那些很常用的重点功能 3 不要电话、QQ、邮件来报告给单独的开发人员，给我一个人发就可以了，我来判断衡量安排。也不要随时报告。每天下班的时候来统一发送，由各个测试小组的负责人来汇总自己组内的测试，并且把重复的问题合并掉 4 每个测试小组的每天的测试报告要连续在一起，不要今天发今天的测试 EXCEL，明天是明天的测试 EXCEL，这样没有连贯性 5 每个问题，要标好功能模块，有测试人，有测试版本号，有测试时间，有测试操作过程，有测试输入数据，有报错截图 6 先测试正常的数据输入，正常的操作流程，是否能全部流程走通，是否数据保存正常，是否保存后的数据还能正确的取出来。那些临界条件测试先不要做。对于功能不易操作、界面不好看、起的窗口标题是否得当，字体是否加粗这些需求不要提。咱们目前阶段的重点是测试问题，不要把需求和找问题混在一起。方法执行下去，问题少了许多。

前几天大家还在抱怨这样的软件简直是烂软件，让他们拿给客户，肯定会被客户打死。

现在呢，才几天功夫，实施顾问已经觉得可以去实施了。

这就是有方法和没方法的区别。

第一批客户的实施终于启动了，实施顾问奔向了全国。

到了真实的客户那里，才发现自己的测试，自己对软件，对业务的理解是多么的肤浅。过去发现的问题原来都是小儿科，真正复杂的问题根本没有测试到。给客户一讲解，客户一问，发现原来很多功能细节没有理解，不知道怎么给客户解释。于是纷纷打电话回来问。而能回答的人只有我，我成了接线生。

我当然不能成为接线生。我一方面仍然要求他们按照过去的测试问题报告流程和方法来报告实施现场中发现的问题，另一方面我自己写了 FAQ 给实施顾问发出去。但是实施顾问仍然问，一个问题重复的问。我说你看 FAQ 的第 XX 行。他说他看了，但没看明白（其实是对客户业务不了解，所以也不明白功能）。我就给他再解释。经过多次解释，我也了解了实施顾问的理解思路和理解层次，于是不断修正 FAQ，使 FAQ1.0、FAQ1.1.1 这样不断发布，几乎天天发布。我现在回过头来想，帮助文件写的好不好，不能你说你自己已经写的很明白了傻瓜才看不懂，不要这样认为，这样根本不解决问题。唯一的方法就是用户理解能力有多低，你就要把帮助写的有多低，让他理解是目的，要不你还能怎样呢，就这样的人，问题还得解决。

随着项目的实施，公司渐渐拢回来不少钱，但是面临了一个瓶颈，这个大项目快做完了，以后有什么活能养活现在这么多人呢。所以，最好的做法就是把现在这个项目产生的软件改改，变成一个产品，卖给其他的客户，卖的越多越好。但是，其他客户我们有关系的并不多，所以要想销售给其他客户，必须拿产品说话。于是，研发部陆续加入了专职的测试人员、文案人员、美工人员，旨在提高产品的质量和包装，希望能卖个好价格。

所以说，专职的测试人员是这么来的。

很多软件公司没有测试人员，其原因就是老板搞定关系，程序员老实干活，项目质量虽然不行，但也能将就把钱结了。既然能赚钱，干嘛要测试人员呢。除非由于质量问题，签不到单子。除非由于质量问题，客户不验收不给尾款。除非公司所有人都测试还是无法达到客户满意的质量。只有这样，才会招聘专职的专业的测试人员。

测试人员一来了，开始工作。但怎么开展测试呢？文档在哪里？

文档只有很老的设计文档，现在软件和文档已经毫无关系。为什么？原因有二：

1 都是程序员，谁来专门写文档。为了公司生存，我身兼数职，到处开会做项目经理或做售前，还管开发人员，还有实施人员给我打电话问软件中某个功能怎么回事，我也分身无术。
2 都是根据实施人员、客户、销售人员、老板反映的需求和 BUG 修改。那些 BUG 和需求 EXCEL 表格倒是有，但没法作为测试案例编写的根据。

测试人员硬着头皮，开始学习软件。

帮助在哪里？没有？

对，没有，因为没有写帮助文件的人。只有打单的时候讲解的 PPT。

测试员晕倒。

晕完继续学习软件，什么是正确的什么是不正确的，测试人员也不知道，当然也不知道 BUG 究竟是什么样。软件质量仍然没有改进。

老板问：这个测试人员是不是没啥能力？要不要裁掉？

我说：不是他能力不行，而是咱们过去为了生存欠了太多东西。我们这会在补过去的课。现在的文案人员正在补帮助。有了帮助，就有了什么是正确的标准。但现在的问题是，文案人员也不了解软件，她写出来的也是自己猜测，所以我已经分出来一个开发人员做项目经理，他目前专门负责把帮助文档建立起来，但是他开发人员出身不擅长写文档，但他熟知软件，所以只有他们两个人搭配才能搞定。但这种磨合，需要时间。

就这样，一边测试人员瞎学习瞎测试，一边项目经理和文案人员不断讲解不断编写不断审核不断修改。

测试人员终于可以编写测试案例了。但他对软件也是初步了解。由于几年发展，软件加入了大量客户的需求，很多细节的东西在帮助中也没有看到，测试人员也不知道有这个功能。所以测试来测试去，其测试结果和实施人员的测试没多大区别，都还是在门外转。

老板又开始沉不住气了，旁敲侧击想裁掉测试人员，觉得他的存在没多大意义，还是实施人员测试好。但是由于专职测试员的招聘是我提出来的，也是我的直接手下，而且这个测试人员也老实，干活勤勤恳恳，老板实在找不出什么把柄把这位开掉。

我说：他来自著名的外包公司，专职做测试，我相信他是专业的。只不过咱们过去缺的太多，所以他想测试，也是巧妇难为无米之炊。咱们可以继续看看。

果不其然，测试人员有其独到的软件测试方法、软件理解方法。很快，测试人员对软件的理解不亚于那些多年的实施顾问，也不亚于程序员。找问题也越来越准确，越来越深入。

当然，其原因也在于这个团队的成长，有专职的项目经理开始书写现有功能需求修改的设计文档。过去的，没有的，就让它过去，就让它缺失吧，但未来，不要成为过去。现在也有专职的文案，不断在修改帮助，加深了许多。测试人员现在比文案人员理解功能更细，更深入，经常提醒文案人员应该把某句话写进帮助中，否则容易被用户忽略，是个不小心就会绊倒的坑。

为了使测试人员更快速的了解客户应用操作方法，更细节的了解个性化的功能，我让测试人员也兼任研发部的技术支持。有服务部的小姑娘无法解决的问题，转到你这里解决。否则，在过去，服务部小姑娘老把电话转给开发人员，本来就几条枪，被客户电话吵的无法安心开发。而且客户发现开发人员接听电话处理问题更有效，所以很多客户都是直接给开发人员打

电话，服务部成了虚架子，而开发人员的开发进度被拖累，叫苦不迭。现在有了测试人员兼任技术支持，这下解放了开发人员。开发的质量和速度提高许多。

但测试人员并没有做技术支持的经验，过了段时间就来和我诉苦，说现在服务部小姑娘啥也不干，都直接把电话转到他这里来，所以他现在已经无法测试了，成了专职的服务支持人员。如果再这样下去，软件质量无法保证，以后的技术支持压力更重，开发部就会成为开发+服务部门。

我给测试人员出了三个方法： 1 经常遇到的问题，就做成 FAQ。下一次还有小姑娘问，直接让她看 FAQ，拒不回答。 2 交给他们方法和思路，不替他们亲自做。亲自看着她，让她服务支持客户。一次不会，再继续这样做第二次，必须让她自己亲自会了。 3 每个星期六定期培训，疑问解答。并且考试。如有讲过后考试还不会者，扣钱。

另外，我也对服务部下了一个考核（当时我已经统管的服务部）：你接待了多少客户问题，解决时间多长，多少个问题转给开发部技术支持了，这些问题的难度级别多高。根据这些指标来衡量服务部小姑娘们的技术解决问题能力。能力差的就辞退。

这几招后，服务部的技术支持能力蹭蹭的提高了。真是没有鞭子不干活。测试人员兼任技术支持越做越轻松了。

我还把版本管理、打包发布交给了测试人员。起源在于有时候客户报告了某个 BUG，程序员一看好改就直接改掉了，改完后就直接联系客户更新了，但是并没有更改软件版本号，也没有做新的打包。于是出现了同一个版本号软件功能表现却不同。而且，由于项目组多了，每个项目组组长都各有各的原因，有时候自己就打了一个包给了客户，随便定个版本号，起的都稀奇古怪，有的叫 beta 版，有的叫 6.0.20050203。这种情形导致了测试人员做测试的时候，开发人员说改了，测试人员说没改。开发人员说已经没有问题了，测试人员说我这里还能重复出来。于是两个人一起查，耗费了两天时间，才查出来测试人员手里的和开发人员手里的不一致。

我又下了几招： 1 开发人员绝对不能接触客户，不能接听客户电话，也不能解决客户问题，更不能给客户更新 2 开发人员不能没有任务分配和设计文档就擅自修改软件，否则记过处分 3 大家一致使用版本管理工具、BUG 管理工具、需求管理工具、任务管理工具。用工具把项目经理、开发人员、测试人员、文案人员绑定在一起，按固化流程推进流转。 4 打包发布统一交给测试人员来做，测试人员来控制是否可以发布，发布的版本号的命名。质量达不到，有权不能发布。

现在，我们的测试已经能做边界测试、版本兼容性测试、系统兼容性测试、压力测试、安全测试、集成测试、破坏性测试。也已经在项目中应用全程测试，测试人员主要参与需求验证、设计验证、代码验证、文档验证、打包验证。

但是，我们现在还没有实现单元测试，开发人员就这些人，项目却多。而且测试人员没有编程能力。我们也没有做更多的回归测试，毕竟测试人员数量配备太少，而项目并行太多。

看机会吧。老板越从软件上赚钱，他才会越舍得投入软件。成本永远嫌多，利润永远嫌少。如果你是一名开发主管，你的老板还没有从你负责的软件中赚钱，而且是很快乐的很大规模的赚钱，而不是他靠他的人际关系和送礼吃饭支撑着，我想，他不会给你一毛钱的。你抱怨也没有用，因为你没有价值，所以投入也是没有意义。

先去证明你的价值吧。

30、蛋白质女孩

我想给大家分享一下关于文案人员，我一下子想起了王文华写过的一篇小说《蛋白质女孩》，因为我们要找的文案

就是这个样子。

“她日月座是狮子和双鱼，同时会讲日文和法语。她早起，起床后先跑半小时，吃了麦片才去公司。她贤慧，每天

做一打火腿三明治，带到公司请同事们吃。她有礼，快递脸上有雨时递上面纸，清洁妇来吸地时抬起椅子。她准时

，和你约会前一天打电话确认，第二天寄卡片谢谢你点的果汁。她纯情，爱像宋词唐诗，意境优美对仗工整；个性

像阿拉伯文，她知道它的存在却不懂是什么意思。她善良，生理时期还抱起大水桶换饮水器的水，没人注意时还认

真做垃圾分类。她负责，影印机塞纸时修理到底，洗完便当后水池一定清理干净。她有礼，咀嚼食物时嘴巴从不张

开，交叉的双腿一定用裙子盖起来。她浪漫，最喜欢的电影是' 第凡内早餐'，失恋后可以好几天不吃饭。她坚强

，撞见我在办公桌上和工读生亲密，她还蹲下来为我捡起地上的笔。”

这就是蛋白质女孩。

如果你要招一个文案，你一定要识别好，否则你很容易招来一个会打字会写诗的女员工，连个秘书的细心、机灵都

没有，你和她说三遍，她只是一个劲的点头，但她还是不明白。写出来的东西，重要的关键的一句没说到，写的没

有一点企业味道，无条理无格式，像是散文。

文案是有了，但对产品一窍不通。她有看不懂设计文档，她又不了解客户行业，她又没有做过实施，客户现实中怎

么应用怎么操作有哪些疑问她也不知道，怎么写？能看不能用的文档我见多了，一篇篇帮助写的很是精美，但真遇

到问题，啥也帮助不了。

为了文案能快速提升她对产品的理解深度，我煞费苦心。

首先让她写产品简介。算是让她对产品有一个框架性的认识，对软件的流程，亮点，实用性

这些关键点进行一个把

握。但写完后，她自己都感觉像白开水，她说我没有感觉到咱们软件的好处，也说不出软件有什么亮点，而且亮

点也不亮，说的勉强。

汗！第一次指导失败。

我又想了第二招。让她兼任测试员。由于她不了解产品，也不懂技术，正好可以以一个普通用户的心态和操作方法

去测试，而且测试的过程中可以深入学习功能，一举两得。

结果是：帮助写完了，让实施人员和服务人员一校验，说没有价值，写的这些现在的客户都会，即使新客户，实施

人员培训完了，都能掌握。我们要的是实施人员实施完毕不在现场的时候，客户不会操作时，不需要打咱们服务部

电话，自己就能知道怎么做。

汗！第二次指导失败。

我又想了第三招。让她给研发部内部全体人员讲，让大家给她提问并补充她的知识。

结果是：小姑娘准备的很紧张，讲的也很紧张，缺乏结构、条理、重心，丢三落四。大家还是很耐心的听完了她的

讲解，并给她提了一些建议。我希望她根据大家的反馈再准备讲一次。但是过了两天，小姑娘跟我怯生生的说：我

想跟你辞职。我真的很笨，可能我不适合做软件文案。我吃不下饭，睡不好觉，昨天哭了一晚上，我知道您是一位

好领导，是为了我好，但我能力不行，我辜负了您的培养。

？？压力太大，把小姑娘吓跑了。看来老革命遇到了新问题，我不仅不会培养文案，而且不会调节小姑娘。

我赶快向小姑娘道歉，表示我拔苗助长，心太急。我看好她的潜力，我希望她能留下来。我说我可以把你调到服务

部去做支持，可能你在那里会成长的更好。

小姑娘留了下来，去了服务部。

一个月后，我和服务部经理吃饭，我询问这个小姑娘的提升如何？

他笑着说：你是不是想让她回去？我早就看出来你是想把人到服务部来锻炼一把就走。可惜，我不放人了。你们这

个女孩，心细、老实、有耐心、有韧性，是做服务支持的好苗子啊。

不过，小女孩还是回到了开发部。

我和她谈心：你希望回到开发部做的第一件事是什么？

她说：把 FAQ 融入到帮助中。

我问她：到服务部一个月，你觉得你最大的提升是什么？

她回答：客户感觉。我过去写东西言之无物，下笔无神，就是找不到这种感觉。虽然我仍然没有见过真正的客户，

没有到过真正的现场，没有看到他们真正的操作，但是从他们的电话中，和给他们做支持的过程中，我理解了他们

的思考问题的思路，他们的应用水平，以及他们也是人，活生生的人，在 QQ 中和电话中也挺逗，不是我过去空洞洞

的给钱买单的那个形象。

我理解了她说我的话，我想，她找到的这种感觉和我们做软件设计时候的拟人法，面向客户场景的设计方法是一致的

。我们也是通过模拟客户场景来达到对客户真实的感觉。

果然，写出来的帮助有料了许多。

但好像还缺少些什么：有一步步的操作截图，有一步步的操作说明，有特殊业务的特殊说明，有 FAQ，有重点注意地

方的重点标注，内容很丰富，但还是缺东西。

我想了想设计文档：有界面草图，有业务流程，有数据流程，有输入要求，有输出要求。

对，缺的是输入要求说明和输出要求说明。

文案按我的要求，把界面上的每个字段都做了解释，形成一张表格，每个字段一行说明。有这个字段的含义，有什

么样的作用，是否为必填，是否有默认值，输入的约束是什么，什么时候是只读的，可能会有几种状态值，如果是

下拉选择的参照录入，参照字典是哪个基础数据维护功能可以维护。

对于输出报表，报表每个字段的含义，报表每个字段的数据来源，字段与字段之间的钩稽关系是什么，此报表字段

还与其他什么报表的什么字段关联。

正好实施部门有一次客户实施，我也安排了文案作为辅助实施人员去跟随实施，到客户现场真正锻炼见识一下。

回来后，小姑娘写起文档信心百倍，下笔有料。她已经成为了一名合格的文案人员。就连销售部门都想要走她，销

售经理一个劲和我说：这个小姑娘心细、踏实、产品也了解，也做过实施到过客户现场，也做过服务支持，文档也

做的漂亮，我们销售部给客户做方案正需要这样的人。

当然，我就不会放人了。但是我也松了个口：如果销售部需要什么文案，可以让她来帮忙做。

她人在开发部是好事

，可以持续深入的理解产品发展。如果到了销售部，她可能就缺乏了这种对产品了解的基础，她又不适合做销售，

那么她以后就会慢慢既不懂产品又不懂销售，成了销售部的废人了。

果然，销售部很多文案，都转来让她帮忙。

我过去最头疼的就是开发和销售的结合。开发和销售老打架扯皮，销售部嫌开发部开发的产品无法销售，开发部嫌

销售部整天好吃好喝好玩什么产品也不懂大嘴一张只懂吹牛，留下了一屁股屎让开发部擦。现在，我有项目经理充当售前，堵住了销售部的吹牛大嘴。我有文案人员充当客户方案编写，让过去销售部夸大空

洞的解决方案有了实质产品感觉。

为了打单方便，文案还制作了演示版，里面有模拟真实客户业务的数据，可以达到完整的演示全部流程和报表表现

。

但是，我还有个问题要交给她。

我发现这样一个现象：不管是实施部门、服务部门、销售部门，对产品的了解还只是停留在产品最初的几个版本认

知基础上。现在软件已经升级了很多版本，加入了很多细节功能，也修补了很多 Bug，但大家都不知道。仍然在按照

自己过去的理解讲。虽然每次发布新版本，都有更新变动说明，都有最新的操作说明，也有操作视频，但没有人看

。这和我们在网上遇到的很多开发人员一样，网上大喊：冰天雪地 360 度空翻裸体在线跪求，非得让人明确告诉他去

看微软某个文档的第几页第几行才算了结。或许，人都大多如此。

其实，我们已经加入了很多特色的功能。其实，我们已经加入了很多处理各种各样情形业务的功能。但很多人告诉

客户，仍然是我们做不到，只有让开发部定制开发了。有时候有很尴尬的事情出现：客户兴冲冲的告诉实施人员，

我发现你们的软件中有个很好的功能对于我们很实用。实施人员呢，反而不知道那个功能。倒是客户自己研究琢磨

出来了。让人汗颜。

我交给文案的任务就是：每当新版本发布，你对照你写的更新变动说明，你的操作说明书，

你的操作视频，你的演

示版，你搞个集中培训。并且你出题，给他们考试，给他们打分。但你给他们做培训的时候，必须有正规的签到表

，正规的培训教材，正规的培训 PPT，掌握好一节课的时间，掌握好一节课的重点，掌握好一节课的快慢与难易程度

的节奏。下了课还必须让他们填写本课的培训反馈。你整天在这里写呀写呀，并不是我的目的。这样的状况，你写

个十年又会如何呢？人需要成长。我希望你能成为优秀的咨询顾问或者培训老师，如果有可能，你也会成为优秀的

市场人员或销售人员。这是我对文案这个岗位的职业发展期许。但是，机会是需要你去铺垫创造，你去抢机会（你

自己铺垫的机会也有可能被别人抢走），你去有能力抓住机会，机会并不是我送给你的，机会是转瞬即逝的，你懂

吗？

文案小姑娘咬咬嘴唇，坚毅的点点头。

这个蛋白质女孩。

32、一分钟先生

有很多网友特奇怪我为什么能有时来写博客，甚至还能接受网友的 IM 交流，问我是怎么做到的。他们都觉得自己

每天忙死了，相信我作为部门的头公司的高层，估计更忙的不见人影，怎么回事呢？

我总结了总结，在此给大家分享一下。

首先，我每天的工作主要干什么？

1 每日接受开发组长报告给我的进度报告、功能需求设计报告，我来提出调整建议和指导。对于报告的问题，我会给

出建议的处理方法。如果需要我出面动手解决，那我就出面。这是每日的例行事务，占了主要的时间。

2 处理手下员工间的流程配合问题，处理部门间的流程配合问题。一般都是根据现在出现的问题重申和强调过去的流

程、职责。如果需要调整流程和职责，就把新的流程和职责重新定义清楚，并且观察几天的

执行情况，直到每个人

都按照新的流程走。

3 定期重新回顾这段时间公司业务的开展情况和面临的困难和未来的动态发展，以重新审视自己过去定义的产品战略

，不断调整，以适合公司发展变化。根据产品发展战略的调整，然后调整内部每个人的职责和分工。

4 定期和每个员工在 MSN 中沟通或面对面沟通，了解他们现在的心理变化、薪水、公司发展看法、职业发展看法，以

自己掌握的信息和自己的经验，对每个员工提出指导建议和方向建议。对于没有个人目标的员工，和他一起交流制

定一个可见时间内他可以达到的目标。如程序员，我会提出，在 6 个月内，让你的代码变的比现在稳定，至少要达到

你自己心中的满意的稳定程度。然后，我会针对这个目标，时常提醒他，不要让他忘记这个目标，鼓励他一定能达

到这个目标。不提醒，人就会渐渐忘记了自己上个月才确定好的目标，甚至由于当前的诱惑和问题，对自己定的目

标发生了怀疑或不感兴趣，就失去了目标。

5 对于每个员工所干的工作，我也会根据他最近出现的问题，及时进行能力方法、技巧的指导。如你应该这样做会更

好。对于如何写稳定的代码、高性能的代码等等，对于如何测试找到更深层次的问题等等，都有技巧的指导，而且

都是结合他现在所遇到的问题和手头的工作。我不会专门拿出一段时间来专门培训。我都是在日常工作中不断指导

，把培训都化在了每日的工作中，尽量做到润物细无声。对于一些新手程序员，我拒绝回答他们的技术问题，我通

常会说：把你们报错的那个信息原模原样输入到百度中，你就会找到答案。你也可以查询微软、SUN 的联机手册，或

者你到 CSDN 上问、搜索。

6 我还会平时看 IT 技术网站、行业网站，思考客户行业发展、IT 技术发展。我来研究软件中的管理思想、管理模型、

管理业务流程优化、考核评价模型。管理软件、数据分析咨询业务、市场推广，很需要这些

模型和文章。每当我研

究出一个管理模型，我就会写文章在内部发邮件，给公司老板、高层、员工推广。

7 偶尔我会跟着实施、咨询团队到客户现场，体会真实的客户应用，和客户面对面交流，交流现在的问题，交流目前

的困境和焦虑，交流未来的发展看法，体会客户真实的想法、思考思路，体会客户的真实应用水平。

这就是我大概的每日工作内容。

我能把时间省出来，最重要是以下几点方法：

1 由于我经常性思考产品发展战略、盈利模式、管理模型、工作职责、流程制度，所以公司的发展一般都是按照计划

和既定流程走的。很多公司也不知道自己做什么才能赚钱，就东一头西一头拉项目，所以也不会有产品发展战略，

而一般老板也是销售出身，想不透软件产品如何发展，而技术总监呢，又不懂这些，可能就是技术牛人而已，甚

至连技术牛人都算不上，只能算个一帮人的头儿。我们过去也是没有产品发展战略，但我是职业经理人，我不能让

公司东一头西一头的随波逐流，所以我常常思考公司现状，从公司的现在的资源和优势和困境来找一条产品发展路

线。就这样不断思考不断尝试不断执行，渐渐的，在我空降公司 3 年后才找到了现在的清晰的切实可行的产品发展战

略。有了切实可行的产品发展战略，随之就有稳定的工作流程与职责，每个人就有了稳定的计划和工作任务。变化

少了，自然也就不会老救火了。很多人说忙，就是老出异常，老救火，公司没有稳定清晰的盈利模式，一直处于抓

项目的状态，每个项目都没有连贯性，当然也就不需要流程，也不需要职责，有了项目大家一窝蜂加班，没有项目

大家闲的要死，这样的状态，即使有流程和职责也是被废掉，根本没有用，执行不了，项目一来就废掉了。

2 我不会去处理事情。我总是授人以渔。我一直强调方法和流程。如果发现新异常，没有流程和职责和明确的人负

责，我就会立即把这些都补齐，让明确的人去处理，而且有明确的处理流程。如果明确的人没有方法和头绪，我会

指导他。但我从不会亲自做。

3 就是因为有稳定的模式和流程和员工岗位，我就从细节中抽了出来。需求调研、功能设计，有项目经理。开发有开

发组长和程序员，测试有测试员、文档有文案人员，实施有实施人员，支持有支持人员，每个事都有专门的处理人

员，而且有明确的流程协作。由于我的平时不断的指导，还有他们每日都处理自己专门的事情，所以他们对自己负

责的事情越来越经验多，处理起来很快，质量也高。我看到很多公司业务不固定不稳定，人经常被抽调来抽调去，

当然经验积累就不深入不专业，工作就不胜任，干活质量和效率都不高。

4 我这个人很专。我清楚的知道自己的优点和缺点，自己想要走的路。所以我十年，一直在开发岗位，而不是去做项

目经理，去做实施，去做售前，去做市场，去做销售等等。很多人尝试了不少工作，但我没有。我知道自己，所以

我总是把自己很优秀的方面练的更加优秀，这样我的缺点就非常明显，因为我没有去想补齐我的缺点，我根本就

没有为我的缺点投入任何精力。所以我这个人看起来很简单，同事们都能很容易看出我的优点和缺点，而且我的优

点和突出。所以对于老板、对于其它同级同事，某件事需要不需要我的参与，需要不需要我做，大家都明白。所

以，我不擅长的事情从来不找我。这样，我也省去了很多乱七八糟的事情，就有了很多的时间。不少人这个也能干

那个也能干，但都干不精，这样的人最容易被别人抽调来抽调去，当然就没有时间了。我只能干很窄的一个领域，

所以我在这个领域精进的很快很高很专业，所以我总是能当这个领域的管理者。

5 我是个公司赚钱导向的人。我研发产品，率领研发团队，我一直贯彻这样的思路，所以整个研发团队全是这个想法

。所以，很多事，一拿这个准则来衡量，这个事到底值不值得做就很清楚了。所以，省掉了不少事。许多人不知道

自己要做一个什么产品，只要别人说的有道理没有办法回绝就做，到头来这个软件就成了一

个四不像，看着什么功

能都有，但这个软件的竞争力在哪里，它的核心模型是什么，很多人回答不出来了。我对产品目标，每个阶段的目

标，产品的核心竞争力，我因为有清晰的产品发展战略，所以我都能把这些要素控制的很好。到底做不做，看是否

符合利益和当前阶段的目标。

6 我这个人不是一个追求完美的人。只要不影响我的任务目标，即使出点小问题小异常，我也不爱管不爱问，就这样

过去了。世界上很多事情，其实要成功，只有那么关键的几个点，我深刻的理解是哪几个点。于是我就把控这几个

点。只要不是这几个点出现的问题，我都大事化小，小事化了。所以在我手下做事，干的比较轻松，还能达到目标

。虽然在有些追求完美的人的眼里看来，每件事情都做的不好。但钱也赚到了，事也按客户要求进度和质量完成

了，而且后续的客户单子也来了，又没有影响未来的客户合同延续。一切关键因素都不影响，我何必花成本和代价

去追求完美，让我的人累的半死呢？

7 我有个我自己的每日任务列表。我每天要做什么事情，我都随时记录下来，然后一条条的做，一条条的消。我看着

不断有任务变灰，就很开心。我清楚的知道我每天要做什么事。

8 我要求手下在每天下班前一个小时把今天的工作进展发过来。然后我一一审阅，给些建议。这样，他们明天该怎么

做，该做什么，今天遇到的问题该怎么解决，在他们下班前就知道了。这样，他们明天一来了就立即动手，根本不

耽误时间，所以工作效率很好。

9 有的员工跟我絮絮叨叨讲了很多，反映他所遇到的问题。我听完后往往第一个校验的就是，你到底要达到什么目的

。很多员工被反问后，是啊，我到底要达到什么目的。最后一重新审视自己的问题，很快就把问题简化了，问题的

症结就找到了，当然解决也就很快了。甚至，弄清楚了目的，发现问题根本不是问题，根本无须解决。省了不少功

夫。

10 我不喜欢员工像讲故事一样来龙去脉的报告。我总是强调，要 1、2、3，这样来。而且你一次报告，不要报告太多

的问题，人不应该关注那么多问题。关注多了，就根本理不清，所以也就解决不了问题。先要把自己的头脑清醒了

，就焦点关注在前三个问题。等前三个问题解决了，再思考以后的。有些员工向我报告问题，总是拔起萝卜带出泥

，一个问题的描述往往会引入另外的问题，然后问题串问题，跟糖葫芦一样。我总是让他打住，就说那是另外一个

问题，咱们现在先别管它，咱们先把你现在说的这个问题说清楚解决完，再处理另外一个问题。

11 咨询有个方法，就是四象限。在时间管理中也有这个方法，重要且紧急，重要不紧急，紧急不重要，紧急重要。

很多人其实是分不清楚这个事情是重要不重要的。我就拿我的目标和赚钱为导向来校验，不赚钱，也不影响未来赚

钱，也不影响我当前核心目标，这个事情就不重要。既然事情不重要，那它怎么会紧急呢。不重要，就认为它是紧

急的。就让大事化小、小事化了。很多事情，随着时间的消磨，也就那样了。你不去处理它，也就过去了，根本对

未来没啥影响。

12 有时我指导训练手下员工，但员工听不懂，当然也不会按照我的方法来做。不少管理者见到这种情况，非常生气

，还不如自己亲自动手做，本来简单的事情被员工搞了两天搞不定，如果是自己，两个小时就能搞定。但作为管理

者，千万不要自己这么做。越这样，越是欲速则不达。每次都是自己亲为亲力，员工闲死，自己累死。我不怕员工

做事做不好，关键点出了异常我可能会亲自处理，其他我都能过则过，而一件事情的关键点往往就是两三个。所以

，一次做不好，咱们其他的事情我继续指导、重复指导。一次听不懂，我继续在下一件事情方法上指导，直到员工

自己也能做的很好。时间，真是一个很奇妙的东西。它能自然改变很多东西。有时候你讲了许多员工也听不懂，时

间长了，他也就明白了。不是不到，是时候未到。就是这个道理。

13 我希望记笔记，不管看书，还是做事，想到一个很好的点就立即记下来。而且我还经常回顾自己的这些记录，把

它们尽量整理成体系和方法，让彼此关联起来。所以我解决问题思考问题的时候，总是有很多参考。而不是从头想

起，也不是一遇到什么事情还得重复寻找资料。我这个人由于太专，所以关注的东西也不多。就看自己关注的东西

，所以范围窄。许多人没有重点，什么热就看什么，而且积累了一大堆资料，反而后来自己都不看。而我，由于关

注的少，所以看的东西自然比别人范围小，看的少也就看的精，而且还定期整理自己的笔记和资料，发现过时了就

删除掉，保留下自己持续关注的资料。我发现，不少人有移动硬盘，里面放了很多资料，但真要做事反而一点资料

都找不到，跟没有资料一样。

14 我喜欢看书和思考问题。过去坐地铁上下班，我都拿本书拿支笔，随时读书和记笔记。一天不读书不总结就觉得

空空的，即使上个厕所也要带本书。很多工作中的管理方法很问题解决思路都是这样想出来的，在工作时间就赶快

去，节省了不少工作时间。有的人记忆力还不好，还没有做笔记的习惯，还想问我有什么阅读理解的好方法？我的

父亲常说：好记忆不如有个烂笔头。其实做笔记不仅仅是个做笔记这么简单的过程，而是在你写字的时候你就会思

考揣摩思考，本来你认为很不错的一段文字准备记录下来，在记的过程中你就会想到自己的问题是否能有所借鉴，

在记的过程中你就会联想其他同类的信息对比，总结出共性和差异点，在记的过程中你就会发现这段话说的比较片

面不像你刚看的那么激动。很多人做不好自己时间管理，就是想做，但只是一个想法，真正做，就觉得这样做太累

了，就放弃了。

15 我常笑说我是个生活不能自理的人。许多平时工作中的例行事情，我都交给我的团队中的文案女孩处理了。她就

相当于我的一个兼职文秘助理一样，帮我处理了许多行政事务。我要去给客户讲某个产品，我给她讲好思路和客户

爱听的重点，她就能把方案处理的符合我明天的讲解。比如我要出差，订票订宾馆只要和她讲一下，她就处理了。

而且她还会告诉我怎么去那家宾馆，地址、路线图、联系电话都有。我很多事情，都是能不用我处理就不要我处理

，我都分配给下属了。

16 我从小爱看书，但是家里也没有多少钱买书。我的一个亲戚家里有许多书，但我不能经常去。但只要去了，我就

赶快看书。为了看的多，就渐渐养成了能很快把握书中的重点和思路的方法，也养成了如何快速鉴别手中的这本书

对自己到底有没有用的方法。所以，现在工作也是，能很快读书，也能很快总结理出思路。对于员工报告的问题，

也能很快找到重点和关键。这个性格也是我能留出时间的原因之一。但这个性格很少有人有，这个是从小的影响，无

法复制，所以我也就放到了最后。

33、灯塔客户

过去一直做项目，也就是说，没有东西，先有方案，方案客户同意签单，才开始调研、设计、开发、测试、安装、

培训、支持。营销部对这种模式也是很熟悉了。

但从前年开始做产品。也就是说，没有明确的客户，也没有特定的客户调研，开发出来客户到底需要不需要不知道

，客户买不买单不知道，客户希望多少钱购买也不知道。这就让营销部没有底儿了。不能让产品闷死在研发部啊。我得想辙。老板和营销部也都认可公司应该由做项目提升到卖产品，但谁也没有经验

，都是做大客户大项目出身的。而且现在的项目也做的不错，有软件单子都逮住，客户关系也不错，客户朋友都主

动介绍单子过来，公司收入和增长发展也不错，于是研发产品就成了一个大家都认可的理想，但销售就是另外一码

事。干吗费那劲呢？

人对自己不熟悉的东西是天生拒绝的，就如同我们黑夜走一条从来没有走过的路一样总以为会碰见鬼一样。

虽然我也内部写了不少文章来阐述新产品，但到底有多少人看，到底大家想法如何，都不得知。我每逢中午吃饭的

时候，我还故意旁敲侧击问，但回答几乎相似：看过了，写的不错。或者是看了，但太忙，没细看。

我也忍不住向老板建议了多次，希望给营销部讲一次新产品。老板嗯了好几次。开季度会议的时候，我在全体高管

会议上又提了出来，这才大家达成一个共识：星期一下午研发部给营销部培训新产品。

培训也做了，大家也都讨论了一下午，营销部对产品也提出来一些建议，我们都记了下来。但我们也修改完了，还

是没有下文。大家可想一个新产品的推广有多难。所以说，一个好产品，研发阶段其实是非常简单的，也是非常短

的时间的，设计、编码、测试都不是具有挑战性的工作。而一个好产品，要获得营销人员认同、老板认同、实施部

门认同、支持部门认同，更要获得客户的认同，是需要非常多的运营和各环节的人员的支持。不少新做开发主管的

网友也曾经和我反应过他们的郁闷，我说这些都是很普通的事情，完成产品研发，只占产品成功的 10%而已。一个产

品死在公司里无人知晓，很普遍。想让大家支持你，凭什么？就凭你是研发他是营销，他必须配合？

到了新年度会议了，我又把新产品的推广销售提了出来。在年度会议期间，我给全体员工又做了一次培训。老板也

参加了培训，感觉产品已经做了这么久还是做的不错的，是时候销售了。于是，开营销部会议的时候，就给营销部

下了一个任务，新产品的销售必须加入新年度的销售计划，并且有销售任务，还要落实到具体人，还要有销售考核

。

这下可捅了马蜂窝。营销部也算是客户打单风里来浪里去，什么人没见过什么阵势没见过什么突然诘问没见过，众

多营销部人员群起攻之，借助他们的营销经验和客户经验，啪啪啪，提出的产品缺陷句句在理。老板也不置可否了

，但仍然下了句话：我们肯定要今年推新产品，不过，可以不作为销售任务，但具体的负责人要明确。

销售部有了具体产品负责人。但负责人只是个销售新手，跑销售项目还只是个助手而已，更没做过产品销售（可想

销售部的打算）。产品负责人新人没见过多大世面很是紧张，忙着安装软件学习软件。

但该产品负责销售人办了一些下面的事情：

1 他试用软件，他感觉不太易用。他认为自己是一个普通的用户，自己感觉变扭客户也应该不舒服。就建议让开发人

员修改。我也没大重视，我也不能说人家啥也不懂瞎胡闹。就这样，开发人员开始改动。从界面的颜色、图标、字

体大小、文字命名、操作方法、说明文档，都按照他的要求改动了一遍。

2 这个认真负责的销售人员，感觉软件不稳定，就召集了销售部一些人员来测试。估计连说明书也没有看，就开始测

试。BUG 倒是没有找到，销售部内部开始 PK。他说他的操作方式更符合客户，他说他理解的流程更正确符合现实。这

个销售人员一开始还和其他销售人员 PK，毕竟这个软件的前期修改都是在他的思路框架下修改的，这次其他销售人

员又提出不同的意见，这不就和他不一致了么。既然他负责这个产品，他就认为自己对这个产品理解更深透。但其

他的销售人员的客户经验比他足，有时候别人说出来的理由他也不知道客户是不是那样想的。于是达成一些协议：

都有理，先记下来，能改的让开发人员都改了。

开发人员问我怎么办？我说静观其变，先让他们内部 PK。

两个月过去，啥销售动静都没有，就在家闷头修改、测试。销售新手成了产品设计师、项目经理，控制了新产品

的功能设计、进度、质量。这真是个奇怪的角色。我曾经在 QQ 群中也见过这种现象，有的网友说他们的组织形式就

是这样，产品的定义归产品部。这个组织形式本来是没有问题的，就连微软的软件设计也是由程序经理提出的。而

微软的程序经理就相当于产品经理，大多为 MBA 出身。但问题在于，微软的程序经理都是非常深刻理解软件的，也是

非常深刻软件营销的。我曾经很惊讶微软的一个管理岗位出身的人居然对微软的开发工具的具体技术有如此深的了

解。而国内，往往是划分了类似的组织结构，却无法配齐相应的人。做产品设计的，往往不懂软件。做软件的人，

往往却只会编写代码。而一个既会做软件，又懂产品的人，如果没有可见的成功案例，也是被别人看作不懂软件不

懂产品的人。

我给老板提了个建议：可以先去客户处调研一下。别老闷在家里，先听听客户怎么说。

于是这位销售听从老板指令，联系自己熟悉的客户去调研。

但调研过程让人大跌眼镜。我听我的开发人员回来反映说简直不是去调研了，而是去销售去了。

这就是一个销售的思维惯性。

当然，最后的结果是客户留下了演示版答应尝试使用。但仍然没有了下文。

季度会议，我提了出来：咱们的软件产品并不是在屋子里闭门造车，咱们的软件设计前期也是借鉴了多家客户的现

有系统，并且在网上和客户交流了很久，也借鉴了很多其他同类的软件。咱们是骡子是马就拉出去遛遛。会不会游

泳，在池子边是学不会的。一脚踢下水，呛几口，死不了，但会很快就学会游泳。

老板也询问了调研情况，但销售新手说不出多少重要的理由，老板于是亲自换人操盘。

新换的负责人是个典型销售型。软件有 BUG？微软还有 BUG。软件不完善？完善了怎么升级呀。

上来就是不断给他的客户打电话，寒暄过后问问现有系统的使用情况，然后根据客户现状引出现在这套产品。客户

一听，好啊，我们正想有这样的功能。老销售说我们这套系统已经好多家试用了，都反响不错，而且能跟现在我们

的系统无缝连接。

电话是打完了，演示版、说明书都给了客户。老销售的三把斧也玩完了。但谁也没做过产品销售，不知道怎么推新

产品。问我接下来该怎么办？

我说，做灯塔客户。他们客户之间都互相走动，消息特灵通。先让他们中的有影响的客户先

用起来，咱们给其他客

户宣传推广的时候就容易的多。而且灯塔客户用的好，咱们的影响力就大的多。如果一开始怕失败怕困难，先做些

小客户，反而费了半天努力起不了多大的水花。还不如咬咬牙顶一项，成了也就成了。我想起了我9年前在其他公司

做灯塔客户时的情景，那时候就是寻找了一家小客户，上线倒是挺快，也没修改多少，但也没有给软件的成熟带来

啥提升，在业界的影响力也没有多大，金额单子也签的少，但最后的支持服务没少打电话，而且客户的IT能力有限

有时还需要去出现场。唉，一切都归结于是第一个客户。客户处处拿这个拿捏我们，说我们拿他们做的实验小白鼠

，就得送佛送到西。而且签的时候就故意找的当地的客户，就是为了做实施的时候成本低。但从此也形成了一种坏

的实施毛病，就是程序员去实施，当场改程序的实施模式。这种模式的形成就在于做灯塔客户是程序员亲自去的，

谁写的代码就谁去做灯塔客户实施，客户有什么需求就当场立即改掉，希望能很快把软件完善起来。但这样做就形

成了惯性了，实施第二家灯塔客户仍然照旧如此，最后灯塔客户都做了好几家还无法交付给实施部门去做。差点把

开发部变成开发、实施、支持全功能的部门。费了好大的劲做过渡转换，如今后遗症还不小。他问我怎么选择灯塔客户？

我说，首先找喜欢创新的客户。产品，也是有生命周期的。一开始，不能找大的客户，也不能找小的客户。咱们的

系统还算初出茅庐，大的客户会控制咱们，影响咱们未来的发展。而小的客户，做完了影响力也不大。所以，我们

要从中不溜的客户中间挑选喜欢创新的客户。不喜欢尝鲜的客户是没有应用的积极性的，反而对新事物天生产生怀

疑和拒绝。

这个创新的客户还需要系统建设方面有影响力。有些客户很喜欢业务创新，但天马行空的，不去想系统能不能实现

，最后反而想的多说的多，系统方面却无法落实。这类客户也经常会见。所以，我们要找的就是喜欢创新、系统

建设重视的客户，而且这个客户需要规模在中等。这就是典型的灯塔客户。只有灯塔客户做起了标杆，有影响力的有分量的大客户在推广的时候才能关注起来。这就好比江湖，江湖老大并不

关注那些小喽啰，而他最关注的人恰恰是第二第三名，因为一不小心这些人就会超过。灯塔客户为生命周期的第一拨，大客户为第二拨，第三拨就是看着创新激进派和大客户们都应用的不错，那些抱着

怀疑的客户才会蜂拥而至。这类客户就如同企鹅一样，谁也不下水，怕水中有海豹。等其他的企鹅下水了没有事了

才纷纷下水。这位销售老手果然重点关注了几个喜欢创新又系统建设的不错的客户。啥也别说了，肯定能搞定，能不能借我一个

星期开发人员，我要到现场去给他们修改开发，肯定签单回来。我有些犹豫，我怕重蹈覆辙。但第一个客户的单子又是如此的珍贵，毕竟要走出第一步。我说去一个星期，但说好了，必须只能是一个星期，不管完成没完成，就一个星期。而且这次去了，还需要跟一位

实施经理。你做项目经理，但你不能越位做产品设计。你负责好客户沟通、协调，上线推动，实施经理和开发人员

调度就行。而开发人员不能直接面对客户，所有的客户需求必须由实施人员收集整理，按照咱们正规的实施流程走

。于是，三人上路了。在此过程中，销售担任项目经理、实施经理、开发人员作为组员。我还尽量控制，但仍然没有

避免客户和每个人都谈需求。我严厉警告开发人员：没有我的确认，你不能开发。即使一个很小的改动，都必须给

我报告回来。哪怕是 MSN 报告或电话报告都行。我还不断让他们各自坚守各自角色。那一个星期，我花了大量的精力在不断区分他们的惧色，在客户要求和他们三

人之间达到平衡，不要让他们三个人都变成需求人员和实施人员。一个星期到了，销售说他已经签了单子，能不能让开发人员再留一个星期，让开发人员开发利索了再回去。我说我要看看还剩下多少活儿。于是我给开发人员报了一下工作列表和工作量估计。我也听

取了实施经理的实施效

果和进度的报告。我也拿开发人员的工作列表找销售去确认了一下看看是否是这样子。三个人都确定好后，我又把

期限宽了三天。三天内，开发人员要完成什么工作我都安排好，做完后离开。实施人员按照正规实施流程走，不能

无限期在现场实施，否则客户永远会说自己不会。要在三天后安排培训、考试，不能实行目前的一对一培训。

那三天，真是很累的三天。销售要说服客户三天后离开，实施要编写培训课程和考试内容，开发要保证按时完成任

务，谁都不轻松，每个人的压力都很大。

终于开发人员和销售回来了。实施经理留在现场，我又派了一位实施培训人员跟进。终于，实施流程和过去又保持

一致了。虽然，销售回来后仍然被客户当作项目经理，但销售尽量的倾听客户，不过实施执行却转给了在现场的实

施经理。开发人员仍然在按照实施流程中产生的需求列表修改着，和以往开发流程、实施流程一致。并没有因为这

次的特殊性而改变。当然，实施经理几次着急上火直接给开发人员打电话，但我都让开发人员自己记录下来，排进

需求管理工具中。这个过程当然痛苦，稍不加管理，开发人员就成了实施人员和支持人员，说不定为了救火还需要

冲到现场。

好不容易，第一家灯塔客户上线验收了。第二家灯塔客户，就是这位实施经理带领培训专员去了，没有跟销售，也

没有跟开发。和过去的实施一样。

我说：你还需要带一名实施经理和培训专员。

实施经理问我为什么？如果再多去人，那两个实施经理怎么分工。

我说：另外一组实施团队，只管培训课程课件制作、培训、考试。你管理好项目进度、需求协调。

于是，四人上路了。

第二家灯塔客户顺利了许多，实施时间也缩短了许多。正规的实施文档、课件都制作了出来，新产品的实施人员也

带了出来。开发人员在办公室，有我做管理指导，有专业测试在每日测试，软件质量也得到保证。

现在，这个产品已经销售的很正常的，开发流程、实施流程仍然很流畅。
走出第一步，千万要把型塑好，千万不要特殊事情特殊处理，否则特殊处理很容易变成习惯处理，最后再也无法回

到正常的流程上来了。
想让开发部不变成实施部，第一步很重要。