

编程课工作总结(汇总5篇)

总结不仅仅是总结成绩，更重要的是为了研究经验，发现做好工作的规律，也可以找出工作失误的教训。这些经验教训是非常宝贵的，对工作有很好的借鉴与指导作用，在今后工作中可以改进提高，趋利避害，避免失误。相信许多人会觉得总结很难写？这里给大家分享一些最新的总结书范文，方便大家学习。

编程课工作总结篇一

1. 完成本年领导安排的工作计划任务和公司项目的维护开发。
2. 改正以前在工作中常犯的编码不良习惯。对公司开发框架技术进行进一步的学习和运用。
3. 利用工作之余时间，加强学习，提高自己处理问题和解决新问题的能力，为后期工作打好基础。
4. 深入掌握java基础知识及数据库技能。
6. 提升团队协作、项目管理、解决项目技术等能力。

本人从事开发进一年，xx年11月开始上班，之前做asp，xx年2月后开始做，由于公司转型，现在带十几个人在一起搞开发，主要是关于准金融行业的行业管理软件。年度总结在会议上也写了，但仅限于对于工作总结及一一年工作计划，自我总结较少，因此今天专门做下自我的年度总结，主要从技术、管理、人生上面谈谈这一年的收获，前辈路过欢迎多多指正。

我在学校里学的是电气自动化，程序基础仅限于c基础课程的一些知识，后来由vb、html转向asp，在asp上花费了不少时间，对asp比较熟悉，后来由于公司业务需要，将开发平台转向，

开始对不是很感冒，以为就是asp的一点扩展（那时还不知道三层架构，数据数据访问全在页面里！），后来招来几个北大青鸟的过来终于意识到的强大之处，经过个人的努力已经逐步掌握了，现在层次上只能讲个人觉得是入门而已，原因是多方面的，待会儿会讲到。

相信从面向过程转向面向对象的同学都有一种感觉：面向对象开始真的有点别扭，涉及到属性，尤其是类之间的各种关系，那时老想用面向过程传递参数多方面啊。于是老在想对象这种东西，从概念中跳中来，以自己的方式去理解才逐渐体会到页面对象的精华来，分层次展现、分级别访问、封装对象之间各种关系逐渐真正理解了，尤其是对象之间的关系，如对象a与对象b两者之间的关系，有些需要完全公开，有些需要隐藏，有些需要通过第三方传递，有些需要给自己的下级可见，有些需要让下级去完成具体操作——这不是现实的实际模型吗？应该这么理解，面向对象来源于现实，它不是一种凭空空想出来的理论，这些对象之间的关系可以将其还原为父子、夫妻、领导下属、同事、朋友之间的关系。相比之下，页面过程往往像是一股脑全部推给用户使用，其中的数据与数据访问方法层次不清晰，在模拟现实上它与面向对象相比更易于入门理解，实质上难于准确直接地表述。

面向对象上另一方面是它的设计模式，在之前的面向过程中对这个设计模式并没有清晰地提出来，面向过程优秀的代码要求高内聚低耦合，从个人的理解上，这仅是对软件开发方法“技”上理论总结；设计模式是达到了“道”的层次，因为它从更大的方向、更抽象的层次来去表述具体的代码模块之间的关系，可以认为设计模式是完全从实际的应用来不断总结得来的经验，之间并没有这种术语，但相信前人肯定也使用到这种思想，它从实际应用于来，当然要应用于实际工作中，认真思考不断总结每个人都会有自己的“设计模式”，可以借鉴前人的思想来去提升自己，不可去为“设计模式”而设计模式。

具体到的实现模型中，真正理解它的机制与方法也就不难理解，记住b/s中离不开post或get[]所有的autopostback[]selectedindex—changed.....都是去调用form传值，加上runat=server的服务器控件打开它生成的源文件也是普通的html标签，微软的让软件开发更容易的思路是很好的，时代在前进，很多年前你使用c写出mis证明你很牛，很多年之后你不在使用c去写“学生管理系统”、“图书馆管理系统”那只能说明你的脑子少一根筋，开发语言都有长处与不足的地方，因为它们适用的场合不同，类似不能拿匕首去跟炮弹比，也不能拿c与php比，程序员都有一种偏执的心理，但一点要记住，你面对的用户才有最终发言权，程序能不能满足需要，易用性、稳定性、成本才是应当首先放到重要位置来去谈的。管理最开始担任管理一职时开发团队加我在内只有四个人，那时只是抱着接受挑战的心理去做管理，加上我本人比较重感情，团队之间关系相处都不错，但严重的问题逐渐显露出来：工作的随意性、团队精神薄弱、工作方式蛮干，印象深刻的是有几个开始承诺项目不能完工，于是最后天天加班，一直做到早上6点，睡一会7点半接着上班，几个同事都是年龄差不多的小伙子，干劲十足。后来随着时间的推移，问题越来越摆在眼前：项目迟迟不能完工，又由于公司待遇方面让新员工感觉不值得，于是形成了老板抱怨员工也抱怨的状况，我在中间两点都要去“消火”，这期间是我们部门相对最累的时间但也是相对感觉最充实的时候，后来，之前的员工跟我说“再也找不到那种感觉了”，这是我能想像的。这期间主要是老总对我十分信任，工作上主要是管理方法上对我指点了不少。后来我逐步体会到，管理应该是“大家定规则去遵守”，而不应该是“人管人”。

人管人很容易陷入一个误区：领导去时时刻刻关注每个员工，这样最后往往后造成员工对领导的敷衍了事，管理松了员工会责任下下降，管理紧了造成员工与领导关系紧张，另一方面领导时间精力有限必然耗费大量的精力在日常的监督中而不能投入到全局的管理中。

于是“定制度一定分工一定进度”，明确日常所有的规章制度，这期间除了公司主要的工作规章制度外其他的日常工作纪律、日常管理等规章制度都是我本人制订，然后征求大家意见最后去贯彻执行。中间也遇到了不少问题，比如开始我们内部是允许使用qq的，后来员工用qq闲聊的时间增多，大大影响了工作效率，最后决心禁止，开始阻力较大（貌似程序员都喜欢挂上几个qq去到群里搞个群主，虽然群里大多都是菜鸟），最后多次开会，逐个谈话，阐明道理，形势逐渐好转。

项目分工上针对技术水平明确分工，制订项目开发计划，由于开始技术都不是很成熟，不少时间我这边强势要求，使用野蛮方法，完不成加班——我陪着加班，这段时间能感觉到员工对我稍有怨言但总体还是认可的。

这期间公司新招人员，人员的增多更使我意识到团队管理的重要性，这期间版本控制、编码规范、文档管理、bug管理等诸多问题都得到一一解决，技术水平上主要是我个人利用空余时间学习新知识充电，然后展开各种培训，主要是photoshop、css、js、sql等方面，培训一方面提升了员工的技术水平，一方面我本人在学习培训的过程中得到的最多，因为这个时候个人要求去思考的会更多，加上我本人对技术兴趣比较深厚，所以后期工作慢慢踏入良性循环。

待遇低、条件艰苦、工作时间长、工作压力大是团队中的难题，这方面公司在某些方面决策层有着严重的错误思想，造成技术人员对公司埋怨增多，在这方面我本人只能以劝架婆的身价去安慰身边的兄弟，因为我明白现在公司的问题与当前中国软件行业的通病一样，盲目追求利益化，不求质量，但求速度，整个社会风气造成软件行业良莠不齐发展，整个中国三四个人的开发团队组成的公司数不胜数，整个程序员阶层生存状况可想而知，疯狂加班、代码质量低下、维护成本大、稳定性差、用户体验差。当然我们本身不能去逃避这

个现状，对于个人来讲任何假大空的口号都是没有意义的，程序员作为技术人员最重要的是心态，以良好的心态去面对各种问题，发现问题、解决问题，发现问题抱怨是解决不了问题的，最主要是解决问题。我在面试员工时反复提到的三点是；工作态度、发展潜力、工作能力，由前向后排序，心态最重要，大公司大家都想削尖脑袋往里挤，但名额有限，所以广大的程序员同志绝大多数都集中在二流三流四流的软件公司，这个时候更应该做好个人职业定位。以我个人的经验，新员工选择一家公司往往最看重工资待遇跟发展潜力，程序员是一种不可以混日子的职业，工作能力很重要，但不是最主要的，工作态度是最重要的，工作态度往往跟个人的思想认识还有职业所在的环境有关，我个人的体会是绝不能太过于情绪化，情绪化无论是对工作还是对个人的发展都是不利的，就是说哪怕是明天你要辞职，你今天都要把今天要做的事情认真用心做好。

对于工作责任心不够的员工是尤其值得关注的，他们往往自我意识过于强烈，追求以自我为中心，很多时候伴随着工作得不到认可、工作感觉不充实不开心，我会主动找他谈话，我认识：你管理的是一个实实在在的人，他不是一台任你摆布的机器，人会有情绪往往是有心结，找到原因，多加开导，用心去关心每个成员。对于实在不适合在本团队发展的员工要做好最坏打算。程序员这个职业与现在的大学生具有一致的特征，一个是从天之骄子沦为多余人，一个是从高智商人士成为今天中国通行的“it民工”。另一方面是程序员往往年龄集中在20—25之间的男性中，情绪化、失落感、心理不成熟等一系列问题都会出现。但这个阶段的人往往也最易沟通，可塑性也较强，适当的引导、合理的方式会比野蛮的管理效果强上百倍。

从我x月x日进入公司到现在已经过去一年了，从一名刚刚结束实习的学生到一名独立的开发人员，角色改变了，职责也改变了。虽然已经预计了工作之中会有很多困难，可是在实际的项目开发中，自己所遇到远远不止自己想象之中的那么

简单、单纯。在开发过程中，难题一个接一个的出现，要做到满足公司的产品需求，成为一名合格的程序开发人员，这个过程还很漫长。每天我都在勉励自己，要好好学，好好做，以后会好的。在此同时我衷心感谢公司给我提供了学习的机会和平台，这也使我对以后的日子多了些许的信心。

编程课工作总结篇二

以上的成功和收获，除了自己的努力，还有公司的支持。这个团队造就了我。我们的团队也是一个务实、努力和创新团队，因为我们的团队成员都很努力。我和软件r&d团队是一个整体，这里的团队总结也是我在这个团队的收获。

务实：公司下达的任务和工作都用心去做。我们团队中没有人在工作时除了工作之外做过其他事情，这些都是真实的实实在在的做跟工作相关对公司有益的事情。相信在伍经理的带领下现在是这样，以后同样也是这样。

拼搏：公司给的每一个任务不管它多难，如果工作没有完成我们会晚上加班，也要尽可能的完成当天的工作。如果工作实在忙，为了赶进度我们放弃周末休息时间也要尽可能的使项目提前。

创新：现在我们开始项目的时候都会进行研讨，一般都会进行一个效率和逻辑的分析与讨论，保证程序正确的前提尽可能的提高程序的效率。

互助：我们小组内只要任何一个人出现技术或其它的问题，我们都会彼此都会尽可能的去帮助他。不会因为某一个人而拖住整个项目滞后。

交流：我们在项目中会及时沟通自己的收获，特别是一些针对性的技术问题。这样可以省了很多重复研究的时间，这是一笔很可观的时间。

在交流中只要我会的，我懂的，我不会去吝啬。我会积极的去与你交流，我的团队名言“人强团则强，人弱团则削”。

编程课工作总结篇三

1998年8月，受微软公司董事长比尔·盖茨的委派，李开复博士来到北京，组建微软*研究院并出任院长。在与莘莘学子的交往过程中，李开复博士看到了青年学生身上的朝气和活力，也看到了不安和困惑，因为*学生应对着高期望的父母、习惯于应试教育的学校和教师以及浮躁的社会心态。“他们虽然有幸出生在能够自由选择的时代，可是时代并没有传授他们选择的智慧”。

于是，李开复博士开设了“开复__”，畅谈人生。“做最好的自己”是李开复对迷茫中的*学生开出的“美国式”思考药方。李开复说：“我出生于*家庭，受教于崇尚自由的美国学校，作为一个充分体验过中西方文化特点的炎黄子孙、融合了中西方文化特点的典型代表，我愿意、也自信有本事为那些渴望成功的*青年供给帮忙，帮忙他们远离困惑，走向卓越”。事实上这是一本畅谈人生价值观的励志之书，它用自己的人生经历和成功事业，捧心置腹传述经验、观察、告诫和期盼。

这使我想起了哈佛校长与南开校长在《对话》节目中的一段精彩对话，哈佛校长：*学生的成绩是要比美国学生好，可是，*学生的想象力、创造力均在美国学生之下，我们有那么多美国学生获诺贝尔奖。

南开校长：*学生还是很听话的，读书也很刻苦，确实独立思考本事差了点。在传统的教育体系中追求的是“大我”而非“小我”，个体的需求往往被忽略。教师要求学生“听话”，考试永远仅有一个标准答案，学生稍有逆反便会标上“问题学生”，等等，这种模式教育出来的人有多少思想？多少创造力？又能作出多少成就和贡献？李开复期望当代青年

要做一个能够从思考中认识自己、从学习中寻求真理、从独立中体验自主、从追求中获得力量的人。

在书中，李开复提出了“成功同心圆”理论，它以诚信价值观为中心，包含进取、同理心、自信、自省、勇气、胸怀等六种人生态度。进取即找回自己，把握自己；同理心即体会他人的情绪和想法，修正自己；自信即鼓励自己，放飞自己；自省即自己反省，改变自己；勇气即不畏失败，挑战困难；胸怀即务实、宽恕、自律、尊重和修养。每一个人都在社会群体中生活和工作，随社会提高而生存、成长和贡献，我们要做一个“德才兼备，以德为先”的人。惟有找到自信和欢乐，找到真正属于自己的成功之路，中华民族才能够拥有更加辉煌的未来。

编程课工作总结篇四

来到xx团队、经过了将近一个月的培训、大体上对业务的流程、规约、制造程序的方法步骤有了必须的了解。期间做过从简单到复杂的几本程序、从仿真到实际收获很大。起先、感到这个人事系统十分之庞大、资料丰富、模块功能繁多。复杂之中又有规律。在此十分感激我的subleaderxx还有老员工们对我们的细心指点。、那段时间常常加班加点地帮忙我们尽快熟悉系统。

培训的最后一本程序、是本工作量比较大的程序。我们也研究讨论过数天、也发现了详细设计的问题。体会到xxxxx公司对程序制造的严谨和细致。处处体现了规范。不是乱来、随心所欲地做程序。如注释地添加、这是多人合作的工作、你写的代码要好好书写注释、以便于以后的修改阅读。在做这本程序时、很多的细节问题常常被我们忽视。让我吃惊的是、画面上的距离竟然都是用尺子去量的、画面上的控件有1px的距离subleader都为我们指了出来、可见这是一个需要细心、耐心、严谨的工作。

在工作中了解到一个原则那就是效率时间优先原则。一切都要有效率、都要在纳品之前、提早发现问题、解决问题。千万不要拖到最后一天、才发现了问题、那样要耽误事情了。采用前紧后松原则、前面多加紧点、后面好宽松点。

之后做过一本比较简单的程序、由于不是很复杂。做的比较顺利。经过自测、还有subleader的测试最终发现了一个javascript的问题、补上了。在做着本程序前、先好好研究了段时间、之后发现了详细设计有错误的地方、也指了出来、发了qa得到了日方的确认。总体上是提前完成了任务。也积累了点小经验。期间最后的步骤sourcereview工作对source进行了修改。有些地方都还不是很规范、总之都要体现按照规约的来、不能随心所欲。

经过一段时间的修正、体会到如果自己写的代码很烂会让下一个改你程序的同事很是头疼、所以必须要提高自己的素质、首先就应注意的要条理清楚、思路明晰。提高素质，写好注释。经过一段时间的修正，体会到如果自己写的代码很烂会让下一个改你程序的同事很是头疼，所以必须要提高自己的素质，首先就应注意的要条理清楚，思路明晰。

掌握进度，随时汇报。了解到我们做的任务都有xx期限，所以要求我们能估计这天做的任务的进度。然后下班前都要跟自己的teamleader汇报。刚开始，我们都很是不习惯，或者是忘记了。没有及时汇报。这样很不好，会导致leader掌握不了你做的任务的动态。以后都就应时刻提醒自己反应状况，提出自己遇到的困难需要哪些帮忙。

测试不马虎，差事不应付。有近一段时间我们做了大批量的测试工作。这可能不必须需求很高的技术含量。却是个考验细心，严谨的任务。甚至有的时候感到很繁琐枯燥。但这个保证软件质量的重要过程。如果测试的马虎，不认真测，软件到用户手上才看到bug是多么不好的状况阿。期间我们都对

不懂的模糊的条目询问了同事和领导。不轻易地乱打对错。做好贴图，贴数据工作。当然做多了也有了点体会，比如100多项的测试，也能够发现里面的规律，有些项目是一组一组的，比如一个动作，测试下来，就能够对好几项的对错进行判别。

中途以前做过一本xxxxx程序，由于接触的同事不多，我自己研究了下参考了给定的程序，跟同事研究讨论后，最终完成了。期间还发现了详细设计的错误，跟日方沟通得到了确认。这让我有了信心，虽然没有接触过的东西，也是能够透过学习完成的。

在空闲的时间subleader也安排了我们自主学习，对xxxxx系统做些技术调查等等。也学到很多，当然还是有很多不明白的地方有待继续学习。

另：以前以前做的.net多是windowsapplication方向的，刚进入公司做的是web方向的所以也是有很多需要学习的。工作中发现，有些动作步骤常常是机械的繁琐的重复的。我们是程序员完全能够自己编写点小软件来服务自己，提高效率。我发现加菜单操作十分的繁琐，要去数据库查东西，复制粘贴要弄个老半天，在空的时间里，我也写了个小程序，能够快速加菜单，很想继续完善实现其它的功能得到subleader及其它几位同事认可。且课余时间还做过个多年的心愿一个麻将程序，有点停滞不前，此刻只完成了个雏形。

总之业务上还有很多需要探索学习的.东西。

a 团体合作，交流协调

做软件是个团队合作的项目，不是一个人单枪匹马能做好的。个性是我们此刻做的人事这么个庞大的系统。刚进公司，由于对大家都不是太熟悉，遇到了问题都是自己琢磨leader鼓

励我们要多交流。此刻大家也很融洽，老员工也很热心地帮忙我们新人，遇到问题我们也能讨论共同进步。这点做的还是挺好。

b□工作的态度

工作态度上还能用心配合大家做好工作的，服从领导的安排，有问题及时反应。并没有恶意地违反规章制度。

尊敬的领导：

程序员，三个月的试用期转眼就过去了。这段我人生中弥足珍贵的经历，给我留下了精彩而美好的回忆。在这段时间里您们给予了我足够的关怀、支持和帮助，让我充分感受到了领导们“海纳百川”的胸襟，在对您们肃然起敬的同时，也为我有机会成为影响力在线的一员而惊喜万分。

这段时间，在领导和同事们的关怀和指导下，我通过不懈努力，各方面均取得一定的进步，现将我的工作情况做如下汇报：

编程课工作总结篇五

先自我介绍一下，本人从事开发进一年□xx届毕业生□xx年11月开始上班，之前做asp□xx年2月后开始做，由于公司转型，现在带十几个人在一起搞开发，主要是关于准金融行业的行业管理软件。年度总结在会议上也写了，但仅限于对于工作总结及xx年工作计划，自我总结较少，因此今天专门做下自我的年度总结，主要从技术、管理、人生上面谈谈这一年的收获，前辈路过欢迎多多指正。

我在学校里学的是电气自动化，程序基础仅限于c基础课程的一些知识，后来由vb□html转向asp□在asp上花费了不少时间，对asp比较熟悉，后来由于公司业务需要，将开发平台转向，

开始对不是很感冒，以为就是asp的一点扩展（那时还不知道三层架构，数据数据访问全在页面里——），后来招来几个北大青鸟的过来终于意识到的强大之处，经过个人的努力已经逐步掌握了，现在层次上只能讲个人觉得是入门而已，原因是多方面的，待会儿会讲到。

相信从面向过程转向面向对象的同学都有一种感觉：面向对象开始真的有点别扭，涉及到属性，尤其是类之间的各种关系，那时老想用面向过程传递参数多方面啊。于是老在想对象这种东西，从概念中跳中来，以自己的方式去理解才逐渐体会到页面对象的精华来，分层次展现、分级别访问、封装对象之间各种关系逐渐真正理解了，尤其是对象之间的关系，如对象a与对象b两者之间的关系，有些需要完全公开，有些需要隐藏，有些需要通过第三方传递，有些需要给自己的下级可见，有些需要让下级去完成具体操作——这不是现实的实际模型吗？应该这么理解，面向对象来源于现实，它不是一种凭空空想出来的理论，这些对象之间的关系可以将其还原为父子、夫妻、领导下属、同事、朋友之间的关系。相比之下，页面过程往往像是一股脑全部推给用户使用，其中的数据与数据访问方法层次不清晰，在模拟现实上它与面向对象相比更易于入门理解，实质上难于准确直接地表述。

面向对象上另一方面是它的设计模式，在之前的面向过程中对这个设计模式并没有清晰地提出来，面向过程优秀的代码要求高内聚低耦合，从个人的理解上，这仅是对软件开发方法“技”上理论总结；设计模式是达到了“道”的层次，因为它从更大的方向、更抽象的层次来去表述具体的代码模块之间的关系，可以认为设计模式是完全从实际的应用来不断总结得来的经验，之间并没有这种术语，但相信前人肯定也使用到这种思想，它从实际应用于来，当然要应用于实际工作中，认真思考不断总结每个人都会有自己的“设计模式”，可以借鉴前人的思想来去提升自己，不可去为“设计模式”而设计模式。

具体到asp.net的实现模型中，真正理解它的机制与方法也就不难理解，记住b/s中离不开post或get所有的autopostback selectedindexchanged.....都是去调用form传值，加上runat=server的服务器控件打开它生成的源文件也是普通的html标签，微软的让软件开发更容易的思路是很好的，时代在前进，很多年前你使用c写出mis证明你很牛，很多年之后你不在使用c去写“学生管理系统”、“图书馆管理系统”那只能说明你的脑子少一根筋，开发语言都有长处与不足的地方，因为它们适用的场合不同，类似不能拿匕首去跟炮弹比，也不能拿c与php比，程序员都有一种偏执的心理，但一点要记住，你面对的用户才有最终发言权，程序能不能满足需要，易用性、稳定性、成本才是应当首先放到重要位置去谈的。

最开始担任管理一职时开发团队加我在内只有四个人，那时只是抱着接受挑战的心理去做管理，加上我本人比较重感情，团队之间关系相处都不错，但严重的问题逐渐显露出来：工作的随意性、团队精神薄弱、工作方式蛮干，印象深刻的是有几个开始承诺项目不能完工，于是最后天天加班，一直做到早上6点，睡一会7点半接着上班，几个同事都是年龄差不多的小伙子，干劲十足。后来随着时间的推移，问题越来越摆在眼前：项目迟迟不能完工，又由于公司待遇方面让新员工感觉不值得，于是形成了老板抱怨员工也抱怨的状况，我在中间两点都要去“消火”，这期间是我们部门相对最累的时间但也是相对感觉最充实的时候，后来，之前的员工跟我说“再也找不到那种感觉了”，这是我能想像的。这期间主要是老总对我十分信任，工作上主要是管理方法上对我指点了不少。后来我逐步体会到，管理应该是“大家定规则去遵守”，而不应该是“人管人”。

人管人很容易陷入一个误区：领导去时时刻刻关注每个员工，这样最后往往造成员工对领导的敷衍了事，管理松了员工会责任下下降，管理紧了造成员工与领导关系紧张，另一方

面领导时间精力有限必然耗费大量的精力在日常的监督中而不能投入到全局的管理中。

于是“定制度一定分工一定进度”，明确日常所有的规章制度，这期间除了公司主要的工作规章制度外其他的日常工作纪律、日常管理等规章制度都是我本人制订，然后征求大家意见最后去贯彻执行。中间也遇到了不少问题，比如开始我们内部是允许使用qq的，后来员工用qq闲聊的时间增多，大大影响了工作效率，最后决心禁止，开始阻力较大（貌似程序员都喜欢挂上几个qq去到群里搞个群主，虽然群里大多都是菜鸟），最后多次开会，逐个谈话，阐明道理，形势逐渐好转。

项目分工上针对技术水平明确分工，制订项目开发计划，由于开始技术都不是很成熟，不少时间我这边强势要求，使用野蛮方法，完不成加班——我陪着加班，这段时间能感觉到员工对我稍有怨言但总体还是认可的。

这期间公司新招人员，人员的增多更使我意识到团队管理的重要性，这期间版本控制、编码规范、文档管理、bug管理等诸多问题都得到一一解决，技术水平上主要是我个人利用空余时间学习新知识充电，然后展开各种培训，主要是photoshop、css、js、sql等方面，培训一方面提升了员工的技术水平，一方面我本人在学习培训的过程中得到的最多，因为这个时候个人要求去思考的会更多，加上我本人对技术兴趣比较深厚，所以后期工作慢慢踏入良性循环。

待遇低、条件艰苦、工作时间长、工作压力大是团队中最大的难题，这方面公司在某些方面决策层有着严重的错误思想，造成技术人员对公司埋怨增多，在这方面我本人只能以劝架婆的身价去安慰身边的兄弟，因为我明白现在公司的问题与当前中国软件行业的通病一样，盲目追求利益最大化，不求质量，但求速度，整个社会风气造成软件行业良莠不齐发展，整个中国三四个人的开发团队组成的公司数不胜数，整个程

程序员阶层生存状况可想而知，疯狂加班、代码质量低下、维护成本大、稳定性差、用户体验差……当然我们本身不能去逃避这个现状，对于个人来讲任何假大空的口号都是没有意义的，程序员作为技术人员最重要的是心态，以良好的心态去面对各种问题，发现问题、解决问题，发现问题抱怨是解决不了问题的。“it民工”是我个人认为it人最没有正确的自我定位的一个称谓，试想一个人连自己都看不起自己的职业，他能做好自己的工作吗？），最主要的是解决问题。

我在面试员工时反复提到的三点是；工作态度、发展潜力、工作能力，由前向后排序，心态最重要，大公司大家都想削尖脑袋往里挤，但名额有限，所以广大的程序员同志绝大多数都集中在二流三流四流的软件公司，这个时候更应该做好个人职业定位。以我个人的经验，新员工选择一家公司往往最看重工资待遇跟发展潜力，程序员是一种不可以混日子的职业，工作能力很重要，但不是最主要的，工作态度是最重要的，工作态度往往跟个人的思想认识还有职业所在的环境有关，我个人的体会是绝不能太过于情绪化，情绪化无论是对工作还是对个人的发展都是不利的，就是说哪怕是明天你要辞职，你今天都要把今天要做的事情认真用心做好。

对于工作责任心不够的员工是尤其值得关注的，他们往往自我意识过于强烈，追求以自我为中心，很多时候伴随着工作得不到认可、工作感觉不充实不开心，我会主动找他谈话，我认识：你管理的是一个实实在在的人，他不是一台任你摆布的机器，人会有情绪往往是有心结，找到原因，多加开导，用心去关心每个成员。对于实在不适合在本团队发展的员工要做好最坏打算。程序员这个职业与现在的大学生具有一致的特征，一个是从天之骄子沦为多余人，一个是从高智商人士成为今天中国通行的“it民工”。另一方面是程序员往往年龄集中在20—25之间的男性中，情绪化、失落感、心理不成熟等一系列问题都会出现。但这个阶段的人往往也最易沟通，可塑性也较强，适当的引导、合理的方式会比野蛮的管理效果强上百倍。