

浅谈软件开发管理体会

作者：杨利梅

从毕业至今，大小的项目做了一些，有不少成功的喜悦，也有很多失败的教训。今年由于工作需要，我以软件项目负责人的身份参加了接入网统一网管系统开发的整个过程。从中学到了不少知识，有许多体会，想将自己的感受写出来，与大家共勉。

软件项目管理是一个庞大而复杂的系统工程，当前业界对于软件开发流程有不少规范和定义，如 CMM 和 ISO9000。在该管理体系的管理下是可以开发出高质量的软件产品。但是由于该体系较适合于大型而且复杂项目的团队开发，真正实施尚需要时间和过程。而我们当前执行的项目，一般只有 10 个人左右，要实施软件工程难度更大。我认为：虽然项目大小不一，但管理方法是相通的，要做好软件开发工作，就必须加强有效管理。

大家知道，“软件危机”起源于一些大型项目的不断延迟甚至失败。与大项目相比，小项目具有以下特点：

- 项目功能相对较少；
- 开发人员较少；
- 开发周期较短。

小项目看起来比较简单，比较容易成功，人们往往容易忽视小项目的管理，其实这是一种误解。

据我了解，小项目开发中容易出现以下问题：

1、开发之前没有认真地进行项目可行性和工作量的估计。

往往由于项目较小，便很草率地制定一个开发日程表，没有认真地估计项目难度，结果实际完成时间与估计完成时间往往有较大差距。

2、没有真正的设计过程。

开发人员少，不同人员的程序之间交互、接口相对少一些。开发周期短往往是几个人从头到尾负责一个项目，几个人碰一下头，讨论一下最基本的数据结构、函数接口便分头去做自己的工作了，没有一份较正式的文档来规范各自职责和项目细节。

这种做法潜在的危險之一是有人可能会对所讨论的接口、结构理解有偏差，可能会造成以后的返工。

另一个潜在的危險是由于讨论时忽略了某些情况，等大家都按时完成分工任务后，才发现各个模块组合起来却无法形成一个完整的系统。其根源在于没有一个负责协调的人员不断监控整个开发过程。

第三个潜在的危險是一旦有人中途退出开发队伍，其他人加入时，难以理解以前别人做好的代码，又要从头做起。另外，没有文档的程序，日后维护和版本升级都比较困难。

3、不经过单元测试而直接进入系统测试。

造成这一现象的原因是每个模块相对比较简单，但是为了测试一个模块需要建立一些测试环境。例如，为了测试一个函数是否正确，应该用一些测试数据去调用该函数，需要编写一些测试数据。但很多开发人员嫌麻烦，觉得反正其他模块也很快出来了，直接用真正的数据来运行几次就行了。针对以上问题，我认为

在开发过程中必须处理好四个关键问题，严格把关，可以大大提高软件的质量。

这四个关键问题为：人员、规范、测试、时间控制。

一、合理配置人员

首先软件开发是一项长期艰苦的工作，所以一个团结、协作的团体才能在规定的时间内完成一个质量上乘的软件项目。团队中的每个人必须积极融入到整个集体中，不能互相推诿，更不能互相埋怨和指责，正确的态度是大家在充分信任的基础上团结协作，互相帮助，主动承担任务，利用集体的智慧获得成功。整个团队就是一部机器，只有每一个齿轮都能正常运作，才能生产出优质的产品。

合理配备人员是成功完成软件开发项目的切实保证。所谓合理配备人员应包括按不同阶段适时运用人员，恰当掌握用人标准。一般来说，软件项目不同阶段、不同层次技术人员的参与情况是不一样的。图一是典型的软件开发人员参与情况与实际人员需求差异曲线图。

如人员配置不当，很容易造成人力资源的浪费，并延误工期。特别是采用恒定人员配备方案时，在项目的开始和最后都会出现人力过剩，而在中期又会出现人力不足的情况。

为开发人员创造出一个人尽其才的环境也是项目成功的重要环节，让他们能得心应手的施展自己的才华，特别在工作安排上要煞费苦心，针对每个人不同的特长，根据项目的具体环境和条件来合理安排人员在恰当的岗位上。

项目负责人是一个团队的核心，其综合素质直接影响项目的成败。合格的项目负责人具有高超的领导才能和强烈的科技意识和较强的业务处理能力；具有敏锐的洞察力，能瞄准目标，实事求是，精心组织，坚决果断，灵活应变，享有信誉；善于制定计划，解决问题，沟通信息；具有良好的市场意识和交际能力。当然同时满足这些条件比较困难，但是他应该具有实现这些素质的条件，并注重经验的积累、素质的提高、能力的培养。并能从以下几方面严格要求和培养自己：

以身作则：只有身先士卒，各方面以身作则，才能得到广大开发人员的认可和信任，才能树立较高的威信。果断抉择：负责人的重要任务是决策，特别是有多种选择的情况下，一个正确的选择往往事半功倍。

善于交际：他必须积极对外联络，充分利用外部资源，例如其他部门做过类似项目者，可以向他们取经甚至直接获得源码。这对一个项目争取时间，避免重复工作很重要。

善于协调：协调几个人的工作比自己完成一段编码更重要。由于协调不力，将影响开发。所以项目负责人除完成自己的编程任务外，必须随时监控各开发人员的工作，包括内容是否与要求发生偏差，进度是否滞后等等。

善于制定计划：在开发前，可将明确的开发任务通过文档传递给每个开发人员，让大家都熟悉设计模型，都清楚自己所做的工作在整个系统中处于什么地位，这样有时候可能会发现设计模型中的漏洞，避免了各人的代码编写完毕之后又要修改的后果。

沟通问题：团队沟通不是技术问题，但却是一个最能影响工作效率的问题。沟通及时、集思广益、步调一致，才能取得胜利。

二、严格执行软件开发规范

软件开发需要严格按照软件规范实施。用手工作坊式的方式来开发软件，其结果必然失败。从项目的用户需求分析、系统分析、编码、调试、测试、发布都需要一步一步完成，不能轻视或忽略任何一步。前部

分没有完成好，不要贸然进行下一步。越是项目起步阶段，越是要注意按照规范进行。

如前所述，因为开发软件项目规模较小，很容易忽视规范化，而随心所欲，没有计划，想到哪做到哪，其最终的结果是失去控制。其实项目小正是实现软件规范化管理的好时机，规模小，涉及的管理方面有限，管理实施起来比较容易。CMM 等规范标准不是轻而易举就能实现的，但是可以借鉴它的思想和方法，先在小项目上实现规范化管理，培养人员的规范和意识，为以后实现大项目的 CMM 等规范打下良好的基础。

特别需要重视软件开发中文档管理。那种认为只要产品做出来可以运行，何必花费许多精力去做文档的观点是错误的。经过实践，我深刻体会到，没有文档会带来很多问题。用文档去引导开发过程，抛弃随心所欲的开发模式。就象工厂工人师傅按照图纸生产零件一样，否则很可能会得到次品甚至是废品，给后来开发者留下一堆没有意义的“垃圾”产品。我认为文档应该是开发中阶段（mileStone）结束的标志，每个阶段后，都需要提交相应的文档，而且要确保文档的质量。

确保文档质量的最有效方法就是评审，提交文档后，项目负责人组织相关人员对该文档进行审核，在充分讨论的基础上进行文档的重新修改和审核直到满足项目要求。文档应该是贯穿整个过程的主线，在不同的阶段，需要不停地对文档进行完善，使之真正成为全体项目人员的智慧结晶。