软件项目管理的重要性

1、什么是项目管理？

项目管理是在一定的约束条件下，以高效率地实现项目业主的目标为目的，以项目经理个人负责制为基础和以项目为独立实体进行经济核算，并按照项目内在的逻辑规律进行有效的计划、组织、协调、控制的系统管理活动。

软件项目管理是软件工程的保护性活动，它先于任何技术活动之前开始，并且持续贯穿于整个计算机软件的定义、开发和维护之中。

软件项目管理的对象是软件工程项目，他所涉及的范围覆盖了整个软件工程过程。

2、为什么要有项目管理？

没有项目管理，项目也有可能成功。但没有管理的项目，很难保证项目的利润空间，对公司来说，亏损的风险就更大。所以我们要有项目管理，以保证公司在总体上是盈利的，特别注意不是每一个项目都要盈利。

另外，有了项目管理，就有了管理改进的基础，无论刚开始项目管理是多么的糟糕，只要有了管理，就有了改进的可能性至于能不能得到改进，以及改进的快慢或好坏，则取决于两个因素：一个是人，特别是各级管理者；另一个是利益。关键是

“利益”，准确的说是“利益的分配”，在权责利明确的前提下，人才能充分的发挥作用。还需要指出的是“利益”是多元的，这里的多元不仅指利益的具体形式，而且指利益的受众是多元的，包括客户方相关人员个人的利益。

2．1开发人员问题

软件工程，这对软件开发员来说是多么熟悉的字眼，但其中的内涵你又知道多少? 再进一步说，在开发中按软件工程来做的又有多少? 大家常在网上听到一些程序员在抱怨说：“我加班加点地写了10万行的代码，所以老板把我给开除了。”这话扎听有点好笑，但细细一想这是他的悲哀。他写的代码虽多，可是关键的又有多少。如果里面的代码只要有80%的是关键的，我想老板是决对不会开除他的。问题出在10万行代码中有多少是关键的。先不说写这10万行代码所花的时间了，就说以后的维护问题，要是你到一个公司，老板首先要求你看完这10万行代码。我想你第一想到就是走人，第二想到的还是走人。因为对于一个没有按软件工程来进行开发的程序，不要说是这么多的代码；那怕是几十行、几百行，读起来也是很难受的事。所以要想写出关键的代码，还是得按软件工程来。

一个好的程序是先写好设计文档再进行编程的，在设计文档的指导下，才能写出安全的代码。在文档的指导下，这样写出来的程序至少不会出现写了10万的代码还被老板开除的情况。如果你不写文档，一开始就写程序，这样你就不会按已设计好的路线走，而是想到哪写到哪。小功能还好说，要是大功能，你想想等你写下一个时，回过头来看原来写的，你早就不知所云了，那时你就觉得好像在云里雾里乱走，修改的代码也就更不安全了。我所说写文档不是说很正规的，那怕你只用一张搞纸划上几画也要画点出来（这是对小功能来说的）。对于大的程序来说，你就必须正规的写了。因为这样才能详细的记下你的设计思想，如果开发一段时间后，感觉须要对一些功能进行修改或变态时，记住别删除原来的，而是在下面进行变更说明。这样再次看文档时你就会更清楚为什么要这么做的原因。一看就明白不是多好。总比你去再想以前的设计花的时间要少，如果删除原来的设计思想，当你再次看修改或变动的功能时，你可能会对其不理解。这是多么可怕的事呀！所以项目开发的最佳流程，就是要先进行写好设计文档，然后根据设计文档进行编码，这样才是一个优秀程序员应有的素质。

2．2、软件项目管理问题

现代的软件开发，技术不是关键

随着日益增长的软件需求和软件系统功能的增强，过去一个人开发的历史已不复存在。现在单枪匹马写程序也只是一种娱乐。我们一般开发的系统都是一个小组才能完成的。所以管理才是开发出好的软件的前提条件，没有管理一定出不来好的软件，当然有管理也不一定出软件的。一个成功的软件不一定是最好的技术，但在它背后一定有一个好的管理。所以现在的软件开发已不像从前把技术放在第一，而是该把管理放在第一位。我在网上看到一篇关于中国软件和印度软件的比较。我现在记的不是太多，但对我影响最深的是他们会去权衡技术和开发效率问题。如现在开发一个软件，用户要求在三个月内完成，你在做系统分析时也认为在三个月能完成。但你没有考虑到一些细节，你写完系统的总体设计，在进行详细设计时碰到要建一张不是太大的路由表。这时大多数国内的设计人员就会想用什么算法，去花很多时间去设计研究新的算法和技术，而人家首先考虑的是系统的运行环境，而这个软件设计了是在（CPU：1.1G，内存：512M）中运行，用户也没特意提出其运行效率要求。所以人家就在内存中开一个大数组来对这个路由表进行操作。从这点看，人家注重的是软件的整体，而不像国内大多数据设计员那样，把个体放在首位。把软件设计的整体放在首位，而不去花太多的时间在不一定成功的技术上。如果花太多的时间在技术上去，这将对系统的按时完成带来影响。我也不是说不该研究技术，我只是说开发中应当以全局为重。如果要加入新的技术，必须在分析时就预算其所需要的时间，并设置技术风险管理。如果风险太大就应当取消用这项技术，改用其它的已成功的技术代替。风险管理这是近来才提出的软件管理方法。它对我们的软件项目有着很好的控制作用。对于一些中、大型系统，它是一把走进成功之门的钥匙。一些常用的软件管理方法：

（1）错误管理

软件开发是一项复杂的活动。一个典型的软件开发项目可能会给我们提供很多的机会去从错误中吸取经验教训。一般的软件项目也会提供少量的错误给我们学习。学过开车的人都知道，教练老是会这么讲：“我希望你们从我身上学习我和前人的的经验，这些经验你们就不要再去试了。如果要试你也许会赔上钱甚至于生命。”虽然软件项目开发不会赔上生命，但是失败的软件项目是一定会赔钱的。所是在软件开发中少不了要对错误进行管理。在项目的错误管理中一般是这么做的：

（2）列出典型错误

典型错误中有人员方面的。如：对有问题的员工失控、挫伤积极性、人员素质低、英雄主义、项目后期加入人员、开发人员与客户之间发生摩擦、不现实的预期、缺乏有效的项目支持、缺乏各种角色的齐心协力、政治高于物质、充满想象等…

典型错误中有过程方面的。如：过于乐观的计划、缺乏足够的风险管理、缺乏计划、在压力下放弃计划、在模糊的项目前期浪费时间、前期活动不合要求、缺少管理控制、缺少质量保证措施、鲁莽编码等…

典型错误中有技术方面的。如：过高估计了新技术或方法带来的节省量、项目中间切换工具、缺乏自动的源代码控制手段等…

（3）列出自己的最差实践

注意典型错误，建立自己的最差实践列表，可以避免在以后的项目中犯同样的错误。

列出项目中的最差实践：

组织机构和其他项目组总结经验，学习他们的错误中得到的经验。和其他组同事交流项目开发中的磨难，学习他们的经验。列出潜在的错误，看到它我们就会尽量避免今后犯同样的错误。

司在发展的同时，也会积蓄一些各方面经验。列出所有的经验，按其分类。系统分析中的经验提供给系统分析，设计人员中的经验提供给管理人员，技术中的经验提供给开发员。这样我们就会有更多的时间花在新的错误的防范上面。开发出来的系统就会一个比一个好。

（4）风险管理

可以在几个层次上定位、管理风险：

危机管理---救火模式，就是在风险已经造成麻烦后才着手处理它们。

失败处理---察觉到了风险并迅速做出反应，但只是在风险发生之后。

风险缓解---事先制定好风险发生后的补救措施，但不做任何防范措施。

着力预防---将风险识别与风险防范作为软件项目的一部分加以规划和执行。

消灭根源---识别和消除可能产生风险的根源。

我们应当着力于预防风险，更好的是消除风险根源。

风险管理由风险评估和风险控制。而风险评估由风险识别、风险分析和风险优先级组成：

● 风险识别：就是提出一个潜在破坏项目进度的风险列表，就像生成错误列表一样。

● 风险分析：评估每一个风险出现的可能性及其影响，判定风险的级别。

● 风险优先级：按风险影响大小排出一个风险优先级，这个风险列表将作为风险控制的基础。

风险控制由风险管理计划，风险化解和风险监控组成。

● 风险管理计划：制定一个应对每个重要风险的方案，同时就确保每一个单独的风险管理计划之间以及与整体项目计划之间相一致。

● 风险化解：每个重要风险所对应计划的执行。

● 风险监控：就是对解决风险的过程进行监控，风险监控还可以包括识别新的风险并将其反馈到正在进行的风险管理进程中等方面的工作。

人员管理：

不同人员之间经验的不同导致绩效差别是有目共睹的，所以人员管理在软件项目中也有较重的分量。很清楚，人力因素极大地影响着生产效率，同时任何关注提高生产效率的组织首先必须有一套良好的人员激励、团队合作、员工选择及培训的机制。这样才能充分发挥人员的自身能动性。为公司创造更多的价值。

除了以上几个面的管理外还有其它方面的管理也决定软件项目的成功与否。如：团队合作、团队结构、生产率工具等等。