

电子 期刊

作者: 思步网会员

主编: panda6174

编委: step365he lily_014
cecilia fishred steplv

校对: lily_014 step365he

第十期

封面设计: steplv

发布: 2008.09.18

著作权说明

本文档为免费电子文档，任何人可以在思步网（<http://www.step365.com>）免费下载。

著作权属于思步网及思步网会员共同所有。

在不对本文档做任何修改的前提下，任何人都可以在互联网上自由下载、传播本文档，也可以放在自己的站点供他人下载。

若通过互联网在线转载其中部分内容，或者通过其他媒体转载本电子书及其中部分内容，必须注明文档来源“思步网 <http://www.step365.com>”和文档作者。

本文档仅供学习交流之用，未征得思步网及思步网会员的同意，本文档不得用于商业用途。

由于编者水平所限，在文档中难免会出现错误，欢迎读者对本文档提出批评建议，意见请反馈至：service@step365.com，谢谢！

阅读注意事项

在阅读本电子文档时，请留意以下字符（字母）所代表的含义。

本电子文档第一部分中：

Q (user name)：Q 代表问题 Question

A (user name)：A 代表解答 Answer

C (user name)：C 代表结论 Conclusion

括号中内容 user name 为思步网会员名

例如：**Q** (steplv) xxxxx, **A** (sungubbi) zzzzz, **C** (fishred) yyyyyy

表示：

提出问题者为会员 steplv，问题内容为：xxxxx

解答问题者为会员 sungubbi，解答内容为：zzzzz

做总结者为会员 fishred，总结内容为：yyyyy

依次类推。

思步在线交流

同时欢迎广大读者加入我们的 IM 群，与众多朋友在线交流：

✧ MSN群: mggroup24287@hotmail.com

✧ QQ群: [19894513](#)（申请加入时，请注明职业）

✧ WebSite: www.step365.com

目 录

著作权说明.....	2
阅读注意事项.....	3
第一部分：思步简讯.....	5
第二部分：你问我答.....	6
1 全员改进过程.....	6
★ CMM/CMMI将带我们走向何方—— 一个CMM从业人员的反思.....	6
★ 产品的需求阶段和项目需求阶段区别.....	12
2 全员改进案例.....	13
★ 帮这个项目经理想想办法：新老产品交替.....	13
3 项目管理.....	15
★ 软件项目管理的十个误区？.....	15
★ 100 个管理学定律.....	18
4 QA/EPG专区.....	23
★ 如何进行质量分析？.....	23
★ 流程不及时启动或不及时处理作为QA该怎么办？.....	24
★ 如何深入理解验证与确认？二者的区别？.....	26
5 CM专区.....	27
★ 关于产品分支管理的讨论.....	27
6 CMMI专区.....	29
★ 神坛之上的CMM——解析CMM和中国软件.....	29
★ CMMI评估过程中的需求分析？.....	32
第三部分：会员原创.....	36
1. 边界值测试要点.....	36

第一部分：思步简讯

思步快讯 1:【征文】与思步网一起走过的日子

从 07 年 9 月 18 日到 08 年 9 月 18 日，思步网即将走过它一周岁的历程，在这期间，凝聚着“思步者”太多的感触，也汇集着会员们辛勤的汗水。也许，我们有太多的感慨需要表达，只因为这不凡的一年。

在这一年里，我们每个人都收获了自己的事业、收获了自己的爱情、收获了自己的友情、收获了……太多太多。在此刻，曾经的历程清晰地萦绕在脑际、浮现在眼前，多么深刻、多么熟悉！

曾经，我们哼着《在路上》这一首震撼的曲子就这样懵懂地上了路，至今还奋斗在生活的路上。回味过去，也许我们有太多的感慨，因为过去的事物就是我们人生的一大笔财富；憧憬未来，也许我们有太多的期盼，因为未来的事物就是我们此生追逐的梦想。

我们每个人都收获着成功与失败、欢笑与泪水。思步网这一年的发展，也充满着太多的艰辛与坚持，因为艰辛，所以坚持；因为坚持，所以奋斗。我们“思步者”始终相信，这一切还将继续下去，直到梦想成真！

值此思步网创办一周年之际，我们组织“与思步网一起走过的日子”征文活动。只要您曾经接触过思步网、感受过思步网，并与思步网一起奋斗与坚持过，您一定有话要说，因为思步网的发展有您的奉献、有您的支持、有您的足迹(step)……

征文题材：题材不限（散文、诗歌……等）

奖励办法：所有参与征文活动的会员，派发金币奖励（奖励金币数：300±200）

征稿日期：即日起至 2008 年 10 月 18 日结束

评稿组成：超级版主、版主、意向会员

征文投递：详见“思步网（www.step365.com）-资讯-思步快讯”栏目

思步快讯 2:【授勋】思步网创办一周年授予会员勋章公告

2008 年 9 月 18 日，是思步网创办一周年的日子，在这一年当中，有太多的会员与我们一起走过了生活的点点滴滴。这一路伴随着会员们默默的支持与鼓励，我们将不会忘记！

有很多的会员对于思步网的发展做出了自己的贡献，TA 们或奉献智慧、或奉献时间、或资助金钱……，有太多的语言，却无法表达无限的谢意。我们挑选了 TA 们中的一些杰出代表，并分别授予“特殊贡献勋章、杰出青年勋章、优秀版主勋章、忠诚会员勋章”。

● 特殊贡献勋章：



jane lily_014 step365he sungubbi scott w7a8 fishred steplv

● 杰出青年勋章:



Scott lily_014 step365he w7a8 panda6174 rebeccazxy steplv

● 优秀版主勋章:



lily_014 step365he xixiaojing666 lee_huo yywilly 一啸长天 steplv

● 忠诚会员勋章:



cecilia iamredeye panda6174 fishred xixiaojing666

还有太多的会员,TA 们对于思步网的发展也倾注了很多的心血,由于篇幅这里不一一列出,同样的,也感谢 TA 们。

同时,我们也热切期望有更多的会员参与进来,与我们一起迈向自己的目标,真真切切地记录我们曾经走过的每一个脚印(step0、step1、step2、step3……step365……step N) 📁 这将是一段美好的记忆!!!

第二部分：你问我答

1 全员改进过程

★ CMM/CMMI 将带我们走向何方—— 一个 CMM 从业人员的反思

Q (steplv):

本人是吃CMM这碗饭的,虽然到目前为止,本人依然认为无论是SW-CMM还是CMMI都是软件过程领域的集大成者,从专业角度上来说确有其不可否认的价值。而且我们也应该看到,CMM/CMMI在中国的推广无论是对软件企业还是从业个人的确起有其一定的积极作用,尤其是在对先进理论和方法的接收和管理意识的提升上是有其客观贡献的。

然而,本人在长期接触各种不同企业的CMM/CMMI实施和评审的过程中,却对整个CMM/CMMI在中国推广的过程产生了越来越多的质疑,本人在这边愿抛砖引玉,将个人的一些感受与大家分享。

CMM/CMMI大跃进的背后是什么?

有一个耐人寻味的数字，2004 年向上海市政府报备的CMM/CMMI项目超过了一百个，请大家注意，这仅仅是一个城市一年的数量，如果按比例放大的话，中国在短短的 1—2 年内就能成为世界上CMM认证最多的国家了。而按照众多的咨询公司和媒体的说法：“通过CMM认证的企业的数字标志着一个国家软件行业的发展水平”，那么我们是不是马上就要成为世界软件行业的领袖了呢？

显然，稍微有一些理性的业内人士都不会这么的乐观。事实上，这种大跃进式的盛况背后，作为CMM/CMMI的从业人员，本人看到的确是另一番的景象：

首先，我们的软件企业对一纸可以装点门楣的证书的兴趣远比对软件过程改进的兴趣要浓厚的多，因此在前者上下的功夫自然远比后再要大得多，尤其是在评估过程中，通常会发挥我们国人的应考特长，对评审中可能遇到的问题进行精心的准备，当然这大多都是在咨询公司的咨询师的循循善诱下进行的。

通过CMM/CMMI的企业，通常会把所谓的“认证”证书挂在公司的醒目位置，而据本人所接触的多家通过CMM/CMMI的软件企业，一年以后仍能“基本”按照执行的只有一家，大多只有部分过程还被坚持执行，而有不止一家的软件企业在拿到所谓的“认证”后不久，干脆将所有的资料束之高阁完事。

我们的目的达到了吗？

无论是咨询公司还是软件公司，在表面上都毫不例外的宣称CMM/CMMI的引入是为了提升企业管理能力，我相信至少大部分的软件公司在计划引入的时候的确是有这么期望的（虽然他们往往对那张证书报有更大的兴趣），那么我们的目的达到了吗？在整个过程中，我们看到的更多的是拼命加班的赶写文档，却很少看到对一个过程的有效性进行反复的研讨；看到更多的是对最终评审结果的高度重视，却很少看见对整个过程实施有效性的认真探究。往往是在拿到那一张四四方方的证书后，大家弹冠相庆然后作鸟兽散，一切仿佛又回到了开始的地方。而所谓的CMM过程，随着时间的推移终于渐渐的变成只是某个书柜中的一叠文档。

谁是最终的获利者？

既然软件公司从中获益甚少，那么谁才是真正的获利者呢？在这个不平衡的政府、软件企业、评估方和咨询方的四角游戏中，获利的首先是那些来自远方的洋和尚——评估师们，他们在国内几天到十几天的服务往往都能带走五位数的美金，当然，那些咨询公司也是收入不菲的，有一个很有趣的数字大家可以比较一下，在 2000 年左右，CMM刚刚引入

到中国的时候，做一个二级的评审需要花费大约 100 万左右的人民币，而现在，做一个远比CMM二级复杂的多的CMMI三级（同时包括二级和三级）的费用却只有 30—40 万人民币左右，其中相关费用的构成完全都是服务的费用，不涉及任何其他因素。显然价格的下降是因为竞争所导致的，这也可以从另一个方面反映出当年这个行业是如何的暴利。不过，无论是暴利的过去还是不那么暴利的现在，最终获益的都只能是那些洋评估师们和土（或者半洋半土的）咨询公司们。

谁是CMM的买单者？

同样，在这个不平衡的四角游戏中，咨询方和评估方既然是获利者，那当然跟任何掏钱的事挨不上边，那么软件企业呢？这些企业绝大多数都不是国营的，赔本赚吆喝的事情是绝不会轻易干的，那么最大的买单者只能是政府了，目前国内绝大多数的企业都是在有政府补贴的情况下才实施CMM/CMMI的，例如：上海市去年为CMM准备了超过 2000 万的补贴资金，这也就是为什么去年上海CMM市场如此繁荣的真正原因。而且很多在签署CMM/CMMI实施合同的时候把这一条最为乙方的一个条款写进合同中，更有甚者，有些企业用阴阳合同的方式在咨询和评审费用以外套取额外的费用，而咨询公司只要生意有的做当然也是顺水人情何乐不为。于是乎，这些软件公司在赚了吆喝的同时还有外快可赚，这是一举两得。而这一切都是由政府买单的，而政府的钱又是哪里来的呢，只能是我们每一个的纳税人。

咨询公司究竟扮演了什么样的角色？

在这场CMM/CMMI大戏的四个角色——政府、软件企业、评估方和咨询方——之中，咨询公司所起的角色颇耐人寻味，其不光是CMM/CMMI的执行者，还是CMM/CMMI的最为有力的推广者。但有一个非常值得回味的事情是，本人曾经接触过两个在国内都颇有知名度的CMM咨询公司，其中一个还是国内CMM咨询行业的领军企业，但是他们在其企业内部组织的软件开发项目均遭受了重大的挫折，有的甚至是彻底的失败，这就有理由让我们怀疑这些为众多软件企业传道解惑的咨询公司们除了为我们获取一张证书外，是否真的有能力来为我们的软件过程改进提供有效的帮助？同时我们更有理由进一步怀疑咨询公司如此卖力的吆喝这场大戏的真正动机是什么？

有一个概念经常被人们有意或无意的混淆，就是基于CMM/CMMI的软件过程改进和CMM/CMMI评审，显然咨询公司们非常热衷的是后者，因为只有后者才能给他们带来丰厚的利润。但是，花上大把银子请几个洋人来为我们的软件企业指点江山显然并不是中国软

件行业目前最需要的。我们目前最需要的是根据本国国情尽快提升整个行业的水平，无论是软件过程还是工程方法，我们都需要的是有针对性的，可操作的，有效的和低成本改进途径。而这恰恰是那些高举CMM/CMMI大旗高歌猛进的咨询公司们很少能够给我们的。

结论

靠政府补贴带动的国内CMM/CMMI热火朝天的“荣景”，在让那些洋评估师们和土咨询公司们赚走大把银子的同时，给中国留下的除了通过评审的数量上的蔚为壮观外，有价值的东西实在少得可怜，而最为严重的是在那些咨询公司的带头鼓噪下，整个业界从政府到企业都似乎把CMM/CMMI当作是让中国软件行业快速壮大并赶超印度并最终成为软件强国的一济灵丹妙药，而忘了低下头脚踏实地的根据自己的实际情况寻找适合自己的软件过程和工程方法。给本已浮躁的行业氛围又在凭添一份虚假的繁荣，这才是我们整个软件行业所应该忧心的。

最后，和大家分享一个可能不是所有人都知道的细节，那张挂在众多公司门口用来光耀门楣的CMM证书，实际上都是由咨询公司的打印机打印出来的，那只不过是一张质地颇为优良的硬纸.....

出处：有效软件工程实践论坛

作者：行者无疆

行者无疆

Nov. 1 2005:

Q (StepIv):

很值得我们反思。为什么 CMMI 目前是这样的一种状态？除了国情等外部环境的因素外，我们自身缺少什么？

A (iamredeye):

给大家 share 一下国外的公司对待 ISO9001 认证的态度，从 management team 传来的声音是：get findings from the auditor by doing it honestly!把我们流程运行的真实情况 show 给 auditor，我们花钱请 auditor 来 audit 是帮我们发现问题的，不是来买证书的！

归根结底是 culture 问题。国内没有一个诚信的土壤。

A (Jane):

"把我们流程运行的真实情况 show 给 auditor，我们花钱请 auditor 来 audit 是帮我们发现

问题的，不是来买证书的！"非常认同!!!!

个人也有这种想法,感觉公司花了好多的钱,请了咨询公司,然后最多就是按照标准的流程走了一下,最后弄个几所谓的证据,就可以顺利过关了.等证一拿到,自己要实施时,还是处处碰壁,最终我们的 QA 人员变成了比文秘都不如的身份.难受.

A (happy895464):

很幸运，我们公司不是这样的，就是 QA 工作真正做起来好难

A (傻丫丫):

政府为了全球的眼睛，通过获证的企业数量证明自己是强国；企业为了用户的眼晴，通过获得的证书数量证明自己是牛家；咨询公司为了企业的眼晴，通过签订的证书数量证明自己是权威；评估公司，为了政府的眼晴，通过发出的证书数量证明自己在干活。

A (xixiaojing666):

深有感触,为了拿证,只有不到 2 级的水平,花杆拿到了 3\4 级的证书,然后按照 3\4 级的流程实施项目,可是组织的能力只有 2 级的水平,你说 CMMI 怎么能运行的下去??

一个没有读过初中的小学生硬要他读高中,能及格嘛!

A (xixiaojing666):

幸运，我们公司过程改进真正的做了些事情。

而我作为我们部门的 QA 负责人员，定会全力以赴。

虽然有很多困难，但我相信前途是光明的

A (lu8023):

幸运，我们公司过程改进真正的做了些事情。

而我作为我们部门的 QA 负责人员，定会全力以赴。

虽然有很多困难，但我相信前途是光明的

A (酷小孩):

我们公司完全就是走形式，没有真正理解 CMMI 的本质，没有给全员做深入的讲解和培训，只是扔过来一堆表，不填就扣绩效

A (jaedonger):

我想起国内餐饮业为什么不能做大做强了。

大家想想 国内 做菜时候的 那些话语“食盐少许，味精适量。。。 ”

估计就不难明白了

A (sungubbi):

记得曾经兴致勃勃地买过一本关于烹饪的书，里面的描述是“生粉二十克、牛肉半斤。。。”，于是这本书的命运是被我束之高阁了。原因是，太不符合我的实际情况了，因为我没有秤、手感也不好不能感觉到轻微重量差别、也不知道生粉和淀粉有啥区别，如果改成生粉半碗、盐一匙之类的，可用性会提高很多，如果再告诉我加一点八角和香料会更适合味重的人、而某种做法更适合感冒的人，那又更好一些。

还有一个笑话，我曾经去一个餐馆点了一道“鱼香肉丝”，结果端出来一看——小鱼干炒肉丝。这个厨师明显不知道川菜中的鱼香肉丝是啥、也一定没有吃过，不过照着名字就做出了一道菜来罢了。

EPG 实施过程的推进，就好比写一本菜谱，一定要让操作者看懂、可操作；并且不同的项目不同的特点，你还得协助他们一起识别他们的需求、他们的口味来对症下药。

A (jaedonger):

确实是这样

这也许就是国人和国外在做 CMMI 上的差距...

所以 SEI 开始调查中国了..

为什么不调查国外?不是国外没有.而是中国做的太嚣张了.不查不行了...

就沿用我的"做饭(菜)"思路:

如果只是做家常便饭,也就无需讲究.毕竟是自己(家人)吃...

但是如果是在外面做厨师类.

最好还是在某些细节下面下下功夫

(CMMI 亦是.)

A (iamredeye):

又看了一下这个旧论坛的老帖子。

我想国内 CMM/CMMI 的失败原因可能包括两个：

一是目的不纯，骗证书。这个不用多说。

二是能力不够，玩不动。正如文中说的那些咨询公司，指点江山头头是道，做自己内部项目却做不成功。还是没有真正领悟这些方法论，弄的形似神不似。更何况方法论仅仅是给一个参考，也许能提高成功的可能性。要灵活运用起来才行。可是难度很大。我们中国人擅长记忆和考试，但是比较死板，不擅长拿来之后改造。

还有其它原因。先不敲了。。。———

另外胡说两句，楼上的说中餐馆不能做大做强是因为“engineering”层面（厨师）不够量化精确。我有点不同看法。中餐未必好吃到哪里去，但决不会因味道差被历史淘汰。其实精确定量的烹调方法并不能做出最好的菜，还是要靠高手靠感觉去拿捏，毕竟烧菜除了 science 还是有点 art 在里面的。软件也是一样，如果用当年泰勒对待工人那一套对待程序员，完全数字驱动。。。

中餐馆不能做大做强还是因为 management 层面（老板）水平不够。软件企业尚且如此，餐饮行业都是些小作坊，管理水平比软件行业的还是差多了。

★ 产品的需求阶段和项目需求阶段区别

Q (songcl):

我个人的理解是：产品开发过程中，需求阶段可以分为商业需求、市场需求、产品需求、软件需求这四个大阶段，但是每一个需求阶段中又可以包括需求分析调研阶段，需求确认、最终形成相关需求文档等过程；而做项目的话，需求阶段则分为用户需求、软件需求两个阶段即可，当然这两个阶段中又各自包括需求分析调研、总结评审、最终形成需求规格说明书的过程。

但是也有人说做项目和做产品的需求阶段没什么区别，本质上是一样的，请大家在此讨论一下，希望能得到各位专家和老师的指点

A (iamredeye):

这个产品和项目的分法是不准确的。不考虑那些不规范的开发方式，你说的这两种都是“项目”。这个几句话说不清楚。

项目最后的输出是产品（当然也可以是一部分“产品”）或服务。product lifecycle VS project lifecycle 这个概念就复杂了，甚至复杂到可能是一个矩阵。我建议你还是先不要追究这些概念了。

回到你的问题上来，本质上这个需求阶段没什么不同，你可以理解为范围大小的问题。

A (chfyx):

产品可以有很多分支，每一个分支可以当作一个项目。

产品可以有很多版本，每一个版本可以当作一个项目。

例如：

针对数码播放器产品，今年做一个项目包含一部分功能，明年针对这个产品在做一个项目，

包含了更多的功能。每一次产品的升级换代都可以当作项目处理。

例如：

针对电信市场的产品，北京电信和江苏电信需求不同，但是都可以以我们的电信产品为基础进行定制开发，这就产生了分支，每个分支就是一个项目

A (scott):

楼主说的产品开发和项目开发的区别应该可以用 chfyx 的回复来解释。

如果真的有产品开发和项目开发的话，那么项目开发应该只是产品开发的一个环节（可能是产品开发生命周期中的开发阶段）。在需求这条线上，整个产品开发的输入是市场给的需求，我们可以叫做原始需求（就是你说的业务需求/市场需求/商业需求，可能是功能，性能，服务，价格因素等等），然后产品开发团队将原始需求转化为产品需求，再转化为系统级规格需求，再转化为系统级设计规格，设计规格以分配需求的形式分配给各个项目开发团队，再有项目开发团队转化为软件需求。

这应该是最复杂的一条线了。

如果只是纯软件开发，那么产品需求可以直接转化为分配需求，分配需求转化为软件需求；如果进一步说产品需求中的内容都可以由软件实现，那可以直接由产品需求转化为软件需求。这里分配需求肯定包含功能和特性以外的需求，比如“性价比必须达到业界最高水平。”

对于所有需求的跟踪过程来说，那是没有什么区别的。但不同的人可能会负责不同的需求。

A (iamredeye):

楼上说的很清楚。

更复杂的情况是一个项目跨了多个产品。long story。。。

2 全员改进案例

★ 帮这个项目经理想想办法：新老产品交替

Q (sungubbi):

从网上看到一个案例，拿来与大家共同讨论讨论：

我们公司是一家软件开发企业。现状是目前有个产品，但是存在很多问题，准备开发新版本，之前版本存在的问题有一部分应该说是开发人员的问题，但是因为我来得时间不长，不甚了解。整个公司都对新版本给予厚望，就目前的情况来说，老板希望技术团队有个大的改造，但是我不想这么做，因为毕竟新人进来不见得合适、而且需要相当一段时间

的业务熟悉期。现在老板问我怎么做？我觉得可以有这么几步：

- 1、老板主持个动员大会；
- 2、完善部分规范、制度；
- 3、增加一些激励措施；
- 4、我同他们加强沟通，了解每个人的具体情况，帮助他们
- 5、加强项目监控

大家一起讨论讨论，该怎么做？

A(zhuhua0724)：

弄清楚团队和产品的问题所在先。

A(lily_014)：

换人不太可取，正如该经理所说，新人不见得合适，况且就算合适也需要一段熟悉业务的时间和团队的磨合。再者，原帮人马对业务相对来说也会较熟悉。

如果我是该项目经理，

通过与团队人员沟通（也可以单独谈话或一次活动在轻松的环境下拉近距离，毕竟该经理也是新来的，首先要获得大家的信任）知道大家的想法，以及针对产品存在的问题作具体的了解。

在相关的情况有所了解之后，采取相应的措施。（比如 人员的激励机制：加薪或升职，加强管理制度等）

老板要给予承诺，且要让项目组的人确切感觉到老板是在重视，而不是口头上说说，给大家画了一张大饼。

加强沟通与管理。

如果确实是因为技术问题，若在新的版本开发中时间允许最好是从架构开始重新来过。然而选择生命周期也是重要的因素。。。

一个好的领导，往往是成功的关键。

作为项目经理，首先要融入群众，不要高高在上的感觉，那样开发人员容易产生抵触情绪。但，说的融入群众 不是说不论何时都与大家一起打闹，工作之余适当的可以多关心关心项目组成员，多了解大家的内心想法。工作中，还是要树立威信，原则性的问题该怎么办就怎么办。

再者 就是由于本次是产品的升级，原有的产品应该是有个使用反馈的，如果以前没做，现在也可以做个调查（说白了就是调研），这个往往是新产品的需求的关键基础。

A(sungubii)：

从产品和团队管理两方面入手

其一：先从产品的需求角度来说，由于这是一个新的产品，面临新的架构，老板、员工、顾客对这个产品都充满着期待，并希望用它来替换老产品，所以项目经理的当务之急是组织调研产品的需求，这也是新进公司的 TA 最好的熟悉产品的机会。TA 可以不急于成立一个项目团队，而是先组建一支需求调研的队伍，分别向最终使用客户、安装人员、老产品的开发团队、公司管理层、顾客调研他们对新产品的期待，并结合公司的技术发展战略，来获得产品需求。只有基于这些需求，TA 才能更好的进行下一步的规划。

基于需求调研和分析的结果，由需求调研组形成用户需求，并初步形成新产品的方案，并求得相关方对方案的共识。基于达成共识的方案，项目经理开始筹划项目的资源、成本、

进度、周期、风险识别等等。

其二：老产品失败的地方就是新产品可以识别出的风险。TA 应该去了解老产品失败的原因，为什么说是开发人员造成的？是出于技术方面、规范方面还是激励方面？TA 应该通过各种渠道尽量获得这些信息，并在项目组建初期通过措施来规避这些风险。项目经理还可以适当从老产品的项目组中调一些人员进入新项目组，或从公司的技术队伍、产品规划部门中申请一些资源加入新的项目组，以确保产品的技术、产品规划与公司保持一致。

其三：建立团队公约，清晰岗位的职责，建立团队工作正常流转的机制，定义团队工作过程规范

补充：

需求调研开始前，建议在公司内部通过一个讨论会来收集公司高层、技术层、产品规划部对新产品的定位。如果没有求得对新产品定位的共识，将会导致需求调研时方向不明确。另外，如果新产品定位不过就是对老产品的重复实现，而在技术架构、功能、风格方面没有明显的改变，可能需要考虑是否有必要立项。

3 项目管理

★ 软件项目管理的十个误区？

Q (fanyuetian):

随着计算机硬件水平的不断提高，计算机软件的规模和复杂度也随之增加。计算机软件开发从“个人英雄”时代向团队时代迈进，计算机软件项目的管理也从“作坊式”管理向“软件工厂式”管理迈进。这就要求软件开发人员特别是软件项目管理人员更深一步地理解和掌握现代软件工程的理论方法，完成思想观念上的转变。笔者在此分析了 10 个在现代项目管理中思想观念上容易陷入的误区，希望能够抛砖引玉，引发大家更多的思索和讨论。

误区 1：在项目的需求分析阶段，开发方与客户方在各种的问题的基本轮廓上达成一致即可，具体细节可以在以后填充。因为无论开始时有多么细致，以后对需求的修改几乎是必然的。分析：这是一种非常危险的思想。实际上许多软件项目失败的最主要的原因就是需求阶段对问题的描述不够细致，导致后来预算超出或者时间进度达不到要求。正确的做法是：在项目需求分析阶段，双方必须全面地尽可能细致地讨论项目的应用背景、功能要求、性能要求、操作界面要求、与其他软件的接口要求，以及对项目进行评估的各种评价标准。并且，在需求分析结束以后，双方还要建立可以直接联系的渠道，以尽早地对需求变动问题进行沟通。(范围的核实和项目验收都要根据范围基准进行。因此前期的范围说明书和范围的基线至关重要)

误区 2: 软件项目的需求可以持续不断的改变, 而且这些改变可很容易地实现。分析: 的确, 在具体实际中由于种种原因客户方很难在需求分析阶段全面而准确地描述所有问题。随着开发进度的推进, 往往会有一些需求的改变。而现代软件工程理论也利用软件的灵活性特点通过各种方式来适应这种情况。不过, 这并不表明“软件项目的需求可以持续不断的改变, 而且这些改变可很容易地实现”。实践表明: 随着开发进度的推进, 实现软件需求更改所需要的代价呈指数形式增长。假定在需求分析阶段实现需求更改需要花费 1 倍的代价; 那么, 在系统设计和编码阶段, 需要花费 1.5-6 倍的代价; 在系统测试阶段需要花费 10-20 倍的代价; 在软件版本发布以后, 甚至可能要花费 60-100 倍的代价。由此可见, 在项目开展过程中, 软件需求的改变应当尽量早地提出。这样才可能花费少, 容易被实现。(不应该称为误区了, 现在估计谁都不会认为需求可以持续不断改变)

误区 3: 软件程序主要由代码组成, 因此编码阶段是整个软件项目的最重要的阶段, 应该给与大量的时间, 并且集中主要的资源。分析: 与以前相比, 由于软件的规模和复杂度的增加, 以及半自动化软件代码开发平台的出现, 现代软件项目管理的中心发生了转移——不是着重编码阶段, 而是着重系统总体/详细设计阶段。一般说来, 在现代软件项目管理中各种资源的合理分配比例是: 项目论证、风险评估阶段 3%, 项目需求分析阶段 8%, 系统总体/详细设计阶段 45%, 编码阶段 10%, 系统测试阶段 34%。(这个跟软件项目的规模密切相关。对于规模小于 2 万行代码的, 或者说采用敏捷或快速开发的, 或者说架构已经确定的改进型号项目, 编码时间至少要占 30%; 而对于源代码规模超过 50 万行的大型软件项目, 重点则是在需求和系统设计上面, 编码时间一般为 10%)

误区 4: 为了便于代码的维护修改, 在系统的详细设计阶段文档工作应该做到写出所有程序的伪码。分析: 通常伪码的最大作用是对程序的算法流程进行描述, 便于人们深入了解程序的功能和实现过程。可见, 在一定程度上伪码的确有利于对程序代码的维护和修改。但是, 我们知道为了保证项目文档和程序代码的一一对应关系, 维护程序代码的时候同时需要对项目文档进行维护。伪码和程序代码是非常接近的, 对伪码进行维护的话, 相当于进行了 2 倍的程序代码维护。工作量是很大的。所以切合实际的方式应该是对一般的程序文档做到程序流程图即可, 对于涉及了较复杂算法的才需要伪码。(应该深刻理解源代码就是设计的一些重点观点和思路, 因此详细设计输出的代码模型一般是不抛弃的, 编码人员可以直接在该代码模型基础上进行编码)

误区 5: 既然在项目人员配置中设置了专门的测试人员, 那么软件所有的内部测试工作全部应该由测试人员完成。分析: 软件程序测试可以分为“白盒法”和“黑盒法”两种

方式。由于使用“白盒法”对测试人员各方面素质的种种要求，在进行程序测试时测试人员总是最优先使用“黑盒法”。他们的工作方式往往是先对程序进行“黑盒法”测试；如果测试没有通过，不得已这才考虑对程序代码进行“白盒法”测试。显然，这种对“白盒法”有意无意的“逃避”，对软件的可靠性和稳定性构成了威胁。如何解决这个问题？一方面需要提高对测试人员的要求，另一方面也需要程序员完成部分的“白盒法”测试（实际上，程序员往往也是进行“白盒法”测试的最佳人选）。(估计很少有人这样认为，所以不应该称为误区)

误区 6：软件项目管理只是相关技术部门的事情，与公司其他部门无关。分析：在竞争日益激烈的今天，软件项目规模大、复杂度高而且时间要求紧迫。要想提高公司的软件项目管理水平，这就需要提高公司的整体参与意识，需要公司各个部门协同作战。例如需要会计部门协助进行项目预算，财务管理和费用控制；需要研究部门（技术委员会）指派专家协助进行各种风险评估，提供技术指导；需要后勤部门提供各种保障。(干系人管理很重要,同时 CMMI 强调的集成项目管理也说明了这一点)

误区 7：在开发进度滞后的情况下，可以聘请更多的程序员加入到开发团队中，通过增加人力资源来赶上进度。分析：在注重团队开发的时代，开发方应该根据目前的软件项目管理水平慎重考虑这个做法。如果新加入的程序员对目前软件项目的应用行业有一定了解，并且可以很快适应了开发方的项目管理方式、软件开发风格、团队协作氛围；那么“新人”的加入是有益的。否则，可能会“好心好意做坏事”。因为尽管其个人能力很高，但是为了使其与大家一起协同工作，开发团队不得不分出人手对其进行与项目有关的技术/业务培训，更重要的（也是难度最大的）是还要引导其融入团队。这可能需要花费开发团队许多时间和精力，很有可能使项目进度更慢。(可以辩证的看，组织的成熟度越高，复用越高，该方法才是可能的方法)

误区 8：技术骨干应该成为项目的项目经理，项目经理一定是所有项目成员中薪水最高的。分析：在“软件作坊”时代，这是一种普遍使用而且效果不错的方法；而在“软件工厂”时代，这种方法却带来各种问题，有时甚至直接导致项目失败。究其原因这主要是因为随着现代软件开发分工的细化，对项目经理的要求也发生了根本的改变——最注重的不是其对某项专业技术的掌握程度，而是其组织、领导、协调开发团队的能力（当然，可以两者均突出最好）。至于项目经理的薪水问题，这和定薪制度有很大关系。通常，项目经理执行的是管理人员的薪酬体系，而其他人员执行的是技术人员的薪酬体系。项目经理的薪水在项目成员中是比较高的，但不一定是最高的。有时候，为了激励技术人员，项目中

的技术骨干得到的酬劳比项目经理要高。(这里跟工种无关,更相关的是效率和产出,但去很难推行)

误区 9: 只有项目经理以及部门主管才会关心项目整体进度,程序员只关心自己的开发进度。分析: 这是一种“官僚”的想法。实际上程序员作为团队中的一员,他不仅仅是在打一份工,更重要的是在参与一件“作品”的创作。在体味工作的辛苦的同时,程序员更重要的是要享受创作的快感。项目经理不应该漠视程序员对“成就感”的追求,应该向每一个人详细描述最终“作品”将会如何美妙和令人兴奋,并且在到达最终目标的路上设立一系列的里程碑。每当项目整体推进到一个里程碑的时候,项目经理应该把这个消息告诉每一位项目成员。实际上,这不仅仅可以让所有的项目成员享受到阶段胜利的喜悦,还可以激发大家更大的工作热情,提高工作效率

误区 10: 为了保证项目继续,为了留住核心程序员,加薪吧。分析: 加薪可以说是很多企业在挽留程序员时所使用的常用方法。这一招可能暂时奏效,不过往往是人留下来了,但副作用也来了——加薪的人未必见得多干活,没有加薪的人却开始消极怠工了。其实,项目的进行过多地依赖程序员的个人技术是“作坊”时代沿袭下来的“陋习”。既然 IT 行业人员的流动是无法控制的,现在项目的执行应该更加注重团体的力量,应该更多的考虑公司整体技术水平和核心技术能力。例如形成公司自己的专家知识库,类/函数库,第三方控件库,拥有自主知识产权的开发平台等。另外,实际上程序员萌生去意的原因很大程度上不是薪水,而是缺少激励和尊重。这需要项目经理使用“老土”一点的办法,找适当的时机对程序员做一做思想工作,向其描述项目的美好未来,让其感受关心和尊重。总之,要多方面着手保证项目的顺利开展,而不是简单地加薪。

★100 个管理学定律

Q (fanyuetian):

一、管人用人育人留人之道

1. 奥格尔维定律: 善用比我们自己更优秀的人
2. 光环效应: 全面正确地认识人才
3. 不值得定律: 让员工选择自己喜欢做的工作
4. 蘑菇管理定律: 尊重人才的成长规律
5. 贝尔效应: 为有才干的下属创造脱颖而出的机会
6. 酒与污水定律: 及时清除烂苹果

- 7.首因效应：避免凭印象用人
- 8.格雷欣法则：避免一般人才驱逐优秀人才
- 9.雷尼尔效应：以亲和的文化氛围吸引和留住人才
- 10.适才适所法则：将恰当的人放在最恰当的位置上
- 11.特雷默定律：企业里没有无用的人才
- 12.乔布斯法则：网罗一流人才
- 13.大荣法则：企业生存的最大课题就是培养人才
- 14.海潮效应：以待遇吸引人，以事业激励人

二、以人为本的人性化管理

- 15.南风法则：真诚温暖员工
- 16.同仁法则：把员工当合伙人
- 17.互惠关系定律：爱你的员工，他会百倍地爱你的企业
- 18.蓝斯登定律：给员工快乐的工作环境
- 19.柔性管理法则：“以人为中心”的人性化管理
- 20.坎特法则：管理从尊重开始
- 21.波特定律：不要总盯着下属的错误
- 22.刺猬法则：与员工保持“适度距离”
- 23.热炉法则：规章制度面前人人平等
- 24.金鱼缸效应：增加管理的透明度

三、以人为本的人性化管理

- 15.南风法则：真诚温暖员工
- 16.同仁法则：把员工当合伙人
- 17.互惠关系定律：爱你的员工，他会百倍地爱你的企业
- 18.蓝斯登定律：给员工快乐的工作环境
- 19.柔性管理法则：“以人为中心”的人性化管理
- 20.坎特法则：管理从尊重开始
- 21.波特定律：不要总盯着下属的错误
- 22.刺猬法则：与员工保持“适度距离”
- 23.热炉法则：规章制度面前人人平等
- 24.金鱼缸效应：增加管理的透明度

四、灵活有效的激励手段

25. 鲶鱼效应：激活员工队伍
26. 马蝇效应：激起员工的竞争意识
27. 罗森塔尔效应：满怀期望的激励
28. 彼得原理：晋升是最糟糕的激励措施
29. “保龄球”效应：赞赏与批评的差异
30. 末位淘汰法则：通过竞争淘汰来发挥人的极限能力
31. 默菲定律：从错误中汲取经验教训
32. “垃圾桶”理论：有效解决员工办事拖沓作风
33. 比马龙效应：如何在“加压”中实现激励
34. 横山法则：激励员工自发地工作
35. 肥皂水的效应：将批评夹在赞美中
36. 威尔逊法则：身教重于言教
37. 麦克莱兰定律：让员工有参加决策的权力
38. 蓝柏格定理：为员工制造必要的危机感
39. 赫勒法则：有效监督，调动员工的积极性
40. 激励倍增法则：利用赞美激励员工
41. 倒金字塔管理法则：赋予员工权利
42. 古狄逊定理：不做一个被累坏的主管

五、沟通是管理的浓缩

43. 霍桑效应：让员工将自己心中的不满发泄出来
44. 杰亨利法则：运用坦率真诚的沟通方式
45. 沟通的位差效应：平等交流是企业有效沟通的保证
46. 威尔德定理：有效的沟通始于倾听
47. 踢猫效应：不对下属发泄自己的不满
48. 雷鲍夫法则：认识自己和尊重他人
49. 特里法则：坦率地承认自己的错误，崇尚团队合作精神，比尔·盖茨说：“团队合作是企业成功的保证，不重视团队合作的企业是无法取得成功的。”建设一支有凝聚力的团队，已是现代企业生存发展的一个基本条件。
50. 华盛顿合作定律：团队合作不是人力的简单相加

- 51. 木桶定律：注重团队中的薄弱环节
- 52. 苛希纳定律：确定最佳管理人数)
- 53. 凝聚效应：凝聚力越大，企业越有活力
- 54. 懒蚂蚁效应：懒于杂物，才能勤于动脑
- 55. 蚁群效应：减掉工作流程中的多余
- 56. 飞轮效应：成功离不开坚持不懈的努力
- 57. 米格—25 效应：整体能力大于个体能力之和

六、决策是管理的核心

- 59. 吉德林法则：认识到问题就等于解决了一半
- 60. 手表定律：别让员工无所适从
- 61. 皮尔斯定律：完善培养接班人制度
- 62. 羊群效应：提升自己的判断力，不盲目跟风
- 63. 自来水哲学：大批量才能生产出廉价产品
- 64. 松下水坝经营法则：储存资金，以应付不时之需
- 65. 巴菲特定律：到竞争对手少的地方去投资
- 66. 吉格勒定理：设定高目标等于达到了目标的一部分
- 67. 卡贝定律：放弃有时比争取更有意义
- 68. 布利丹效应：成功始于果敢的决策
- 69. 普希尔定律：再好的决策也经不起拖延
- 70. 沃尔森法则：把信息和情报放在第一位!
- 71. 哈默定律：天下没有坏买卖!
- 72. 隧道视野效应：不能缺乏远见和洞察力
- 73. 青蛙法则：时刻保持危机意识
- 74. 坠机理论：依赖“英雄”不如依赖机制
- 75. 奥卡姆剃刀定律：不要把事情人为地复杂化
- 76. 帕金森定律：从自己身上找问题

七、创新是企业的生命

创新是企业发展动力的内核，是市场竞争的必然结果。企业只有创新才可以打破常规，突破传统；只有不断创新，才能在激烈的竞争中永远立于不败之地。

- 77. 达维多定律：不断创造新产品，同时淘汰老产品

78.路径依赖：跳出思维定势

79.跳蚤效应：管理者不要自我设限

80.比伦定律：失败也是一种机会

八、竞争决胜的智慧与策略

21 世纪是一个充满竞争的时代，企业生存的最大武器就是竞争。在这场较量中，对竞争方法、竞争策略以及竞争手段的管理，将成为企业决定胜败的关键因素。

81.犬獒效应：让企业在竞争中生存

82.零和游戏原理：在竞争与合作中达到双赢

83.快鱼法则：速度决定竞争成败

84.马太效应：只有第一，没有第二

85.生态位法则：寻求差异竞争，实现错位经营

86.猴子—大象法则：以小胜大，以弱胜强

九、成也细节，败也细节

细节的不等式意味着 1% 的错误导致 100% 的失败。许多企业的失败，往往是由于细节上没有尽力造成的。把任何细节做到位，企业就不会存在问题。

87.破窗效应：及时矫正和补救正在发生的问题

88.多米诺效应：一荣难俱荣，一损易俱损

89.蝴蝶效应：1% 的错误导致 100% 的失败

90.海恩法则：任何不安全事故都是可以预防的

91.王永庆法则：节省一元钱等于净赚一元钱

十、打好营销这张牌

92.凡勃伦效应：商品价格定得越高越能畅销

93. “100-1=0” 定律：让每一个顾客都满意

94.鱼缸理论：发现客户最本质的需求

95.长鞭效应：加强供应链管理

96.弗里施法则：没有员工的满意，就没有顾客的满意

97.250 定律：不怠慢任何一个顾客

98.布里特定理：充分运用广告的促销作用

99.尼伦伯格法则：成功的谈判，双方都是胜利者

100.韦特莱法则：从别人不愿做的事做起

4 QA/EPG 专区

★ 如何进行质量分析？

Q (xixiaojing666):

我们目前的质量分析做的很简单，只是按照 PA、过程域、NC 的严重程度进行简单的统计。

大家是如何进行质量分析的？听说可以使用工具、图表、数据等等？

A (fishred):

- 1、一方面根据度量数据进行分析（度量数据来源度量计划）；
- 2、另一方面，针对发现的项目问题进行分析，一般根据 PA 来分析；

至于工具嘛，我们目前用 Excel 自带的统计、图表功能；只有缺陷管理用工具（TD）进行统计了；

A (stop-start):

乍一看这个标题，我的第一个反应是项目的质量分析，再细看一下原来 LZ 想交流的是 QA 所做的质量分析，如果只是 QA 所做的质量分析的话，感觉与“全员过程改进”板块中，sungubbi 最近提出的“一起聊聊：如何量化评价项目执行情况”这个话题类似，她这个话题涵盖了“QA 对过程执行情况的评价”与“如何使用测量的结果”两个问题。LZ 可以参考一下那边的一些交流内容

A (stop-start):

另外我很赞同 fishred 的“质量分析分为两方面，一方面是根据项目度量数据进行的，一方面是针对 QA 发现的问题进行的”这个观点，我想再把根据项目度量数据进行的质量分析这个话题引申地具体一点，当然，以下都是个人的一些观点，希望得到各位的指正和补充：

根据项目度量数据进行的质量分析，主要是针对项目实际数据超出计划预估值规定偏差范围时所做的分析，比如规模、进度、缺陷数等，有的要求比较严的项目还会开展缺陷的根原因分析，这一般都是在项目阶段结束及项目结项的时候实施。分析的主体责任人是项目负责人，QA 在这时的职责主要是核对数据，如果有异常数据出现，提醒或协助项目负责人进行分析。这种分析一般用 excle 文档就可以实现，根原因分析一般会运用到鱼骨图这种工具。

A (lily_014):

商业目标不同，侧重点也不同。

进行质量分析，我理解的是度量（不知理解是否正确），根据不同的目标定义相应的质量指标。至于如何进行质量分析，同意 stop-start 说的结合 sungubbi 的贴子讨论。要形成了相应的性能基线才能够分析。否则也只是进行了数据的采集工作。即也就是进行了针对某个具体项目的进度、成本等估算的偏差。个人意见。

★ 流程不及时启动或不及时处理作为 QA 该怎么办？

Q (cecilia):

目前对那些不及时启动流程或不及时处理流程的，很困惑，不知道应该怎么来解决这种现状。大部分人员启动流程后，就不闻不问了，反正流程会走向下一个节点负责人，都由 QA 来跟催了，这样下来导致一个流程从开始启动到最后办结的时间很长。（为此，有些人以此为借口，说一个评审流程需走好久，说走流程太麻烦，反倒以此说延迟了进度）

请大家说说自己的看法，帮我提提意见吧，一直都很困扰，如何让大家都能自觉地按规范及流程来做事呢？

A (iamredeye):

什么是流程呢？文档？做事方式？建议：

- 1) 不要把文档当流程；
- 2) 不要认为流程就是标准，不妨认为流程是 guideline；
- 3) 如果流程完全被“鄙视”，一定要反省流程的制订

A (cecilia):

公司制定流程，就是为了让开发的各个环节不会失控，让每一个成员都能在预计时间里完成相应的任务。

当然流程肯定不是最完美的，但也需启动之后运行了才能发现流程中的不足，再进行改进。

A (Scott):

你说的意思是流程已经工具化，而大家不能及时通过工具走完流程吧？

我觉得这很正常，目前的解决办法，一是添加邮催功能，二是谁发起的流程谁负责关闭（不太建议的做法，考核发起人关闭流程的有效性）。

其实反过来也像 iamredeye 说的，你们的流程太复杂或者不适合，在我们公司也是一样的，一到审核的时候大家就催缴流程走完，文档提交，怪只能怪领导太愚蠢，做工具的人没脑子，还能怎么样呢～～

A (cecilia):

流程制定出来了就要用，不用又怎么知道是否合适呢。

A (iamredeye):

我知道很多公司都是这样的，流程就是要无条件 follow 的，虽然实际上遵守的情况还是有（很大？）折扣的。这个公司文化有关，所以我不好说什么是对什么是错的。

不过不妨参考一下这种思路：（很多情况下）把流程当作一个 guideline，遵守的情况总是介于 0 和 100 之间。项目根据实际情况综合各方面意见选择 compliance 的度。所以总是允许存在一定的 flexibility 的。

换句话说讲，即使不承认流程规定和实施之间的 gap，这个 gap 也是客观存在的。所以不妨换个角度思考。

A (xixiaojing666) :

制订出来的流程先固化，不管它有什么缺点，先强制大家做起来，既然流程制订出来了，我想优点肯定大于缺点的。等大家习惯这个流程之后，在漫漫的优化！

A (cecilia) :

目前使用的 FIOA 并不能实现并行的操作流程，为此对于流程来说也是一个缺陷，为了实现并行的过程，只好让启动者多选择几次相关执行人，如果使用串行，那对于整个工作都是有影响的。

现在主要做的就是督促相关人员启动新的流程，并教其如何使用，以便能在工作当中广泛的使用，万事开头难。

A (Scott) :

先固化再优化需要很好的执行力环境，软件人员最不喜欢的就是强制，你越强制可能得到结果越是相反的。

另一方面，作为一个 QA，你不是想整天都在那里教授和提醒软件人员该启动流程了，该走流程了...工具是 guideline 的一种，是一种辅助性的内容，就像你可以用 VB 写模式，你也可以用 VC 写模式。如果一件事只能用一种方法来做，那它就是流程而不是指南，而且是很不灵活的流程。

同时，太灵活的流程也不好，你觉得串行很慢，可并行的组合就是很多的，入口多、出口多就很难管理，就和软件人员多头管理一个道理。

我对工具的看法是可以提升到流程层次，但一定是先培训并得到认可才实施。如果部分内容不能认可，那就要试用，然后再改善。流程是需要 PDCA 的，如果工具是流程，那么也需要 PDCA，请问你们工具上次 PDCA 是什么时候？

如果工具是指南，那就是可选的，不是强制的，比如你们有评审工具，但是也应该有 Excel 的，甚至 TXT 的，只要能达到充分、尽早发现缺陷的目的就可以。

QA 一味强调工具、强调执行，只会让你和项目越走越远；当然，反过来不强调会让你离管理层越来越远。所以，要善于平衡。

A (jane) :

这个，好像跟我公司比较相似哎，不过，我们公司连个可以走的流程都没出来。只能有个开头，中间就没有戏了。

我个人觉得，推广流程的事情，是需要慢慢引导，比如 QA 以辅助的形式，帮助项目组将流程走下来，通过这个过程，可以发现一些流程中的问题，改进后，再继续引导改进后的流程。当项目组成员慢慢习惯后，那就成功了。当然要养成这个习惯是相当不容易的，因为习惯也有好坏之分，作好充分思想准备吧，这是条艰难的路。

C (jane) :

流程需要 QA 人员推动它顺畅流通，这需要一定的技巧，在强制与随意之间随机生成。无论是流程的适应到达什么程度，都不能把它当成是一个强制的标准去推。说俗点就是，别人会觉得，站着说话不腰疼。当然，对于一些非常重要的过程，还是要采取强硬态度，做起来。不然，可能等于质量没有任务保证，在钢丝绳上跳舞那么悬。

★ 如何深入理解验证与确认？二者的区别？

Q (songcl) :

测试中既包括验证测试也包括确认测试;而评审也可以理解是一种验证和确认的方式。

验证: 我们正正确地制造产品吗?

确认: 我们正制造正确的产品吗?

顺着此定义理解: 验证是侧重于 SQA 的职责, 即保证过程正确执行, 以正确的方式来制造产品; 而确认则是可以理解为测试者的工作, 确认一下是否符合软件需求。但总感觉还不够深入, 请高手更深入更精辟的阐述。

A (fanyuetian):

验证: 验证是检查某样东西是否符合之前已定好的标准, 如: 文档评审, 要检查的东西是文档, 检查标准就是文档的评审标准, 又如: 测试软件, 要检查的东西就是软件, 检查的标准就是软件的规格说明, 包括功能说明, 性能要求等。

确认: 检查软件在最终的运行环境上是否达到预期的目标。一般来说, 就是调试、验收测试等, 这些工作都是在真正的软件需要运行的环境上进行的, 在最终环境上运行软件, 确保软件符合使用要求

注:

Validation 更多是从用户的角度或者可以是模拟用户角度来验证产品是否和自己想要的一致。从 SP 来看可能更容易理解这个;

Verification 更多的是从开发方的角度来做评审、测试来验证产品的需求、架构设计等方面是否和用户要求的一致;

A (lily_014):

楼上的说得好, 通俗易懂

理解和区别这两个概念就好了

A (songcl):

分别从用户和从开发方的角度来理解, 区分得更明确一些! 谢谢 fanyuetian!

A (lee_huo):

val, 保证做对的产品; 注重客户满意度的思想。

Ver, 保证做的对; 实质是需求吻合的思想。

C (cecilia):

以下为 CMMI-V1.2 中对验证和确认的定义:

验证(Verification, VER)的目的, 在于确保选定的工作产品符合其指定的需求。验证包括产品及中间工作产品的验证, 将其与选定的客户需求、产品需求及产品组件需求加以比较。

确认(Validation, VAL)的目的, 在于展示置于预期环境中的产品或产品组件, 可满足其预期的使用需求。

确认证明所提供的产品符合预期的使用需求, 而验证说明工作产品是否适当的反映了特定需求。换言之, 验证确保“你把事做对了” (do things right), 而确认确保“你做了对的事” (do right things)。

通过以上定义, 我想大家应该能理解验证与确认的真正区别, 通过后期跟进项目实施, 过程检查, 项目确定等环节, 一定可以深入地理解的。

5 CM 专区

★关于产品分支管理的讨论

Q (sungubbi):

求教：以下问题，与大家共思考。如果有配置高手能够写成原创就更好啦。

同一个产品分两条线进行开发（例如一个开发新功能，一个进行日常维护开发），并在开发进展到某一个阶段后，再归并形成新的产品版本。由此引发下疑问：

- 1) 产品分支开发的前提：什么情况下比较适合进行产品分支管理？对产品的特点、分支的周期、工作团队是否有要求？
- 2) 分支管理的工具：现有的配置工具哪些比较适合进行分支管理？是否有独立于配置工具的 Merge 工具？
- 3) 文档又如何实现分支与归并管理？
- 4) 使用配置工具分支 merge 时，工作量有多大？常见问题？
- 5) 产品产生分支后，管理上需要注意什么？

A (iamredeye):

可能这些问题太大，所以大家很难回答。分支管理比较强大的工具大概是 ClearCase 了。你也问了文档相关的问题。我在以前的旧论坛问过类似的问题。如果我没有理解错的话，这是 product vs project 的 CM，我感觉比较难。你有答案了不妨分享一下！

A (iamredeye):

一什么情况下比较适合进行产品分支管理？

不同的产品针对不同的市场或客户

一现有的配置工具哪些比较适合进行分支管理？

我知道最好的就是 clearcase。也比较复杂，管理成本高。

一文档又如何实现分支与归并管理

这个比较难，也是我想知道的。

一使用配置工具分支 merge 时，工作量有多大？常见问题？

应该比较容易有冲突—》rebaseline; delicate branch planning

一产品产生分支后，管理上需要注意什么？

Release management 和 project management 都要跟上

A (fishred):

—什么情况下比较适合进行产品分支管理？

不同的产品针对不同的市场或客户/并行开发，产品分批上线

—现有的配置工具哪些比较适合进行分支管理？

我们在用 CC、PVCS

—文档又如何实现分支与归并管理

这个我们没实现，也没这个需求，主要精力放在代码版本上

—使用配置工具分支 merge 时，工作量有多大？常见问题？

工作量还是比较大，要打标签、去对应标签的代码、进行代码合并、版本控制、处理冲突等，当需求回退时要合并版本等；

—产品产生分支后，管理上需要注意什么？

每月制定分支计划、配置审计等

A (step365he): 产品的可维护性和可集成性越好，尤其是产品由较好的较标准的组件构成，分支管理越不易出错。

A (xixiaojing666):

使用分支的朋友赶快进来讨论拉。

—什么情况下比较适合进行产品分支管理？

相同的产品针对不同的市场或客户/并行开发，也就是所谓的特殊版本，当一些特殊版本变成通用版本的时候，应当就是合并的时候

—现有的配置工具哪些比较适合进行分支管理？

目前正在实践

—文档又如何实现分支与归并管理

我觉得这个和代码的分支与合并应该一致吧！

—使用配置工具分支 merge 时，工作量有多大？常见问题？

工作量具体有多大正在实验中，但是我觉得常见的问题应该是合并出错吧

—产品产生分支后，管理上需要注意什么？

我觉得还是平时的配置审计，不过频率也就是关切度加强一下。

A (iamredeye): to xixiaojing666

感觉起来是一致的，但是仔细想想就不那么简单了。比如说，代码是纯文本的，合并起来理论上没大问题；文档呢？word, excel, UML, MPP...

Q (xixiaojing666): to iamredeye

我觉得文档合并都应该是同类型的进行合并吧？

A (iamredeye):

是的。我不是说要把 UML 搞到甘特图里嘛。

C (坠入深海):

同一个产品分两条线进行开发（例如一个开发新功能，一个进行日常维护开发），并在开发进展到某一个阶段后，再归并形成新的产品版本。

1、什么情况下比较适合进行产品分支管理？

不同的产品针对不同的市场或客户/并行开发，产品分批上线的时候；

2、现有的配置工具哪些比较适合进行分支管理？

CC、PVCS 等，CC 比较复杂，管理成本高；

3、文档如何实现分支与归并管理

比较有难度；

4、使用配置工具分支 merge 时，工作量有多大？常见问题？

工作量还是比较大，要打标签、去对应标签的代码、进行代码合并、版本控制、处理冲突等，当需求回退时要合并版本等；

5、产品产生分支后，管理上需要注意什么？

发布管理和项目管理都要跟上，每月制定分支计划，配置审计等。

这些问题还是比较有难度，重要的是要探索出适合自己公司的最佳做法，尤其在分支合并的时候处理冲突会花费比较大的工作量，一般来说，产品的可维护性和可集成性越好，尤其是产品由较好的较标准的组件构成，分支管理越不易出错。

6 CMMI 专区

★ 神坛之上的 CMM——解析 CMM 和中国软件

Q (steplv):

CMM/CMMI 是中国软件外包走向世界的不二法门？——自欺还是欺人？

能够帮助中国的软件外包企业走向世界一直是咨询公司们不遗余力推广 CMM/CMMI 的一个重要的说词，那么果真就是这样吗？

目前外包市场上主要分为“对日外包”和“欧美外包”，日本人对我们中国人好像从来就不怎么相信，给我们的外包绝大多数都是比较初级的编码工作，有些甚至将厚厚的伪码都拿来过来，我们做的基本上都是些体力活，同时承接对日外包的公司大多有日资的背景，且

大多都延用了日本企业的那一套管理和协调机制。

相对来说欧美人对我们国人的期望要高一些,不过本人曾经有幸在境外参加了一个旨在为国外发包商和国内软件企业牵线搭桥的科技展会,接触了不少有意向中国软件公司发包的外商,他们最关心的是两个事情,一个是对知识产权的保护,另一个则是人力资源上是否具备和甲方进行有效沟通的能力。关于开发的管理,有一个老外的话让我感到深深的悲哀,他说,如果在美国,一个企业如果拿出任何能够证明其公司管理能力的资料,他都不会有任何怀疑,而在中国:"No, I only believe my eyes!"

有一个很有趣的事例是,国内有一家非常知名的CMM咨询公司的老板的一个美国朋友应邀来到中国,希望为其在美国的一个大项目在中国寻找开发商,该公司拥有大量的实施过CMM/CMMI的客户,其中不乏专门从事软件外包的,按说在其中寻找一个满意的开发商应该是唾手可得,然而事与愿违,所有的尝试都在第一个环节就不得不终止,原因非常简单,因为所有的候选企业都没有派出一个能在技术、业务、管理和商务层面和美方有效沟通的人来,更有甚者,有些公司干脆拿了一大堆的中文资料就来和老美沟通,结果可想而知。最后,这位老美不得不放弃了这个计划。

显然这一切都和CMM无关。

软件过程改进及其基础构建——我们在空中楼阁中舞蹈

所谓的“软件过程改进”,应当按照在一个循序渐进和有序的过程进行,当初SEI制定CMM的应用对象都是非常大型的航天和军工软件的承包商,这些企业基本都具有相当的规模和良好的规范化基础。显然,对这样的企业实施基于CMM/CMMI的软件过程改进是有意义的。反之,如果是一个完全缺乏规范管理的软件公司来说,连基本的软件工程规范和质量管理理念都不存在,盲目追求所谓CMM的某个级别只能是空中楼阁。

不幸的是,这种完全管理无序的公司在国内实施CMM/CMMI的企业中还不是少数,显然,这也并不是SEI当初制定CMM所考虑的范畴。

对于这样的企业,首先应该进行的并不是去运行什么基于CMM/CMMI的软件过程,而是扎实的完成软件过程的基础构建(这也是过程跃迁的一个重要组成部分),甚至非常有必要在实施CMM/CMMI前在企业内部引入PSP/TSP,然后在循序渐进的进行有针对性的软件过程改进。

然而,非常遗憾的是,本人看到不少这样的企业在短短不到一年的时间里就能通过所谓的CMM或CMMI三级(现在,国内企业对二级大多已经觉得不过瘾了,往往一上来就直接上三级)评审,效果如何可想而知。实际上这只不过是一场由咨询公司导演,软件企业主

演的“评审”秀而已。

CMM/CMMI中有一个最为核心的思想就是：只有好的过程才能导致好的结果，上述的现象倒是该思想的一个非常好的反面实例。

好的过程=好的软件？——瞎子摸象故事的现代软件工程版演绎

CMM/CMMI最原始的出发点是重过程管理而轻工程方法，这对于最初的CMM应用对象——美国军方软件的承包商或许是有其合理性的，但是对于其他众多的软件企业，由其是国内的软件企业就并不是那么的合理了，个人认为国内的软件的特点是规模有限，业务领域较广，技术周期较短，软件工程基础较差，对于这样的企业，有效的工程方法往往比有效的软件过程更能让企业获得即时的效果。

软件开发管理是一个非常复杂的系统工程，其涉及到企业技术和管理的方方面面，而且其各个因素又是相互关联和制约的，不均衡的过分强调其中的某一个或某几个方面都是不科学的，也是难以取得良好效果的。

CMM实施——削足适履的痛苦旅程

CMM/CMMI的初期培训往往是最令人愉快的，参与者在了解了大量以前没接触过的知识后往往对CMM燃起了高昂的热情，但是随着进入过程定义和项目的试运行就会渐渐发现身陷没完没了的文档编写和无休无止的各种会议，加班成了家常便饭。这时候，早先的热情早已荡然无存，而早先咨询公司宣称的能为企业带来无比效益的“先进过程”也成为了一件非常不合尺寸的紧身衣.....等咬紧牙关捱到了预评估，结果往往是当头一棒——和要求相距甚远，而正式评估的日子又已经迫在眉睫，怎么办？于是乎大家发挥国人的应试特长，将CMM正式评审变为一场期末考试，这时候，咨询公司往往扮演“枪手”的角色，将评估中可能遇到的问题及其答案罗列出来，而软件公司中的相关人员则将这些答案牢记于心。最后，在通过一场大家心照不宣的所谓“评估”，终于大功告成。

CMM评审——对皇帝新衣的自我陶醉

在上海某个高科技园区，有一个软件公司的外墙上高高的悬挂着一块牌子——“CMM5级企业”，这在专业人士看来恰恰是对CMM本质的曲解，非常遗憾这种曲解竟然是发生在一个通过5级评审的企业。

SEI在他的网站上明确声明：The SEI does not "certify" organizations. 即SEI不对任何组织进行“认证”，它仅仅通过获得其授权的评估师们对企业的过程能力进行“Appraisal(评审)”

CMM本身只是一个可以被用来评估企业能力成熟度的过程框架，而并不是什么“标准”，

CMM的重点在于软件过程的改进而不在于什么“评审”，至于所谓的“认证”那更是扯不上边的事情。

所以当我们舍本求末，将CMM作为一块可以证明其高贵身份的金字招牌到处炫耀时，令我们不由得联想起安徒生笔下的那个酷爱新衣的光屁股的皇帝

结论

CMM/CMMI本身是一套非常有价值的过程模型，但简单的将其图腾化却是整个中国软件行业的悲哀，反观CMM/CMMI的发源地——美国，除了和军方有业务往来的软件企业会寻求通过CMM/CMMI评审外，其他多数的企业并不怎么在乎是通过了CMM/CMMI的三级还是四级，像是著名的微软、甲骨文等知名企业都没听说过和CMM/CMMI沾过边，但这也丝毫不影响以他们为代表的美国软件企业在整个行业中独领风骚。

所以，我们认为，只有有效的而不是最权威的，才是最好的。

A (lee_huo) :

哇，这段分析太精辟了!! 认同~~~我们公司也是过了三级，虽然没有参与整个评估过程，但从后期的写文档来看，我意识到，好多都是形式，咨询师也只是为了让公司应付过去，而不是真正解决问题，没意思!

A (z dq2601) :

我们中国需要的是产品，不是一套为他人做嫁衣的过程，来繁荣我们的软件业。

A (lily_014) :

精辟之著。。。我个人比较推崇敏捷，就是不知道他两能否结合来实施，这样的话取其所需，并不是所谓的所有过程都要遵循于 CMMI。

★ CMMI 评估过程中的需求分析?

Q (jaedonger) :

需求分析的目的在于理解提出要求的组织对于这次评估的商业需要，评估小组领导将收集信息来帮助评估发起方对照评估目标和他们的商业目标。通过需求分析，可使评估人员在对评估目标，约束，输出和范围形成共同理解的基础上对下一步评估做出正确的决定。

在进行需求分析之前，应确保满足以下两个进入标准：评估发起方已经决定使用SCAMPI方法；能够提供评估要求综述的人有时间接受访问。发起者、初始要求和约束、过程相关

的历史信息是需求分析的三个输入因素，评估输入是需求分析的输出因素。

确定评估目标

我们知道，以满足商业需要为出发点的过程改进最为关心的三个因素是减少费用、改善质量、缩短产品面市时间。为此，本阶段所必需的实践是：

1. 标明评估发起者和相关的利益分担者，并在他们之间建立经常性的交流；
2. 将商业目标和评估目标文档化；
3. 确保评估目标与商业目标的一致性；
4. 确定评估使用方式（内部过程改进，供应商选择，过程监视），并将其文档化。

此外，在本阶段评估小组领导和发起者之间至少有一次交流。在某些情况下，还必须通过其他方式确保他们之间存在经常性的面谈。

确定评估约束

评估约束是由评估小组领导和评估发起方或者高级管理人员讨论得出的。它是一个不断反复的过程，以在满足评估发起者提出的要求、评估所采取方法的限制和对资源的要求之间达到平衡，最终达到评估输入参数的优化。为此，本阶段所必需的实践是：

1. 建立高层费用和日程安排约束；
2. 确定评估包含哪些过程域和哪些组织实体；
3. 确定对评估结果的最小期望和最大期望，或达到某一特殊的目的；
4. 和评估行为的利益分享者商谈约束条件和目的，确保评估活动的可行性；
5. 将商谈好的约束文档化。

同样，在本阶段评估小组领导和发起者之间至少有一次交流。在某些情况下，还必须通过其他方式确保他们之间存在经常性的面谈。此外，在评估早期阶段标识的费用和日程安排的约束应该是针对高层而言的，是一种系统的估计，而不是详细的估计。

确定评估范围

在评估过程中，由参考模型范围和组织范围决定了评估的范围。无论使用阶段式表示法或者连续式表达法，在过程改进执行的早期，模型范围都应该被确定并文档化。评估小组领导有责任保证发起者能够考虑到评估范围中所涉及的过程域和采取的模式表示法。评估的输出应该由他们根据实用价值来决定，在需求分析活动中制定，并且尽可能在那些可选择的模型范围内做出选择。组织范围定义了评估中调查的边界，例如：对于每一个项目的实践完成情况、为了完成组织级目标所做的实践，可被选来作为组织代表和过程执行的背景。为此，本阶段所必需的实践是：

1. 确定评估所使用的参考模型范围和表示法，并将其文档化；
2. 确定评估期间调查的组织单位，并将其文档化。

参考模型应包括过程域和相关的评估小组调查的最大能力等级或者成熟度等级（例如：评估范围内的过程域的共性目标）。参考模型表示法的选取应该在确定评估目标时就讨论过

了，这是因为表示法的选取可能影响到评估目标的实现。

评估模型的范围至少应该包括一个过程域。所有的共性目标和特定目标应包含确定的过程域能力等级或者成熟度等级；过程域内单个的目标不能被排斥在外。

通常来说，评估期间被调查的组织单位的确定应该满足以下条件：至少有两个被调查的过程是可以作为目标证据源，而且，要能获得组织使用的生命周期的典型覆盖。组织单位实例的选取可以通过调查表，或者通过和组织人员讨论得出的概要信息来决定。对于组织级别的过程实现（例如：组织培训），不要求多个实例。

评估中将调查的典型实例也会驱动提供目标证据信息源的参与者的选取。评估参与者（名字，角色）在初期所做的决定应该作为组织范围确定的一个部分，和评估发起者或者高层管理者磋商。这在后面的详细评估计划中将得到进一步确定。

确定评估输出

本阶段的目标是确定特定的评估输出。有些评估结果是必需的，而附加输出是可以剪裁的。在确定评估输出之前，应清楚地了解下面的问题的答案：

1. 评估中将产生什么级别；
2. 文档化评估结果，是否要写最终报告；
3. 是否要产生和报告关于怎样致力于特定发现的建议。

为此，本阶段所必需的实践是：与评估发起方一起检查要求的输出；与评估发起方一起检查并选择可选的过程改进评估方法的输出。要求的过程改进评估方法输出包括：评估记录；评估发现的事项综述；CMMI干事数据。

尽管可以选择向评估发起方之外的任何人公布定级结果，但因为在ARC中规定了，至少过程域的目标和调查的过程域都必须定级。因此，评估发起方至少要得到如下评估结果：

1. 最终发现，包括评估小组对每个调查的过程域的文档化的强项和弱项陈述；
2. 计划内的、评估小组对相应评估对象的定级描述。

是否达到评估输出的决定，包括将报告的定级，应在评估输入中写明。此外，评估发起方可能要求附加的定级输出来作为评估的结果。可能选择的典型的定级结果包括：

1. 成熟度等级或者能力等级评定；
2. 过程域满意 / 能力等级剖面；
3. 实践定级；
4. 可选择：使用“部分满意Partially Satisfied”用于过程域定级；
5. 15504 过程剖面；
6. 特定学科定级（例如：SE或者SW）；
7. 项目级别的发现或者定级；
8. 其他（非典型）的期望的输出。

评估发起方可能也要求其他的产品作为评估结果，可能要求的典型的产品有：

- 1.评估最终报告;
- 2.基于评估结果,采取行动的建议;
- 3.过程改进活动计划。

获得评估输入的许可

本阶段的目的是确认评估发起方对评估输入的正式批准,并且这些信息集合置于变更管理之下。为此,本阶段所必需的实践是:

- 1.记录评估输入记录的要求信息;
- 2.获取评估输入记录的发起方的正式批准;
- 3.管理评估输入的变更,获取发起方对于变更的承认。

评估输入可能是在计划中逐步产生的,但必须在数据收集开始之前得到正式批准。因此,评估输入至少应该包含如下必需的信息:

- 1.评估发起方的身份,发起方和被评估组织单位之间的关系;
- 2.评估目的,包括相应的商业目标;
- 3.评估参考模型范围;
- 4.被评估组织单位;
- 5.评估过程背景,包括:组织单位的大小和人员统计情况,应用领域、大小、危险程度和复杂度;组织单位产品和服务的高优先级特征(例如:面市时间,多功能,可靠性);
- 6.评估约束,包括:关键资源的可用性(例如:人员,资金,工具,便利设施,日程安排约束,评估可用的最多时间,评估之外的特定过程域或者组织实体,评估期望的最大、最小或者特定样例大小或覆盖,评估结果的所有权归属以及使用的限制,一致同意的信息结果的控制,评估结果与相关来源的属性。
- 7.使用的CMMI模型标识(版本,学科,表示法);
- 8.将成为评估小组领导者的主任评估师的身份和联系方式;
- 9.评估小组成员的身份和联系方式,以及他们各自特定的评估责任;
- 10.评估参与者和支持人员的身份(名字和组织关系),以及他们各自在评估中的特定责任;
- 11.为实现评估目标,评估期间收集的任何附加信息;
- 12.包括将产生的定级的计划评估输出描述;

- 13.预期的进一步活动(例如:报告,评估活动计划,再评估);
- 14.计划的SCAMPI剪裁和相关的折衷,包括组织单位的样例大小或覆盖;
- 15.评估使用的方式(例如:内部过程改进,供应商选取,过程监视)。

需求分析作为过程改进评估方法的第一步,是进行准确评估的前提条件。当满足下面三个退出标准时,我们认为需求分析阶段可以结束:评估发起方和权威的SCAMPI主任评估师进行了初步的接触;主任评估师已经访问了发起方组织的成员;评估输入已经被评估发起者证明并且置于变更管理之下。

值得注意的是,评估发起者在过程评估方面的经验将推动这一过程的剪裁选择。一个没有

评估经验的发起者将需要大量的信息和合作的咨询，以提供有意义和完整的评估需求；而有经验的发起者很可能会提供很有针对性的需求分析。

(test) 测试专区

第三部分：会员原创

1. 边界值测试要点

作者：思步 step365he

软件黑盒测试的常用方法有路径分析法，边界取值法，等价类划分等方法。而边界取值一般就是针对输入的数据，我们取刚好达到这个范围边界的值以及刚刚超过这个范围边界的值进行测试。日常工作中，我们常将边界取值和等价类划分结合使用；如果没有等价类，则使用一个或若干个合理值。

比如有这么个例子：一个输入文件有 1—255 个记录，我们测试时，可以尝试输入 0，1，255，256 个记录（这些是边界取值），以及输入 2，10，254 个记录（这些是合理值）。

我们再看一个例子：一个输入信息，有着年月日三个输入子项。假定年份的合理范围是 2000—2010；月的范围是 1—12；日的范围是 1—31。如果不采用边界取值，我们需要 $13 \times 14 \times 33 = 6006$ 个测试用例。如果采用边界取值结合合理值，年份可以取 1999，2000，2001，2009，2010，2011；月份可以取 0，1，11，12，13；日子可以取 0，1，30，31，32；据此我们计算需要 $6 \times 5 \times 5 = 150$ 个测试用例。和 6006 个相比，大大的降低的测试用例数目。不过存在一个问题，当我们编写 150 个测试用例的背景是在一个只有输入信息的程序中，工作量不大；如果是一个简单的信息输入界面，光年月日就耗费 150 个用例，则整个用例数目还是相当可观的。

我们可以将边界取值法做个改进。在已经获得的边界值中，筛选出我们真正需要的测试用例。在一侧的边界值中，我们选择年月日三个输入项同时为假的输入值（只需 1 个），接着选择三个输入项中 2 个同时为假（只需 3 个），接着是三个输入项中 3 个分别为假（只需 3 个），这一侧我们共计有 7 个‘假’的测试用例。我们到另一侧边界值，同理我们可获得 7 个‘假’的测试用例。再加上我们随机选取一个三个输入项同时为真的输入值（选取一个合理值的测试用例）。筛选后，我们只需要 15 个测试用例就能完成。

这里有个问题，从 150 个用例缩减到 15 个，其中会不会遗漏掉原本不应该被遗漏的用例？可能会有。这个可能性以及可能的大小根据项目规模和项目种类不同而不同。但是这个方法可以使得我们可以在最短的时间内将尽可能多的 bug 发掘并修复，降低风险。