

会员自己的 **精品** 读物

记录我们的足迹 分享成长的点滴

思步 电子期刊

作者：思步网会员

责编：lily_014

编委：坠入深海 jane
sherry rebeccazxy
amandating

校对：sungubbi

主编：steplv

第三期

封面设计：steplv

发布：2008.02.18

著作权说明

本文档为免费电子文档，任何人可以在思步网 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 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

括号中内容 user name 为思步网会员名

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

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

依次类推。

前 言

俗话说：“没有不胜利的闭幕”。的确，在广大的会员齐心协力的努力和奉献下，“思步电子期刊”在创刊期、第二期都圆满地划下了感叹号，相信“思步电子期刊”给大家的工作和生活之余也添上了绚丽的一笔。为了回报一直以来给予思步关心与支持的广大会员朋友们，我们的期刊不会拉下帷幕，过程会一直演绎下去.....

创办“思步电子期刊”的目的始终不会变：将会员在思步网中的探讨与交流，在不改变原意的前提下，通过一定的规律整理、汇集起来，并免费分享给广大的会员，真正达到：知识“从会员中来，到会员中去”的这一理念。打造出属于“会员自己的精品读物”，记录我们的足迹，分享成长的点滴。

07 年是收获的一年，值得大家去回味和享受硕果的滋补；而新的一年是充满期待和挑战。本期电子期刊是思步送给大家 2008 年的首份贺岁大礼，同时也希望大家一如既往地支持思步，使得其成长得更茁壮。自思步创办至今，获得广大会员朋友们的支持，在不懈努力下思步一天天地壮大，为了共同的目标、心中的那个梦想、对过程改进的那份热情与执着，使得我们不得不倍加努力让思步走得更好、更远。既然思步给大家提供了这个好的平台，愿广大的会员朋友们好好珍惜，把它当成您心中的舞台，写下我们人生中的舞步，把过程的奇迹刻画在思步。思步，愿一路上有您！

再次感谢思步团队的各位成员：steplv、sungubbi、step365he、jane、坠入深海、sherry、xixiaojing666、w7a8、rebeccazxy，感谢各位的无私奉献，希望我们继续加油，生命因服务而美丽！

此外，还要特别感谢思步网会员：amandating，她在本期电子期刊的制作过程中，付出了努力。

由于篇幅所限，无法将思步网中的所有内容都收录在本次期刊中，第四期、第五期、第 n 期……我们会坚持下去，奉献给大家。

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：mgroup24287@hotmail.com

✧ QQ 群：[19894513](#)（申请加入时，请注明职业）

迎新春、迎吉祥、迎幸福。听，新年的钟声在耳边回荡着即将敲响，思步团队全体成员在这里给大家拜个早年，提前祝大家春节快乐、阖家欢乐！因今年接连着不期而至的冰雪天气使得盼望回家心切的我们路途的艰辛，甚至很多在外打拼无法回家的朋友们，不要因为身在他乡独自一人而感到孤独，可以跟亲朋好友欢聚一堂享受一个难忘的节日，或许岁月将往事退色，或许空间将彼此隔离，但思步团队会想着大家。笔者此时也是夹杂着复杂的心情，也因天公不作美而不能回家的一份子。在此，思步团队全体成员祝福大家！

Lily_014

2008-2-1 北京

[lily_014 联系方式]

WebSite: www.step365.com

Email: Kren@step365.com

QQ: 50913696

致 谢

衷心感谢众多思步网会员！有了你们的大力支持，思步更有信心走好！

感谢你们在思步网，或贡献自己的宝贵经验，或热心解答其他会员的问题。有了你们的支持，我们才能把思步网越办越好，把思步电子期刊竭力打造成“会员自己的精品读物”！

在本期电子期刊中，我们收录了以下会员的原创文章、交流与探讨的互动内容：

清水至真、Scarlet、happy895464、Tyrone、QONE、wsming、littlecool、smilelight、jessie、mimcat、fishred、沟通无限、如你的愿、jnpanda、zdq2601、flyee_cn、stop-start、lilyli

感谢你们的真诚付出！我们将为上述会员加上金币以感谢你们对思步网的贡献，也希望大家继续支持思步。

队伍的壮大少不了新鲜血液的注入，我们忠诚地希望越来越多的朋友加入到我们的大家庭中，正所谓“众人拾材火焰高”，思步敞开心胸，期待您的加入！

思步网管理团队

2008-2-1

更多精彩请加入：

WebSite: www.step365.com

MSN群: mgroup24287@hotmail.com

QQ 群: 19894513（申请加入时，请注明职业）

目 录

著作权说明.....	2
阅读注意事项.....	2
前 言.....	3
致 谢.....	5
第一部分：你问我答.....	7
1 QA-质量保证.....	7
★ SQA应该做那些事?	7
★ QA计划对项目而言是否增值?	8
2 EPG的工作.....	9
★ SPI过程改进的前期准备是什么.....	9
★ CMMI裁剪探讨.....	10
★ 是否需要给全体员工培训CMMI模型规范.....	12
3 PM-项目管理.....	15
★ 如何管理外包项目.....	15
★ 项目组成员如何考核的讨论.....	17
4 RD-需求开发.....	18
★ 需求开发具体做什么?	18
5 RM-需求管理.....	20
★ 关于需求规范化.....	20
6 CM-配置管理.....	25
★ 关于配置审计与配置状态报告的几个问题.....	25
★ 对于如下三类配置项你们是如何进行变更控制的? 采用一套变更流程? 还是有不同的变更流程?	33
★ 如何进行数据备份.....	35
7 TEST-软件测试.....	36
★ 单元测试执行不好, 怎么办?	36
★ 能否介绍一下DRE模型呢?	38
8 思想碰撞, 故事中有深意.....	39
★ 你该走那条CMMI之路?	39
第二部分：专家文章.....	44
1 过程改进 (CMMI)	44
★ 青出于蓝-在模仿和超越之间铸就伟大.....	44
第三部分：会员原创.....	55
1 RD-需求开发.....	55
★ CMMI思想结合RMC实践之——“需求开发”流程定制.....	55

第一部分：你问我答

1 QA-质量保证

★ SQA应该做那些事？

Q (steplv) :

一直以来，有一个很传统的说法：EPG 相当于一个立法机关，只负责立法；SQA 相当于一个执法机关，负责核对规则、规范的执行情况。

但本人更倾向于另外一种说法：SQA 首先是一个 Teacher，要教会并引导大家按照正确的规范去做；其次是 Doctor，要及时就执行 QA 活动时遇到的问题总结分析，抓住病灶，加以解决；再次是 Lawyer；最后才是 Police，这时候才真正按照各种规章制度来核对规范的执行情况。

期待各位思步网会员，就自己在 SQA 工作中的实际情况简单探讨一下，从两个方面来进行：

- 1、组织所定的各种规范与制度执行的难易度
- 2、各种规范的遵守及遇到问题的实际解决方案

A(flyee_cn):

一套新的规范上线的时候，总会遇到有些抵触的，这个时候就需要过程的推动者强势一点，或者制定一些奖惩措施（呆的第一个公司曾经这样做过，效果不错 ^_^），习惯成自然，慢慢就好了。

对于过程的制定和推行者，一定要为用这些过程的人员考虑，制定过程的人一定要自己先走一边，模板要实际填一边，不能把自己的都很难填的东西交给项目组，要让项目看到你确实是在为他们考虑；推行过程的人首先必须要对这些过程很熟悉，切实深入项目，跟项目一起来执行好这些过程并提高效率和质量，过程是死的，人是活的，把握尺度，活用过程，软硬兼施，和气相处。

其实过程在全球性的公司或者涉及到跨部门的公司更容易被执行

A(jane):

Flyee_cn 说得挺通俗易懂的，谈谈我的看法，其实本人还是比较在意各种规章制度的，因为好多时候硬性规定后，大家都会为了这个制度而走一下流程，真正意义上的效果都没有实现。而一旦实施人员看不到效果，很有可能会形成一种恶性循环，然后又得回到起点，重新进行过程改进。过程改进其实又是一个很漫长的过程，不过 Flyee_cn 提到的奖惩制度倒是不错的，只是这处度又是很难把握的。所以能否再谈谈。

A(flyee_cn):

当时的奖惩制度实施的时间比较短，3-5 个月，主要是奖励，在月会的时候明确那个项目到底在哪方面做的好，有实际的证明才能给奖励，基本上会惠及到项目组的每个成员，金

额不大，效果挺明显，在实施的过程中，公司会有侧重性的去领导某些项目区提高，这样 3-5 个月下来，基本上所有的人都能得到奖励，流程（那个时候只做 2 级）经过这个时间段也能稳定在项目组里面了。

★ QA计划对项目而言是否增值？

Q (Scarlet) :

最近听人提起过程改进是否增值的话题，先开个小口，与大家一起讨论 QA 计划是否增值。各类项目的 QA 计划内容几乎接近，很多可以复用，看起来对项目组也没什么可增值的地方。期待大家参与讨论……

A(happy895464) :

单单文档是不增值的，QA 计划千篇一律，格式都一样。不过，区别就在内容上，每个项目的情况都不一样，所要选择的过程也不一样，所以 QA 计划也不一样！QA 的职责决定了 QA 计划是增值的！我觉得好的质量是：合格的产品在合格的时间内提交给客户，创造出合格的价值

A(Scarlet) :

多谢 happy895464。我在听到这个问题时，也觉得但就文档而言没什么增值的，想到 QA 的价值也有些失落。确实 QA 计划的内容更值得关注，选择的过程以及审计策略和力度上反应了 QA 关注的重点，改进的重点，也许对某个项目而言增值少，但是从组织级整体而言还是增值的。

A(xixiaojing666) :

QA 计划的格式当然是一样的了，但是内容我估计没有相同的。因为 QA 计划是依据项目计划制订的，估计没有完全一样的项目计划吧。

怎么能通过 QA 计划来判断对项目组有无可增值的地方呢？我觉得一个合格的、变通的 QA 人员对项目组肯定有增值的地方。

1、首先她检查你的过程的符合性，只有过程作的好，才有可能产出高质量的产品。想象不出一个完全不按照公司的流程去走的项目或是偷工减料的项目能产出高质量的产品。

2、她还帮助你进行项目管理的指导，及时提醒和指导项目的工作。

呵呵，所以我觉得 QA 工作是增值的。

A(sungubbi) :

QA 计划是 QA 工作安排的重要依据，只不过目前我们大多数的 QA 计划脱离了项目计划而存在，这样会导致计划脱离实际，对 QA 的直接指导意义不大，如果跟着项目的计划变更更新又过于频繁，从而觉得 QA 计划价值不大。

所以我在想关于 QA 要做的事情、要做的测量、工作的原则已经在组织的文件中有了明确的定义，那么 QA 就不必有一份独立的计划了，完全可以将 QA 的行动计划与项目的进度计划（PROJECT）直接结合，根据项目的实际进度安排 QA 的相关活动。当然，如果项目

的 PROJECT 计划缺乏时效性，那么 QA 计划也无从谈起了，只能制定每周计划了。

2 EPG的工作

★ SPI过程改进的前期准备是什么

Q (xixiaojing666):

有朋友的一个公司要准备过 CMMI L3 了,她问我前期准备是什么,应该怎么做?

我觉得有以下两个方面:

- 1、进行咨询公司的评估与选择
- 2、按照咨询公司的日程安排进行学习

欢迎大家补充。

A (坠入深海):

我觉得要做如下准备工作:

- 1、先在网上找一些 CMMI 方面的相关资料，首先自己有一个大概的了解（比如上思步网找一些入门级的资料）
- 2、在公司内部首先要获得领导的支持，领导支持对于推动 cmmi 工作非常重要
- 3、有了一定的了解之后就是对咨询公司的选择了，尽量选择一个好的、负责任的咨询师接下去等项目启动了就可以按照咨询公司的要求进行了，呵呵

A (QONE):

- 1、争取高层支持：包括过程改进相关人员的设置。
- 2、争取 CMMI 咨询评估费用。可以公司自筹。政府也有对获得 CMMI 认证企业的适当补贴，可以查阅一下相关政策，能争取到政府补贴当然更好。
- 3、考虑一下公司实施 CMMI 的周期，打算什么时候进行评估？国内通常是 12-15 个月，国外要长一点，根据 SEI 公布的结果平均需要 26 个月。
- 4、调研一下支撑 CMMI 的工具，选择合适的工具能有效促进过程改进工作的实施，在评估通过之后也能够比较好的在公司内实施 CMMI3 的质量体系。
- 5、调研一下咨询公司，选择实力雄厚的咨询公司，选择经验丰富、认真负责的咨询师。
- 6、确定支撑工具和咨询公司后，按照咨询公司的计划实施就行了。
- 7、实施 CMMI 过程中，要调动公司上下对 CMMI 过程改进工作的理解和配合，否则很难实施好。

A (w7a):

从初始级到 L3 的么？

- 1、原始需求的整理及获取经营层的支持;
- 2、请咨询公司做个初步诊断;
- 3、成立 EPG 小组

★ CMMI裁剪探讨

Q (wsming):

一直学习 CMMI，但是对于具体项目的裁剪还是感觉很迷惑，在这里提出来，希望大家能够多多讨论，发表自己的意见，互相学习。

- 1、1 年以上的项目怎么做裁剪?
- 2、3-5 个月的小型项目又怎么裁剪?

在这里还有个矛盾的问题，SEI 要求每个 PA 都需要有实践，如果裁剪掉了相应 PA 的实践，那不是和 SEI 相冲突了吗?

A (sungubbi):

裁剪需要考虑：项目的规模大小、工期大小、团队大小、团队成员素质与技能情况、用户特殊要求、紧急程度，以及项目所采用的生命周期模型。

根据项目的以上特征，结合项目实际情况，进行适当的裁剪

A (step365he):

个人观点——如果组织级别过程实施的较好，那具体到项目的时候，裁剪才能顺势而为。必须先建立一个指导过程改进的思想体系，制定出基本的原则——保证最基本的组织层面的世界观要一致。无论如何裁剪，核心不能丢。并且我认为核心有两点：为组织的商业价值服务，为开发人员服务。

另外一点很重要：每次裁剪后，必须检验其效果，从而得到反馈和总结!

A (littlecool):

在讨论“如何做裁剪”以前，我建议我们先讨论一下“裁剪”的涵义。

裁剪的英文原文是 Tailoring，是“裁缝、缝制衣服、适应、适合”的意思；特别的，根据美国传统辞典（双解）的解释，Tailor 意为——

- 1) To make (a garment), especially to specific requirements or measurements.
- 2) To fit or provide (a person) with clothes made to that person's measurements.
- 3) To make, alter, or adapt for a particular end or purpose

特别注意以上黑体字标注的部分。

了解了 Tailor 的涵义之后，我们发现，其实“裁剪”并不是指拿着一把大剪刀“咯嚓咯嚓”的把模型、标准和/或方法论砍来砍去，而是“量体裁衣”使模型、标准和/或方法论根

据组织/项目自身的实际情况而怎样应用的问题。所以，我个人觉得使用“裁剪”来翻译“Tailoring”的并不是很贴切，因为：“裁剪”不仅仅可能是根据组织/项目的规模、工作量、业务模型……等属性来简化标准的活动，也可能是在标准和模型基础上的更加得深化、丰富化和精细化。

如果同意这个观点，wsming 所担心的矛盾就很好解答了——“裁剪”主要是指如何对模型的具体化的运作。

举 2 个例子：

1、OT 的 SG1, “Establish an Organizational Training Capability”, 根据实际的需要，组织完全可以采用 People CMM 模型中“Competency Management”和“Competency Development”的若干过程域的实践，构建组织的技能架构和学习地图（Learning Map），从而更加准确的识别组织的培训需求、更加有效地培养组织自己的培训能力。

以上是“在标准和模型基础上得更加深化、丰富化和精细化”的例子，下面再举一个“简化”（就是通常意义上大家所理解的“裁减”的意思）。

2、PP 的 SP1.2, “Establish Estimates of Work Product and Task Attributes”, 一般情况下，CMMI 在这里的推荐做法是先估算项目的规模，将规模估算的结果作为工作量、成本、进度的估算输入。但是，如果一个组织有很好的关于工作量分布的模型，这个模型的粒度已经细化到 Task 一级，例如：开发一个复杂度为“中等”的 Web Page 工作量平均是 1 人天，关键地，关于“复杂度中等”在组织内有着严格的定义（根据 Web 页面的控件数量、关联页面的数量、外部接口的数量、数据库访问的数量……等等“属性”的量化的定义），这时候也可以认为该组织建立了关于工作产品和任务属性的估算模型，规模的估算可以简化。

A(smilelight):

拿不准的问题建议简洁处理

关于 CMMI 裁剪，建议还是把问题处理得简洁一些。

感觉裁剪的基本思路是从组织标准过程经过裁剪形成项目定义过程，这对于积累过程财富、提高过程成熟度很重要。如果只是想通过 CMMI MLx 的评估裁剪应该不难做；如果真是想做得很有项目特点、充分促进项目绩效则很不容易。

要做裁剪，自己组织的标准过程、裁剪指南和裁剪案例总是要有的。到底该花多大的力量去定义标准过程和裁剪指南、去做裁剪就要看组织的具体情况和约束了。如果不清楚该花多大力量、做到什么程度，那基本建议还是把问题处理得简洁一些，着重做有意义的事，少花力量去做作秀的事。例如自己先学习，接受 Training，然后召集组织内相关的牛人接受一些 CMMI 过程域、实践（不妨视其为“业界标准软件过程”）及裁剪方面的 Orientation，讨论、归纳一下常用哪些标准过程，选用项目过程时考虑哪些因素，这些因素各自的权重分值和综合评价的操作方法等等，形成裁剪方面的过程定义文件。开始阶段这些文件建议做简洁些！如果真想做好裁剪就当真去执行、维护、改进这些文件，日积月累就会越来越好。

做好裁剪不光是裁剪活动本身的问题。例如，若组织的过程要素本身就不稳定，或项目定义过程等本身执行力度弱，那裁剪是难得真正做好的；建议不如先去处理这些问题，用不着花太大力量当真做裁剪

A(tyrone):

要 ML3(含以上)的过级其实是不需要会或做裁适活动的，只要能产出项目管理计划就行了，也不会有评估师去探究，所谓 Project defined processes 到底对不对，至少我参加过几场 ML3 评估(担任 ATM)，不同的评估师都不会去看裁适的结果，甚至于许多评估师也看不懂 WBS 与 Project defined processes 到底有什么不同。为了这个我还曾与其它公司的评估师吵过架。

CMMI 这个框架有一个很大的缺陷，就是效益的问题，效益不好是因为裁适只被提到，却没有裁适过程，没有相对产出的要求，因为 CMMI 所描述的主要是工程开发与管理上的要求，所以到底有没有裁适，裁适得对不对，评估师不会去探究，多半咨询师也不会去教，甚至不具有相对的知识与经验。

在过程标准中，则不同了，像是 IEEE 1074、ISO/IEC 12207、ISO/IEC 15288、ISO/IEC TR 15271、IEEE/EIA 12207 系列等工业与国际标准，都含有一个裁适的过程，这个过程描述了裁适的活动及工作项目，并且要求有一项产出一裁适的理由阐述(rationale)，那是个纪录，由项目经理说明，针对项目的特性，TA 挑选了那些过程、那些活动、哪些工作项目、做了什么样的调整、有哪些产出，并且说明为什么要选这些过程、活动、工作项目，为什么做那样的调整，并且记录在项目管理计划里或成为一项独立的文档。如此，制度的执行也更有效益，不会变成像 CMMI 推行常见的弊病，在完全没有裁适，或不知道怎么裁适情况下，多做了许许多多的工作，然后再来怪制度很耗成本与时间，过级之后就把所有制度通通锁到柜子里。(不过，现在 SEI 聪明了，说是怕过级之后就荒废了，所以过级的公司每三年要 renew 资格，但不知这种规定大家买不买帐，因为这个制度的过级与维持还真正够贵的)

★ 是否需要给全体员工培训CMMI模型规范

Q (jessie):

这个题目大家可以说说自己的看法，没有标准答案。

我个人一直倾向于不给全体员工，特别是给工程师培训 CMMI 的规范。原因是我觉得他们不需要熟悉规范，也不需要知道 CMMI 里面有几个 PA，每个 PA 大概要做什么。我觉得熟悉 CMMI 模型，是 EPG 的事情。我倾向于给大家培训公司的流程和规范，让大家知道自己在日常工作中，需要做些什么，把流程和规范落实起来。当然，如果他们想更深入了解，我会告诉他们为什么要这么做。但是，切忌跟大家说是因为 CMMI 要求这么做，因为我们要过级，所以必须这么做。要这么说，希望大家不执行过程流于形式都不行了，因为你已经给了大家这么一个信号。我主要从执行是为了方便他们日后的工作，或是跟他们讲道理，从保证项目成功和提高产品质量入手。引导工程师从项目经理和公司领导的角度去看问题，要求他们换位思考，没有不想当将军的兵，这样他们会容易接受并执行。

A (step1v):

这个主题很好，凡是从事 SQA 或者 EPG 的人员，肯定会遇到这样一个问题，个人很同意你的观点“倾向于给大家培训公司的流程与规范”。

其实不论是从公司角度还是从个人角度，都想有个很清晰的流程摆在那里，并且把这种流程演变成为一种良好的行为习惯。(当然，流程没有不变的，一定要实时地进行优化。)

华为任总说过一句话，大体意思是：规范的东西，“先僵化、再优化、然后再固化”。想想看，这句话就已经说明了为什么现在国内的部分中小企业不能很好的实施 CMMI 的原因，

因为一开始就想着“优化”，所以会越“优化”越乱，最终 CMMI 成了鸡肋，丢也不是，不丢也不是，很是尴尬。

先给员工一个很好的流程规范（当然是结合公司实际的），引导他们按这些规范去做（一定要引导，不要强推），慢慢就会将这些规范变成良好的习惯的。

A (w7a8):

我们做工作，应该知其然更知其所以然！

CMMI Overview 的培训是必不可少的，目的是让公司或项目组成员知道 CMMI 到底是做什么的，不让大家觉得多么的神秘！

其次是 OSSP 和 PDP 的培训，准确来说是 PDP 的培训，因为针对培训对象的所在项目不同，培训的内容都会有所调整。

A (tyrone):

个人的看法与 Jeesie 是一致的。应该以公司所订的制度程序培训，CMMI 的框架，可以不提就不需要去提，因为，CMMI 是管理者的事，而不是工程人员的事，工程人员只是依规范在做事而已。能否循着 CMMI 所提供的一个成长阶梯步步升高，这是管理者必须要去思考的与努力的。

CMMI 本身是一个管理的框架，在告诉管理者，一个组织要做到什么，做好什么，所以要在组织中建立什么样的制度，这些制度需要被落实，然后才能够谈改进，不能落实的制度，谈改进这档子是缘木求鱼的。

w7a8 所说“做工作，应该知其然更知其所以然”，这句话的本意是没有错的，重点是，“知其所以然”所指为何？CMMI 里所涉及的系统工程、软件工程、项目管理等等 disciplines，大家都可以在坊间的书中查考到，其实并不难理解，如果要知其所以然，我反倒会是建议大家去研读这些领域原汁原味的论述，而非 CMMI 这个由实践所集成的框架，因为您看不到真正精华，只有指出该做什么事而已，就算在 V1.2 版本已经提供更多的说明与例子，但仍然是不足以清楚让你全然理解“其所以然”的。个人研究所研习的就是“系统工程管理”及“企业管理”，简言之，就是 CMMI 的全部还多，甚至于也涵盖了 P-CMM，当我回头看 CMMI 这个模型的时候，就觉得一般人要从这个模型去理解 CMMI 如果没有相当的经验与背景，其实并不容易，我曾经估计，那也要花个两年的时间吧。因为还有许多扮演“桥梁”角色的学问需要去学习。

如果教大家通过去学习 CMMI 就有一层人员会逃脱责任--管理者，因为大家都去听课了，所以这些工作就变成第一线工程师的责任了，因为每一层的管理者竟“权责下授”了，由工程人员去制订流程、订模板、订准则，管理者只是准备接收成果而已，说得好听点是说：因为那是你们要用的，所以由你们来订是最恰当不过了，但话说回来，从管理的角度，希望大家做到什么，才对组织的营收与成长有帮助的角度，来定义各项流程与模板、准则，不是更加恰当？大家可以仔细去思量，每个 PA 都有 GP，这些 GP 都是管理性的活动，而 SP 是工程人员要做，但是管理者要盯着工程人员要做到的，从这个角度去理解，我想很清楚，到底谁该去理解 CMMI 的培训应该那些人参与了。

但以上的论述，并不排除工程人员对于 CMMI 模型学习的参与，因为诚如 Jessie 所说的“没有不想当将军的兵”，理解管理者所要的，所想的，工程人员也会知道自己该如何表现

了。

A (steplv):

Tyrone 的解读很精辟，让人受益匪浅。

w7w8 所说的“知其所以然”，诚如 Tyrone 所说，这个从一般的情况下而言，普通的工程师并不一定非得知其所以然，他们只需要理解这样去执行会给工作带来多少的效率及最终的成果质量的高低，这应该是每个一线的工程师及 PM 所应该关注的地方。当然，“知其所以然”，也不会有坏处，只会加分，只要有些精力及相关体系知识。

我之前说过：我同意 jessie 的观点，倾向于不给全体员工培训 CMMI 规范，而给 TA 们培训公司的研发流程体系。这样做的重要性及适用性 Tyrone 已经说过了，我不再重复，不过我还是要强调的是：公司的研发流程体系一定要结合企业实际而有所积累，并做适当的及时变更。

至于 Tyrone 所说的：没有相当经验与背景的一般人员，要理解并运用 CMMI 模型，需要两年时间。我想知道的是：这个“两年”时间是业内的统计数字还是其他相关的统计，是不是一般情况下都是两年时间？

也许大家很奇怪我为什么这里一定要抓住这个词：“两年时间”，因为这里也有一些是刚刚从事 CMMI 相关的人员，这样可以至少给他们一个自我的计划段时间，并做好为之奋斗的准备。

另外，我还有个请求，Tyrone，请您对一个刚从事 CMMI 相关职业的新手一些建议，告诉 TA 们应该去学习、关注那些知识体系，需要具备何种素养，这样才能使 TA 们更进一步理解 CMMI 规范。谢谢！

A (tyrone):

我说两年的时间，其实是我念研究所研习系统工程两年的时间，那是一个系统化的教程，包括企业管理、组织管理、人事、财务、战略、市场营销管理等商业管理面，以及产品开发、系统工程与分析、研发管理、生产管理、维护管理、质量管理、科技管理、统计过程控制等向面的深度学习。其实，在那之后，个人还有管理领域的研究：包含国际标准、工业标准、美国军方标准、CMM/CMML 框架及相关素材的研译，乃至参与台湾之 CNS 标准，有关于信息技术部分的制订工作。这些也都是学习的一部分。这个时间加起来其实更长达十年以上。而我所指的两年的时间，其实就是指念研究所的那段时间而已，但是那个时间里，其实只是学到个别的知识，将这些知识完整地融合、应用，并产生效益，那又需要实践及工作经验的积累了。这样子看来，时间就会更长了。但是，别气馁，活到老学到老，学习是人生不该停止的一项活动，就算到了现在，我仍然如同一个道场里的学生，不断努力学习当中。

加油吧，别在意那“两年”，我说两年的时间，主要是提醒负责主导过程改进的公司领导，不要急功近利，设定一些不是很恰当的目标（像是要在一年之内通过 CMMI ML3 评估，又不给参与流程制定的人员，充分的学习机会与时间），应该按部就班，在潜移默化中真正改善公司的体质，提升公司每个成员的能力。

A (littlecool):

简单的表明一下我的观点：不需要。孔老夫子讲，“民可使由之不可使其知之。”话虽然难听了一些，但是蕴涵着至上之理啊……

A (sungubbi):

我的看法：不需要

我的做法：对所有员工普及质量控制与过程控制的观点、意识培训，通常在新员工培训时进行。培训的内容主要是什么是质量、质量产生于各个环节、产生于各个工作环节的 BUG 导致的成本放大、上下游工作环节，但一定要将培训做得生动和有趣。

同时在员工进入新岗位一段时间后，进行再次培训，培训重点则针对他们的岗位的规范要求以及在试用期间他们对流程规范的疑惑，而且需要注重告知这样做的目的和意义。这样的培训会更有针对性一些，效果也相对比较好一些。

3 PM-项目管理

★ 如何管理外包项目

Q (sungubbi):

要上一个外包项目，做为乙方，在外包项目中，应用正常的项目管理方法外，还需要注意什么呢？做为接包方，我的想法还是希望首先能控制住需求的范围，必须得到明确的界定，并排定优先级。同时形成比较明细的计划，得到分包方的确认。在实施过程中，进行每周至少一次的用户沟通。甚至希望能够将该项目的过程定义结果提供给用户进行确认，避免在开发中，用户提出其它的要求。

另外，做为承包方，是否应该争取让用户多参与项目的活动呢？以便获取用户对项目的认同感。

A (tyrone):

您的问题我有些小困惑，我想应该是：

您是承揽方(合同的乙方)，所称分包方，应该是委托方(合同的甲方)，是用户的外包、或者委外(outsourcing)案的合同执行管理单位。首先要说的是，这种项目的管理与组织内部项目(如产品开发及维护)并无二致，但是有些问题可能需要特别注意。

1. 合同内容应包括变更处理的策略、方法与程序，哪些层级的问题需要修改合同，哪些直接修在需求规格里就成了。建议凡会影响到产品架构、对时程成本有重要冲击的变更应该要修订合同，例如某个需求变更会使成本或时程增加 10% 时，这个增加与变动最好区分产品来列计，如果您的项目里有三个系统要交付，其中一个架构变动达 10% 以上、或成本或时程增加达到 10% 以上时，就要修订合同。

2. 合同里的权利义务划分很重要，例如，客户要配合实施的评审或对于问题的裁决，要明订处理的期程(duration)，因为客户评审或裁决使用的时间，或者根本悬而未决，所造成的时程延误或产品开发困难的情况，是不能全数由乙方来承担的。另外将权责明订在合同当中，也可以敦促一下甲方要配合相关的活动。不能因为甲方相关人员的逃避与不作为，而造成项目难以进行。当然甲方要驳回送审文件时，也不能是以笼统的「不符合需求」来拖延，有任何问题就要直接指出问题所在及其严重性，然后以充分的理由驳回。

3. 需求管理很重要，但是如果需求不是很明确宁可采用演进式开发方式，不要采用瀑

布式或是渐增式(或者迭代式)开发方法, 否则将使需求不断蔓延, 最后项目会很难收拾。

4. 避免程序员直接面对客户或使用者代表。你可以相信客户或使用者代表的参与具有重要意义。但是, 如果是因为需求不明确, 客户弄不清自己要什么, 所以让程序员到第一线面对客户, 直接 coding, 这种项目是必死无疑的。也许项目能做完, 但是项目肯定是失败的。因为项目成功的定义是一如期、如质、如预算完成。让程序员到第一线面对客户, 会发生的问题第一个就是需求失控, 然后时程与成本接着失控, 最后产品的质量失控(至少产品可能因为不断修修改改, 废码太多, 体型变得很庞大, 执行起来没有效率, 另外也可能潜藏许多未能及时发现的逻辑问题)。应该面对客户的, 不是项目经理、商务经理、要不然就是系统分析师。遇到问题带回公司解决, 千万不要当场解决。尤其是复杂的问题, 当场解决可能很花时间、而且也不见得能够很全面地处理, 会不会衍生出其它的问题也不知道, 有的问题弄不好会让整个系统 Crash 掉, 所以还是把问题带回公司再研究解决。

5. 有关于文件的模板与格式的问题, 这个也常常会遇到。项目里要使用的文件模板是什么, 首先可以请甲方来决定, 甲方如果没有, 那么乙方可以先提供自己公司制度中的文件模板, 在甲方的同意下采用, 或者经甲方的修改后采用。但是无论如何, 一个重点, 这些事情都要在合同议订阶段, 要不然就是在项目规划阶段全部谈定, 项目经理在项目规划阶段, 就要把这些项目里要产出的文件与使用的模板, 放在项目管理计划的数据管理计划中。在项目的前期就把这类的问题搞定, 以免在后期因为文件的格式、风格等问题, 造成验收上的困难。

6. 让客户或用户的代表参与项目开发的工作是有益的作法, 但必须在项目启动阶段就要建立参与人员, 项目之成败大家荣辱与共的共识, 否则, 客户或用户代表的参与就会造成需求蔓延的问题, 从而使得项目的时程、成本与质量失控。因此, 除了在项目启动阶段建立任务说明(Mission Statement)、共同愿景(Shared Vision)之外(这些其实就是一份文件, 让所有参与项目的人员(含客户或用户的代表)大家签署, 给出承诺), 在项目的每个评审会议(含验收测试活动)中, 由评审会议主席, 不断重复提及, 或者在乙方向甲方索取合约中已列示、或开发期间甲方允诺提供, 对于项目开发成败有重要影响的数据与信息时, 列于索取数据及信息的公文中, 提醒甲方配合, 以免客户或用户代表, 因为小枝节的问题, 而一直抗拒合作。开发工作是需要甲乙双方共同合作方能完成的。

7. 在项目执行当中, 所有的活动应该都要预留一些 buffer, 或者做好有效的工期规划, 这可由工作事项的前置时间(lead time)去考虑, 千万不要变成, 像是, 在项目计划中预计于 11 月 20 日需求规格要订为基准, 但需求规格却是一直等到 11 月 20 日才定稿, 使得基准订定所需要的评审活动(Peer Review 一周的时间)与提交给客户核准的时间(可能需要两周的作业时间)通通没有预留, 这样的话, 只是让项目的期程一延再延而已(以上面例子而言, 实等于项目进度已经落后三周了), 这个时候, 乙方是不可以怪甲方没有及时核准需求规格订为基准的。

8. 项目的实行, 要采用工程与项目管理的最佳实践, 包括风险管理、人力资源管理、配置管理、质量管理、量测与分析。在产品交付客户之前, 务必先在内部完成所有的测试与评审工作, 不要以为客户或用户会测试, 所以内部的测试工作就省免了。这常常会是公司商誉受损的首要原因。

9. 项目经理应随时带着合同、WBS、工作条款(Statement of Work)、项目管理计划、需求追溯表。尤其是在举行联合评审、项目管理评审(project management review, PMR)时, 务必带着前述的文件, 项目经理也应该多抽空详读合同, 尤其是罚则, 要先知道如何保障自己

公司的权益，当然，理解罚则不是要去设法钻合同的漏洞，而是要知道所有可能的后果，以便进一步做好风险管理。

10. 可以的话，项目经理要根据项目管理计划，做出项目的月工作计划及周工作计划，以便有效管理项目。这些工作计划可以只简略提列工作事项的起迄时间、本月或本周应达成的状态、负责人员。项目的工作很繁忙，提列工作项目及负责人员，有助于你依据负责人员的特质，去管理每个人的工作及产出。有些人是比较被动的，或者对于时间没有什么概念，您就得对 TA 施予严密地控制，千万别到预计完成时间的当下才问完成了没有，我可以百分之百地告诉您，一定没有完成。再者，您也可以对技术能力水平不是很高的工作人员，授予关注，适时找人协助，以免因为 TA 遇到工作瓶颈不敢说，而使得项目进度受到影响。

11. 与客户或用户的代表讨论需求变更时，一定要控制在合同的范围(时程与成本)内。项目经理可以在权责范围之内，允许变更，但要让客户明白，合同原来是怎么写的，您同意的变更，可能让项目时程延误及系统 integrity 上的可能的风险，这是您所要的承担，您尽可能排除问题，以达成客户的「期望」。设法让客户或用户代表知道，从合同的角度，TA 们欠您一份情。藉此建立双方的互信基础，要给 TA 们的讯息是：您很希望尽可能满足 TA 们的要求，只是合同上是有限制的。请记得，需求变更超出合同范围，实质上是一种违约的行为，也会造成后面验收的困难(验收失据)。就算客户或用户代表认为出钱的是老大，TA 们说了就算，但是您也应该让 TA 们知道，有一天超出合同的范围部分发生问题时，是没有办法救济的，除非变更时一并把合同给修了。但通常客户是不会愿意修约的。而走上修约一途时，项目经理就可以把变更的问题提升给管理部(法务、财务等人员)或公司管理层级去处理了。

不知道那个项目的形式是怎样的，您们是「成熟甲方」的外包，还是下包？我的意思是，所谓下包指您的甲方还有一个甲方，而外包是指那是甲方的一个内部项目，两种情况的风险是不太一样的。我深信外包的情况下，甲方可以采用较高成熟度的供货商管理做法，来要求贵公司；但是，如果是下包的情况下，而甲方的甲方是不成熟的 acquirer，则另当别论了。但无论如何，前面提到的那些都还是适用的，至少可以保护自家公司的权益。如果甲方的甲方不是很成熟的话，可能需要建议您的甲方，多与甲方的甲方多多交流，沟通观念与建立共识。

至于项目管理，PMI 的 PMBOK 离实务有很大的距离，不论是硬件或软件产品的开发都是一样，因为 PMI 的 PMBOK 的原始架构基于建筑业的 Context。可以使用 PMI 为美国国防部所做的 PMBOK Extension for DoD，里面谈到与系统工程的一些连结与注意事项。

★ 项目组成员如何考核的讨论

Q (wsming):

最近公司正在对公司考核制度进行调整，在这里，想问问大家有没有好的办法对项目成员进行考核？一般项目组中有需求开发人员、开发人员、编码人员等，还有相对比较独立的测试人员、QA、CM 等。

我们现在大致的思路是项目承包制，公司、部门和项目经理协商确定承包金额，之后由项目经理对承包金额进行分配使用，最后项目经理对成员进行考核，确定利润分配。

希望大家能积极讨论，寻求一个更好、更合理的方法！

A (steplyv):

你是站在公司的角度来考虑还是仅仅站在某一个部门的角度来考虑？你们的公司组织架构是项目型的？职能型？矩阵型？

要考核项目成员，无非是以下几个因素：

- 1、工作效率
- 2、工作量多少
- 3、工作质量

其实，还有很多是隐性的东西包含在里面，比如：团队精神、学习积极性、不当操作造成的影响等。

以上这些，仅仅是实实在在的，并没有考虑企业的文化，仅仅是从员工技能方面的考核，并没有考虑其他的，比如：员工纪律情况、加班情况等。

从你说的情况来看，你说是实行“项目承包制”，你把权利全部给了 PM，其实，你又如何明确地约束 PM 呢？

你把你想要做的事情说的详细一些吧，这样大家就可以有针对性地给你一些建议。以上，请参

A（清水至真）：

考核者的目的是什么，直接关系到考核的内容。就我所接触，一般公司考核都是关系考核员工的奖金发放，直接关系到员工的 MONEY。如果是这些考核，steply 说的各方面都可能在考核的范围之内。

如果是项目经理考核下面的成员，我个人则觉得，还是手把手教的考核方式是最好的。也就是说遇到问题就给他们辅导，帮助他们成长。他们出现问题直接告诉他们错在哪里，怎么处理可能更好些。这样的方式可以使团队就有向心力和战斗力，达到真正完成目标的考核。也就是考核更高层面是帮助员工成长，高效完成既定的目标。

4 RD-需求开发

★ 需求开发具体做什么？

Q(mimcat)：

由于现在公司的特性，在需求开发部分涉及的不是非常多，只是应用了其中的一小部分信息！所以想向各位请教一下，各位都是怎么进行需求开发的？

查看相关的资料，都只是说做什么，没说为什么。导致我只知道可能干些什么，而不知道为什么和具体怎么干，所以想请大家也回答一下：需求开发的作用和必要性，以及一般的做法。

A(sungubbi)：

在 CMMI1.1 中对需求开发的描述：

“需求开发”过程的目的是产生和分析顾客需求、产品需求和产品构件需求。

“需求开发”这个过程方面包含 3 组基本惯例。第 1 组惯例是定义完备的顾客需求集合所要求的惯例；这个需求集合将用于开发产品需求。第 2 组惯例是定义完备的产品和产品构件需求集合所要求的惯例；这个需求集合用于产品和产品构件设计。第 3 组惯例是在定义、派生和理解这些需求时用于执行必要的分析的惯例。这 3 组惯例可能彼此间发生一些递归互动作用，它们与在“技术解决”过程中开发的优选产品概念和候选解决方案的定义也存在递归互动关系。

所开发的需求将成为设计的基础。需求开发工作包括：

- 1) 收集和协调共利益者的需要；
- 2) 产品的生存周期需求的开发；
- 3) 建立顾客需求；
- 4) 建立与顾客需求一致的产品和产品构件初步需求；
- 5) 提取、分析和通报顾客需要、期望和限制条件，以获得顾客需求，从而在共利益者之间就所要满足的内容达成共识。

这个过程方面涉及所有顾客需求，而不仅仅是产品一级的需求，因为顾客也可能提出特殊的设计需求。

要把顾客需求进一步精练成产品和产品构件需求。除了顾客需求外，还可能从所选择的解决方案中派生出产品和产品构件需求。

在整个产品生存周期里，需求可能发生演变。要针对所派生的和分配的需求对设计决策、后续的纠正措施以及从生产、集成、验证、确认、产品运行、支持以及处置等得到的反馈进行分析。通过分析来理解、定义和选择所有各个层次的需求。分析工作包括：

- 1) 需要和需求分析；
- 2) 操作概念的开发；
- 3) 所要求的功能度的定义；
- 4) 为处理成本和可供给性而开发相应的制造和支持概念。

功能度定义（也称为功能分析）不同于软件开发中的结构化分析，也不假定是面向功能的软件设计。功能度定义是在面向对象的软件设计中与定义服务相关的活动。功能的定义、功能的逻辑分组以及与需求的关系等，统称为功能体系结构。

在产品体系结构的比较细节的层次上可能需要不断进行递归分析，直到细化程度足以推进产品的详细设计、采办和测试。作为对需求、操作概念（包括功能度、支持、维护、和处置）以及制造或生产概念的分析的结果，将产生更多派生的需求和以下需要予以注意的事项：

- 1) 各种各样的限制条件；
- 2) 技术制约；
- 3) 成本和成本驱动因素；
- 4) 时间限制和进度驱动因素；

- 5) 风险;
- 6) 顾客或最终用户没有明确指出的隐含的问题;
- 7) 由开发者的专有的业务考虑、法律和法规引出的因素。

通过操作概念的反复演变,建立分层次的逻辑实体(功能和子功能、对象类别和子类别)。对需求加以精练,进行派生,然后分配到这些逻辑实体上。这些需求和逻辑实体进一步被分配给产品、产品构件、人、相关的过程或服务。

邀请所有相关的共利益者介入需求开发和分析,可以使他们了解需求的进展演变情况。

在软件工程中,又通常将需求开发划分为需求的获取、分析和定义。通过需求的获取,获取共利益者(例如客户、具体使用产品的用户、进行产品维护的人,又或者乙方的老板、销售、运维人员等)的对本产品的需求,通过需求分析将获取到的需求进行过滤、归纳、分析和汇总形成软件的需求,通过需求定义活动将软件需求通过文字结合图形予以呈现。

5 RM-需求管理

★ 关于需求规范化

Q (jane):

想请教一下,在项目开发的过程中,需求方面的规范化活动应该包括什么?

A (tyrone):

“需求方面的规范化活动”这个命题太广泛了。您几乎可以将需求开发(RD)及需求管理(REQM)都纳进来谈。而规范化活动,则是去订规章、制度、程序、与应有的产出物,让项目团队成员可以遵循,进而完成项目。从您的命题来看,是去订定项目中与需求有关的所有过程、活动、工作、及产出物。所以,首先您要识别,在项目中应履行哪些工作才能获得需求、需求要在何处管理、怎么管理、用什么管理、谁来管理...,接着是规划需求的活动,包括获得需求、开发需求、管理需求等等的活动,这些活动由谁执行、怎么执行、用什么方式执行、使用什么工具执行、如何回报结果及成效、用什么方式回报、由谁回报...,然后把规划的结果记录下来,经过评审的工作,订为项目的需求工作基准。

A (jane):

谢谢 tyrone 的专业回答,命题是大了点,主要是自己对需求这一块的把握无从下手,您提到的“在项目中应履行哪些工作才能获得需求”这一点,我之前也查过一些资料,但公司的实际情况是需求调研的人员一般就一个人,而我们面对的客户群比较特殊,客户在配合我们需求调研工作时,经常会有推泄责任,或者找借口延期的事情发生;导致需求调研工作进展难。当然客户有自己的工作,他们选定的那些项目组人员对于我们标准的需求文档都会很懒得看,这又给需求确认产生不少麻烦;而糟糕的是需求评审这一步公司内部首先就没能做到位;

个人想如果改变不了客户，只能改变方法，就只针对需求获得，如问卷调查，不过这个问卷的设置应该有什么要求？还有就是想用某种工具能无数据的演示一下；还有得到需求后，该如何表达，能让客户更乐意接受；这三个都是很头大的问题，希望能帮忙提个建议！！

A (tyrone):

问卷的设计旨在指导需求调研的人可以找出客户完整的需求，问卷内容的设计最好能够从系统工程切入，以下是一些例子：

1. 请说明系统要达成的目标为何？系统如何达成其目标？
2. 请说明系统作业要处理的业务流程有哪些？（以 operational concept 与 scenarios 表达）
3. 请说明系统需求的背景(例如法令规章的要求，或属于组织发展策略的指导)？
4. 请说明系统的主要产出有哪些？产出的格式有哪些？是否有范例？或既有表格？
5. 请说明系统主要输入有哪些？输入数据的格式有哪些？是否有范例？或既有表格？
6. 请说明输出数据的策略为何？实时或者批次产生？
7. 请说明与系统有关单位有哪些？这些单位各扮演何种角色？
8. 请说明系统的各项业务流程在何种条件下开始？
9. 请说明系统的各项业务流程在何种条件下结束或完成？
10. 请说明系统的各项业务流程间的顺序为何？
11. 请说明系统产生之输出数据是否有验证的需求？
12. 请说明对于输入系统之数据的验证需求？(例如，数据格式、数据类型、身分证字号的真实性)
13. 请说明使用单位的组织状况及签核流程？此部分是否需要与系统结合？
14. 请说明谁是系统的使用者？
15. 请说明使用者的教育程度？以及对于信息系统使用的经验要求。
16. 对于使用者教育训练的需求为何？
17. 对于系统维护人员的教育训练需求为何？
18. 对于系统操作支持人员的教育训练需求为何？
19. 请说明使用者在何种条件下才会使用系统？
20. 请说明对使用者接口的要求为何？(颜色、配置方式、风格如何？)
21. 请说明系统对于使用者要求的反应时间为何？
22. 请说明预期系统及其组件每天的作业时数为何？

23. 请说明预期系统及其组件每月的作业时数为何？
24. 请说明系统的开关机程序为何？
25. 请说明系统的备份程序为何？
26. 请说明系统备份的策略为何？
27. 请说明系统备份的作业要求为何？
28. 请说明系统的数据来源为何？进入系统的方式为何？
29. 是否有资料转置的需求？
30. 请说明系统运作所需要的作业环境为何？包括网络环境、操作系统等。
31. 系统是否有与其它系统共通作业(interoperation)之需求？
32. 请说明系统的信息安全要求、隐私权保护的要求为何？
33. 请说明系统可维护性的要求，或系统维护每年的时间限制为何？(多少小时以内)

一般客户项目组人员，最怕的就是担责任了，所以改变一下调研最后划押的思维，之前大家认知的可能是，签了名就代表认可所有的需求是完整的，而且是要负责任的，所以受访者一定不太愿意签名认可的。但是如果加注一句话：“所记内容与本人所述一致”，会比较容易获得受访者的认同，也愿意签名，因为这只是代表说与记的是一致的而已。

有关于演示，如果是我的话，我是可以接受 PowerPoint 演示，这个部分应该与客户先谈好，调研时的演示要如何做。当然有些客户想还没有开发就要有网页，这个还是需要事先沟通的。

一般很多人，会想要使用 Use Case，不反对，但不是所有的客户都看得懂，使用 Flowchart 也有相同的问题。表达的方式，也是一样要与客户先沟通确定，然后再以客户想要看到的方式去表达是比较保险的。

A (jane):

真是受益非浅!!! 谢谢 tyrone! 辛苦了!!

需求调研工作进展顺利些的话，调研人员接下来写需求说明书。写需求说明书，是不是让开发人员一看就知道如何来设计系统呢？总感觉跟使用说明书啊，或者设计说明书（公司现在没做设计一工作）类同。所以在写这些文档经常会把这个程度搞混，能说明一下吗？？再次感谢!!

A (tyrone):

在美国军方的一系列软工标准，所使用到的 DID (Data Item Descriptions)，对于几项生命周期文件的命名如下：

1. SRS-Software Requirement Specifications (软件需求规格书)
2. SDD-Software Design Descriptions (软件设计说明书)
3. IRS-Interface Requirement specifications (接口/界面需求规格书)

4. IDD-Interface Design Descriptions (接口/界面设计说明书)
5. DBDD-Database Design Descriptions (数据库设计说明书)
6. SSS-System/Subsystem Specifications (系统/子系统规格书)
7. SSDD-System/Subsystem Design Descriptions (系统/子系统设计说明书)

从上面大概可以知道与客户接近,与产品有关的,直接定义出来可以做为最后验收依据的都是规格(specifications),而承接需求,可以用来依样构筑出产品的,其实是设计说明,程序员应该是依据设计说明去实现软件产品的程序代码。需求规格则是用来和客户确认其对产品的要求:含有产品应该有的外观(form)、与构件间的关联(fit)、应该具备的功能(functions)等等数据化的、明确的、具体描述。也是设计人员的产品设计依据。

在 SDD 之中,可以包括软件的架构设计及细部设计、也可以再把 IDD、DBDD 的内容放进来。也就是在展现基于需求,每个功能会在那些软件的构件上实现,还有各个构件如何实现,基于完成功能需要,各个构件间如何互动(方法、程序、时序.....)。还有,这些被拆开分别设计的部分,最终将如何集成起来,要经过什么样的集成程序,才能得到一个满足最上一层级要求的完整产品。

文档各有其目的,当然有些公司或基于项目执行上的需要、甚至可能是客户要求的,而把文档合并,例如所有的系统文文件就是一本文件,涵盖了产品规格、软硬件需求、架构设计、软件细部设计、数据库设计等等,但是无论怎样的合并,应该要掌握“井然有序”、“符合工程开发的层级原则”撰写,从产品需求-->产品架构-->软件需求-->软件架构设计-->软件细部设计,同时在这些层级间建立起追溯关系,否则,当问题发生,要在最短时间以最有效率的方式,找到问题、评估风险,或者做出纠正措施都有问题。

A (jane):

居然一直忙着写文档,刚花了点时间仔细理解了 tyrone 的话,是不是可以这样子理解,从需求调研中得到的是需求可以整理成需求规格书,或者说是需求说明书,这两个文档只是形式上有些差异,内容上就要说明外观(form)、与构件间的关联(fit)、应该具备的功能(functions)等等数据化的、明确的、具体描述。就你提到的这些项,我再详细诠释一下,外观这应该只是一个参考性的东西吧,在没有设计好之前肯定是拿不出什么外观样式的,如果是按照以前的界面,那是不是最后还是要把这些都换成现实中的界面呢??数据化、明确、具体描述是不是就应该把具体的数据流程,以及每个流程中的功能按钮,以及这些功能按钮是如何操作的都写出来??这样子一来完全可以省略设计这一流程了;其次需求说明书中有可测性的要求,而测试用例是根据需求说明书来写,这就是说,需求说明书的内容可以跟用户使用手册相比了,天哪,又一绕回来了~~

可不可弄个实例,就比如日志记录这一个模块,记录的内容也可以确定的,主要就是修改呀,删除呀,新增的操作,这在需求说明书中应该如何来写呢??

A (tyrone):

呵呵, Jane 想得太多了,事实上,在需求阶段里,绝大多数的内容来自于合同里的需求表列,还有就是在调研阶段,对需求表列中不清楚的事项,经由对需求提供者的访谈与确认的结果,这些都是需求规格的内容来源。

我举一个简单的例子，曾经在台湾某政府机关的项目里，合同的需求表列有一个项目：提供系统问题的回报与处理机制。这个机制承包商的开发人员认为，小事一椿，在应用系统主页上提供一个菜单项，产生问题回报单填表功能，然后储存到数据库中，系统维护人员可以透过另外一个画面看到这个问题报告。因此，也没有再与需求提供者做深入探讨。等到要验收的时候，客户说，这个需求其实是要将该机构目前所有的 MIS 问题回报渠道合一纳入管理。因此，这个项目延后两个月才完成验收。

用这个例子一则先提醒大家，在进行项目的时候，千万别以自己的想法去揣摩客户的想法，或以自己认定的想法做为客户的想法，项目中，需求逐项确认绝对没有错的。

再者以这个需求项目为例，如果事先已经知道要与所有既有系统连接，收拢各系统的问题报告，那么，首先你要了解与那些系统有关系，与这些系统各是透过什么技术，使用什么接口，这些接口或技术方法就是 Fit(例如，使用 JDBC、Middle-ware、ODBC，与每个系统之间的都要描述)，功能(function)包括：产生一个填写问题单的画面、传送问题单、显示问题单、处理问题单....。至于 Form 指的是外观，当然一开始没有画面，但是可能基于某些要求，存在了 Form 的数据，例如：下拉菜单是放在横眉上，还画面的左侧；背景的色系用白色、蓝色、粉色系、字型采用几号字...，当然可能还没有具体的布局出现，或者有了，那可能就可以指定出长、宽、高，按钮的规格(size)....。所谓数据化，就是刚刚谈到的颜色、长、宽、高、字体为何、几号字等等有明确的值出来。当然这个还没有到达像数据流设计的地步。所谓明确，是指不要使用到形容词，例如：快速，就应该以“X 秒可以完成”来替换，可以处理“大量数据”，就应该以“X 笔数据”来替换。非常 User-friendly 就应该具体找出所谓 user-friendly 的定义是什么，是指操作性、还是整体画面感觉的舒适度，如果是指操作性，就要去界定操作的顺序是什么，由左到右，还是由左到右？或者有什么特殊的操作规定，整体的画面舒适度，那就要去定义画面的基础色调、字体的大小、需不需要特别的属性？例如要粗体字还是斜体字？菜单中的提示用字的用语及修饰的注意事项，哪些用语要被采用，或者哪些应该要排除的？

当然前面的数据化或者具体化的陈述，才有可能达到较好的可测性，因为你知道速度是多少秒，你可以用码表去量测、你知道是什么颜色，大家都不会有疑问(除非有人有色盲)，字体也有一定的量尺可以验证，而技术也可以透过标准技术的规范可以验证。

需求规格是设计的开始，是客户所认为的产品最终应该要符合的基准要求，这其中可能会因为层级的关系而会纳入不同的东西，像人工操作的需求、硬件的需求、软件的需求，但是，它的内容不会是人工操作程序设计的结果、硬件设计的结果、软件设计的结果，千万不要把它与设计说明混在一起，以免造成困扰。这样的建议，不代表不能将所有需求规格与设计说明合在一个文档中，当你要将所有东西放在一个文档中时，需求与设计等层次关系还是要明确地用章节区分出来，免得造成文档使用者的困扰(细部设计负责人员，可能要到处去找架构设计的结果)。

A (littlecool):

可不可弄个实例，就比如日志记录这一个模块，记录的内容也可以确定的，主要就是修改呀，删除呀，新增的操作，这在需求说明书中应该如何来写呢？

Answer: 使用用例 (Use Case) 规格划地描述需求是一个不错的选择。如果是用户需求，就使用业务用例来描述；如果是产品/软件需求，使用系统用例来描述。

A (tyrone):

以下是我的看法，仅供参考：

不管在需求规格或是需求说明中，不会写到修改、删除、新增的操作的。如果是我，在需求规格书中，大概只是写下面的几个字："日志记录应具备新增、删除、修改的功能"。

我的认知是，需求规格还写不到模块层次，因为那属于 High Level Design 设计说明的内容。大家如果有需要，我大概可以提供一些美国军方在做 COE(共通作业环境)的一些 SRS(需求规格书)给大家参考。

A (jane):

认真阅读后，个人对两位的评论触动很大。to tyrone: 您用具体的内容对 FORM、FIT、FUNCTION 的详细解释，使本人对需求说明书的要求有进一步认识；由于是自己未从事过开发工作，所以不能很确切表达需求，也不能确定自己所写的需求是否符合实际设计的需要？所以在这里本人问的问题要是很“幼稚” 尽请束谅。言归正传，您回复我的那例子的写法，个人有点困惑，如果只简单地把新增、修改、删除这些功能写上去，这个需求说明不是很模糊，比如修改的字段包括哪些也没说明；删除时需不需要提示，也没说明；删除可否多条记录一起删也没讲。

A (tyrone):

To jane, 所提到"比如修改的字段包括哪些也没说明；删除时需不需要提示，也没说明；删除可否多条记录一起删也没讲。"这个分成两段来说明。

1. "修改的字段包括哪些"，这个应该是写在 Change Requests 里面的内容，不是 SRS 的内容。然后 SD 的人员，依据这个 change request 的内容，相对修改到设计文件里。

在此要注意一件事情，所有对现有产品做修改或提升的维护项目，所有要增修的部分，均应视同「变更」来处理，将合同中的需求一个项目以一张 change requests 来执行、追踪与管制，因为每个需求的变更处理可能是不同的(层级可能不同，有的要从需求规格着手、有的要从架构设计着手或细部设计、有的只是改 code 而已)，但可能也有关联性，但是原则上，为便于管理，仍然是一个变更需求，一张 change requests。

2. "删除需不需要提示"，以及"删除可否多条记录一起删"，这两项会写到 SRS 里面，只是很可能合同(需求表列)里没有写，因此，如果 SA 已经有先前一些经验，TA 可以利用访谈时与需求提供者确认是否有这些需要，如果有，那当然就可以纳入需求规格书之中了，如果没有，就不会列入需求规格书之中。但是，一件事情，SA 是否可以自作主张把"删除提示"及"可以多条记录一起删"不经访谈就放到需求规格书中呢？我觉得是不可以的，因为这些工作在 SA 是一句话，但在 SD 及 Coding、测试阶段则有更多的 effort，所以，SA 不能只是考虑自己展现对客户"聪明"及"善解人意"而已。以上请参考

6 CM-配置管理

★ 关于配置审计与配置状态报告的几个问题

Q (sungubbi):

在我们公司，配置审计工作在执行了一年后从最初的好奇和尝试慢慢的越来越形式化，因为从审计这项工作中，配置员和项目组感觉发现无法从中获得好处，我的问题是：

关于功能审计

1) 功能审计是否可以与相关的评审相结合? 也就是在完成相关的需求、设计、用例等评审的同时, 完成功能审计, 输出的结果就是一份评审报告? 对于评审发现的问题的修改情况是否也同样需要进行功能审计。

2) 或者功能审计独立于评审, 通过抽查产品输出之间的一致性, 并通过检查是否经过评审来判断其正确性? 形成独立的审计报告。

以上两种方式, 哪种更为合适? 通常由谁来完成功能审计更具有可执行性?

关于物理审计:

在配置项进出配置库时, CM 工程师已经完成了日常的具体审计。是否仍然要求定期的对配置库进行物理审计, 比如一月一次? 除了形成审计报告外, 是否还有其它更有效的留痕方法?

关于配置项状态报告:

使用 EXCEL 的方式来汇报配置项状态 (当前状态、变更历史、版本、所处位置), 在每一次的检查中我们均可以发现这份报告是不够完整的, 或者不够及时的, 而项目组也未从这份报告中获得好处, 想要完整及时地更新这份报告需要较多的投入, 而如果只是定期形成单靠人工就存在遗漏。(我们公司属于产品类项目, 即研发一个产品面对着上百家客户, 用不断的版本升级替代旧的版本, 通常旧的基线被新的基线替换, 较少产生基线的变更)

请问, 在没有工具的情况下, 配置项状态报告可以简化到什么程度? 如果缩小配置状态报告的范围, 是否只需展示基线的内容? 是否可以只汇报异常, 而不汇报所有配置项的内容

A (fishred):

偶认为配置审计的内容每个公司的要求不一样, 可以在项目组内部约定个审计的要求;

Q (sungubbi):

那么应该达到谁的要求、什么样的标准呢? 如果是达到配置组的要求, 配置组一定会要求都取消。如果是达到项目经理的要求, 项目经理由于没有从中感受到它的重要性, 那么他们也会要求去掉这两个活动。配置员总是笑笑说: 瞧, 我们的审计和配置状态报告, 就是为了 QA 检查而做的。这种情况, 也许大家都会遇到吧。

A (step365he):

这个问题和我在 SE 体系区的软件测试板块发的帖子讨论的主题有相似之处, 作为他山之石给笨笨鱼参考:)

功能审计是否可以与相关的评审相结合? 也就是在完成相关的需求、设计、用例等评审的同时, 完成功能审计, 输出的结果就是一份评审报告? 对于评审发现的问题的修改情况是否也同样需要进行功能审计。

个人觉得根据项目规模大小而定。如果小项目, 可以采用非正式审计, 对于发现的问题使用非正式或者口头报告就可以。一般我这会采用邮件留底。

大项目, 因为有正式流程, 因此少用形式化的手续, 但是审计这个必要的得使用 (包括对于评审发现的问题审查); 况且报告是评审提炼出的精华。我的体会是恰当的形式能提高开发人员的能力, 特别是分析能力和全局观。

另外，物理审计是个什么工作？

A (tyrone):

功能性配置审计(FCA)与物理性配置审计(PCA)

有关于功能性配置审计与物理性配置审计目的、准备事项、审计时的评审事项如下所述：

功能性配置审计(FCA)

- FCA 用以验证配置项的实绩，满足其预划性能的要求。
- 对于极大型、复杂的配置项与系统来说，FCA 可采取渐增方式来完成。
- 在所有的渐增 FCA 完成后，应举行最后(总结) FCA，并将 FCA 的状况记在会议纪录及认证书中。
- 虽然对于每一项配置项或系统仅须做一次 FCA，但是，在配置项或系统生命周期中，还可能执行许多类似 FCA 的作业。
- 有许多第一级的工程变更建议书(ECP)，会将新的设计融入到已订为基准的设计之中。新设计要项的性能，须经验证，以确保该项设计不会降低配置项或系统的性能。

功能性配置审计准备事项：

- 供识别每个配置项规格之配置识别、配置控制需求的矩阵，以交叉参照：
 - 测试计划书、测试程序，以及每项需求测试的结果
 - 为验证需求而展示、检视及分析的书面化成果
- 可资应用的规格书、设计、时程表、验证测试计划书及测试程序、验证测试结果、以及演示、检视及分析的文件等

功能性配置审计评审事项：

- 评审规格、验证过程与结果
 - ◆ 测试计划书/程序遵守规格的要求
 - ◆ 测试结果、分析、仿真等等，证明配置项的需求，和规格的要求相同
 - ◆ ECP 均已融入配置项中，并已经过验证
 - ◆ 接口(界面)需求已通过验证
 - ◆ 配置文件反映出配置项的配置，其测试数据已经过验证
 - ◆ 将提供之配置项的数据被取样出来，以确保这些数据参照到可资应用的性能与测试的需求
 - ◆ 对于计算机软件配置项，
 - 数据库、储存体的配置、时机与顺序与指定的需求一致
 - 软件系统营运与维护的文件已完整

- 测试结果与文件反映出正确的软件版本
- 内部的 QA 审计已满足要求

◆ 暂时性的偏离，依核定的变异要求书记载于文件中

物理性配置审计(PCA)

- PCA 用以检验配置项的实际配置，以验证相关的设计文件与可交付之配置项的设计相符。
- PCA 亦用以对卖方用于配置项产制之支持过程做确认。
- PCA 亦可用于对已完成 FCA 后重新设计之配置项要项，亦能满足配置项的性能规格要求。

物理性配置审计准备事项：

- 配置项目详细规格的最终草案
- FCA 会议纪录
- 设计(蓝图)、工程/制图手册
- 交付评审之配置项识别(特别的序号)
- 可不受限地进入用以检视、制造、生产、组装、测试的设施
- 可取得工程设计(蓝图)、检视报告、过程表及其它可用数据所参考到的所有文件。
- 评估与验证所需之工具和检视装备，以及测试软件

物理性配置审计评审事项：

- 产品基准
 - ◆ 依据构成配置项或计算机软件配置项产品基准的规格及设计文件，对配置项或计算机软件配置项的配置进行正式检验
 - ◆ 保证如工程蓝图(如下)所映的适当组件，已真正安置并正确标示
 - ◆ 确定产制中的配置，正确地反映已发行的工程数据
- 工程设计(蓝图)或计算机辅助设计(CAD)描述(设计细节)评审
 - ◆ 评审蓝图(或计算机辅助设计描述)代表号及相关制造指示的正确性，以保证制造指示(硬件依此制造)反映所有设计细节，同时纳入已授权的工程变更。
 - 设计(蓝图)号码及制造指示的修正版与正确发行的设计(蓝图)或计算机辅助设计描述相符
 - 设计(蓝图)与修正版在发行纪录中正确显示；设计(蓝图)未整合的变更应少于五项
 - 制造指示的数据清单与设计(蓝图)分部清单相符

- 命名法、组件编号及序号的标示均正确
- 所有奉核定的变更均已纳入其中
- 主要总成(Assembly)的组件(Component)参考数据及其它特征,从最顶层蓝图到每一分部具有连贯性
- 必要的核定事项俱全

Sungubbi 提到“关于功能审计.....以上两种方式,哪种更为合适?通常由谁来完成功能审计更具有可执行性?”

审计的定义是由具有适切之独立性的单位来执行的活动,看看工作产品是不是遵守相关的需求、计划、以及协议书;而评审则是对产品工作产品提供意见或者是核准的过程;此两者的目的并不相同,执行的要素也不同,并不能够被合并起来。但是在此处有一个问题必须厘清,那就是软件的功能审计通常是在软件已经完成之后,就已完成之软件产品相对于需求规格去实施的,辅以测试纪录、测试用例等等,功能审计仅实施一次即可,亦即在产生产品基准之前。而相关需求、设计、用例等的评审,通常是在需求分析完成前、设计阶段完成前实施的,所以并不存在合并的问题,因为功能审计和需求与设计评审的发生时间点不相同。

Sungubbi 还提到“关于物理审计:在配置项进出配置库时,CM 工程师已经完成了日常的具体审计。是否仍然要求定期的对配置库进行物理审计,比如一月一次?除了形成审计报告外,是否还有其它更有效的留痕方法?”

物理审计的标的物并不是配置库,而是配置库中的配置项目,拿完成的产品与设计信息核对,在技术上是是否相同、一致、完整,通常对于软件而言,也只需要实施一次物理审计即可,当然可以结合在设计阶段去实施,但不论多少次,最后在产生一份给客户的产品基准前,要实施一次总结性的物理性审计。

至于 sungubbi 提到的“关于配置项状态报告:请问,在没有工具的情况下,配置项状态报告可以简化到什么程度?如果缩小配置状态报告的范围,是否只需展示基线的内容?是否可以只汇报异常,而不汇报所有配置项的内容?”

建议分各客户,建立一份配置状态报告的网页,参与项目的人员以及客户可以按照权限查阅到所有配置项目的最新情况,配置经理随时依据最新的状况去更新该网页,这个网页可以符合多种配置管理的要求。例如配置状态报告、配置项目总表(inventory)、在这个网页上也可以提供对于最新版本的文件或代码提供连结,方便工作人员取得最新的版本,而这样结果也就构成一个简单的配置管理系统。

还有 jane 提到的“配置项真的很重要吗?我目前公司里,都没有体现出配置管理的活动~~~~”,ISO/IEC 12207:1995/AMD 1:2002 中,有关于配置管理过程的目的与成功实行的结果摘述如下:

F.2.2 配置管理过程

目的:

配置管理过程的目的是在建立与维护过程或项目之所有工作产品的完整性(integrity),并让相关的各方可以取用。

结果:

配置管理过程成功实行的结果为：

- 1) 配置管理策略已经订定；
- 2) 过程或项目产生的所有配置项，均已鉴别、定义，并订为基准；
- 3) 配置项的修改与发行受到控管；
- 4) 修改与发行已让相关的各方可以取用；
- 5) 配置项的状态与修改的要求，已列入纪录及报告；
- 6) 配置项的完整性与一致性已予确保；以及
- 7) 配置项的储存、处理与交付已受到控管。

配置项的划分没有固定作法，一个项目里可以把一个产品当成一个配置项，或者一个产品划分为数个配置项，完全看管理的方便性。

配置管理对于软件产品为主的公司非常重要，尤其你是以一个核心产品，然后依据不同客户的需求去 **Customize** 时很容易就看到它的好处，如果没有配置管理，你可能在为客户 **build** 一个产品的时候，有可能抓错 **components**；某个客户新上的版本有问题时，你可能就找不到或回不到上一个可用的版本了；而当客户回报一个错误，工程人员要修改时，有可能就会改到另一个客户的产品，造成另外一个客户在上新版的时候，发生不可预测的状况，所以无论如何，一定要把配置管理建立起来。

A (sungubbi):

谢谢林老师的回答，您提到：“建议分各客户，建立一份配置状态报告的网页，参与项目的人员以及客户可以按照权限查阅到所有配置项目的最新情况，配置经理随时依据最新的状况去更新该网页，这个网页可以符合多种配置管理的要求。例如配置状态报告、配置项目总表(**inventory**)、在这个网页上也可以提供对于最新版本的文件或代码提供连结，方便工作人员取得最新的版本，而这样结果也就构成一个简单的配置管理系统。”

从这句话里我有几个思考：


1、建立这样的一套系统也好，或者只是使用表格来反应配置项状态也好，应该都有一个粒度。我认为在您的这个例子中，展示给用户的那块，粒度可能只需在 **FUNCTION** 级，用户可以看到所提供的产品功能构成和产品的实现情况即可。而当进一步与文件或代码连接时，粒度则需进一步的细化，才会更符合项目进行管理的需要，但不需要细化到每一个具体的文件。

2、如果我们反映的是产品的状态，那么这份状态报告是否就可以只是针对基线产品来建立，也就是反应了基线产品的状态和变更情况？这份报告的维护时机，可以是在基线形成中途的小里程碑时进行，或者在基线形成时建立（但这样的话，这份报告只能反应配置项的组成和当前状态，而给项目组用于进度参考、配置项变更参考的目的似乎就不太强了。）

3、以 **OFFICE** 这样的软件为例（这个产品的开发周期过长与我们的实际情况相差较大，假设适当地缩短几个版本的研发周期控制在每半年推出一个新的版本），在 **OFFICE** 的 98、2000、2003 时分别发布时，以及在新版本研发周期中推出的 **SERVICE PACK** 是，是否只需要推出版本时才建立配置状态报告呢？

关于配置审计，按照您的解释，我可以对我们现在的做法进行较大的简化，应该会减轻配置员的工作量。不过不同的咨询师间总是给出不同的答案，例如我们之前的咨询师不仅强调基线生成时的物理和配置审计，甚至要求日常的审计（例如每月一次），甚至包括组织财富库方面的审计（我认为 QA 已经可以实现该工作，无需配置员再进行），这些模型之外的要求，往往让我们很无奈。

A (tyrone):

 sungubbi:

1、建立这样的一套系统也好，或者只是使用表格来反应配置项状态也好，应该都有一个粒度。……，但不需要细化到每一个具体的文件。


2、如果我们反映的是产品的状态，那么这份状态报告是否就可以只是针对基线产品来建立，也就是反应了基线产品的状态和变更情况？……

对于以上内容，其实，配置状态报告主要在于向相关的 stakeholders 说明每一个配置项目的现况，有哪些订为基准了，哪些还没有，这些还没有的是处于哪个工作阶段，是“起草”、“评审”、“核准”、还是“修改”；起草或修改人员是谁、预计何时发布新版本；最新版本是哪一个、最新基准的版次是什么、基准发行的日期、新版本与旧版本的差异有哪些，这些目的就在于让 stakeholders 知道自己手头上的工作依据是不是改变了，如果改变了他就必须取得新的版本，以免事情越做越错。对于客户而言，也可以经由这个状态报告，知道项目工作的进程是不是令人满意。

以下是 ISO/IEC 15298 第 10.12 节 Configuration status report，所提出的内容建议：

配置状态报告提供受控配置项之状态，包括基准、发行识别符、以及最新的软件配置项版本。它可包括项目的变更数、发行版本数、及发行版本的比较。

你可以用项目或(客户的)产品为基础发布一个网页，就把前面建议的内容放到网站上即可，涵盖的是项目内或(最终交付)产品有关的所有的配置项(甚至于包括开发环境的版本、DBMS 的版本、目标环境的操作系统版本等)。采用网页方式的配置状态报告的好处是，你只要配置项有新的状态就更新。但是如果为了工作量分配的关系，也可以采取周期性方式处理更新，那这个时候就得要有一个纪录暂存的机制了，要不然可能会造成正确数据的流失。

 sungubbi:

3、以 OFFICE 这样的软件为例……分别发布时，以及在新版本研发周期中推出的 SERVICE PACK 是，是否只需要推出版本时才建立配置状态报告呢？

应该是说，推出版本时才建立的是另外一个文件—版本说明书(SVD, Software Version Description)，这个文件与配置状态报告并不相同，这个 SVD 所含的内容是与先前版本的差异状况，技术变革的情况，环境的需求、使用者应该注意的事项，如何安装，已知的问题，如何规避这些问题状况的发生，还有与该新软件的相关技术问题等等。而配置状态报告基本上是在经过审查、核对，持续地把各配置项的状态弄清楚后发布，以协助工程人员做好工程

开发的工作，并供客户了解工程开发的进度，应避免与 SVD 搞混了。实务上，有时候 SVD 可能就是一个文字文件格式的 readme.txt 而已，随新版本交付给客户。

请把握一个重点，从国际标准 ISO 9001:2000、ISO/IEC 12207:1995 含 AMD 1:2002 及 AMD 2:2004、ISO/IEC 15288:2002、IEEE 所有的系统工程与软件工程标准、CMMI 框架，都要求过程要经过裁适的，以适合组织(这个组织可以是公司、部门、项目组织)的规模与工作范围。而在国际标准 ISO/IEC 12207:1995 及 CMMI 其实主要的焦点组织是项目组织，因为都谈到生命周期模型及生命周期过程，也就是到达产品或软件的获取、开发、运作与维护等阶段的工作，这就回到产品的开发项目组织上，也就是一个项目团队，而我们在谈裁适的时候，ISO/IEC 12207 谈两层次的裁适，第一个是组织，第二个是项目，以 ISO/IEC 12207 提供的生命周期过程裁适出公司合用的组织标准过程；到了项目时，再以组织标准过程为项目裁适出项目组织可以执行的项目已定义过程，这个方式与 CMMI 里所强调的并无二致。CMMI 里(ML3)谈的是，公司订出 OSSP 放在 PAL 中，项目经理在规划项目时，以这个 OSSP 去订出项目的已定义过程纳入项目计划里。

“强调基线生成时的物理和配置审计”并没有问题，只是组织的规模做不做到？如果你采用的开发生命周期是 Evolutionary 或是 Incremental 的开发方式或者 iterative 的开发方式，这些方式都是公司在面对不同项目时可以选的，而且采用 Evolutionary 或 iterative 开发的产品是一项内部核心技术的研发，只分配了两个人，因为“基准是产品/工作产品，在工程上的某个时间点经过核准而产生，成为下一个阶段的工作依据”，这是基准的定义。基于这个基准，每次都要物理审计及功能审计，尽管内部的开发项目可能没有太多的时间压力，可是一直实施这些配置审计还是非常耗时间的，在 MIL-HDBK-61 及 MIL-STD-973(这两本是美军有关于配置管理的标准与手册，很多其它标准中的配置管理过程，是以这两份文件为蓝本的)就提到至少在交付前实施这两项审计即可，但是它并没有排除分阶段的多次审计，问题在于你的项目有多少人，产出是不是很复杂，如果你希望对于配置管理中有关于审计有更深入的理解，除了可以看前述两份美军文件的功能审计及物理审计之外，还有工业标准 IEEE STD 1042 配置管理实践建议，在 1042 这个标准中，后面有四个例子，不同的情况，采用不同的作法，甚至于不实施配置审计，而由其它的审计来代替，或由客户主导实施配置审计及功能审计，值得参考。

至于审计的频率(日常审计，每周一次或每月一次)全视公司层级或项目层级的产品、工程进程、参与的人数而定，如果产品经常被修改，可以加强审计的频率、但此同时也要看看有没有充分的人力与时间以应付审计，包括审计规划、审计执行、后续的问题处理、提出审计报告等等，如果没有，光是审计，却做不了矫正的措施，意义也不是很大，因为高频率的审计，为了绩效的问题，鸡毛蒜皮的小问题都会浮上台面，到底矫正不矫正？矫正就多花时间，不矫正又觉得审计根本没有效用。这些都是制度设计者(包括项目管理的规划者一项目经理)要好好思考的。

咨询师的话可以听，但不见得全盘接收，另外，花些时间研究一些标准文件，包括国际标准、美军标准、工业标准及国家标准，会让你对于过程及过程如何运用有更深入的理解，从而可以判断咨询师说的可采用度到底有多少了。

A (sungubbi):

关于配置审计的讨论，对于要做怎样的改进，现在我已经心中有数了，谢谢 TYRONE。

关于配置项状态报告,我认为配置项目状态报告从产品部件的角度展现了项目的进展情况。通过配置状态项目报告,项目组可以很清楚地看到,当前这个产品,哪些是已经开发完成的,哪些在修改中,哪些还未得到开发,同时通过该报告还可以了解到配置的稳定性。应该说,它产生于项目开发过程之中,并且为项目的进度控制服务,但这也对配置项状态报告的及时性提出了要求,如果不能做到实时更新,那也需要定期进行更新吧(例如每周或每两周)。但目前维护的工作量较大,导致配置项的跟踪总是缺漏,或者在发布时才能够提供,而背离了配置项状态报告的初衷。如果没有使用工具,或者像 TYRONE 推荐的那种建立网页的方式,我感觉这项工作的形式化将仍存在一段时间。

Q (tyrone):

能请教一下何谓「非正式审计」? 谢谢!

A (sungubbi):

也许是说进行审计,但不会像正式审计一样留下审计报告吧,呵。对于小项目,我们通常在产品发布时进行一次审计,中间就不会再安排了。大项目则分阶段进行审计,发布时要求功能审计和物理审计。

A (tyrone):

所有的活动应该都要有报告出来的,报告只需要说重点就行了。如果做一件事情而没有留下纪录,要说有做过还是不会有人相信的,更何况,所做的每一件事还都有成本存在,虽然几分钟的时间,但是累积起来还是很可观的。而没有留下纪录本身,只会让人产生“可能没有实施”的疑问而已。

所以,“非正式”的概念,在国际标准或工业标准里,指的不是没有留纪录(没留纪录会成为 non-conformance 或 non-compliance),而是执行时的规模、参与的人员、采用的程序、方法,并未完全遵循组织或项目计划里订的标准。

所提到“对于小项目,我们通常在产品发布时进行一次审计,中间就不会再安排了。大项目则分阶段进行审计,发布时要求功能审计和物理审计。”其实正确的说法,这些都属于对于项目的 tailoring。这个观念与作法是正确的,符合 CMMI 及各项国际与工业标准过程的运用精神,但重点是,这些 tailoring 的结果(小项目要实施多少次、范围多大的审计,大项目要实施多少次、实施的周期、时机、范围多大)都要列入项目计划里,以便在项目执行时遵循。

★ 对于如下三类配置项你们是如何进行变更控制的? 采用一套变更流程? 还是有不同的变更流程?

Q (xixiaojing666):

1. 项目计划
2. 软件工程的配置项(交付客户的配置项如 SRS、概要设计、详细设计等……)
3. 测试阶段的代码(经过了单元测试)

A (sherry):

这个我们公司会根据项目的规模、性质、客户的规范性走不同的变更流程。

1、项目计划的变更，我们会根据项目规模的大小、变更弧度的不同走不同的变更流程。对于一些小变更或不影响里程碑的，一般不通过评审、直接以风险的形式提前上报领导，有的项目在公司规定的变更范围内是不需要做变更通知；对于一些大变更，或者对关键里程碑影响较大的，我们需要走正规的变更流程，甚至需要进行计划变更评审、审批等流程。

2、软件工程的配置项：提交给客户的配置项一般在项目计划时都跟客户协商好的，因为从公司的角度出发，对于一些原先要提交给客户，后来不提交的，我们一般不做变更流程，只要能拿到验收证书即可；但如果是原来计划不提交的，但现在又要提交的，则需要由客户向项目经理提出申请，再由项目经理提交变更申请（包括提交的原因、保密程度等）、领导审批等流程。

3、测试阶段的代码（经过了单元测试）：代码我们不作为变更流程来走，而是根据缺陷、补丁管理的要求进行实施。

A (sungubbi):

从配置项本身的变更流程上，计划变更和产品的变更我们共用一套。所谓变更，就是对已经受控的配置项提出的修改请求，此时不论是计划还是产品，均需按规定的变更流程申请到配置项后方能进行修改。暂且定为《变更规定》吧

但从变更的申请、评估和分析、批准的角度来讲，我们又独立制定了文件来分别予以描述计划变更和产品（或者基线）的变更的要求，在文件中当评估通过后决定变更时引用《变更规定》

对于经过了单元测试的代码的修改，我们的不把它定位于变更的范畴，而是依据缺陷管理的要求实施，首先是由专人负责评估是否需要进行修改，然后指定开发工程进行代码修改并标识，配置工程师根据标识正确提取那些经过修改的代码集成到下一个集成包去。

A (xixiaojing666):

结合我们自己的做法以及各位的回答,我得到了一些启示:

1. 项目计划的变更,如果是大变更需要进行申请和审批流程.但是流程文件和软件工程的配置项变更分为两套不同的模板.如果是小变更则项目经理直接批准,直接修改项目计划,只要留下修改记录就可以,不必召开会议等等
2. 软件工程的配置项(交付客户的配置项如 SRS、概要设计、详细设计等)则是遵循 CM 过程中的变更流程,进行变更控制。
3. 测试阶段的代码（经过了单元测试），不走变更流程,按照缺陷管理的流程来实施。

另外产品的变更，我们使用另外的变更流程和变更分析模板。（这个变更过程写在了维护指南里）

还有个问题需要请教,各位的计划变更有使用变更分析表?(目前我们的计划变更没有任何变更分析表)而且这个计划变更的流程在哪里规定的?我们写在了 PP 过程里,软件工程的配置项的变更流程写在了 CM 过程中,产品的变更我们写在了维护指南里。

各位都是怎么做的?我怎么总感觉有些乱呢?大家帮我理理!

A (iamredeye):

change mgmt 是项目管理的范畴, 所以考虑这个问题首先从项目的管理的角度去考虑, 而不应先去考虑到 deliverables 的层面, 否则会理不清思路。project mgmt 主要从 change 对 scope, cost 和 time 的 impact 来考虑, 尤其是 scope 类型的 change。然后定义出多大的 change 由什么人来决策, 怎样 communication 等等。最简单就是一个流程, 包括了各种可能的分支, 就香程序设计里的流程图。至于什么样的 documents 会受怎样的影响, 这些是细节了。

btw, 你的 3 种分类也有点奇怪。1 和 2 我还可以理解, 2 和 3 就奇哉怪也了。。当然最后到 deliverables 层面可能会这样处理, 不过开始这么考虑是不是想复杂了? 比如我问一个问题: 发现了一个 bug, 虽然最终体现在 source code 里, 但也许是 design 的问题, coding 本身没问题, 那改 design specs 是不是 defect mgmt 的一部分呢? 当然 design 可能某些 release 的人不关心, 但是从项目的角度看, design 也好 code 也好, 都是 change mgmt 实施到后面的一些细节了。

★ 如何进行数据备份

Q (xixiaojing666):

各位是怎么进行数据备份的呀? 我们的做法:

1. 首先将数据按照更新程度进行划分, 比如需要每天备份、需要每周备份、需要每月备份的;
2. 然后使用文件压缩软件, 将要备份的数据进行压缩;
3. FTP 方式将压缩好的 rar 文件转移到目的地;
4. 使用 OS 的计划任务将批处理文件在你定制好的时间内自动执行。

注: 2 和 3 可以通过一个批处理文件实现。

感觉这种方式优点简单、方便; 缺点是只适合小 rar 包, 只能实现冷备份。

大家有没有什么更好的时时备份方法呢?

A (rebeccazxy):

俺们跟你们大致相同, 俺们的服务器是 linux 系统, 做法如下:

1. 首先也是数据类型划分, 分为组织级和项目级, 然后分别规定备份频率和备份内容;
2. 写个备份脚本, 实现数据的打包 (.tar 包)、本地备份和传输 (SSH 方式);
3. 利用 Linux 的 cron 来定时执行备份脚本。

目前打的包大小一般都是在 2-3G 左右, 暂时没有发现什么问题。只是因为异地备份, 所以受网络影响较大, 不过一般放在夜间进行, 倒也没什么问题。

异地备份, 是通过 Linux 的 SSH 传输来进行的。

首先, 实现定时本地备份, 这个不用多说;

然后，设置 SSH 传输，这个分为以下两步（假设源数据服务器是 A，备份服务器是 B）：

1. 设置 B 机器的用户（一般是 cvsadmin）可免认证登陆 A 机器，这个主要通过统一两机器的公钥和私钥实现；
2. 通过脚本实现 B 机器从 A 机器中取备份数据到本地；
3. 通过 cron 实现定时运行 2 的脚本。

7 TEST-软件测试

★ 单元测试执行不好，怎么办？

Q (sungubbi):

一个场景：开发组在开发完成后，未能执行好单元测试工作，就提交代码进行产品集成和测试，导致集成、测试时因无法构建或测试发现低级问题而无法继续。在开发组内部，有些开发人员是有进行单元测试和模块联调的，但他们不愿意留下记录，而那些未进行单元测试的人员在代码编写后就直接提交了，导致无法区分开发组是否真的做过该项工作。

曾经采取过的措施：1) 强调在开发人员开发完成时，必须进行自我单元测试。——不能实现自律，问题依然存在。2) 开发组短期内的构建和联调，由开发组完成——一个别项目组做到，但开发人员会依赖组内构建和联调，而忽略单元测试工作。3) 由开发组留下进行了单元测试的脚本、测试重点与测试数据。个别项目做到，会留下记，但不一定真正执行单元测试。

依然存在的问题：单元测试未真正落实，在进入产品集成和确认测试时，仍然无法顺利构建，在测试时发现低级错误。

求解：面对这种情况，怎么办？怎样能够促使开发人员自觉、有效地完成单元测试？是否执行过单元测试，QA 在检查时如何验证？

我的一个小想法：进行每日自动编译和自动测试，时间定在每个工作日的下午 4 点左右，在自动编译和自动测试过程中发现的问题，由开发人员当天解决（需占用晚上时间加班），迫使开发人员在开发提交代码前就进行单元测试。

A (沟通无限):

单元测试要用行政的规定来做，作为开发人员，考核的一个标准，和项目奖金挂钩。应出一个单元测试的规范。如果达不到就单元测试的规范标准的，嘿嘿，就从项目奖金里扣。建议 QA 在遇到问题时需要前、后都想一下，向前想为什么会出现这个问题，例如不按规定做单元测试，是意识（执行）、能力、技术、过程、产品结构等那些方面的原因，哪些原因最重要，如果开发组本来就做不好这件事，不只是意识问题，就要先考虑怎样报告问题，包括协助开发组解决问题的原因。向后想这个问题的影响，有什么措施能够解决问题，促进业绩，例如提高代码质量、减少签发后的缺陷；如果由于能力、产品结构等方面原因想不到有效措施，那报告问题的方式就更需要斟酌。QA 不好强烈要求项目团队去做那些按规定要做，实际做了也收效不大的工作。

A (如你的愿):

开发组提交代码前有签发机制, 开发组负责人保证提交的代码是经过本组验证的, 以后发现的缺陷要被证实记录、统计、分析、评价等等。

A (step365he):

执行的情况如果要求留下记录”, 其实很容易留下, 很多 xunit 工具都支持, 代码检查的时候也将 xunit 包含。

A (lily_014):

这个问题好像普遍都存在, 哎, 以前的老员工好点, 有时对于主干的代码会进行单元测试, 但后来招的新人基本上没有这样的习惯。

不过鱼的想法我也很赞同, 代码提交到配置库的准则是执行过单元测试并通过的. 否则不让提交. 至于有无执行的记录的话, 整个单元测试代码生成工具可否?

A (lilyli):

我们以前也存在这个问题, 现在基本没有了。解决的方法是:

1、开发人员完成开发后, 自己进行单元测试, 没有问题了, 提交源码至 server;

2、PM 或开发经理进行集成测试。这个步骤很重要。开发人员是归属 PM 或开发经理管理的, QA 去直接交流未必有效。让 PM 或开发经理做提交给 Tester 前的测试, 他们发现问题就会直接去跟 Engineer 说, 这样效果会好很多;

3、经过了集成测试后的程序, 提交给 Tester 进行测试。此时会先“冒烟”, 发现明显问题立刻“退货”。这样 PM 或开发经理也会注意提高自己的测试力度。

这样三个步骤下来, 收效还算不错, 现在三个环节上的人员责任感都高了很多, Tester 的怨言也少了很多。

A (stop-start):

我们公司现在也在探索如何执行好单元测试, 因为部分项目单元测试执行不好, 以至系统上线或升级时问题频出, 追究原因, 有很大一部分是因为开发人员所测试的项目不全或不对引起的, 因此目前我们打算先从以下几个方面入手:

1、开发了自动抽取 C 语言代码测试用例的工具, 开发人员要简要记录这些用例的测试结果, 如 OK, NG, 问题的简要描述。

2、对于没有自动抽取工具的情况, 要求开发人员必须列出单元测试用例, 项目组长要对用例进行适当的确认, 可采取抽查、互查等方式; 开发人员同上一条一样记录测试结果。

3、对于没办法做单元测试的模块, 要求项目组进行代码走查并留下走查记录, 现在对于个别重点项目, 走查是必须过程, 但走查的范围可以定制, 不需要对全部代码进行走查。

我们这样做是考虑首先要保证测试内容的正确性, 用例正确了测试才有意义。但目前这种方法的弊端是: 1、增加开发人员的工作量, 对于时间紧的项目组会比较抵触; 2、结果记录也有可能是开发人员未经测试自己填上去的, 组长、QA 也无法跟踪是否确实做过单元测试, 只能依赖开发人员的自觉行了。

我们将跟踪这种做法一段时间，看看效果如何，如果不好，那只能像以前个别部门那样，在后面的阶段如果发现 CT 测试遗漏的问题，就要追究原因和相应的责任人了。当然，我们更希望经过一段时间，我们可以找到更简化的操作方式，减轻项目组的工作量。

个人是认为如果能实现自动测试那是最好的，但目前所知的自动测试的工具并不能完全实现单元测试的需求，大部分还是依赖开发人员的手动测试，大家有没有更多的有关自动测试方面的经验可以共享……期待中

★ 能否介绍一下DRE模型呢？

Q (sungubbi):

能否简单介绍一下该模型的运用目的、内容、运用的时机呢？

A (jnpanda):

DRE (Defect Removal Effectiveness) 模型中文一般翻译成缺陷清除率或缺陷清除效率。DRE 是用来衡量缺陷清除活动（比如评审、测试等）效率，理论上说来， $DRE = \frac{\text{清除活动中实际发现的缺陷数}}{\text{清除活动中应该发现的缺陷数}}$ 。

但是，我们知道，“清除活动中应该发现的缺陷数”一般来说就是产品中所有的缺陷，实际上是不可能知道的，这就需要一种近似计算。实际计算中，我们用（当前活动发现的缺陷数+此后发现的缺陷中应该在此活动中发现的缺陷数）来近似。

举例来说开发分为需求、设计、编码、测试 4 个阶段，每个阶段有一次缺陷清除活动，分别是需求评审、设计评审、代码审查和验收测试。假设需求评审发现了 10 个缺陷（这 10 个缺陷一定是需求阶段引入的），设计评审时发现了 10 个缺陷，其中有 5 个是需求阶段引入的，代码审查时发现了 10 个缺陷，其中 3 个是需求阶段引入的，测试中发现了 10 个缺陷，其中 2 个是需求阶段引入的。那么需求评审时总共应该发现的缺陷数近似等于 $10 + 5 + 3 + 2 = 20$ 个，需求阶段的 $DRE = 10 / (10 + 5 + 3 + 2) = 50\%$ 了。同样的方法，可以计算设计阶段的 DRE，但是这是需要注意，设计阶段应该发现的缺陷数，除了设计阶段本身引入的缺陷外，还应该包括从需求阶段漏过了的 $10 - (5 + 3 + 2) = 0$ 个缺陷。以此类推。要计算所有阶段的 DRE，可以构造一个矩阵，横向表示缺陷注入的阶段，纵向表示是各个阶段对应的缺陷清除活动，如下表，就可以计算了。

	需求	设计	编码	测试	合计
需求评审	10				10
设计评审	5	5			10
代码检查	3	3	4		10
验收测试	2	2	2	4	10
合计	20	10	6	4	40

从上例中我们可以看出两个问题：

一是要计算 DRE，我们就必须知道每一个缺陷是在哪个阶段被发现的（这个属性一般情况都会记录）以及该缺陷是在哪个阶段注入的（这个属性需要对每个缺陷进行分析，很容易被人忽略）。

二是只要一个项目全生命周期结束，才能够准确计算个阶段的 DRE。那么 DRE 有什么作用呢？（to be continued）

注意我们上面例子中的那个矩阵，根据最下面以行的数据，我们可以知道需求阶段引入的缺陷大该是 50%（20/40），设计阶段是 25%，编码阶段是 15%，测试阶段是 10%。根据相关研究，当过程趋于稳定时，这个比例也是趋于稳定的。那么当我们有了一个新的项目，我们就可以进行缺陷估算了。

第一步：是估算总的缺陷数，目前这方面的研究也有很多，比如说 COCOMO 模型中就有一个子模型，叫 COQUALMO，但是我们也不一定要用这种参数模型，最简单的模型：估计缺陷总数=估计产品规模*历史缺陷密度。假设这一步我们估计县项目有可能 100 个缺陷。

第二步：可以根据我们刚才提到的分布，估算出需求阶段注入的缺陷大概应该有 50 个（100*50%）。设计阶段亦然。

第三步：根据 DRE，我们可以估算清除活动发现的缺陷数，例如需求评审预期发现的缺陷数应该是 25（50x50%）个。

至此估算大致完成，当我们实际执行项目时，如果需求评审发现的缺陷数远远低于 25 个，则有两种可能，一是产品质量特别好，而是评审的效果变差了，这是我们需要其它的度量帮我们分析，但是至少我们知道这里有问题了。反之亦然，若发现的需求数高于 50，则可是评审效果提高或产品质量特别差（一般是后者）。特别的，若发现的缺陷数远大于 100，则说明我们原是估算出现了问题（一般是产品开发出现严重问题），我们可能需要重新估算。

8 思想碰撞，故事中有深意

★ 你该走那条CMMI之路？

Q (step1v):

有如下一个 CASE:

“有一群小朋友在玩耍，而那个地方有两条铁轨，一条还在使用，一条已经停用，只有一个小朋友选择在停用的铁轨上玩，而其它的小朋友全都在仍在使用的铁轨上玩。很不巧的，火车来了，小朋友们已经来不及躲闪，而你正站在铁轨的切换器旁，因此你能让火车转往停用的铁轨，这样的话你就可以救了大多数的小朋友，但是那名在停用铁轨上的小朋友将被牺牲，你会怎么办？”

好了，我们一起从 SPI 过程中涉及的不同角色、角度、方法等方面来探讨，到底我们走那一条路是正确的？

可以从以下多个方面来考量：

1、这个 CASE 本身，是不是还存在其他的可能，为什么旧铁轨没用了？作废了？还是旧铁轨更危险？（当然，这个不同的人有不同的看法）

2、抛开 CASE 本身，联想企业实施 CMMI 过程中的一些策略与选择，是不是我们只站在自己的角度考虑问题？有没有站在市场的角度？或者 BOSS 的角度？（当然，可以有很多

的角度与策略)

3、其他的。(这个大家随便怎么说都可以, 只要有道理, 并说明理由)

我的本意是:

1、直接看这个 CASE, 给出一个选择, 并说明理由

2、除了直接看待这个 CASE 外, 结合企业在 SPI 过程中不同的选择, 以及理由。

大家可以从这两个方面来探讨!

A (z dq2601):

告诉我们做事要从实际出发, 更具具体的利益和目标而定具体方针和策略, 不要墨守成规, 中规中矩;

从决策人的角度考虑, 任何决策都有危险性, 决策要谨慎。

从小孩的角度, 必要离群, 做事要依靠群体, 团队精神。

A (steplv):

如果我们把这个 CASE 当一个项目来讨论, 那我可以先起个头, 大家接着来一起探讨。

一、做为一个项目, 我们可以从以下几个方面来思考:

1、范围: 我们要知道这个 CASE 具体包括那些东西, 也即, 我们要把需求弄清楚。

2、时间: 是不是有足够的时间, 可以让火车停下来, 或者把小孩抱离铁轨?

3、成本: 如果选择了不同的轨道, 我们要评估损失有多少, 一个生命? 三个生命? 抑或是火车上的(人、物)?

4、质量: 火车本身的质量如果存在缺陷, 它会不会熄火? 铁轨是否都完好, 有没有脱轨的问题?

5、风险: 我们遇到的风险是: 生命没了? 物品损失了? 还是重大机密产品报废了? 等等。我们得评估风险, 并制订相应策略, 是规避风险、解决风险、拒绝风险还是接受风险?

6、人员: 在这个 CASE 中会出现那些干系人? 为什么会出现这些情况, 干系人的职责是否明确?

7、沟通: 我们的沟通是否出现了问题, 从列车长、搬道工、以及铁路沿线管理人员、小孩、小孩父母等来分析。

8、采购: 出现了这个问题, 是不是我们得增加一些设备, 比如: 防护栏? 警示灯? 还是?

9、整体: 看来我们的整理管理与沟通等方面, 出现了严重的问题, 为什么小孩会突然出现在铁轨上? 为什么没有防护栏? 为什么没有警报? 等等。

二、那么, 如果从 CMMI 实施、应用的角度来看, 是不是也可以去分解?

1、企业的商业目标

2、企业的战略

3、企业的文化

4、企业的人员

5、企业的组织制度与体制

6、…………等等

A (sungubbi):

我第一反应：选择切换轨道，救那几个孩子，而放弃另外那个
然后又有些痛心：我在“眼前的利益”和“牺牲规则”中选择了“眼前的利益”而纵容不守规则。这也恰恰是我们在做过程改进工作中，各项目组最容易犯的错。

又有些思索：如果增加警示牌，是否能够避免这个悲剧？（标识和风险提示）

废弃的铁轨通向何方？谁知道？改道后会发生情况有人能够预知吗？操作轨道的工人知道吗？——企业中谁是这个角色，QA 吗？还是管理者？

怎么样才能够把损失减到最小？是否可以急停？可否救出小孩？——紧急措施

除了警示牌，还有什么样的预防措施？

A (step365he):

我看了这个案例，觉得做出的任何选择都是错误。

因此我们要回到案例的开头，先问自己有没有足够的勇气面对并承担自己的决策（以及执行）带来的后果？

如果这个答案是肯定的，才能从这个案例中汲取养分——组织需要系统论的思考（做出的决策，或称为战略才能经受考验），组织需要有良好的控制（对执行的过程，或称为战术），必须有及时可信的反馈机制对出现的偏差进行修正。

ps1. 假设决策出现问题，这是一个话题。

假设决策没问题，而过程出现问题，这个案例还提出一点——“相关人士”会“不由自主”的根据“自身利益”做出相应的行为或者选择。

ps2. 这个案例另一个层面的解读是在遵守规范和打破条框之间的选择

ps3. 个人观点，仅供参考

A (step1v):

To step365he: 你提到“有没有足够的勇气面对并承担自己的决策（以及执行）带来的后果？如果这个答案是肯定的，才能从这个案例中汲取养分——组织需要系统论的思考（做出的决策，或称为战略才能经受考验），组织需要有良好的控制（对执行的过程，或称为战术），必须有及时可信的反馈机制对出现的偏差进行修正。。。。。。ps2. 这个案例另一个层面的解读是在遵守规范和打破条框之间的选择”，同意你的观点，这个主题中，除了暴露出体制的问题外，更多的倾向于人的问题，是不是在一定的岗位选择了合适的人？是不是让合适的人出现在关键的岗位？

走回到现实，目前绝大多数企业对于 QA 岗位的重视程度，就完全可以从 TA 们要那些人承担这一岗位而看出。是不是让那些所谓的项目助理、文档管理员、秘书、或者其他不受重用的人去承担？抑或者让那些刚刚走出校门的大学生去承担？为什么会选这些人，因为这些人薪水不高，公司养的起闲人！可否取消这些岗位？不行，因为 ISO 9000 中明确要求要有质量保障部，不然，外审就不通过，所以，这些人是有用的。（注：这句话是错误的，后来经林老师一说，才知道错了，本来想删除了，但为了保持后续疑问及解答的源头，所以我保留了下来，对大家造成的误解，请见谅！）好矛盾呀！处于用与不用之间，搁在中间的感觉比较不爽。

A (tyrone):

窃以为，如果是选择了 2 或者是做了其它那么多的考虑，其实就与这个主题的意旨不符了。

个人的看法是，如果以此来比喻该选择什么样的 CMMI 道路，则根本不必去理会有多少小孩子在使用中的铁轨(经过设计、测试、可行的一条道路，比喻 CMMI 给大家的方向与实践建议)上玩耍，因为，你可以这么说，那列火车就是你的企业，那些在使用中之铁轨上

玩耍的小孩，你可以看成是，反对你的企业推动 CMMI 的障碍。对于这个企业而言，要走好正轨，火车的司机员(Czar 与 EPG Leader)，应该是鸣汽笛警告(宣达公司的奖惩与绩效考核制度)所有的小孩(那些障碍)赶快远离火车所行驶的铁轨(车来了，远离行驶中的轨道与公司一同奋斗，挡在前面碍事就等着被消灭吧!!)。这个时候，如果负责转侧器的工人(可以比喻为管理咨询师、或者是公司的领导)把火车导向旧轨道(比喻现行让多数从业人员受到很多痛苦的旧的道路)，或者是命令火车停下来，都是不对的，因为小孩子要是知道在铁轨上玩耍火车会停下来，而且不会有人采取警告、驱离与惩罚的行动时，他们更会大刺刺地强占住铁轨，让铁路运输做不下去，要不然就是走向原来有问题老路上了(因为走下去死伤惨重，会失去那些“看似很重要的”人力资源(但其实可能是鸡肋)，所以让车子走上原来的旧轨道，牺牲掉守规矩，与公司有共愿景、原来就愿与公司一同奋斗的人员(由在旧轨道上玩耍的小孩所譬喻))。许多企业推动 CMMI 失败，不就是因为不知道如何面对抗拒变革(RTC)，而终使过程改进毫无所获！

我个人以为公司成立 QA 部门根本就是管理阶层在推卸其对于公司质量的责任。质量是公司里每个分子的职责，公司的领导，负起对于公司每位成员的质量教育、培训与监督，使得每个人都重视自己的工作质量，那么就可以将公司产品与服务的质量做到最好。公司并不需要成立质量部门。公司成立质量部门只是让公司高层与领导，误认为有了质量部门、公司的产品与服务就自然而然有质量了，而他们对于质量的责任也了结了，所以一个真正具有高质量的公司是不需要成立质量部门的。当然这并不意味着，不成立质量部门的公司，其质量一定很高。主要还是要看公司在质量保证活动上的成熟度，他们的质量保证活动是自发性的，还是因为有外力的压迫。如果是因为有外力才执行质量活动，显然一定做得不好，因为公司的各个阶层与质量保证活动的执行者总会想要偷斤减两的。

ISO 9001 并没有要求一定要成立 QA 部门，至少我之前服务的单位导入 ISO 9001 时就没有成立 QA 部门，而我服务的单位就是政府部门的信息中心。每次我们要做质保或审计的时候，是将散在各组中，接受过正规 ISO 审计员课程并且认证合格的人员集合起来，在管理代表的授意下实施审计(把握工作的要领：审计员要回避对与自己工作或利益有关之业务的审计工作)，我常常参加单位里质量审计活动(因为具有 ISO 审计员的资格)。而我们只有在单位的管理组中，有一位负责质量活动的参谋人员，年度的质量活动均由该员依计划发起、编组与执行，乃至最终的检讨改进。

我在执行机构的软件委外(Software Acquisition)时，在我们信息中心里成立了一个项目管理小组，而我就是这个项目管理小组的领导，这个项目管理小组负责委外软件需求的获取、发展、然后外包，并执行外包管理的工作，从对供货商的选择、所有联合评审、审计、验证确认、及至于最终的验收工作，均由这个项目管理小组成员共同执行，也没有什么 QA 人员在我们这个管理小组当中工作。所有 QA(质量保证)应执行的工作，都是由我们这个小组的成员共同执行，我们执行 QA 活动的时候，我们都是 QA 人员。

以前在某政府机构各阶层工作时，年度里都会有上层或上几个层级的业务主管单位人员组成督导小组，来看看我们工作执行的情况如何，是不是符合工作程序的要求、工作产出是否符合规定，有没有造假不实之处，这些也都是审计的工作，但重点是，这些人可都不是什么 QA 部门的人员喔，因为机构里根本没有 QA 部门(政风单位或督察部门并不算，因为他们的工作内容与 QA 活动并不相同)。所以一个公司需要有 QA 的活动，却不见得一定会有 QA 部门。

ISO 9000: 2005 质量管理体系—基本原理与词汇(ISO 9000: 2005 Quality management systems -- Fundamentals and vocabulary)，主要在说明质量管理体系的基本观念与词汇术语释义。在基本观念理强调的是，管理者的职责、过程导向、文件化、达成质量管理目的所需要的资源及这些信息的提供等等要点，在本文件中亦未明确提到导入 ISO 质量管理体系的公

司要成立所谓「质量保证部门」。如果 ISO 需要律定这样的要求，也会在「ISO 9001 质量管理体系—要求(ISO 9001: 2000 Quality management systems – Requirements)」这份文件中律定。但是实际上，均没有这样的规定。主要原因就在于，质量管理体系本身就是过程导向的标准，ISO 认为组织要达成质量的目标，满足客户的需求，必须透过过程的实作来达成。至于达成质量目标、满足客户需求的过程，是否需要成立一个专责单位来执行，这项决策 ISO 留给导入质量管理体系的公司自行决定。因为，成立一个专责单位去执行一个不见得是属于工作周期频繁，几乎都是在消耗成本，而且如果公司制度走上轨道、所有成员都建立起深刻的质量观念，而且身体力行时，不见得真正发挥效益的单位，对于任何一个公司来说，都是很大的成本开销的，所以，ISO 并不会强制去要求导入的公司一定要成立质量保证部门，实质上，就算这么规定，也会引起很大的争议的。



第二部分：专家文章

1 过程改进（CMMI）

★ 青出于蓝-在模仿和超越之间铸就伟大

作者：何丹 博士

一、追随大师的智慧而不是足迹— CMMI 的模仿与超越

“吾尝终日而思矣，不如须臾之所学也；吾尝跂而望矣，不如登高之博见也。登高而招，臂非加长也，而见者远；顺风而呼，声非加疾也，而闻者彰。假舆马者，非利足也，而致千里；假舟楫者，非能水也，而绝江河，君子生非异也，善假于物也。”

一遍遍品味着荀子的《劝学》，我更体会到了为什么 CMMI 模型越来越得到广泛的认可和应用，而且那么多世界第一流的公司即便在拥有的自己成熟的体系后，仍然在使用 CMMI 来进一步提升自己的流程能力。因为 CMMI 是经过 SEI 收集和整理的业界优秀公司的过程改进最佳实践的大全，借鉴别人的经验，可以大幅度缩短我们走向成熟的时间。

但为什么也有不少公司并没有从 CMMI 改进中获得实际利益呢？我问着自己。CMMI 能带给我们的仅仅是别人的一些优秀经验吗？如果是这样，那么教育界的一些大师为什么在不断反省着：“教育不应只是拿昨天的教材，教今天的学生，去面对明日的世界。”教育更是要让学习者学习“学习的方法”和“思考的方法”。CMMI 同样是昨天业界优秀实践的总结，我们把这些 CMMI 实践去教给今天的管理者和员工，那么他们能够应对日新月异的纷繁世界吗？

想想我们的近邻韩国和日本，韩国提出“东道西器”，以本国文化为本，以西方管理为辅，从而韩国的工业化和高科技产业实力迅速提升，典型的代表就是三星。而日本有史以来的崛起都是在模仿中国和西方的文化和精神，但又有一定的超越或不同。世界上美好的东西，决不是完全的复制品。一位优秀的画家即便在临摹伟大作品时都一定有自己的灵活性和独创性，否则就变成匠人，而是既在模仿之中，又在模仿之外。伟大似乎总喜欢诞生于模仿和超越之间。没有扬弃、没有打破、没有超越，我们永远就只是奴隶，只是仆人，就只能成为前人的依附。没有一位伟大的导师希望将我们制造成傀儡。同样，我想那些创建 CMMI 模型的大师们（如 CMM 之父 Watts Humphrey）一定不希望把我们复制成罐头人才、罐头团队、罐头公司。

如何对 CMMI 进行模仿，又如何超越？我不断反思和总结着，在漫漫咨询和评估之路上下求索着……

二、指月之手—顿悟要靠自己

2.1 指月之手—顿悟出 CMMI 改进的灵气

释迦牟尼佛曾经在楞严经中做过一个有趣的比喻：“如人以手，指月示人。彼人因指，当应看月。若复观指以为月体，此人岂唯亡失月轮，亦亡其指。何以故。以所标指为明月故。岂唯亡指。亦复不识明之与暗。何以故。即以指体，为月明性。”

大意是说：如果把真理比作天空的一轮明月，很多人不知为什么对明月视而不见，因此佛陀慈悲地用手指给大家看，但很多人却把佛陀的手指当成真明月本身，这样手指和月亮的价值都失去了。

根据佛学的观点，真理是不立文字的（即无法用文字描述）。因此虽然佛陀讲经 49 年，但在离开这个世间前告诉须菩提和其他弟子说自己一部经书都没有讲：“若人言如来有所说法，即为谤佛，不能解我所说故。须菩提，说法者，无法可说，是名说法。”因为所有的经文都是指月的手。

CMMI 何尝不是如此，CMMI 实际上是很多公司过程改进走向成功路标的汇总和大全，而并非过程改进真理本身，只有通过不断的学习和思考，透过 CMMI 的各个实践看到其本质，才能模型背后大师们的思想和灵感相通，才能把干巴巴的 CMMI 变得含情脉脉，把机械的流程变得生机勃勃。

当年林语堂曾经在《论趣》中提到徐志摩从美国 Clark 大学转学到英国的剑桥大学后，才真正成为通达的学者。林语堂谈到：

青年人读书，最难是动了灵机，能够知趣。灵机一动，读书之趣就来了。无奈我们这种受考试取分数的机械教育，不容易启发一个人的灵机。我曾经问志摩，“你在美国念什么书？”他说在 Clark 大学念心理学。就是按钟点，摇铃上课下课，念了什么书！后来到剑桥，书才读通了。”剑桥的教育是导师制，导师一礼拜请你到他家谈学问。就是靠一支烟斗，一直向你冒烟，冒到你的灵魂冒出火来。与君一席话，胜读十年书。灵犀一点通，就可以了。禅师有时敲一下你的头，你深思一下，就顿然妙悟。现代的机械教育，总不肯让人学思并重，不肯叫人举一反三，所以永远教不出什么来。

不止读书要读出那份灵气，而且过程改进也要悟出那份灵气。这样我们才能真正做流程的主人，才能真正按照业务目标裁减和优化流程。虽然 CMMI 以及过程改进人人可学，但无人能教，因为真正的顿悟要靠自己。

2.2 削足适履还是量体裁衣——僵化实施 CMMI 的代价

2.2.1 裁缝列维的故事

全球著名咨询师——杰拉尔德·温伯格在给客户咨询时曾经讲过一个有趣的故事[1]：

从前有一天，一个人找到裁缝列维，因为他听说列维能做出便宜的民族服饰。衣服做好了，这个人试了试，发现衣服根本不合身，“看”，他说，“上衣后摆太长了”“上衣后摆太长了。”没问题，列维回答，并指导他怎样弓起后背以把上衣松弛的部分挺起来。

“那右边袖子怎么办？长了三英寸。”

“没问题，”列维重复道，并给他示范怎样斜着身子伸长右臂以使得袖子合适。

“那裤子怎么办？左腿太短了”。“没问题，”列维第三次说道，并上前教他怎样提起大腿，这样尽管走路跛得厉害，可是衣服看起来合身了。

没有太多的抱怨，这个人蹒跚着走上街头，有点儿被列维欺骗的感觉。有个陌生人叫住他问：“打扰一下，我想知道，你穿的是新衣服吗？”

这个人很高兴有人注意他的新衣服，“是的，为什么这么问？”

“哎，我正准备做套新衣服。谁给你做的？”

“列维，——沿着这条街一直走。”

“好的，非常感谢。我真应该找列维为我做衣服。”

“为什么？”

“能为你这样的瘸子做出这么合适的衣服，更别说我了，他可真是天才。”陌生人说着就匆匆离去了。

在有些公司执行 CMMI 的过程中，也是采用了列维的方式按照 CMMI 来定制衣服，而不按照公司的“身材”（即业务模式）量身订做。

2.2.2 僵化实施 CMMI 的痛楚

想起自己在 SEI 参加培训时，SEI 的学者一再强调说：对 CMMI 主任评估师和咨询顾问最大的挑战不是 CMMI 模型，而是对客户业务运作模型的深刻理解，然后去灵活地解释 CMMI，才能给客户带来真正的价值。然而我看到一些公司却是以相反的方式在实施 CMMI。

一些公司邀请我去做 CMMI 预评估和正式评估，进入这些公司后，发现公司高层虽然不懂 CMMI，但对过程改进相当重视，因此聘请了一些 CMMI 专家来指导公司的改进工作，定义流程、在项目中推广和实施。为了表明对 CMMI 的重视，高层还要求公司各个方面的专家在具体开展各项工作前，要得到 CMMI 专家的认可，以确保满足 CMMI 的要求。比如公司需求分析师在做需求分析前必须向 CMMI 专家咨询 CMMI 所要求的需求分析方法，测试人员进行测试前也必须向 CMMI 专家咨询 CMMI 所要求的测试方法。似乎 CMMI 专家是个全才。既然有老总的大力支持，这些 CMMI 专家也就当仁不让做起了老师，将 CMMI 过程域（Process Area - CMMI 过程域）中的各个实践（SP1.1, GP1.1... ..）逐字逐句翻译成中文给相关人员讲解一通，然后，那些需求分析师和测试人员等相关人员带着似懂非懂的神情离去，心里自我批评着：也许自己的理论知识不够，所以一时难以理解，但只要照着老师的讲解做就是了。

再去看看公司的培训计划，几乎全部是 CMMI 各个过程领域相关的流程培训，没有公司业务相关的培训、没有技术和工程相关的培训、没有管理理念和软技巧的相关培训，绝对是又红又专的 CMMI 公司。我不禁困惑地问道：“你们公司是做 CMMI 工具的吗？”

“不是的？”

“那么为什么所有工作都是围绕着所谓的 CMMI 展开，过程改进的目的到底是为了 CMMIX 级别还是公司的业务发展？”

“我们的 CMMI 专家说必须这样做才能满足 CMMI 的要求。”

“那你怎么不问问所做的工作是否给公司、给客户带来真正的价值？是否给员工带来能力的极大提升呢？这难道就是 CMMIX 级的精髓吗？即便这样拿到

CMMI X 级的证书，那么又会给公司留下哪些持续改进的文化基因呢？”我忍不住再次问到，知道不会得到答案。

我所知道的最优秀的人做任何事情（What to do）之前都要问问自己“为什么？—

WHY?”而不是不加思考地做着别人要求的事情(“What to do?”), 牛顿在苹果落到头上时, 问苹果为什么苹果不往上飞, 而是往下落, 最后发现了万有引力定律。我们却通常会把它吃掉, 我们无法成为牛顿, 而只能成为多吃一个苹果的人, 原因就在于我们没有思考 WHY 的习惯。

也许《把信送给加西亚》这本畅销书在某种程度上对开发人员是一种毒害, 正是由于从来不问为什么, 才造成了需求不清晰, 造成了开发中大量的返工和成本失控。

僵化实施的 CMMI 是可怕的, 使 CMMI 变得如此枯燥和乏味根源是什么, 也许正是因为我们仅仅懂得 CMMI 皮毛, 而没有深入理解 CMMI 背后那些大师们的鲜活的思想, 因此只能削足适履而痛苦地改进。

三、成为高尚的医生而不是卖药人

3.1 专家的锤子法则

温伯格在《咨询的奥秘》中对锤子法则(The Law of The Hammer)是这样定义的[1]:

在圣诞节收到锤子做礼物的孩子会发现每样东西都需要敲打。

只有这一件工具的专家也许会把所有的东西都当成螺丝钉, 从而不停敲打。

那么 CMMI 难道不是工具吗? CMMI 的过程改进是采用 BOTTOM-UP 的改进模式, 就好像一个收集了很多成功公司的工具形成的工具箱, 25 个过程域相当于 25 个具体的工具, 如: 需求管理“剪刀”、配置管理“钳子”等等。这些工具有其适用范围, 并不能解决各公司所有的问题。

但随着 CMMI 的逐步流行, 很多公司就一拥而上, 都开始用 CMMI 的进行过程改进, 把它当成了取之不尽百宝囊或灵丹妙药, 不管企业存在什么样的问题, 通通用 CMMI 解决。不管是什么企业, 不管是什么问题, 都从 CMMI 的 25 个过程域的相应流程定义开始起步, 需求管理流程、项目计划流程… … 就像从工具箱中按顺序逐一拿出相应工具, 而不关注到底要解决什么问题。真不知道用这些工具最后会把企业修理成什么样子。

3.2 从问题域出发— 成为流程的医生、企业的医生

对于我们任何一个学过或做过系统或软件开发的人员, 都曾经有老师和同仁告诉我们系统或软件的开发要从问题(Problem)域开始, 找出客户或产品真正需要解决的问题, 然后再寻找和制定解决方案(Solution)。而我们为什么在做过程改进的时候恰恰忘记了这一点, 先从 CMMI 箱中(Solution 域)找出工具, 而最终不知道到底解决的是什么问题。

通常, 我的客户会毫无保留地把企业最真实的一面展现给我, 希望我能更多地帮助他们挖掘出企业中深层次的问题, 并帮助他们提供一些解决的思路和方法, 每到此时, 我能深刻感受到双肩上那沉甸甸的分量, 这来自客户的信任也来自我对自己新的挑战。我知道客户把我当成了企业的医生而不是 CMMI 卖药人, 把我当成了通才而不是工具的专家。在大多情况下, 我发现企业的主要问题还不仅仅是 CMMI 能够解决的, 有的公司企业文化和价值观完全不被员工接受从而士气低落、有些公司考评体系不公平、有些公司知识管理和共享存在不少问题、有些公司招聘和提拔标准很不清晰、有些公司在客户关系和客户满意度管理方面存在极大的问题… … 在这些情况下, 在制定公司过程改进策略时, 也许应该先改进文化和价值观、先改进考评体系、先改进知识管理和学习型文化建设、先改进招聘和提拔标准、先建立客户关系管理和满意的调研流程。也许应该采用的改进工具是 6SIGMA 中的 DMAIC、

关注企业员工文化和能力的 People CMM、或关注精细化管理的 LEAN 方法，虽然这些内容都不是 CMMI 的范畴，但这样改进也许才会对企业和员工有最大的裨益。

改进是为了解决公司今天的问题，去增强明天的企业竞争力，而不是为了定义一些目前对公司无关紧要的流程，去通过 CMMI X 级的认证。生命的厚重体现在能够给他人和团队带来希望、带来价值，而不是祈求一个外部的认证去给自己脸上贴金去证明内在的贫乏。

然而，成为通才和医生对公司过程改进领导人或咨询师提出了很大的挑战，因为如果一个人仅在专业上做太久，他的视野会变得比较狭隘，误以为自己那个小小的角落已经代表了整个世界。因此需要尽快补充自己的知识和经验，但即便几方面专业能力都有所增强，可能只能变成几个方面的专才。正如管理大师德鲁克所说：一个人不会因为多增加几项专业能力，就变成通才，通才和医生需要视企业为整体，需要无私和开阔的视野去根据公司业务方向和发展方向去寻找公司过程改进的灯塔，否则很容易导致公司的过程改进触礁，无法引领公司在茫茫的过程改进大海中朝着业务目标航行。

四、站在巨人的肩膀— 过程改进三部曲

4.1 高山仰止— 改进的真谛

管理大师德鲁克说[2]：二十世纪中制造业的生产率提高了 50 倍之多，而我们最需要感谢的人就是泰勒（Winslor Taylor），在他之前，别人也研究了很多方法，但都没有使生产率获得明显提高。而正是他的创意，使得我们把艺术的劳动工程化，从而成就了这次飞跃。其实他的系统化改进方法非常简单，只有以下三个步骤：

(1) 分解—科学地将工作分解成各个活动；

(2) 改进—专注各个局部动作或局部工作的改进，寻找业界最佳实践，从而可以大幅度提高绩效；

(3) 整合—对所有动作进行系统化地设计和整合。

让我们反思工业化整个进程，完全是按照这三个步骤来进行的。比如以前是手工的汽车作坊，产量不高、质量不稳定，完全是依赖于工人师父个体的技能，而且工人要经过很多年的学习才能掌握汽车各个环节的手工生产。而我们按照泰勒的第一步，首先把工作分解成各个零件的加工制作、组装等各项工作；然后按照第二步，各个工人只分配其中的一小部分的工作，重复性地一次次去做，有专注才更容易成为专家，才更容易积累经验和相互学习，如果有必要还可以针对这个环节形成自动化生产来提高效率；最后第三步，设计和工艺工程师根据不同产品的要求系统化地设计出产品的流水线。

软件行业虽然发展比制造行业晚了很多，但也在按照这三步曲来改进着：以前是软件作坊，后来逐渐有了需求分析员、架构师、编码人员、测试人员，共同协作完成一个大型的软件产品。

但泰勒的方法我们真的理解了吗？还有进一步的改进空间吗？让我们再次详细分析各步骤的内涵。

1. 分解—使得我们把一个系统的工作流程分解后，形成简单的一个个环节和步骤，使得问题简化，便于进一步突破；

2. 改进—对各个环节和步骤分而治之的改进模式有很多的好处：

- 1) 专注— 每个人的精力是有限的, 只有首先专注某个环节和步骤的研究, 并积极主动地和外界交流学习, 寻找这个方面的最佳实践, 才能迅速成为专家;
 - 2) 积累和重用— 很容易让相关人员在某个具体环节形成知识交流和积累, 积累可重用组件, 或开发出自动化系统来代替这个环节和步骤;
 - 3) 80—20 原理的改进— 从系统的角度识别出对系统影响最重要的 20% 的环节和步骤, 重点突破来进行改进。
3. 整合— 由极少数的对全流程和系统完全了解的通才根据客户的需求、市场需求以及项目特点, 艺术性的选择和整合各个环节和步骤, 从而将开发或生产流程定制化成最高效的体系, 把项目的优势发挥到最佳。

很多人怀疑泰勒的这种模式不适合现在的个性化开发和生产, 只使用于大批量的生产模式, 其实这是对第三步的误解。德鲁克解释说, 我们常用的英语似乎是很复杂的, 那么用这三个步骤, 我们可以把英文分解成简单的 26 个字母, 而 26 个字母的反复重用和组合, 又形成一个个单词、美丽的文章和诗篇。因此, 这个时代实际上是对分解和整合的大局观人才的要求更强了, 这两个步骤都需要优秀的通才。

4.2 大芬村的油画和麦当劳的汉堡包— 泰勒改进思想的精彩展现

一位 CEO 参观过大芬村 (中国油画村) 后对我说: 这次参观给他带来很多启迪, 原来纯粹的艺术都可以工程化、产业化, 那么软件业应该更容易做到, 大芬村是真正读懂了泰勒的思想。

他说, 在大芬村, 你只要给他们一张经典油画的图片, 他们很快就可以用很便宜的价格 (200 元) 给你临摹出一张较高质量的油画。因为大芬村有通才和专才, 通才类似于我们软件开发中的项目经理和需求分析师, 他拿到照片后, 首先分析照片中所含有的基本绘画元素, 如花朵、人脸、山峦等。然后他设计出绘画的步骤, 再找那些专门画花朵、人脸、山峦等专才去组成绘画流水线, 这样一副比较精美的油画就这样完成了。也许, 它比不上梵高和达芬奇的原作, 但对于我们普通人来说已经足够了, 而且是在我们可以承受的价格范围内。非常有意思的是这种通才和专才组合的绘画模式, 通才的作用体现在过程改进三步曲中第一步和第三步, 也就是对一幅画的元素分解以及整合方式的设计。而专才只要关注第二步, 就是具体绘制出自己最擅长的那一部分, 也许是脸部表情、也许是红花绿叶。所以通才和专才都非常明确自己的职责以及能力的要求, 通才并不需要很高的绘画能力, 但一定要是一个杰出的鉴赏家和管理者, 而专才也并不需要经过中央美术学院经过写生、临摹、考试等多年的浸淫, 因为非常专注, 所以容易成功。而顾客也高兴地免去了通常油画中所附加的培训费 (通常艺术家多年积累和培养所需要的成本其实都包含在价格中)。

大芬村的组织形势与一个乐队的组成何其相似, 乐队的指挥是通才, 对音乐有深刻的理解, 但可能并不是优秀的小提琴家或大提琴家。而乐队的成员则是专才, 只深刻地理解自己的乐器以及演奏方式, 对其它乐器可能仅仅略知皮毛。同样, 速食业巨头麦当劳的成功也是把食品加工的艺术工程化, 分解和重新设计了全流程, 因此取得了巨大的成功[3]。

汉堡包其实很早就在美国境内出现, 特别是二战以后, 在各个大城市的街道拐角处, 汉堡包摊犹如雨后春笋大量涌现。但汉堡包的质量完全依赖于厨师的艺术水平。但是, 不同的是麦当劳将改进思想运用到了昔日那种毫无规划、夫妻老婆店的经营模式里。重新设计了产品制作的整个过程; 接着, 又设计和改进了操作工具, 使每一块肉、每一片洋葱、每一个圆

面包、每一根炸薯条在精确定适合完全自动化的程序中被生产出来，大小完全一模一样。最后，麦当劳着手研究客户真正最关心的因素，如：产品的质量、产品的可预料性、服务速度、绝对整洁以及亲切友好，并针对这些实际要求，制定出相应的标准，按照标准进行员工培训，同时将员工的工资与这些标准挂钩。从而，一个速食业的巨无霸出现了。

4.3 Process Pattern 和 CMMI

前些年，软件的 Design Pattern（设计模式）一经推出，就得到软件业界的一致好评。其最大的成果就是针对某些问题，可以马上采用一些成熟的优秀设计思想，从而形成知识的重用，大大提高了生产率。

而反观泰勒的思想，给软件开发和系统开发以及公司运作体系提供了那些启示呢？我们是否可以从公司运作方式、从软件和系统开发的特点出发，把这个庞大的体系分解成多个环节和步骤，在每个环节和步骤上总结出 Process Pattern，以促成过程 and 知识最大限度重用呢？

CMMI 正是这种 Process Pattern 思想的展现，它把组织运作和开发体系分解成需求管理、项目计划、项目跟踪和监控、测试等 25 个环节（25 个过程域），然后在各个环节上去寻找业界的优秀 Process Pattern，从而最终形成 CMMI 这个模型。

而各个公司业务特点不同，不一定非要分解成这 25 个环节。也许还有人力招聘、文化建设等更多的环节；也许一些 CMMI 要求的、但对公司目前不是很重要的环节可以合并或删除。然后通过项目结项总结、部门定期总结等方式，针对各个环节整理和积累出真正属于自己的、更具体的 Process Pattern（比如适用于组织的估算方法、经验教训以及样例库、重用组件库等）。同时，通过建立各种学习和培训机制（如兴趣小组、知识论坛、个人培训计划等）把积累的 Pattern 尽快传递到每个人，使员工能力尽快提升。

制造业中各公司的实力对比一定要看固定资产的多少，而软件和系统开发公司都是高科技知识型公司，人才以及人才所具有的能力和知识才最重要的资产。众所周知，知识的最大特点是第一次产生的成本很高，而整理后知识重用（COPY— PASTE）的成本非常小。因此知识型公司更迫切需要对 Process Pattern 的积极地总结和积累，然后，更有艺术性和创造性地整合和裁减这些 Pattern 去适应公司的发展和项目的特点，使公司和项目的利益最大化。那么制造行业可以提高 50 倍的生产率，在高科技行业里的生产率一定可以提升得更多，也许是 500 倍乃至 5000 倍，因为人的能力的可拓展性远远超过了制造行业的机器。

4.4 系统思考： Why - What - How 的过程改进模式

CMMI 告诉我们在哪些方面可以进行改进（即 What to do），但没有告诉我们 Why（公司的业务目标），也没有告诉我们 How（具体改进的某个方面的手段和方法），因为各个公司的特点各不相同，因此 CMMI 模型的设计者无法穷举出这些目标和特点，只能提供给我们其他优秀公司的 What。但对任何一个公司要进行改进时，最佳的方式就是把公司当成一个整体，通过系统思考，识别出改进的目标(Why)，按照 80—20 原理确定出最需要改进的环节（What），然后确定各个环节的最佳改进方案（How）。

系统思考是管理大师彼得·圣吉在《第五项修炼》[4] 首次明确提出的，刚一提出，就引起世界上有识之士的广泛共鸣和反思。因为很多公司在做改进的时候都是采用西医头痛医头、脚痛医脚，有些暂时的效果，但不明显。我们是否忘记老祖宗教给我们的天人合一的价值观并失去了系统思考能力，中医学的传统技术是否正在被逐步流失。

而要真正采用泰勒的改进思想，必须具备的是系统的大局观，并明确出系统改进的目标，

然后再识别系统中对目标的达成最有价值的环节进行持续改进。如果缺乏系统目标的设定以及系统的大局观，改进中即便付出了巨大的努力，也无法使系统形成强大的激光，而只能是散射的灯泡。

因此，我们如果希望改进能够给公司和项目带来最大的价值，就要按照 why - What - How 的方式进行改进。

1. Why:

- a) 确定改进的业务目标（如客户满意度提升、利润率提高等）；
- b) 绘制出系统改进蓝图，将系统分解成各个相互关联的环节；

2. What:

- a) 从业务目标出发，并针对公司的现状，对各个环节进行排序，从而识别出最需要改进的重点环节（比如最需要改进的是需求分析和测试环节）；
- b) 再将这些重点环节整合，形成过程改进的整体规划或计划来有序地开展改进工作。

3. How:

- a) 有些重点环节的改进可以参考一些业界优秀的模型（如 CMMI，P-CMM）。
- b) 同时在公司内部或外部寻找相应的专才在重点环节进行研究、整理出更适合公司的具体 PATTERN(方法论、重用组件、样例、经验教训等)
- c) 然后进行试点推广。

为了更清晰地解释 Why-What-How 方法，我们用下面一个小例子来说明：（参见图 1 所示的系统改进行军图，箭头为行军路线— 本图绘制时参考了文献[5]）：

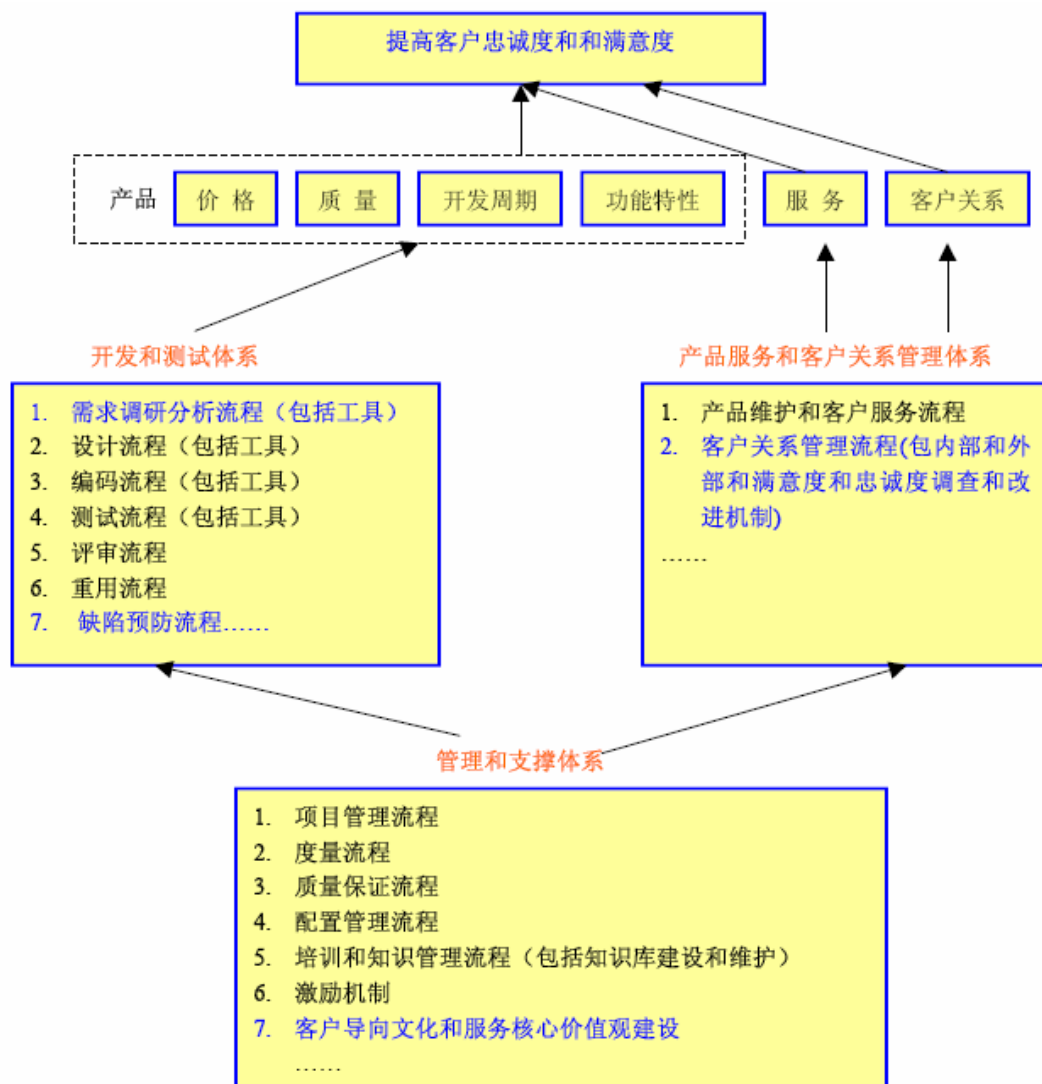


图 1. 系统改进行军图

（注：由于篇幅所限制，本图做了很多简化）

1. Why

a) 公司确定了今年的一个业务目标是大幅度提高客户满意度。

b) 为保证过程改进的目标和方向和公司业务目标的一致，我们绘制出图中所示的过程改进系统行军图，从图中可以了解客户满意度提高的全局视图。客户满意度的高低依赖于产品开发的 4 个要素（功能特性的多少、成本、质量以及开发的进度）、服务质量以及客户关系。而产品四个要素在很大程度上依赖于公司的“开发流程体系”的成熟度，而服务质量以及客户关系管理又严依赖于公司的“服务和客户关系管理体系”的成熟度。这两大体系要高效运作，就需要一个优秀的“管理和支撑体系”平台。

c) 在系统行军图中我们已经把三大体系都分解成各个流程和工具等环节

（注：其实应该含有更多环节，这里做了很多简化工作）

2. What

d) 从业务目标和公司的现状我们识别出最需要改进的环节为

- i. 需求调研和分析流程（包括工具方法等）
- ii. 缺陷预防流程（包括缺陷预防库的建设）
- iii. 客户关系管理流程(包内部和外部和满意度和忠诚度调查和改进机制)
- iv. 客户导向文化和服务核心价值观建设

e) 根据这些重点环节，并根据公司目前的资源情况，我们制定了过程改进的路标和规划（比如：第一季度关注需求相关的改进，第二季度关注缺陷预防的改进……）

3. How

f) 分别成立 4 个专家小组对 4 个方面进行研究，总结公司的经验教训，并寻找外界的最佳实践，而需求和缺陷预防的改进我们可以参考 CMMI 模型。

g) 如果我们在哪方面没有真正的专家（比如：客户关系管理方面），那么我们可以考虑参加外界的交流，并聘请外部咨询师来协助公司在这个环节上进行改进。

五、总结一怀着谦虚的赤子之心，做个拾贝的孩子

牛顿（Newton）是古典物理学的奠基人，当别人称赞他时，他谦虚地说自己不过是站在了巨人的肩膀上，因此看得更高更远。

基于 CMMI 的过程改进之所以在有些公司没有取得很大的成功是因为对 CMMI 教条化地执行，是因为没有深刻理解 CMMI 背后大师的思想，没有去积极主动地寻找巨人的肩膀，没有如饥似渴地汲取更多的知识和经验营养，去寻求顿悟。

真理也许是最简单的，其实大师泰勒几乎在一个世纪前就用非常简单的语言告诉了我们改进的真谛：

- (1) 整体出发，系统思考，把系统工作分解成各个环节；
- (2) 分解后在各个环节寻找最佳实践并持续改进；

(3) 像音乐大师灵活运用 8 个音符来谱写乐章，针对目标和问题把各个环节有机整合去达到目标并解决问题。

在我们谦虚地学习 CMMI 改进的时候，我们不要忘记 CMMI 不是真理本身，我们需要在模仿和超越之间铸就伟大。像当年爱因斯坦在一次演讲中提到的：牛顿先生，对不起，我把您的理论突破了。因此我们做任何事情前都可以问问 WHY，用 Why-What-How 的方式持续改进着、创造着、学习着、快乐着… … 改进是一条没有终点的路。耳边响起了大师牛顿的话：“我好像是在海边玩耍的孩子，时而拾到几片美丽的贝壳，并为之欢欣。那浩瀚的真理海洋仍展现在面前。”我想，爱因斯坦是另一个拾贝的孩子，而且拾到了更美的贝壳。让我们也加入拾贝的行列，怀着一颗谦虚的、快乐的赤子之心，把那些优秀 Process Pattern 串成珍贵的永恒。

“君子曰：学不可以已。青，取之于蓝，而青于蓝；冰，水为之，而寒于水… … ”再一次捧起荀子的《劝学》读着，似乎品味出更深刻的含意… …

参考文献:

- [1] 杰拉尔德.温伯格,《咨询的奥秘》—成功提出和获得建议的指南,2004 年 5 月—清华大学出版社。
- [2] 彼得.德鲁克,《21 世纪的管理与挑战》,2001 年 12 月,九州出版社。
- [3] 彼得.德鲁克,《创新与创新精神-Innovation and entrepreneurship》—2002 年 9 月,上海人民出版社、上海社会科学院出版社。
- [4] 彼得.圣吉,《第五项修炼》,2001 年 6 月,上海三联书店。
- [5] 宝利嘉顾问/著,《战略执行-平衡记分卡的设计和实现》,2003 年 8 月,中国社会科学出版社。



第三部分：会员原创

1 RD-需求开发

★CMMI思想结合RMC实践之——“需求开发”流程定制

作者：steplv （吕斌）

提起项目研发过程中的需求，人们的第一反应往往是：需求很难把握呀……这变化也太快了吧……很多的感慨，凡此种种，既有现实当中需求变化的速度之快原因，也有本身在需求获取及管理流程上存在的不足。笔者既非需求管理领域的专家，也非经验十足的 PM，更不是老道的流程高手。

在本文中，笔者仅仅依据自己浅浅的理解，来谈谈对于需求开发过程这一方面的陋见，不当之处，恳请广大读者批评指正。

在工作中，笔者始终认为要紧紧抓住活动、任务、角色、工作产品这几方面来展开。即：目前处于什么活动状态？要执行什么任务？由谁来执行？输入输出的产品是什么？只有弄清楚了这之间的关系，才能有效地解决工作中遇到的难题。笔者曾经的主管跟笔者说过一句话：一定要懂得如何分解工作。这句话一直对笔者起着积极的影响作用，当然了，至于如何分解，看每个人的造化了。

好了，言归正传，下面，笔者就简单的介绍一下，如何定制“需求开发”这项活动。

一、需求开发之——需求调研

笔者在上文中说过，要明确状态、任务、角色、产品。这一条主线将一直贯穿在本文中，这里不再赘述。

step1：明确该活动中涉众角色及对应的任务及负责的产品。

在这里，笔者假设有以下几个角色执行这一活动：项目经理、需求分析员、客户。



图 1：需求分析员

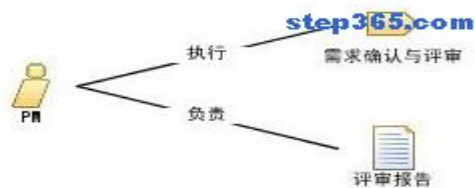


图 2: PM



图 3: 客户需要提供的资料

step2: 进一步细化各干系人在具体任务中的输入输出产品。

在这里，千万别忘了需求分析员还有一个关键的动作：“进入产品需求定义流程”，一定要形成需求开发动作的连续性，“产品需求定义”过程，在下文中详细描述。



图 4: 需求分析员详细工作图



图 5: 项目经理详细工作图

step3: 分清楚工作输入输出产品之间的依赖关系。

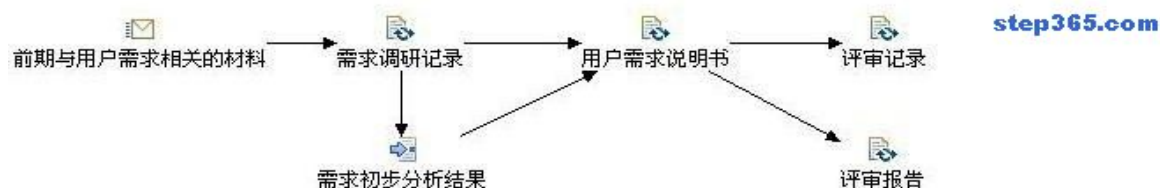


图 6: 工作产品依赖图

step4: 生成完整的“用户需求调研”流程图。

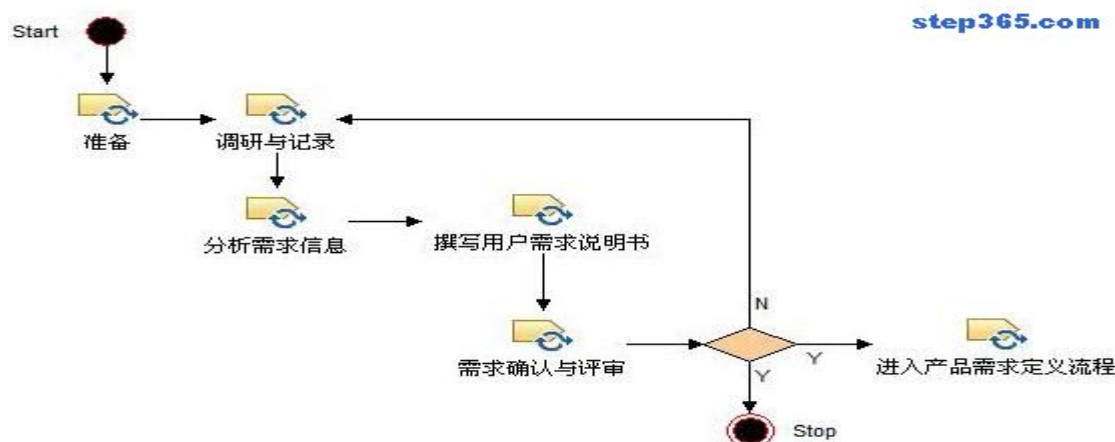


图 7: 用户需求调研流程图

至此，该流程已经初步成形，但是，人们往往会忽略一些项目管理中的支持活动及相关角色的工作活动，比如：SQA、SCM、软件测试、项目助理等这些角色的工作。

举例说明：因在本环节中涉及需求的评审工作，所以，评审的记录不可忽略，到底由谁来承担这一任务呢，不同的企业用不同的作法，在此，笔者让项目助理来承担。



图 8: 项目助理工作成果对应关系

将相关角色的任务、工作产品对应关系补充完整后，用户需求调研活动结束。在完成用户需求调研后，也许大家应该都意识到进入实质性的产品需求定义阶段了，其实笔者在上文中就已经预留了一个伏笔，是的，笔者明确了一条任务是：“进入产品需求定义流程”，这一动作起一个承前启后的作用。

那么，现在我们就一起进入“产品需求定义”活动吧，Let's go!

二、需求开发之——产品需求定义

Step1: 明确该活动中涉众角色及对应的任务及负责的产品。



图 9：需求分析员

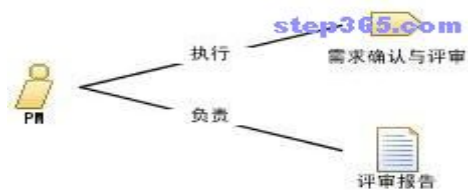


图 10：PM

Step2: 进一步细化各干系人在具体任务中的输入输出产品。



图 11：需求分析员



图 12：PM

Step3: 定制整体流程

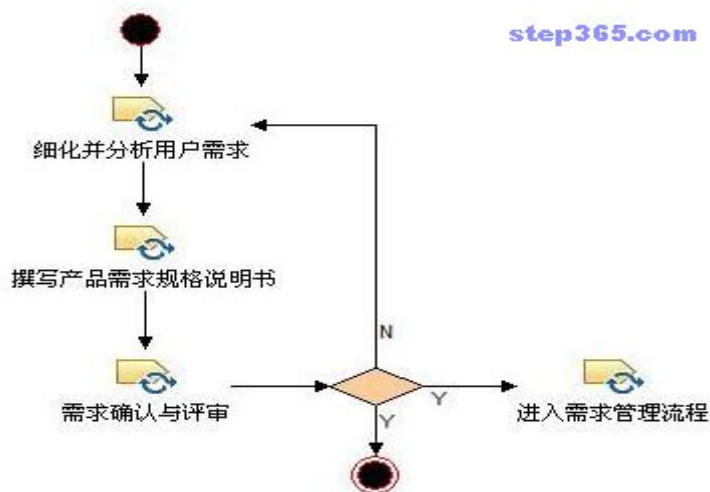


图 13：整体流程

还有其他的一些管理与支持活动，你也可以补充进去，这里笔者将不再赘述。That's over，到此，产品需求定义结束了。

OK，通过本文，你是不是对于自己定制一个或多个软件工程活动有大概了解了呢？你是不是已经决定定制 CMMI 中相应的 PA 流程了呢？那么，决定了就要去做，现在开始吧。

另外，需要说明的一点是：RUP 一直推崇的是迭代的思想，所以，不要孤立地去看这一项活动，要从完整的角度去理解。在本活动中有多少次迭代，每次迭到都要达到什么样的迭代目标，这些根据自己企业的实际情况来定制。

说来说去，其实也很简单，定制流程，首先要抓住几个关键点，就是：活动、角色、任务、工作产品。只有准确地定位了这些信息，才从真正意义上定制适合于企业实际的软件工作活动流程。

哈哈，估计大家可能没有注意到这一个动作：“进入需求管理流程”，没错，下次再分享“需求管理”活动的相关内容。

（注：RMC 全称是 Rational Composer Method，是一款 IBM Rational 的软件产品，也是一种开发方法与理念，笔者需求强调的是：工具并不是主要的，关键是思想，思想对了，用什么工具都可以（当然是相关的），只要能很清晰地表达出你的思想就可以了。）