

一、CMM/CMMI 不是软件企业唯一的选项

- 1、一边按照 CMM/CMMI 做各种需要的文档，一边还在按照老传统做什么调研、方案设计、调试，跟 CMM/CMMI 并不合拍。这是问题之一。
- 2、CMM/CMMI 是一个评估的依据，也是一个过程改进的框架，并不是一个标准。这是问题之二。
- 3、CMM/CMMI 体现了西方的“三权分力”的思想。SEPG 相当于立法机构，而软件项目组相当于行政机构，SQA 人员相当于司法机构。三者相互制约。这种软件项目经理和技术主管职位不分、职责不明的做法怎么能够做好 CMMI？这是问题之三。
- 4、CMM/CMMI 不是为软件研究性项目设计的，而是为产品项目设计的。需要搞清楚你的项目是研究性的，还是产品性的，不要盲从 CMM/CMMI。这是问题之四。
- 5、CMM/CMMI 不可以以强权方式实施。不能越级做 CMM/CMMI。需要收集不同项目的过程实践经验，经总结分析才能形成组织的过程。这是问题之五。

二、实施 CMM 时必须解决的认识问题

在企业的开发能力中过程、技术(含工具、方法)、人员都是主要的因子，都需要全面提高，只关注一个方面，而忽略了其他方面，都是有害的。

在开始实施 CMM 时，最容易犯的一个错误就是“唯管理论”或孤立地只抓过程改善，忽略了开发技术与人员的提高，过分强调管理的作用，实施了半年或一年后，发现企业的生产能力并没有得到明显的改善，这时反对的声音就会成为主流，过程改善就难以继续进行了。

管理就是预防，管理的作用是隐性的，不都是立竿见影的，大家要有耐心。在实施 CMM 时，企业的管理层在开始时往往会对过程改善期望值太高，希望短时间内效果显著，上面我们谈到了，效果显著与否不是由一个方面的要素决定的，需要多个因素共同改善。而管理的最大作用是预防，防患于未然。任何的管理的改善都是符合 J 曲线的，即在改善的初期企业的运行效率可能会下降，甚至可能会出现一些混乱的局面，不过渡过了这段时间就会看到效果。所以在改善的初期大家要有这个思想准备，要有耐心。

坚持活学活用，以我为主。机械照搬 CMM 的条文是在实施 CMM 时常犯的错误。CMM 是软件工程经验的集大成，是从实践中总结出来，用以指导实践的，CMM 本身也在更新版本，不断完善。每个企业都有自己的特点，就象微软的 MSF，那是微软自己内部的管理过程标准，是微软的产品开发经验总结，有些内容是 CMM 中没有的，完全可以借鉴过来使用，所以只要可以提高企业自己的软件管理水平，就应该大胆地来尝试。

要改良式不要革命式。让大家在“小步快跑”中接受变革，这样风险最小，效果最好。期望在短期内通过 CMM 评估，单纯追求市场上的轰动效应。恐怕由于没有实效而得不到大家的认同而难以将这种“水平”持续下去。一个企业引入 CMM 之后会从本质上影响企业的文化，改变大家的思想与做事方法。

CMM 与企业的创新文化是不矛盾的。企业在推行 CMM 时，过分机械，没有从实际出发，不能与实践紧密结合，挫伤了开发人员的积极性。软件企业必须形成创新的文化，事实 CMM 本身也是一种软件工程管理的创新，而技术创新是必须进行管理才能使其有效地转化为生产力，转化为企业的实际效益，达到效益最大化，这是最根本的。

要勇于实践，也要允许犯错误。CMM 就是软件工程经验与教训的总结。在实施 CMM 的过程中，肯定会走些弯路，甚至于要犯错误，由此许多人会议论纷纷，一直会反映到高层经理处，这时不要犹豫，要敢于尝试，更不能因为有困难就打退堂鼓，现在大家都是“摸着石头过河”。要少说不，少说难，勇于实践，有错就改。对于软件企业的领导尤其要注意这一点，不要因为过程中的一些实践失败，就对项目经理、SEPG 等人员有偏见，要提倡这种文化。管理过程改进是组织内所有人的事情，而并非仅仅是 SEPG 的事情。按照 CMM 专家的建议，在一个组织内专职从事软件过程改善的人数应为组织总人数的 2-3%，根据这一建议，我们企业内一开始就配备专职的软件工程过程组（SEPG），这些员工专职负责企业的软件过程改善工作，另外我们根据需要组织一些技术任务组（TWG），他们会兼职的参与特定过程规程、标准的制定、试点和修改完善工作。在这种情况下，可能会出现如下问题：

SEPG 成了最忙的人，TWG 的任务往往会由于那些兼职的人员以工作忙为理由一拖再拖，最后还是由 SEPG 的成员替代 TWG 做工作；企业的非开发人员对管理过程改进的效果一下没有明确地感受到，甚至看到由于加了些新的活动可能使项目拖期可能会更严重，于是他们可能就会将这些抱怨反馈到企业的高层经理，在推行过程中经常会听到：我这个项目时间太紧，当前不适合使用 CMM；

高层经理迫于市场的压力，甚至可能会提出不合实际的项目工期等等。推行 CMM 不仅仅是管理人员的事情，每个人都要积极参与。要改变原来的一些做法：即 SEPG 是在使劲的推进 CMM 的工作，而不是大家自觉自愿的来实施 CMM。从 SEPG 的角

度来看，要做好培训的工作，首先要解决的大家的思想认识问题，这还是比较难的，有些人的思想还是比较顽固的。

当然管理首先要解决的是思想认识上的问题，不但在主观上要解决，在客观上也要有措施，光说不练是不行的，光练不说也是要否定的。我曾经遇到过类似的问题，有的开发人员或者项目经理在口头上是可以接受变革的，会配合工作的，但是在具体操作，很可能又会遇到事实上的否定，这时作为 CMM 的推广人员要尽快提出实施的具体措施，尽快落实。任何变革都要涉及到企业内的权利的再分配，不要忽视企业政治，这是客观存在的，所以一定要预防那些光说不练者。

三、理解 CMM

直到 198x 年，人们还错误地认为，“只要有好的 SE 方法/工具，就可以开发出高质量的软件，就可以提高软件生产率”。现在，人们认识到，“如果软件开发组织不能良好地定义/管理其软件过程，它就往往不能从 SE 中充分获益，从而也得不到预期的结果”。即只重 SE 方法/工具，不重 SE 管理，不行。有句名言叫“拥有工具的傻子还是傻子”（呵呵，还有句名言叫“格言总是情绪的”，因此认为自己 SE 意识不强的朋友，不要认为这句话是骂你的哟），用来说明重视 SE 管理的重要性。

软件过程 = 工程过程 + 管理过程

软件工程学 = 方法/技术+工具/环境+标准/规范+管理/控制

前二者关注工程过程，后二者关注管理过程。

过程：“为实现给定目标所执行的一系列步骤”。

软件过程：“人们用以开发/维护软件及其相关产品的一系列活动，包括《软件工程活动》和《软件管理活动》，自然，其中一定会涉及有关的方法/技术”。其中“相关产品”指项目计划、设计文档、代码、测试用例等，它们在 CMM 中常被称为“软件工作产品”。这里提《软件过程》“一定会涉及有关的方法/技术”。于是，应该这样理解，即“CMM 本身不是特定软件过程的定义”，CMM 只是建议如何“一步一个台阶”地改进《软件过程》。软件过程能力：《软件开发组织/项目组》通过执行其《软件过程》能够实现《预期结果》的《程度》。软件过程性能：《软件开发组织/项目组》通过执行其《软件过程》所得到的《实际结果》。软件过程能力是《最可能的预期结果》，可对《软件开发组织/项目组》而言。软件过程性能是《已得到的实际结果》，可对《软件开发组织/项目组/某个软件项目》而言。软件过程成熟度：一个特定软件过程被明确/有效地定义/管理/测量/控制的程度。

CMM 侧重于软件开发组织中的有关软件过程的宏观管理，for 软件开发组织。PSP 侧重于软件开发组织中的个体软件过程的微观优化，for 软件开发个人（和小型群组）。为什么把个人和小型群组并列呢？因为它们的共同实质是“合作的规模小”。个人开发也需要“合作”哟，你把需求分析做完向自己交接，自己和自己“合作”嘛。CMM 和 PSP 二者相互补充/相互支持，因为如果众多个体没有好的过程意识和高的过程能力，整个开发组织的软件能力成熟度也不可能高。从事纯粹的软件开发的组织，其实是比较少的，这些组织，往往还进行软件采办业务/系统工程产品开发业务等，于是出现了一种将系统工程、软件工程、软件采办等集成在一起的 CMM——它就是 CMMI

我国的 SE 专家基于 CMM 1.1，参照 CMM2.0 和有关资料，提出了符合我国国情的 CSCMM。CSCMM 与 CMM 的主要区别：在等级 1 和等级 2 之间插入了一个“基本级”，使成熟度等级更加均匀。意义：以免软件开发组织长时间达不到 2 级，会失之失去信心。0. 初始级。没有定义软件过程。

1. 基本级。定义了软件过程但执行可能不一致。
2. 可重复级。定义了软件过程且执行基本一致。
3. 已定义级。工程过程/管理过程都明确/妥善定义，并且文档化/标准化，成为软件开发组织的《标准软件过程》。组织内所有软件开发项目均采用该《标准软件过程》的《经批准的剪裁版本》。
4. 定量管理级。在开发过程中，能详细采集软件过程/软件产品的度量数据，从而对《成本/进度/质量》定量预测/定量控制。（可见，“定量管理级”这个名字比“已管理级”更传神。）
5. 优化级。能根据《实践总结/新的技术》，对《标准软件过程》不断改进。

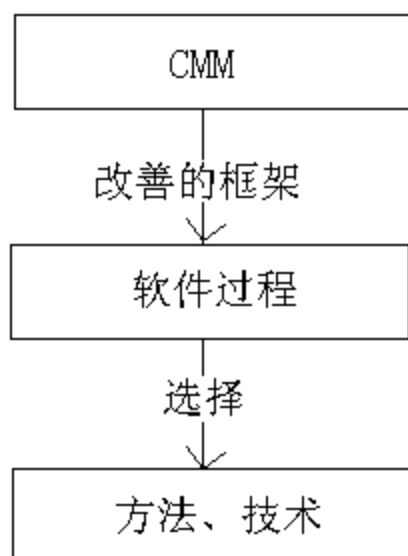
温昱注：上面体现的“定量”比“定性”更难更高级的思想，和数学/化学倒是一样的。另外，《标准软件过程》由开发组织来定，CMM 并不对“用什么方法/技术”做建议。其实，既然优化级就是“能根据《实践总结/新的技术》对《标准软件过程》不断改进”，更能看出 CMM 是独立于任何老技术/新技术的。呵呵，永不失效。

在软件开发组织中，软件过程要规范化/具体化。在软件开发组织中，要通过正式文档/文件和培训，将其软件过程准确无误地告知所有员工，新员工要进行特别培训。CMM 的组织结构：它推荐在最高领导之下设立 SEPG（软工过程组）、SQA（质量保证组）、SEG（软工组）。提示：三个组构成了软件开发中的“立法”、“监督”、“执法”体系，体现了西方文化的法治观念。

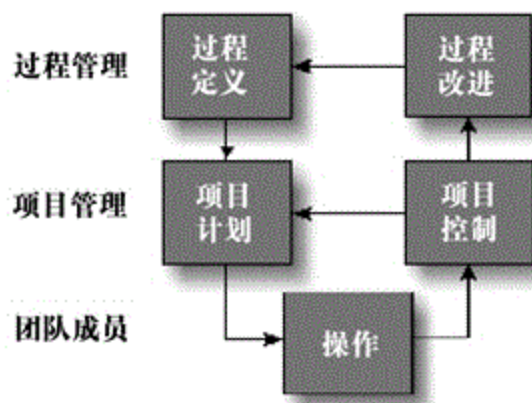
CMM(Capability Maturity Model 软件能力成熟度模型)为软件企业的过程能力提供了一个阶梯式的进化框架，它基于过去所有软件工程成果的过程改善的框架，吸取了以往软件工程的经验教训。它指明了一个成熟的软件组织在软件开发方面需要管理的那些主要工作、这些工作之间的关系、以及以怎样的先后次序一步一步的做好这些工作，使软件组织走向成熟。是目前国际上最流行也是最实用的软件生产过程标准，理解 CMM 需要注意以下几点：

1. 他仅指明该做什么，而没有指明如何做，他不是方法论（温昱注 1：见后），但我们在学习 CMM 时，可以从中学到分析问题的方法。
2. 他仅指明该做的关键内容，他仅描述软件过程的本质属性，而非面面俱到。抓问题主要方面的思想贯穿在整个 CMM 模型中。
3. 软件过程是指软件工程过程、软件管理过程和软件组织的过程三者的有机结合（温昱注 2：见后）。软件工程过程是我们理解的常规的软件的需求分析、设计、编码、测试等过程，软件管理过程是指为使软件过程顺利进行而进行和管理活动的集合。上述两个过程是以软件工程组为主的活动。软件组织的过程是企业级的对软件的组织活动，是以企业为主的活动。
4. 他是从软件过程的角度考虑问题，而非关注软件开发工具。这与框架软件生存周期无关，也与所采用的开发技术（温昱注 1：见后）无关。
5. CMM 为改善整个企业的软件过程提供了指南，而非针对某个具体项目。SW-CMM 并不能保证在这个过程框架下，产品开发百分之百的成功。产品的成功是多种因素的组合，例如市场等因素。
6. CMM1.1 是针对大型软件企业（500 人以上）的，对小型的软件企业（50 人以下）需要裁减。
7. SW-CMM 的过程的不断改进基于许多小的、进化的步骤而不是革命性的创新。
8. 基于 CMM 的过程改善投资力度大、周期长，而技术投资则可能在短期内有较快回报。单独依靠技术改进可能在短期内有较快回报，但最终可能一无所获。

温昱注 1：方法论是通过一组选定的《方法》和《技术》，来解决问题的策略。方法论=方法(们)+技术(们)。因此，这里提 CMM“不是方法论”，“与所采用的开发技术无关”，是正确的。我见过有人提“CMM 是方法论”，不对。当然，软件过程和方法/技术有关哟。



CMM和具体方法、技术无关



温昱注 2: 软件工程 = 工程过程 + 管理过程。软件工程学 = 方法/技术+工具/环境+标准/规范+管理/控制。前二者关注工程过程，后二者关注管理过程。但是，这里却提“软件过程是指软件工程过程、软件管理过程和软件组织的过程三者的有机结合”。我认为它和前一观点并不矛盾，后一观点只是把《管理过程》分成了《软件工程组的管理过程》和《整个企业的管理过程》，又换了个名字叫“组织过程”而已。

四、软件企业如何有效地推行 CMM

软件项目管理是针对软件开发进行的项目管理，它既有项目管理的共性，也有其特殊性。它的特殊性主要表现在软件项目的开发过程及其项目的最终产品--软件产品上。

国内开展软件项目规范管理的时间并不长，软件企业各级管理者对软件项目管理的认识也很不够。目前很多软件项目的成功主要归功于技术高手的个人努力，或者碰巧由一位有能力的项目经理来管理项目，偶然性的因素很大。随着市场竞争的日趋激烈，市场环境的日益成熟，特别是在中国进入 WTO 后，国内软件企业与国外软件企业的竞争，以及开拓国外市场的需要，软件项目管理不完善的问题便越来越突出，软件项目管理显得越来越重要。

为此，许多企业引进了目前世界上较完善的公认软件业标准 CMM（软件能力成熟度模型 Software Capability Maturity Model），希望通过 CMM 的实施来提高公司的软件项目管理水平。

但是，由于对软件项目管理的认识不足，人们对 CMM 的期望值也很大，对 CMM 的实施普遍存在以下误区：

1. CMM 能很快提高企业的软件产品质量;
2. CMM 能解决软件开发过程中的所有问题;
3. 迫于市场压力去拿一张 CMM 评估证书, 而不去考虑 CMM 的真正作用;
4. 技术水平比管理水平更重要, 当技术水平提高时, 再考虑实施 CMM
5.

当抱着以上想法去实施 CMM 时, 其效果便可想而知。而当有些企业认识到软件项目管理不能立竿见影地解决他们面临的问题时, 他们当初对 CMM 实施的信心便开始动摇, 就有可能走上形式化的死循环。

那么, CMM 是什么呢?

从内容上看, CMM 标准分 5 个级别, 每一级别由一些关键过程域 (KPA) 组成, 也就是说, CMM 的管理方式是基于过程的管理方式。每一个 KPA 都有目标 (GOAL) 要求, 要通过 CMM 某级别的评估, 必须达到本级别 所有 KPA 的所有目标要求, 以及本级别以下级别的所有 KPA 的所有目标要求。如过 CMM 三级, 要达到 CMM L3 的要求, 也要达到 CMM L2 的要求。

对于如何达到目标要求, CMM 标准又规定了以下五方面内容:

1. 执行约定: 实施本 KPA 的方针要求与高级管理者的承诺与支持;
2. 执行能力: 实施的先决条件, 包括组织结构、资源、培训等方面的要求;
3. 执行的活动: 为实现一个 KPA 要求所必须的角色和规程。包括制定计划, 进行工作, 跟踪, 并在必要时采取纠正措施;
4. 测量和分析: 采集数据表明过程的状态, 预防问题的发生。
5. 实施验证: 确保已建立流程的实施。包括三级验证, 即高级管理者、主管和项目经理, SQA 对过程相关活动的实施进行验证。

总之, CMM 规定了要达到的目标, 实施需要的条件 (约定、能力), 需要做的工作 (过程活动、测量和验证)。而具体如何去实现, 则须根据公司实际, 可以八仙过海, 各显神通。正如, 规范化管理不会制约开发人员的创造力, 而是使开发人员的创造力在正确且明确的轨道上, 得到更充分、更有效的发挥, 实施 CMM 的真正目的, 是使公司的软件项目管理潜能, 在借鉴成功企业的经验, 结合本公司实际后, 得到完全的展现, 从而保证软件开发过程和软件产品的质量。

推行 CMM, 相当于在企业内引入一种新的软件项目管理的方式, 是软件项目管理的一场变革。它的成功, 有赖于大多数组织成员, 特别是各级管理者的赞同、支持和配合。所以, 在实施 CMM 之前, 公司应对面对的变革阻力有充分的认识和准备。

首当其冲的, 便是对 CMM 的正确认识。正如前面所介绍的, 由于 CMM 推广在国内才起步不久, 其真正作用目前还没有得到充分的证实。所以, 过高、过分的期望, 或者是怀疑、抵制的情绪, 普遍存在。由于 CMM 实施的是一项长期的管理工作, 不能一蹴而就, 所以前者也会对 CMM 的真正落实实施起阻碍作用。因为, 当他们意识到 CMM 不是万能的, 不能满足他们过高、过分的期望时, 他们反过来会否定 CMM 的意义和作用, 从起初热情的拥护者转向坚决的反对者。所以, CMM 的正确认识是推行 CMM 前必须达到的共识。

解决对 CMM 正确认识的方法可以通过理智而循序渐进的宣传和培训活动来实现。为什么要强调理智呢? 正确的认识是为了有一个正确的态度。当一个人的情绪过于激动时, 就不能冷静地、理智地思考问题。情绪的影响能较快地产生反响, 但持续的时间不长。反之, 理智、冷静的接受的观点, 能较长时间的保持。CMM 实施是一项长期的工作, 不能凭一时的冲动, 需要持久、稳定的推动力。所以培训和宣传要考虑采用条理清楚、说理充分的方式。培训和宣传工作也不是能一步就到位的, 有一个对 CMM 标准逐步深入了解的过程。而且循序渐进的宣传和培训就像不断加深的记忆, 起到了强化的作用, 有助于人们对 CMM 的认识深入而持久, 不会轻易改变。

其次, 是管理与技术的对立与统一。CMM 是软件项目管理的一个标准, 软件项目能否成功, 技术因素也是关键。技术水平直接影响着软件项目管理的方式方法, 对管理起着制约作用。反过来, 成功的软件项目管理则对技术的进步起到保障的作用, 巩固技术改进的成果, 使技术的积累与提升沿着正常的轨道有效地发展。没有管理的技术进步, 也是不能持久的。

目前, 绝大多数企业的项目经理都是技术出身, 所谓“技而优则仕”。重技术, 轻管理的现象普遍存在。而且技术对项目质量立竿见影的效果, 也使项目经理对软件项目管理的缓慢而持久的对项目质量的保障作用持怀疑态度, 或者说有只关注眼前利益, 而不顾长远的利益的心态。

这里, 我想补充说一下, 为什么 CMM 标准 2 级只有管理方面的内容, 而到 3 级才引入软件工程、技术管理方面的概念。CMM 标准的 5 级, 就像是台阶, 每上升一级, 代表者软件过程能力的成熟, 软件项目管理水平的提高。下面的台阶是向上走的基础。CMM2 主要关注的是需求的管理 (注意: 是管理, 而不是需求分析等技术)、项目策划、项目跟踪与监督、软件配置管理和软件质量保证。这里, 没有技术的内容在里面。为什么呢? 因为只有建立了管理的机制, 技术的进步才能有保障; 只有有了管理的基础, 才能实现技术的积累与提升。

项目经理是实施 CMM 的中坚力量, 项目经理管理意识的提高是实施 CMM 成功的关键。为此, 企业应建立有效的引导、激励机制, 加强项目经理的管理知识及其应用的培训, 并逐步建立有效的项目经理选拔培养制度。

既然是变革，必然就有新观念、新概念的引进。前面已谈到 CMM 的正确认识对消除变革阻力、有效推行 CMM 的重要性。这是在较高层次上对 CMM 进行抽象后的认识，它起到的是统一思想的作用。在具体的实施过程中，会遇到 CMM 标准中的各种新的名词（实际上就是新的概念，新的管理思想的引入），包括 CMM 的组成结构。概念的理解是 CMM 标准理解的重要组成部分，概念不理解，就不能很好地理解标准要求。如何在操作中体现和落实这些思想，是我们实施过程中要去克服的。如果不能很好地处理这个问题，容易使人沮丧，从而有可能逐步丧失对 CMM 实施的信心。

有人认为，只要 SEPG 成员（SEPG：软件工程过程组 Software Engineering Process Group，负责 CMM 实施的小组，主要工作有组织过程的制定、维护和改进的组织工作）理解 CMM 标准要求（包括理解 CMM 标准中的各种概念）就可以了。其他人员，包括各级管理者、项目经理、开发人员不需要理解这些要求和概念，只要告诉他们如何去做就可以了。本人对这种说法持否定态度。如果不理解 CMM 标准要求，不理解各种概念的真正含义，只知其然，而不知其所以然，实施者如何能把 CMM 标准在具体工作中真正落实。就像只告诉你怎么走，而不告诉你往那里去，你将如何处理遇到的意外情况？如果所说的行不通，怎么办？你又如何根据实际情况有效地到达目的地？

也许在某些行业，这种“只告诉你如何做，不告诉你为什么要这么做”的方式是有效的，但这种方式绝对不适用于软件企业。为什么？因为人的差异。我们要根据不同的人采取不同的激励方式。因为 CMM 推行本身就是引导和激励企业全体开发人员的持久的支持和参与，所以要根据开发人员本身的价值和他们的追求采取相应的 CMM 推行措施。开发人员的整体素质相对较高，从事的又是创造性很强的软件开发工作，自主性相对较强。让他们理解标准要求，理解各种概念，对于调动他们的积极性，从而有效的实施 CMM 是非常有帮助的。

另一方面，CMM 标准的核心管理思想，过程改进，也只有在全员实施者理解标准要求的基础上，积极参与的过程中才能实现。SEPG 制定的初始过程文件可能是最好的、理想的，但绝对不可能是最适用的、最可行的、最可操作的。这需要各级实施者在使用文件定义过程的基础上，结合标准要求对过程进行改进，从而找到符合标准要求的、适用于本企业的软件管理方式。

所以，对各级人员进行 CMM 标准理解的培训非常重要。培训可以分为不同层次进行，一层一层逐步从抽象到具体。培训的方式也可以有演讲、座谈、案例讲解、讨论小组、实际操作中的指导等不同的方式。

总而言之，有效推行 CMM 的关键，在于充分理解 CMM 标准的要求后，结合企业软件项目开发的实际，找出适合于本企业的软件项目管理方法。CMM 标准的实施，重在解决企业软件项目开发过程中存在的问题，实现软件开发过程的持续改进。企业应当充分认识到推行前及推行过程中可能遇到的困难和阻力，并根据企业实际，采取有效的措施。只有这样，才能使推行 CMM 真正成为企业提高软件项目管理水平，保证软件产品质量，从而提高企业市场竞争力的有效手段

五、CMM“六步曲”-CMM 实施技术问题分析

一些正在实施 CMM2 级的企业发现有大量的重复性工作要做，原因何在？没有做好需求开发是产生这一问题的主要原因！

1 需求管理与需求工程

需求开发和需求管理是需求工程的两部分，如果没有做好需求开发，那么从需求管理的角度看就会出现重复性的工作。导致需求开发欠佳的主要原因有以下几点：

◆ 缺乏良好的需求规格说明编写模板

分析一些企业的 CMM 实施过程，从表面上看，它们的确遵循了先推荐方案再进行评审的基本选择原则，但由于缺乏经验，实际选定的方案常常缺乏客观性，同时在企业的工程和管理机制里又缺乏实践反馈的方法和过程来不断地改进原有的方案。一般来说，大家在一起工作的时间长了，就会形成一种“默契”，而这很可能给以后的工程和管理工作埋下很多隐患，一旦出现意见分歧时，这种默契就不复存在。如果按 CMM 的要求去做，大量类似的重复工作就会因此出现。改进的方法之一是在整个工程和管理过程中，既保持文档和产品的一致性，又反向追踪需求规格说明更改的程度，并持续改进需求规格说明编写模板。

◆ 较严重地忽略了非功能性需求

目前，国内的软件客户很少主动提出非功能性需求，但随着客户的逐渐成熟，软件客户对软件的非功能性需求也会越来越高，这就对软件开发商提出了更高的要求。不做好非功能性需求的规格说明编写工作，同样会陷入大量重复工作的包围之中。如果缺乏非功能性需求的规格说明，将会使一些基础问题直到软件生命的中期才被发现，这将导致大量的文档和产品需要更改，由此带来严重的工程和管理难题。改进的方法之一是调用有相当软件调试和维护背景的资深人员参与需求规格说明的编写，他们的丰富经验往往可以较好地弥补设计开发人员在这方面存在的不足。

◆ 缺乏对需求文档的配置管理

采用两个需求规格说明编写模板是一种不错的做法：一份给软件客户看，一份留给软件开发小组内部使用，前者的目标是让客户较容易理解，后者则更加专业化。在这种情况下，两个需求规格说明都应纳入配置管理的范畴以便从管理的角度保持其一致性。这还不够，从工程角度考虑，企业还应该形成一套从前者到后者的转化规则。尽管这两个模板的表现形式可能是自然语言，但一

个尽可能严谨的规则将大大缩小转化过程中人为自由发挥的空间。需要注意的是，这套规则的建立应从一个项目开始，从基础做起，逐渐完善。例如，首先确定项目的基本名词和动词集合，并规定语句书写规则。

◆ 需求规格说明缺乏可测性

在需求说明应具备的几个特性里，为什么单单挑出可测性呢？在需求说明编写阶段，主观性对其他特性的影响较大，而一个独立且有经验的测试组对可测性的掌握是从独立于需求规格说明的测试文档出发的。从测试的角度看，很多需求说明是不可测的，这就要求重写这些需求说明，直到可测性得到保证。测试组要求的往往是简洁且准确的说明，而这恰恰是开发人员做得不够好的几个方面之一；另一方面，目前无论是国内的市场还是企业，对测试人员都不够重视，软件企业很少招聘测试人员。实际上，优秀的软件测试人员对保证软件质量非常重要，一般来说，测试部门的经理应该由具有软件开发经验、做过软件开发管理且有相当测试经验的资深人员担任。处理好设计和测试人员的关系是众多国内软件企业应该进一步重视的问题。

◆ 缺乏较好的需求规格说明转化规范

需求规格说明转化的目的是把用自然语言书写的需求说明转化为更准确的中间形式，这一转化过程也被称为“软件建模”。一般来说，建模可以使需求说明的某些方面更形式化一些，并使设计更加清晰地保持需求继承。通常，不做需求规格说明转化或缺乏较好的需求规格说明转化规范，将造成不同程度的需求说明丢失，从而增加后续管理工作的难度。需求管理的根本目的是为其后的工程和管理建立基线并保持相关及衍生文档和产品与需求的一致性，因此需求工程完成得好坏对需求管理实施的工作量有很大影响。

2 配置管理与工作产品的转化

软件配置管理的目的是保证项目生成的产品在软件生命周期中的完整性，它需要一个较好的工具，当找不到较好的商用软件工具覆盖该关键域的实践时，许多国外软件企业会自行开发一些工具来弥补不足，并且取得了很好的效果。国内软件企业在实施该关键域时也会使用一些工具，但存在的典型问题是：有太多的 SCCB（软件配置控制委员会）活动。

配置管理是在软件生命周期中建立和标识软件工作产品并控制基线的更改，这将保证软件工作产品的完整性和一致性。但是，作为配置项/单元标识的软件工作产品通常为典型的软件生命周期中的工作产品，这些产品具有一个共同特点：一个产品通常是由另一个产品转化而来。从一些企业配置管理下的工作产品来看，存在的主要问题是缺乏较好的可转化性。在这里，“较好的可转化性”是指把一个产品转化为另一个产品时有较规范的转化规则可循，其目的是最大程度地保证一种工作产品能被忠实地转化为另一种工作产品形式，从而最大限度地降低最初的软件需求在转化过程中出现遗漏和被错误解释的可能性。企业在实施这个关键过程域时，应由 SCCB 记录工作产品的更改以及引发这些更改的原因，这些数据能很好地帮助企业找出问题的症结。一般来说，引发类似问题的原因主要有以下 3 点：需求规格说明书编写不好或不全；工作产品模板定义不好；工作产品之间转化缺乏规范定义。

3 项目计划与数据收集和分析

项目计划是 CMM 实施一开始就涉及且最后才能相对完善的关键过程域，它主要包括软件规模估计、工作模块计划、人力资源计划、进度安排和其他资源计划。在其他关键过程域的实践相对稳定之前，项目计划的实践总是处于需要改动的状态。一般来说，期望在 CMM 实施之初就有一个可靠的项目计划是不现实的，因为这需要经历若干项目的实施才能获得有效数据并据此制定未来项目的计划。我们知道，配置管理可以保证项目生成的产品在软件生命周期中的完整性，因此，为了更好地实施项目计划，我们可以把用于项目计划的大部分数据放在对应的工作产品配置管理之下，必要时，还可将工作产品进一步细化，以保证对应的项目计划数据的准确性。项目完成后，我们还应该对项目计划的数据进行收集和分析，在此基础上制定下一个项目计划时，准确性就能大大提高。通过对若干项目进行同样的实践，项目经理就有了比较可靠的数据用于制定未来的项目计划。通常，项目跟踪和监督实施不好的原因很大程度上是由于项目计划的频繁更动，同时缺乏良好的项目跟踪工具，使项目管理人员逐渐失去跟踪项目的兴趣。

4 质量保证与实践反馈

实践反馈是质量保证体系得以有效运作的驱动力，企业应该为所有项目建立一条从 SQA 到项目经理以及更高层经理的反馈渠道。实践反馈是 SQA 组与高层经理相互沟通的过程，SQA 组定期向高层经理汇报 SQA 的活动，并及时与项目组沟通，使项目组能尽早改进工作；当沟通不畅、发现项目组运作不力或发现组间协调困难时，应及时报告高层经理，通过高层经理的协调及时进行修正。有些项目经理认为自己心里有一套计划，只要按计划进行就可以按时保质完成项目，但事实并非如此，在项目组之间的协调问题上，高层经理的作用是非常明显的。如果仅仅为满足 CMM 的要求而虚设高层经理，这种做法是不可取的，因为如此一来，实践反馈是不完全的。SQA 组在 CMM 实践中犹如一个司法机构，但这还不够，它还应该为改进过程管理提供资源。SQA 组不一定完全由专职人员组成，也可调配一些擅长软件开发方法和软件过程管理的人员参与主要的 SQA 活动。

5 同行评审

从理论上讲，同行评审这个关键域的实施并不难，但实际上大多数企业都掌握得不够好，主要表现在以下方面：

◆ 评审时组间争论过多或过少

这一问题在不同企业的表现也不同。调查表明，争论较多的情况是工作产品的输入/输出不清楚，组间缺乏沟通的公共平台，因此组间只有通过较多的讨论甚至争论才能弄清其他组的需求。遗憾的是，事后大家并没有坐下来认真讨论如何改进原工作产品的模板形式或表现形式，因而也就无法从根本上解决问题。另一种较极端的情况是，评审一个组的工作产品时，其他组很少发表意见，尽管有些问题是十分明显的。通过调研发现，这实际上是企业文化的问题。一种普遍的想法是“等我们实际做的时候自然就清楚了”，但实际情况往往事与愿违，这使企业的工作效率大打折扣，但又不易被管理层意识到。无论是哪种情况，最终的原因是：“项目甚至企业缺乏持续改进过程管理的意识。”

◆ 缺乏心理训练

做好同行评审的最大挑战是克服心理障碍。简单地说，同行评审就是被别人挑错或挑别人的错。因此，评审会就像是答辩会，必须做好充分的准备。当角色互换，自己成了挑错方时，则应该把被评审的工作产品看成是自己在较早前完成的，现在再进行一次修改，且修改完成后，自己要拿它去参加评审答辩。经历几次这种心理角色换位，就会逐渐适应。如果大家都这么做，同行评审就会形成良好的氛围，这对形成健康的企业文化将起促进作用。

◆ 竞争与合作意识不充分

从另一个角度看，同行评审又是竞争与合作的最佳表现场所和形式，凡在这种场合讲话有理且意见中肯的人逐渐成为团队的核心人物。在这种竞争的环境中，合作是基础，同行评审的目的就是在合作的前提下尽早且有效地排除工作产品中的缺陷。把握好竞争与合作的尺度，将有益于企业文化的发展，否则有可能出现恶性循环。如何把握呢？从大量的案例看，多数消化少数是较好的方法，因为文化是不可创造的。

◆ 考虑不全面

同行评审存在的另一个问题是评审时仅注意工作产品内容本身，大家面对面地弄清内容后，却忽略了如何改进工作产品的表现形式，使新表现形式下的工作产品可更好地用书面形式表示，进而可减少面对面沟通的需求。当然，面对面的沟通并不是不好，但如果一个工作产品需要太多的口头表达才能被理解，则原因只有两个：书写不清楚或模板定义不好，如果是后者则情况更糟。

6 缺陷预防与度量

缺陷预防的目的是为了识别产生缺陷的原因并防止其再次发生。一些实施低级别 CMM 的企业通常都采用一些度量（metrics）来预防缺陷，包括软件大小、软件设计错误、编码错误、测试错误、设计评审覆盖、编码评审覆盖、产生测试覆盖、与过程原因相关的缺陷、与项目原因相关的缺陷等。个别企业选用了一些难度更大的度量。大多数情况下，这些企业并非要达到更高级别的 CMM，而是从产品需求的特性出发，对工作产品进行缺陷分析和预防。其过程通常是：获取数据、数据整理、度量、发现原因并确定过程的改进措施，其典型例子包括设计复杂性测试及测试深度、模块复杂性测试及测试深度等。这类企业的软件产品一般具有以下特点：软件产品规模较大，通常在软件产品交付给用户后，通过相当长时间的不断维护，稳定性才能达到满意程度。如果在早期对可能产生较多错误的软件模块进行识别，加强对这些模块的早期关注和测试，就可较早地使系统达到稳定。这种方法常常用在大型软件开发中，但真正用好的并不多，主要原因有以下几点：

◆ 忽略了使用度量的环境

大多数工作产品的度量都是为某一种特定的设计方法或编程语言设计的，忽略了这个因素，度量就容易失去准确性。此外，软件产品不同的行业特性往往也是造成度量产生偏差的原因。

◆ 忽略了对度量参数的修改

一些度量参数是在原来的实践环境下确定的，当在新环境下使用时，其中的参数很可能需要进行修改，才能使度量的准确性得到保证。

◆ 忽略了对相关性的研究

使用的度量与缺陷在本地的相关性越高，度量的价值就越大。获取这种相关性的方法一般是对本地的历史数据进行相关性研究，企业只有在确认满意的相关性后才可将度量用于缺陷预防。

六、软件工程与能力成熟度模型 CMM

软件项目失败的主要原因有：需求定义不明确；缺乏一个好的软件开发过程；没有一个统一领导的产品研发小组；子合同管理不严格；没有经常注意改善软件过程；对软件构架很不重视；软件界面定义不善且缺乏合适的控制；软件升级暴露了硬件的缺点；关心创新而不关心费用和 risk；军用标准太少且不够完善等等。在关系到软件项目成功与否的众多因素中，软件度量、工作量估计、项目规划、进展控制、需求变化和风险管理等都是与工程管理直接相关的因素。由此可见，软件工程管理意义至关重要。软件工程管理和其它工程管理相比有其特殊性。首先，软件是知识产品，进度和质量都难以度量，生产效率也难以保证。其次，软件系统复杂程度也是超乎想象的。CMM 框架可用 5 个不断进化的层次来表达：其中初始层是混沌的过程，可重复层是经过训练的软件过程，定义层是标准一致的软件过程，管理层是可预测的软件过程，优化层是能持续改善的软件过程。

任何软件开发单位在致力于软件过程改善时，只能由所处的层次向紧邻的上一层次进化，即软件过程的进化是渐进的，而不能是跳跃的。而且在由某一成熟层次向上一更成熟层次进化时，在原有层次中的那些已经具备的能力还应该得到保持与发扬。

CMM 家族包括 CMM 集成产品集、SA-CMM(软件获取能力成熟度模型)、SE-CMM(系统工程能力成熟度模型)和 IDEAL 模型。其中 CMM 集成产品集为工业界和政府部门提供了一系列集成产品，以支持软件过程和产品的改善；SA-CMM 用于单位获取和采购基于软件的应用系统的软件过程，美国国防部、陆军、海军和一些商用单位都已采用 SA-CMM 对他们的获取能力进行评估；SE-CMM 是描述一个单位为保证实现一个好的系统工程的主要元素；而 IDEAL 模型则是一个单位用于启动、规划和实现过程改善措施蓝图的模型，概括了建立一个成功的过程改善项目的必要步骤，其中 I 代表 Initiating(启动)、D 代表 Diagnosing(诊断)、E 代表 Establishing(建造)、A 代表 Acting(措施)、L 代表 Learning(学习)。

需要注意的是，并不是实施了 CMM，软件项目的质量就能有所保障。CMM 不是万能的，它的成功与否，与一个组织内部有关人员的积极参与和创造性活动是密不可分的，而且 CMM 并未提供实现有关过程域所需要的具体知识和技能。PSP 为基于个体和小型群组软件过程的优化提供了具体而有效的途径，例如如何制订计划，如何控制质量，如何与其他人相互协作等等。在软件设计阶段，PSP 的着眼点在于软件缺陷的预防，其具体办法是强化设计结束准则，而不是设计方法的选择。绝大多数软件缺陷是由于对问题的错误理解或简单的失误所造成的，只有很少一部分是由于技术问题而产生的。而且根据多年来的软件工程统计数据表明，如果在设计阶段注入一个差错，则这个差错在编码阶段要引发 3~5 个新的缺陷，要修复这些缺陷所花的费用要比修复这个设计缺陷所花的费用多一个数量级。因此，PSP 保障软件产品质量的一个重要途径是提高设计质量。

TSP 指导项目组中的成员如何有效地规划和管理所面临的项目开发任务，并且告诉管理人员如何指导软件开发队伍始终以最佳状态来完成工作。TSP 实施集体管理与自己管理自己相结合的原则，最终目的在于指导一切人员如何在最少的时间内，以预定的费用生产出高质量的软件产品；所采用的方法是对群组软件开发过程的定义、度量和改进。实施 TSP 的先决条件有 3 条：首先，需要有高层主管和各级经理的支持，以取得必要的资源；其次，项目组开发人员需要经过 PSP 的培训并有按 TSP 工作的愿望和热情；第三，整个单位在总体上应处于 CMM 二级以上。在实施 TSP 的过程中，首先要有明确的目标，开发人员要努力完成已经接受的委托任务。在每一阶段开始，要做好工作计划。如果发现未能按期按质完成计划，应分析原因，以判定问题是由于工作内容不合适或工作计划不实际所引起，还是由于资源不足或主观努力不够所引起。开发小组一方面应随时追踪项目进展状态并进行定期汇报，另一方面应经常评审自己是否按 PSP 的原理工作。开发人员应按自己管理自己的原则管理软件过程，如发现过程不合适，应及时改进，以保证用高质量的过程来生产高质量的软件。项目开发小组则按集体管理的原则进行管理，全体成员都要参加和关心小组的规划、进展的追踪和决策的制订等工作。

总之，单纯实施 CMM，永远不能真正做到能力成熟度的升级，只有将实施 CMM 与实施 PSP 和 TSP 有机地结合起来，才能发挥最大的效力。因此，软件过程框架应该是 CMM/PSP/TSP 的有机集成，其相互关系可用图 1 来表示。

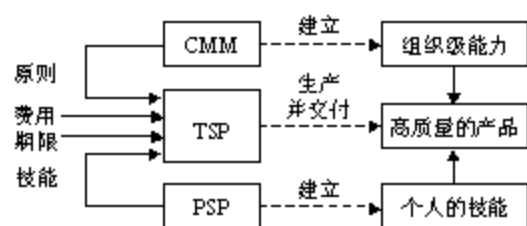


图 1 软件过程框架

目前国内对软件工程管理存在的最大问题是认识不足。管理实际上是一把工程，需要高层管理人员的足够重视。据国外有些大公司的介绍，他们在软件工程管理方面的投资一般占软件开发费用的 10% 左右，这些都需要得到高层管理人员的支持。而且软件过程的重大修改也必须由高层管理部门启动，这是软件过程改善能否进行到底的关键。此外，软件过程的改善还有待于全体有关人员的积极参与，否则不仅他本人将失去从软件过程改善中获得提高的机会，甚至还会成为过程改善的阻力。

除了要认识到过程改善工作是一把工程这个关键因素外，还应认识到软件过程成熟度的升级本身就是一个过程，且有一个生命周期。因此，过程改善工作必然具有一切过程所具有的固有特征，即需要循序渐进，不能一蹴而就，需要持续改善，不能停滞不前；需要联系实际，不能照本宣科；需要适应变革，不能凝固不变。而且我认为，要将 CMM/PSP/TSP 引入软件企业，最有效的途径是要对单位主管和主要开发人员进行系统的培训。建议这些中小企业可以以 CMM 为框架，先从 PSP 做起，然后在些基础上逐渐过渡到 TSP，以保证 CMM/PSP/TSP 确实在企业中生根开花。总之，我们必须从软件过程、过程工程的角度来看待 CMM

的发展，从经济学的观点来分析这个过程的价值。

七、CMM 五级标准

第一级：初始级

在初始级，企业一般不具备稳定的软件开发与维护的环境。常常在遇到问题的时候，就放弃原定的计划而只专注于编程与测试。

第二级：可重复级

在这一级，建立了管理软件项目的政策以及为贯彻执行这些政策而定的措施。基于过往的项目的经验来计划与管理新的项目。

第三级：定义级

在这一级，有关软件工程与管理工程的一个特定的、面对整个企业的软件开发与维护的过程的文件将被制订出来。同时，这些过程是集成到一个协调的整体。这就称为企业的标准软件过程。

第四级：定量管理级

在这一级，企业对产品与过程建立起定量的质量目标，同时在过程中加入规定得很清楚的连续的度量。作为企业的度量方案，要对所有项目的重要的过程活动进行生产率和质量度量。软件产品因此具有可预期的高质量。

第五级：（不断）优化级

在这个等级，整个企业将会把重点放在对过程进行不断的优化。企业会采取主动去找出过程的弱点与长处，以达到预防缺陷的目标。同时，分析有关过程的有效性的资料，作出对新技术的成本与收益的分析，以及提出对过程进行修改的建议。

CMM 第一级：初始级

◆ 特征

- (1)软件过程的特点是杂乱无章，有时甚至混乱，几乎没有定义过程的规则或步骤。
- (2)过分的承诺，常作出良好的承诺：如“按照软件工程方式，有序的工程来工作”；或达到高目标的许诺。但实际上却出现一系列问题。
- (3)遇到危机就放弃原计划过程，反复编码和测试。
- (4)成功完全依赖个人努力和杰出的专业人才，取决于超常的管理人员和杰出有效的软件开发开发人员。具体的表现和成果都源于或者说是决定于个人的能力和他们先前的经验、知识以及他们的进取心和积极程度。
- (5)能力只是个人的特性，而不是开发组织的特性。依靠着个人的品质或承受着巨大的压力；或找窍门取得成果。但此类人一旦离去，对组织的稳定作用也消失。
- (6)软件过程是不可确定的和不可预见的。软件成熟性程度处于第一级软件组织的软件过程在实际的工作过程中被经常的改变（过程是随意的）。这类组织也在开发产品，但其成果是不稳定的，不可预见的，不可重复的。也就是说，软件的计划、预算、功能和产品的质量都是不可确定和不可预见的。

◆ 过程

- (1)极少存在或使用稳定的过程
- (2)所谓“过程”，往往是“就这么干”而言。
- (3)各种条例，规章制度互不协调，甚至互相矛盾。

◆ 人员

- (1)依赖个人努力和杰出人物。一旦优秀人物离去，项目就无法继续。
- (2)人们的工作方式如同“救火”，就是在开发过程中不断地出现危机，以及不断的“救火”。

◆ 技术

引进新技术是极大风险。

◆ 度量

不收集数据或分析数据。

◆ 改进方向

- (1)建立项目管理过程，实施规范化管理，保障项目的承诺。
- (2)首要任务是进行需求管理，建立客户与软件项目之间的共同理解，使项目真正反映客户的要求。
- (3)建立各种软件项目计划，如软件开发计划、软件质量保证计划、软件配置管理计划、软件测试计划、风险管理计划及过程改进计划。
- (4)开展软件质量保证活动（SQA）。

CMM 第二级：可重复级

◆ 特征

- (1)进行较为现实的承诺，可按以前在同类项目上的成功经验建立的必要过程准则来确保再一次的成功。
- (2)主要是逐个项目地建立基本过程管理条目来加强过程能力。
- (3)建立了基本的项目管理过程来跟踪成本、进度和功能。
- (4)管理工作主要跟踪软件经费支出、进度及功能。识别在承诺方面出现的问题。
- (5)采用基线（BASELINE）来标志进展、控制完整性。
- (6)定义了软件项目的标准，并相信它，遵循它。
- (7)通过子合同建立有效的供求关系。

◆ 过程

- (1)软件开发和维护的过程是相对稳定的，但过程建立在项目一级。
- (2)有规则的软件过程是在一个有效的工程管理系统控制之下，先前的成功经验可以被重复。
- (3)问题出现时，有能力识别及纠正。承诺是可实现的。

◆ 人员

- (1)项目的成功依赖于个人的能力以及管理层的支持。
- (2)理解管理的必要性及对管理的承诺。
- (3)注意人员的培训问题。

◆ 技术

建立技术支持活动，并有稳定的计划。

◆ 度量

每个项目建立资源计划。主要是关心成本、产品和进度。有相应的管理数据。

◆ 改进方向

- (1)不再按项目制定软件过程，而是总结各种项目的成功经验，使之规则化，把具体经验归纳为全组织的标准软件过程。把改进组织的整体软件过程能力的软件过程活动，作为软件开发组织的责任。
- (2)确定全组织的标准软件过程，把软件工程及管理活动集成到一个稳固确定的软件过程中。从而可以跨项目改进软件过程效果，也可作为软件过程剪裁的基础。
- (3)建立软件工程过程小组（SEPG）长期承担评估与调整软件过程的任务，以适应未来软件项目的要求。
- (4)积累数据，建立组织的软件过程库及软件过程相关的文档库。
- (5)加强培训。

CMM 第三级：确定级

◆ 特征

- (1)无论管理方面或工程方面的软件过程都已文件化、标准化，并综合成软件开发组织的标准软件过程。
- (2)软件过程标准被应用到所有的工程中，用于编制和维护软件。有的项目也可根据实际情况，对软件开发组织的标准软件过程进行剪裁。
- (3)在从事一项工程时，产品的生产过程、花费、计划以及功能都是可以控制的，从而软件质量也可以控制。
- (4)软件工程过程组（SEPG）负责软件活动。
- (5)在全组织范围内安排培训计划。

◆ 过程

- (1)整个组织全面采用综合性的管理及工程过程来管理。软件过程和管理活动是稳定的和可重复的，具有连续性的。
- (2)软件过程起了预见及防范问题的作用，能使风险的影响最小化。

◆ 人员

- (1)以项目组的方式进行工作。如同综合产品团队。
- (2)在整个组织内部的所有人对于所定义的软件过程的活动、任务有深入了解，大大加强了过程能力。
- (3)有计划地按人员的角色进行培训。

◆ 技术

在定性基础上建立新的评估技术。

◆ 度量

- (1)在全过程中收集使用数据。
- (2)在全项目中系统性地共享数据。

◆ 改进方向

- (1)开始着手软件过程的定量分析，以达到定量地控制软件项目过程的效果。
- (2)通过软件的质量管理达到软件的质量目标。

CMM 第四级：管理级

◆ 特征

- (1)制定了软件过程和产品质量的详细而具体的度量标准，软件过程 and 产品质量都可以被理解和控制。
- (2)软件组织的能力是可预见的，原因是软件过程是被明确的度量标准所度量和操作。不言而喻，软件产品的质量就可以预见和得以控制。
- (3)组织的度量工程保证所有项目对生产率 and 质量进行度量、并作为重要的软件过程活动。
- (4)具有良好定义及一致的度量标准来指导软件过程，并作为评价软件过程及产品的定量基础。
- (5)在开发组织内已建立软件过程数据库，保存收集到的数据，可用于各项目的软件过程。

◆ 过程

- (1)开始定量地认识软件过程。
- (2)软件过程的变化小，一般在可接受的范围内。(3)可以预见软件过程中 and 产品质量方面的一些趋势。一旦质量经度量后超出这些标准或是有所违反，可以采用一些方法去改正，以达到良好的目标。

◆ 人员

每个项目中存在强烈的群体工作意识。因为每人都了解个人的作用与组织的关系，因此能够产生这种群体意识。

◆ 技术

不断的在定量基础上评估新技术。

◆ 度量

- (1)在全组织内进行数据收集与确定。
- (2)度量标准化。
- (3)数据用于定量地理解软件过程及稳定软件过程。

◆ 改进方向

- (1)缺陷防范，不仅仅在发现了问题时能及时改进，而且应采取特定行动防止将来出现这类缺陷。
- (2)主动进行技术变动管理、标识、选择和评价新技术，使有效的新技术能在开发组织中施行。
- (3)进行过程变动管理，定义过程改进的目的，经常不断地进行过程改进。

CMM 第五级：优化级

◆ 特征

- (1)整个组织特别关注软件过程改进的持续性、预见及增强自身，防止缺陷及问题的发生，不断地提高他们的过程处理能力。
- (2)加强定量分析，通过来自过程的质量反馈和吸收新观念，新科技，使软件过程能不断地得到改进。
- (3)根据软件过程的效果，进行成本/利润分析，从成功的软件过程中吸取经验，加以总结。把最好的创新成绩迅速向全组织转移，对失败的案例，由软件过程小组进行分析以找出原因。
- (4)组织能找出过程的不足并预先改进，把失败的教训告知全体组织以防止重复以前的错误。
- (5)对软件过程的评价和对标准软件过程的改进，都在全组织内推广。

◆ 过程

- (1)不断地系统地改进软件过程。
- (2)理解并消除产生问题的公共根源，在任何一个系统中都可找到：由于随机变化造成重复工作、进而导致时间浪费。为了防止浪费人力可能导致的系统变化。要消除“公共”的无效率根源，防止浪费发生。尽管所有级别都存在这些问题，但这是第五级的焦点。

◆ 人员

- (1)整个组织都存在自觉的强烈的团队意识。
- (2)每个人都致力过程改进，人们不再以达到里程碑的成就而满足，而要力求减少错误率。

◆ 技术

基于定量的控制和管理，事先主动考虑新技术、追求新技术。可以实现软件开发中的方法和新技术的革新、以防止出现错误，不断提高产品的质量和生产率。

◆ 度量

利用数据来评估，选择过程改进。

◆ 改进方向

保持持续不断的软件过程改进。

CMM 总结：五层结构图

我们看到，在第五级上技术和过程的改进像普通商业活动一样有计划、有管理地进行。由于组织不断致力于改进过程的能力，所以软件开发组织的能力可持续改进。这种改进不仅表现在对存在的软件过程逐步改进，不表现在采用新技术和新方法方面的革新。初始级…>有纪律的过程…>可重复级…>标准一致的过程…>确定级…>能预见的过程…>可管理级…>不断改进的过程…>优化级

八、实施软件质量保障体系 CMM/TSP/PSP 的建议

软件产业的发展，在经历了从 70 年代开始以结构化分析与设计、结构化评审、结构化程序设计以及结构化测试为特征的结构化生产时代，到 90 年代中期，以 CMM 模型的成熟模型和日益为市场接受为标志，已经进入以过程成熟模型 CMM、个体软件过程 PSP 和群组软件过程 TSP 为标志的以过程为中心的时代，而软件发展第三个时代，及软件工业化生产时代，从 90 年代中期软件过程技术的成熟和面向对象技术、构件技术的发展为基础，已经渐露端倪，估计到 2005 年，可以实现真正的软件工业化生产，这个趋势应该引起软件企业界和有关部门的高度重视，及早采取措施，跟上世界软件发展的脚步。软件生产转向以改善软件过程为中心，是世界各国软件产业或迟或早都要走的道路。软件过程改善是当前软件开发技术的核心问题。

软件过程的三个流派：

CMU-SEI 的 CMM/PSP/TSP

ISO 9000 质量标准体系

ISO/IEC 15504 (SPICE)

CMU-SEI 的 CMM/PSP/TSP

20 世纪 80 年代中期国际软件产业界对软件的研究十分重视，因为在采用软件工程方法克服软件危机的过程中，人们认识到，软件是否完善是软件风险大小的决定因素。这方面的研究取得了重大的突破，其标志是 1987 年美国 Carnegie Mellon 大学软件工程研究所(CMU/SEI)以 W.S.Humphrey 为首的研究组发表的研究成果“承制方软件工程能力的评估方法”，该成果在 1991 年发展成为 CMM（软件过程能力成熟度模型）。软件过程能力成熟度模型被国际软件界公认为软件工程学的一项重大成果。软件目前，软件能力成熟度模型 2.0 版已经修订问世。CMM 在软件工程的实践方面已有很大的影响，在工业界已得到广泛接受。不仅已用于军事控制系统，而且已用于全球经济领域的主要组织。有数千个组织在利用 CMM 的软件过程改进。在美国，关于 CMM 模型的教程已经作为参考和研究的对象出现了，这样做是为了让 CMM 模型极其相关问题引起工业界的更密切地关注。基于 CMM 模型的工具如成熟度问题集，软件过程评估训练和软件能力评价训练已经在 CMM 中渐渐得到修订。近期的关于 CMM 的活动主要是发展关于 CMM 模型的不同版本。由于 CMM 并未提供有关实现 CMM 关键过程域所需的具体知识和技能，因此，美国 Carnegie Mellon 大学软件工程研究所(CMU/SEI)以 W.S.Humphrey 为首主持研究与开发了个体软件过程 PSP (Personal software process) 和群组软件过程 TSP(Team Software Process)，形成 CMM/PSP/TSP 体系。

ISO 9000 质量标准体系

最初的软件质量保证系统是在 70 年代由欧洲首先采用的，其后在美国和世界其他地区也迅速地发展起来。目前，欧洲联合会积极促进软件质量的制度化，提出了如下 ISO9000 软件标准系列：ISO9001、ISO9000-3、ISO9004-2、ISO9004-4、ISO9002。这一系列现已成为全球的软件质量标准。除了 ISO9000 标准系列外，许多工业部门、国家和国际团体也颁布了特定环境中软件运行和维护的质量标准，如：IEEE 标准 729-1983、730-1984、Euro Norm EN45012 等。

ISO/IEC 15504 (SPICE)

CMM 的方法很快就引起了软件界的广泛关注，1991 年国际标准化组织采纳了一项动议，开展调查研究，在此后引发了一系列的研究工作，现已取得重要成果，产生了技术报告 ISO/IEC 15504《信息技术-软件过程评估》，预计于今年产生正式标准。从该技术报告的内容来看，其基本的目的和思路，均与 CMU/SEI 的 CMM 相似。

目前,学术界和工业界公认美国 Carnegie Mellon 大学软件工程研究所(CMU/SEI) 以 W.S.Humphrey 为首主持研究与开发的软件能力成熟度模型 CMM 是当前最好的软件过程,已成为业界事实上的软件过程的工业标准。

国内

学术界:中国生产力促进协会、北航 SEI、中科院研究 SEI 等科研机构已于近几年在北京、上海、广州和深圳等地先后举办过多次报告会和研讨会,组织过课程学习和应用实验,开展了软件过程方面的研究与开发工作,并发表了多篇的研究成果和学术论文,在软件质量保障平台支撑环境也取得了一定的成果。

产业界:近两年来,CMM 在我国获得了各界越来越多关注,业界有过多次关于 CMM 的讨论,国务院发布的《鼓励软件产业和集成电路产业发展的若干政策》对中国软件企业申请 CMM 认证给予了积极的支持,在第 17 条规定“对软件出口型企业 CMM 认证费用予以适当支持。”2000 年中国村电脑节上还有 CMM 专题论坛,吸引了众多业内人士。鼎新、东大阿尔派、联想、方正、金蝶、用友、浪潮、创智、华为、东大阿尔派等大型集团或企业等都是从 1997—2000 年起批企业都在进行研究、实验或实施预评估。其中鼎新公司从 1997 年着手进行 CMM 认证工作。1999 年 7 月通过第三方认证机构的 CMM2 认证。东大阿尔派公司于 2000 年 10 月通过第三方认证机构的 CMM2 认证。2001 年 1 月,联想软件经过英国路透集团的严格评估,顺利通过 CMM2 认证。

总体来讲,国内对软件过程理论的讨论与实践正在展开,目标是使软件的质量管理和控制达到国际先进水平,中国的软件产业获得可持续发展的能力。专家分析,在未来两三年内,国内软件业必将出现实施 CMM 的高潮。从这一趋势看,中国的软件企业已经开始走上标准化、规范化、国际化的发展道路,中国软件业已经面临一个整体突破的时代。

软件质量保障体系的实施

根据一直以来对国际上软件过程理论与实践的发展、尤其是近几年来着重在 CMM、PSP 和 TSP 以及 ISO 软件过程标准草案等方面的研究工作,国内专家学者建议,软件过程的改善应该从三方面着手进行:

- o 软件能力成熟度模型 CMM(Capability Maturity Model)
- o 个体软件过程 PSP(Personal Software Process)
- o 群组软件过程 TSP(Team Software Process)

三者各有侧重,但互为补充。

CMM

- o 迄今为止学术界和工业界公认的有关软件和管理实践的最好的软件过程。
- o 为评估软件组织的生产能力提供了标准。
- o 为提高软件组织的生产过程指明了方向。

CMM 软件过程成熟度模型概要*

1、比较

在介绍 CMM 内容之前,首先概述一下不成熟软件组织与成熟软件组织的差异。在不成熟的软件单位,软件过程一般由实践者及其管理者在项目进程中临时拼凑而成,因而推迟进度和超出预算已成为惯例,产品质量难以预测,有时为了满足进度要求,常在产品功能和质量上做出让步。

然而,一个成熟软件组织具有在全组织范围内管理软件、开发过程和维护过程的能力,规定的软件过程被正确无误地通知到所有员工,工作活动均按照已规划的过程进行。并通过可控的先导性试验和费效分析使这些过程得到改进,对已定义过程中的所有岗位及其职责都有清楚的描述,和通过文档与培训使全组织有关人员对其已定义的软件过程都有很好的理解,从而使其软件过程所导致的生产率和质量能随时间的推移得到改进。

表 1 给出了不成熟和成熟软件组织的比较,这种比较分析不仅是形成软件能力成熟模型的基础,也有利于理解该模型。

表 1 不成熟软件组织与成熟软件组织的比较

名称 内容	不成熟的软件组织	成熟的软件组织
软件过程	临时拼凑、不能贯彻	有统一标准，且切实可行，并不断改进；通过培训，全员理解，各司其职，纪律严明。
管理方式	反应式（消防式）	主动式，监控产品质量和顾客满意程度
进度、经费估计	无实际根据，硬件限定时，常在质量上作让步	有历史数据和客观依据，比较准确
质量管理	问题判断无基础，难预，进度滞后时，常减少或取消评审、测试等保证质量的活动	产品质量有保证，软件过程有纪律，有必要的支持性基础设施

2、CMM 的一些基本概念

（1）软件过程：人们用于开发和维护软件及其相关过程的一系列活动，包括软件工程活动和软件管理活动。

（2）软件过程能力：描述（开发组织或项目组）遵循其软件过程能够实现预期结果的程度，它既可对整个软件开发组织而言，也可对一个软件项目而言。

（3）软件过程性能：表示（开发组织或项目组）遵循其软件过程所得到的实际结果，软件过程性能描述的是已得到的实际结果，而软件过程能力则描述的是最可能的预期结果，它既可对整个软件开发组织而言，也可对一个特定项目而言。

（4）软件过程成熟：一个特定软件过程被明确和有效地定义，管理测量和控制的程度。

（5）软件能力成熟度等级：软件开发组织在走向成熟的途中几个具有明确定义的表示软件过程能力成熟度的平台。

（6）关键过程域：每个软件能力成熟度等级包含若干个对该成熟度等级至关重要的过程域，它们的实施对达到该成熟度等级的目标起到保证作用。这些过程域就称为该成熟度等级的关键过程域，反之有非关键过程域是指对达到相应软件成熟度等级的目标不起关键作用。归纳为：互相关联的若干软件实践活动和有关基础设施的一个集合。

（7）关键实践：对关键过程域的实践起关键作用的方针、规程、措施、活动以及相关基础设施的建立。关键实践一般只描述“做什么”而不强制规定“如何做”。整个软件过程的改进是基于许多小的、渐进的步骤，而不是通过一次革命性的创新来实现的，这些小的渐进步骤就是通过一些关键实践来实现。

（8）软件能力成熟度模型：随着软件组织定义、实施、测量、控制和改进其软件过程，软件组织的能力也伴随着这些阶段逐步前进，完成对软件组织进化阶段的描述模型。

3、CMM 模型概要

软件开发的风险之所以大，是由于软件过程能力低，其中最关键的问题在于软件开发组织不能很好地管理其软件过程，从而使一些好的开发方法和技术起不到预期的作用。而且项目的成功也是通过工作组的杰出努力，所以仅仅建立在可得到特定人员上的成功不能为全组织的生产和质量的长期提高打下基础，必须在建立有效的软件工程实践和管理实践的基础设施方面，坚持不懈地努力，才能不断改进，才能持续地成功。

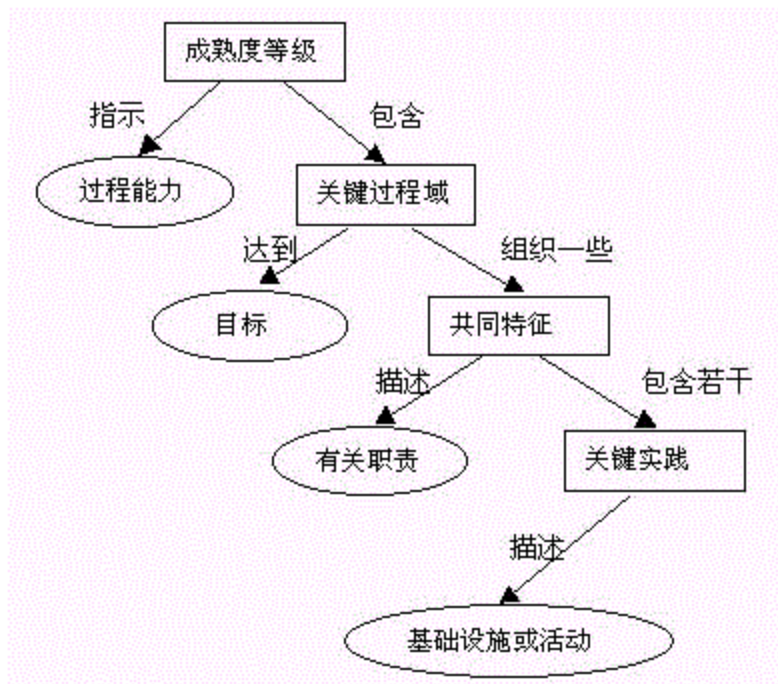
CMM 提供了一个框架，将软件过程改进的进化步骤组织成 5 个成熟等级，为过程不断改进奠定了循序渐进的基础。这 5 个成熟度等级定义了一个有序的尺度，用来测量一个组织的软件过程成熟和评价其软件过程能力，这些等级还能帮助组织自己对其改进工作排出优先次序。成熟度等级是已得到确切定义的，也是在向成熟软件组织前进途中的平台。每一个成熟度等级为连续改进提供一个台基。每一等级包含一组过程目标，通过实施相应的一组关键过程域达到这一组过程目标，当目标满足时，能使软件过程的一个重要成分稳定。每达到成熟框架的一个等级，就建立起软件过程的一个相应成分，导致组织能力一定程度的增大。

下面表 2 给出了 CMM 模型概要，表中的 5 个等级各有其不同的行为特征。要通过描述不同等级组织的行为特征：即一个组织为建立或改进软件过程所进行的活动，对每个项目所进行的活动和所产生的横跨各项目的过程能力。

表 2 CMM 模型概要

过程能力等级	特点	关键过程域
1 初始级	软件过程是无序的，有时甚至是混乱的，对过程几乎没有定义，成功取决于个人努力。管理是反应式（消防式）	
2 可重复级	建立了基本的项目管理过程来跟踪费用、进度和功能特性。制定了必要的过程纪律，能重复早先类似应用项目取得成功。	需求管理 软件项目计划 软件项目跟踪和监督 软件子合同管理 软件质量保证 软件配置管理
3 已定义级	已将软件管理和工程文档化、标准化，并综合成该组织的标准软件过程。所有项目均使用经批准、剪裁的标准软件过程来开发和维护软件。	组织过程定义 组织过程焦点 培训大纲 集成软件管理 软件产品工程 组织协调 同行专家评审
4 已定量管理级	收集对软件过程和产品质量的详细度量，对软件过程和产品都有定量的理解与控制	定量的过程管理 软件质量管理
5 优化级	过程的量化反馈和先进的新思想、新技术促进过程不断改进	缺陷预防 技术变更管理 过程变更管理

4、CMM 的结构



软件机构的最终质量保证模式可以用下图 1 说明，图 1 给出软件质量计划、质量控制、质量改进一个简单循环，其实，它归纳出 CMM 的真正内核，所以，可以说 CMM 的模型是一种新兴管理思想：连续改进（Continuous Improvement）循环的体现。

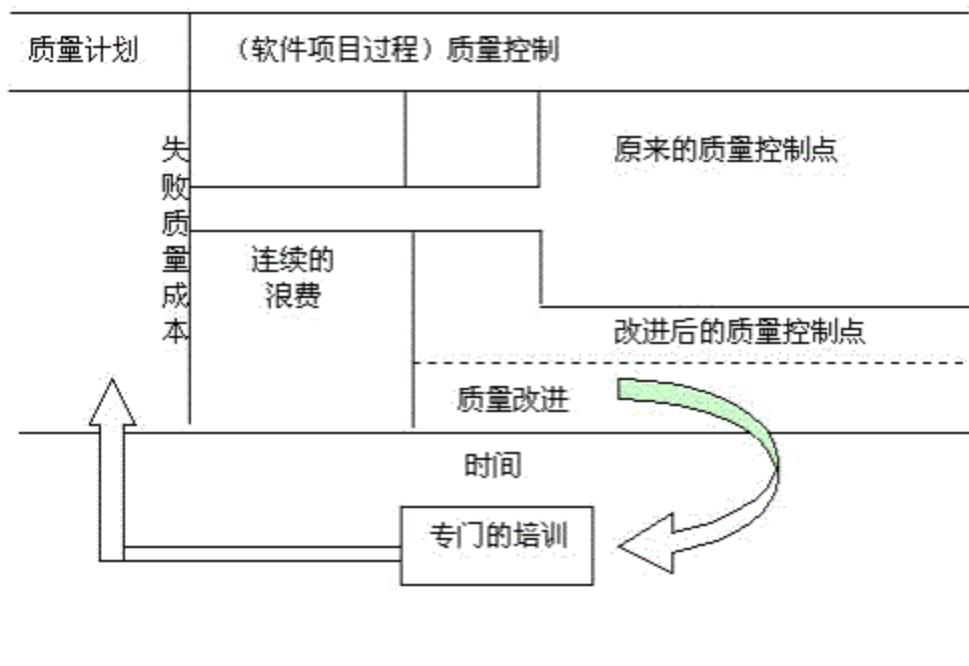


图 1

CMM 的作用

- n 科学地评价软件开发单位的软件能力成熟等级
- n 帮助软件开发单位进行自检，了解自己的强项和弱项，从而不断完善和改进单位的软件开发过程，确保软件质量，提高软件开发能效率。

CMM 实施的思考

根据 CMM 的基本原理、基本内容和基本方法，对 CMM 提出 4 个问题供大家思考：

1. 过程成熟度需要多长时间？多少费用？对企业有何好处？
2. 影响基于 CMM 的软件过程的成败因素是什么？
3. CMM 是否会导致过度官僚主义？是否会使组织变得更保守，不愿冒风险？

4. 有无合适的、易理解的框架（不仅仅是告诉“我们做什么”，而且告诉“我们怎么做”）可指导所有软件组织进行 CMM 改进？

这些针对 CMM 提出的问题与争论，国外进行了一些调查工作，但国内基本上没有这方面的专业调查和研究，以后再根据国内企业对 CMM 的认识、认证的增强和增多，这些问题会得到更科学的解答。

现给出国外针对上述问题的一些调查结果：

问题 1：成熟度提升一级建议安排 1 年到 2 年，费用问题国内外相距太远不好比较。对企业的好处问题给出下表说明：

表 1 CMM 的改进数据

类别	范围	平均
生产率提高/年	9%—67%	35%
上市时间（提早/年）	15%—23%	/
发布后的缺陷（减少/年）	10%—94%	39%
收益比	4.0:1~8.8:1	5.0:1

问题 2：影响过程改进失败的因素有：无法实施计划和跟踪、突发事件或危险造成、时间和资源限制造成、知道应该做什么而不知道如何做造成。

问题 3：大部分（84%-96%）不认为会使组织变成官僚主义机构、难于创新和不敢冒风险。

问题 4：这需要不断总结经验，提出办法。

在国内要想取得过程改进成功，作者认为：

- 1、软件过程改进必须有高级主管的支持与委托，并积极地管理过程改进的进展。
- 2、中层管理的支持很重要
- 3、责任分明，过程改进小组威望高
- 4、基层的支持与参与极端重要
- 5、如何利用定量的可观察数据，尽快使过程改进成果可见，从而激励参与者的兴趣
- 6、为企业的商业利益服务，并要求有成功的过程改进相符的企业文化变革

如果企业出现如下情况，过程改进肯定就失败：

- 1、高层领导机构态度不明确，见解不一致
- 2、各部门只管自己，互不通气，互不支持
- 3、对以前不成功的过程改进冷嘲热讽
- 4、项目成员认为软件过程改进会影响实际工作，而不支持软件过程改进活动

结论：CMM 不是万能的，它的成功与否，与一个组织内部有关人员的积极参与和创造性活动是密不可分的。

CMM 是对软件工程的工业实践所需的有关目标、方法和实践的最佳有效描述。问题是如何在一个实验室或者产业环境中做到 CMM 规则的应用？

CMM 是一个致力于组织过程改进的框架，问题是如何才能确保 CMM 使工作有效而且便利？

未提供有关实现关键过程域所需要的具体知识和技能。

因此，个体软件过程 PSP(Personal Software Process)也就应运而生。

PSP 概述

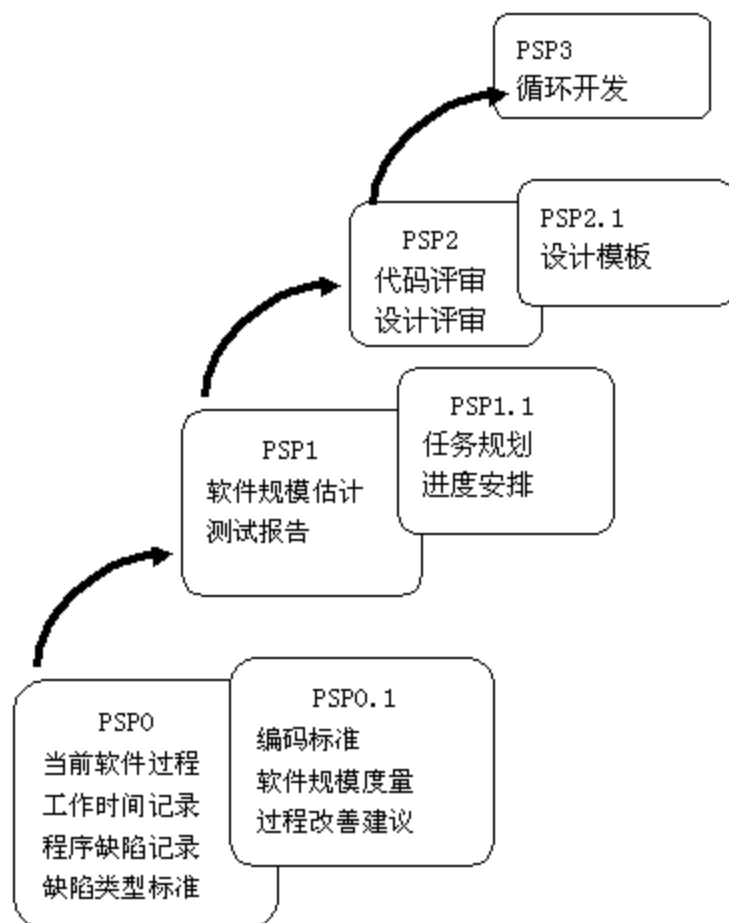
个体软件过程(Personal Software Process, PSP)是由美国 Carnegie Mellon 大学软件工程研究所(CMU/SEI)的 Watts s. Humphrey 领导开发的，于 1995 年它的推出，在软件工程界引起了极大的轰动，可以说是由定向软件工程走向定量软件工程的一个标志。PSP 是一种可用于控制、管理和改进个人工作方式的自我改善过程，是一个包括软件开发表格、指南和规程的结构化框架。PSP 为基于个体和小型群组软件过程的优化提供了具体而有效的途径，例如如何制订计划，如何控制质量，如何与其他人相互协作等等。在软件设计阶段，PSP 的着眼点在于软件缺陷的预防，其具体办法是强化设计结束准则，而不是设计方法的选择。根据对参加培训的 104 位软件人员的统计数据表明，在应用了 PSP 后，软件中总的差错减少了 58.0%，在测试阶段发现的差错减少了 71.0%，生产效率提高了 20.0%。PSP 的研究结果还表明，绝大多数软件缺陷是由于对问题的错误理解或简单的失误所造成的，只有很少一部分是由于技术问题而产生的。而且根据多年来的软件工程统计数据表明，如果在设计阶段注入一个差错，则这个差

错在编码阶段引发了 3—5 个新的缺陷，要修复这些缺陷所花的费用要比修复这个设计缺陷所花的费用多一个数量级。因此，PSP 保障软件产品质量的一个重要途径是提高设计质量。

个体软件过程 PSP 的现状

- o 从 1993 年开始，美国、欧洲、澳大利亚等地已先后有 20 多所大学开设了讲授 PSP 的课程。
- o 在工业界，PSP 也先后在 Motorola、HP、AIS 等公司推广使用。
- o 北航软件工程研究所于 1997 年开始，在北航计算机科学与工程系率先讲授了 PSP 课程，并组织了 PSP 应用实验。

个体软件过程 PSP 的演化*



个体软件过程 PSP 的内容

PSP 与具体的技术（程序设计语言、工具或者设计方法）相对独立，其原则能够应用到几乎任何的软件工程任务之中。PSP 能够：

- (1) 说明个体软件过程的原则；
- (2) 帮助软件工程师作出准确的计划；
- (3) 确定软件工程师为改善产品质量要采取的步骤；
- (4) 建立度量个体软件过程改善的基准；
- (5) 确定过程的改变对软件工程师能力的影响。

个体软件过程 PSP 支持环境

北航软件工程研究所在研制的基于 Internet 的“个体软件过程支持环境”，支持个体软件过程的定义、运作、度量、分析和优化，支持 PSP 在实际软件开发项目中的应用，支持 PSP 概念和方法的推广普及，支持软件工作人员软件工程方面素质的提高。

个体软件过程 PSP 的作用

- 1 使用自底向上的方法来改进过程，向每个软件工程师表明过程改进的原则，使他们能够明白如何有效地生产出高质量的软件。
- 1 为基于个体和小型群组软件过程的优化提供了具体而有效的途径。其研究与实践填补了 CMM 的空白。
- 1 帮助软件工程师在个人的基础上运用过程的原则，借助于 PSP 提供的一些度量和分析工具，了解自己的技能水平，控制和管理自己的工作方式，使自己日常工作的评估、计划和预测更加准确、更加有效，进而改进个人的工作表现，提高个人的工作质量和产量，积极而有效地参与高级管理人员和过程人员推动的组织范围的软件工程过程改进。

群组软件过程 TSP 概述

致力于开发高质量的产品，建立、管理和授权项目小组，并且指导他们如何在满足计划费用的前提下，在承诺的期限范围内，不断生产并交付高质量的产品。

TSP 指导项目组中的成员如何有效地规划和管理所面临的项目开发任务，并且告诉管理人员如何指导软件开发队伍。始终以最佳状态来完成工作。TSP 实施集体管理与自己管理自己相结合的原则，最终目的在于指导开发人员如何在最少的时间内，以预定的费用生产出高质量的软件产品，所采用的方法是对群组开发过程的定义、度量和改进。

实现 TSP 方法需要具备的条件

- o 需要有高层主管和各级经理的支持，以取得必要的资源
- o 整个软件开发小组至少应在 CMM 的第二级（可重复层）。
- o 全体软件开发人员必须经过 PSP 的培训，并有按 TSP 工作的愿望和热情。
- o 开发小组成员应在 2 到 20 个人之间。

在实施 TSP 的过程中，首先要有明确的目标，开发人员要努力完成已经接受的委托任务。在每一阶段开始，要做好工作计划。如果发现未能按期按质完成计划，应立即分析原因，以判定问题是由于工作内容不合适或工作计划不实际所引起，还是由于资源不足或主观努力不够所引起。开发小组一方面应随时追踪项目进展状态并进行定期汇报，另一方面应经常评审自己是否按 PSP 的原理工作。开发小组成员应按自己管理自己的原则管理软件过程，如发现过程不合适，应及时改进，以保证用高质量的过程来产生高质量的软件。项目开发小组则按集体管理的原则进行管理，全体成员都要参加和关心小组的规划、进展的追踪和决策的制定等工作。

按 TSP 原理对开发小组的基本度量要素

- o 所编文档的页数。
- o 所编代码的行数。
- o 花费在各开发阶段或各开发任务上的时间（以分为单位）。
- o 在各个开发阶段中引入和改正的差错数目。
- o 在各个阶段对最终产品增加的价值。

度量 TSP 实施质量的过程质量元素

- o 软件设计时间应大于软件实现时间。
- o 设计评审时间至少应占一半以上的设计时间。
- o 代码评审时间至少应占一半以上的代码编制时间。
- o 在编译阶段发现的差错不超过 10 个/KLOC
- o 在测试阶段发现的差错不超过 5 个/KLOC。

CMM、PSP 和 TSP 组成的软件过程框架

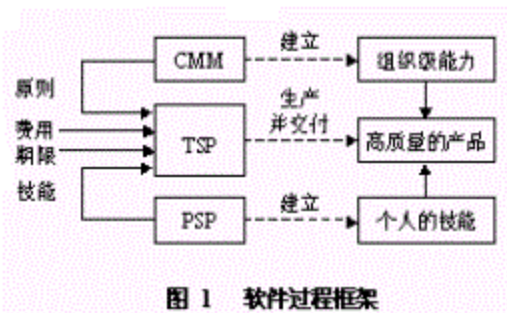


图 1 软件过程框架

1 CMM 是过程改善的第一步，它提供了评价组织的能力、识别优先改善需求和追踪改善进展的管理方式。企业只有开始 CMM 改善后，才能接受需要规划的事实，认识到质量的重要性，才能注重对员工经常进行培训，合理分配项目人员，并且建立起有效的项目小组。然而，它实现的成功与否与组织内部有关人员的积极参加和创造性活动密不可分。

1 PSP 能够指导软件工程师如何保证自己的工作质量，估计和规划自身的工作，度量和追踪个人的表现，管理自身的软件过程和产品质量。经过 PSP 学习和实践的正规训练，软件工程师们能够在他们参与的项目工作之中充分运用 PSP，从而有助于 CMM 目标的实现。

1 TSP 结合了 CMM 的管理方法和 PSP 的工程技能，通过告诉软件工程师如何将个体过程结合进小组软件过程，并将后者与组织进而整个管理系统相联系；通过告诉管理层如何支持和授权项目小组，坚持高质量的工作，并且依据数据进行项目的管理，向组织展示如何应用 CMM 的原则和 PSP 的技能去生产高质量的产品。

级别	CMM 的 18 个关键过程域	提供者
第 5 级	缺陷预防	PSP
	技术变更管理	PSP
	过程变更管理	PSP
第 4 级	定量的过程管理	PSP
	软件质量管理	PSP
第 3 级	组织过程焦点	PSP
	组织过程定义	PSP
	培训大纲	无
第 2 级	集成软件管理	PSP
	软件产品工程	PSP
	组织协调	TSP
	同行专家评审	PSP
	需求管理	TSP
	软件项目规划	PSP
	软件项目追踪和监控	PSP
	软件子合同管理	无
	软件质量保证	TSP
	软件配置管理	TSP

1 CMM/TSP/PSP 代表了目前国际上软件过程工程研究方面最先进的成果，它们对促进软件生产的科学化管理，提高软件 生产能力意义重大。

1 软件的生产过程及其它的许多子过程、软件的开发者和用户、以及系统的使用中存在着巨大的变化和不同，要使一个 软件过程对软件生产的改善真正有所帮助，其框架应是由 CMM、TSP 和 PSP 组成的一个完整体系，即从组织、群组和个人三个层次进行良好的软件工程和管理实践的指导和支撑。总而言之，单纯实施 CMM，永远不能真正做到能力成熟度 的升级，只有将实施 CMM 与实施 PSP 和 TSP 有机地结合起来，才能发挥最大的效力。

建议

过程改善工作必然具有一切过程所具有的固有特征，即需要循序渐进，不能一蹴而就而就需要持续改善，不能停滞不前；需要联系实际，不能照本宣读需要适应变革，不能凝固不变。我认为，将 CMM / PSP / TSP 引入软件企业最有效的途径首先要对单位主管和主要开发人员进行系统的培训。Carnegie Mellon 大学软件工程研究所曾尝试让软件工程师通过自学的方式来进行，但实际上，只有不到 20% 的人能够坚持到底。另外一个有效的途径是自顶向下的课程培训，即从高层主管依次普及到下面的工程师。

CMM 就如同软件生产的一面镜子，以之来衡量自己可以折射出企业发展的水平以及具体的差距。借鉴 CMM 的方法，改进软件研发项目管理，提高软件开发水平。软件研发项目管理是影响软件研发项目全局的因素，而技术只影响局部。建议以 CMM 为框架，先从 PSP 做起，然后在此基础上逐渐过渡到 TSP，以保证 CMM / PSP / TSP 确实在企业中生根开花。我相信只要坚持改善软件工程的管理，并在实践中总结适合自身的经验，一定能取得很好的效果。

不过强调一点，我们必须根据自身的实际制定可行的方案。不深入研究就照搬别的企业模式是很难起到提高软件产品质量水平的真正目的的。

九、系统分析员备考之 CMM 篇

引子

CMM 理论和知识是最近几年的热点，在最近两年的系统分析员上午试卷中都有一题考察 CMM 知识的，一般有 3-5 分的样子。估计未来的系统分析员考试还会有这方面的考题。即使不考，我们的系统分析员也应该掌握这方面的知识，因为将来从事

的系统分析与设计的工作也离不开 CMM 理论和知识，因为即使我们所在的公司不去进行 CMM 评估，CMM 理论知识对于我们不断的进行公司的软件过程改进有一定的借鉴意义，从而有助于软件质量的提高，进而提升公司产品的市场竞争力。

摘要

本文是根据这两年试题中涉及 CMM 知识而特为广大考友搜集整理的关于 CMM 的基础知识的文章。主要内容是有关 CMM 的基本概念、CMM 的基本框架和对 CMM 的正确态度等。希望这篇文章对你有所帮助，谢谢。

CMM (Capability Maturity Model, 软件能力成熟度模型) 是于 1984 年美国国会与美国主要的公司和研究中心合作创立的一个由联邦资助的非盈利组织——软件工程研究所 (Software Engineering Institute, SEI) 的一个早期研究成果。该模型提供了软件工程成果和管理方法的框架，自 90 年代提出以来，已在北美、欧洲和日本成功地应用。现在该模型已成为事实上的软件过程改进的工业标准。下面我们来一起学习有关 CMM 的一些基础知识。

一、CMM 基本概念

过程 (Process)：为实现既定目标的一系列操作步骤[IEEE-STD-610]。

软件过程 (Software Process)：指人们用于开发和维护软件及其相关产品的一系列活动的、方法、时间和革新。其中相关产品是指项目计划、设计文档、编码、测试和用户手册。当一个企业逐步走向成熟，软件过程的定义也会日趋完善，其企业内部的过程实施将更具有一致性。

软件过程能力 (Software Process Capability)：描述了在遵循一个软件过程后能够得到的预期结果的界限范围。该指标是对能力的一种衡量，用它可以预测一个组织（企业）在承接下一个软件项目时，所能期望得到的最可能的结果。

软件过程性能 (Software Process Performance)：表示遵循一个软件过程后所得到的实际结果。（与软件过程能力有区别，软件过程能力关注的是实际得到的结果，而软件过程性能关注的是期望得到的结果。由于项目要求和客观环境的差异，软件过程性能不可能充分反应软件过程整体能力，即软件过程能力受限于它的环境。）

软件过程成熟度 (Software Process Maturity)：是指一个具体的软件过程被明确地定义、管理、评价、控制和产生实效的程度。所谓成熟度包含着能力的一种增长潜力，同时也表明了组织（企业）实施软件过程的实际水平。随着组织软件过程成熟度能力的不断提高，组织内部通过对过程的规范化和对成员的技术培训，软件过程也将会被他的使用者关注和不断修改完善。从而使软件的质量、生产率和生产周期的到改善。

CMM 是软件过程能力成熟度模型 (Capacity Maturity Model) 的简称，是卡内基—梅隆大学软件工程研究院为了满足美国联邦政府评估软件供应商能力的要求，于 1986 年开始研究的模型，并于 1991 年正式推出了 CMM 1.0 版。CMM 自问世以来备受关注，在一些发达国家和地区得到了广泛应用，成为衡量软件公司软件开发管理水平的重要参考因素和软件过程改进事实上的工业标准。

CMMI (Capability Maturity Model Integration) 即能力成熟度模型集成，这也是美国国防部的一个设想，他们想把现在所有的以及将被发展出来的各种能力成熟度模型，集成到一个框架中去。这个框架有两个功能，第一，软件获取方法的改革；第二，建立一种从集成产品与过程发展的角度出发、包含健全的系统开发原则的过程改进。

关键过程 (区) 域 (Key Process Area) 是指一系列相互关联的操作活动，这些活动反映了一个软件组织改进软件过程时所必须满足的条件。也就是说，关键过程域标识了达到某个成熟程度级别时所必须满足的条件。在 CMM 中一共有 18 个关键过程域，分布在第二至五级中。

关键实践 (Key Practices)：是指关键过程域种的一些主要实践活动。每个关键过程域最终由关键实践所组成，通过实现这些关键实践达到关键过程域的目标。一般情况下，关键实践描述了该“做什么”，但没有规定“如何”去达到这些目标。

软件过程评估（Software Process Assessment）是用来判断一个组织当前所涉及的软件过程的能力状态，判断下一个组织所面向得更高层次上的与软件过程相关的课题，以及利用组织的鼎力支持来对该组织的软件过程进行有效的改进。

软件能力评价是（Software Capability Appraisal）用来判断有意承担某个软件项目的软件组织的软件过程能力，或是判断已进行的软件过程所处的状态是否正确或是否正常。

软件工程组（Software Engineering Group）：负责一个项目的软件开发和维护活动的团体。活动包括需求分析、设计、编码和测试等。

软件相关组（Software Related Groups）：代表一种软件工程科目的团体，它支持但不直接负责软件开发或维护工作，如软件质量保证组、软件配置管理组合软件工程过程组等等。在 CMM 的关键实践中，软件相关组通常应该根据关键过程域和组织的上下文来理解。

软件工程过程组（Software Engineering Process Group）：是由专家组成的组，他们推进组织采用的软件过程的定义、维护和改进工作。在关键实践中，这个组织通常指“负责组织软件过程活动的组”。

系统工程组（System Engineering Group）：是负责下列工作的个人的团体：分析系统需求；将系统需求分配给硬件、软件和其他成分；规定硬件、软件和其他成分的界面；监控这些成分的设计和开发以保证它们符合其规格说明。

系统测试组（System Test Group）：是一些负责策划和完成独立的软件系统测试的团体，测试的目的是为了确定软件产品是否满足对它的需求。

软件质量保证组（Software Quality Assurance Group）：是一些计划和实施项目的质量保证的团体，其工作目的是保证软件过程的步骤和标准是否得到遵守。

软件配置管理组（Software Configuration Management Group）：是一些负责策划、协调和实施软件项目的正式配置活动的团体。

培训组（Training Group）：是一些负责协调和安排组织培训活动的团体。通常这个组织负责准备和讲授大多数培训课程并协调其他培训方式的使用。

二、CMM 的基本框架

任何一个软件的开发、维护和软件组织的发展离不开软件过程，而软件过程经历了不成熟到成熟、不完善到完善的发展过程。它不是一朝一夕就能成功的，需要持续不断的对软件过程进行改进，才能取得最终的成效。CMM 就是根据这一指导思想设计出来的。该模型为了正确和有序地引导软件过程活动的开展，建立一个能够有效地描述和表示的软件过程的改进框架，使其能够对各阶段软件过程的任务和管理起指导作用。该模型一产品质量的概念和软件工程的经验教训为基础，指导企业如何控制开发、维护软件的生产过程和如何制定一套与之相适应的软件过程及管理体系。

（一）分级标准

CMM 模型描述和分析了软件过程能力的发展程度，确立了一个软件过程成熟程度的分级标准，如图 1 示。一方面软件组织利用它可以评估自己当前的过程成熟度，并以此提出严格的软件质量标准和过程改进的方法和策略，通过不断的努力去达到更高的成熟程度。另一方面，该标准也可以作为用户对软件组织的一种评价标准，使之在选择软件开发商时不再是盲目的和无把握的。

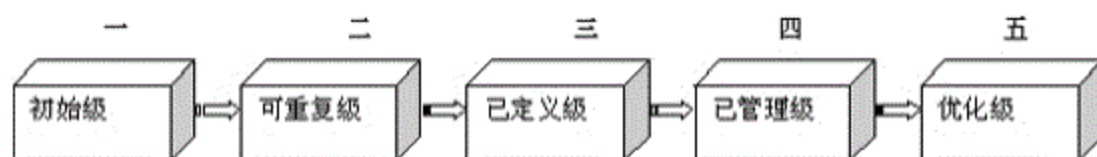


图 1 软件过程成熟度的级别

CMM 的分级结构可以描述为：

①、初始级：软件过程的特点是无秩序的，有时甚至是混乱的。软件过程定义几乎处于无章法和步骤可循的状态，软件产品所取得的成功往往依赖于极个别人的努力和机遇。

②、可重复级：已建立了基本的项目管理过程，可用于对成本、进度和功能特性进行跟踪。对类似的应用项目，有章可循并能重复以往所取得的成功。

③、已定义级：用于管理的和工程的软件过程均已文档化、标准化，并形成了整个软件组织的标准软件过程。全部项目均采用与实际情况相吻合的、适当修改后的标准软件过程来进行操作。

④、以管理级：软件过程和产品质量有详细的度量标准。软件过程和产品质量得到了定量的认识和控制。

⑤、优化级：通过对来自过程、新概念和新技术等方面的各种有用信息的定量分析，能够不断地、持续地对促进过程进行改进。

除第一级外，每一级都设定了一组目标，如果达到了这组目标，则表明达到了这个成熟级别，自然可以向下一级别迈进。CMM 体系不主张跨级别的进化。因为从第二级开始，每一个低级别的实现均是高级别实现的基础。

（二）CMM 的主要内容

CMM 为软件企业的过程能力提供了一个阶梯式的进化框架，它采用分层的方式来解释其组成部分，如图 2 示。在第二至第五个成熟等级中，每个等级包含一个内部结构的概念，关于内部结构详细描述将在下面 CMM 内部结构的一栏中进行。

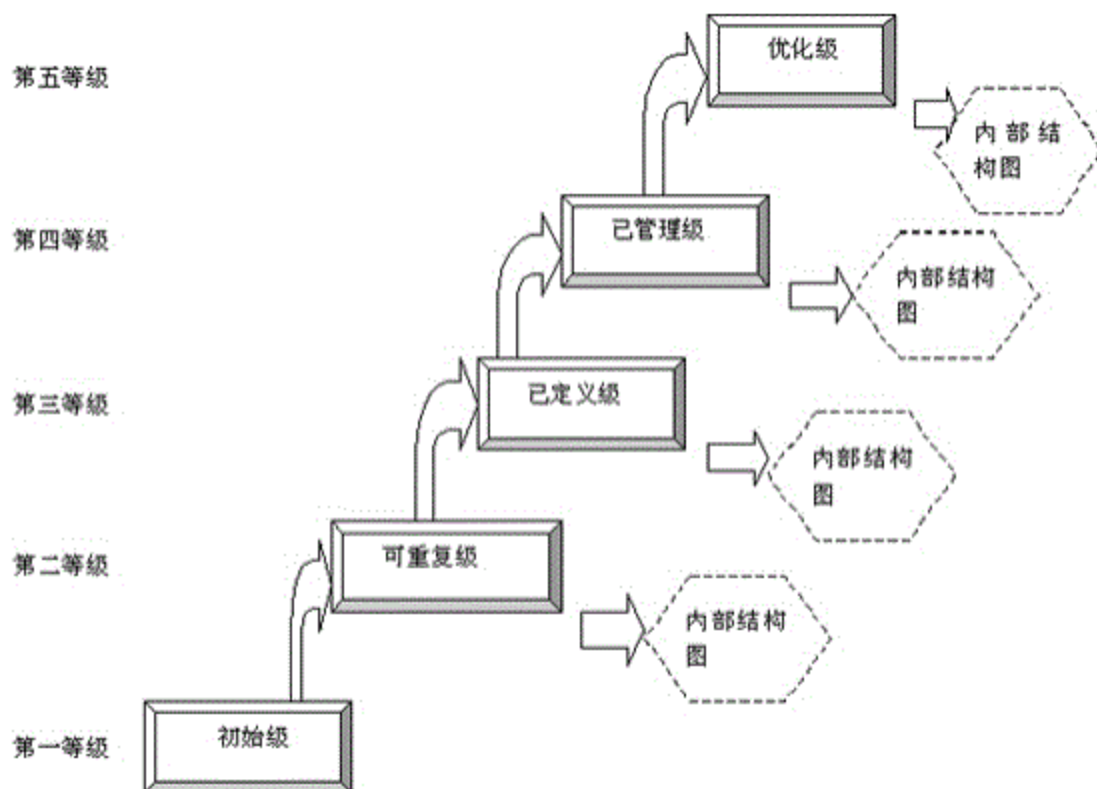


图 2 CMM 的五个成熟等级

每一级向上一级迈进的过程中都有其特定的改进计划，具体情况如下。

初始级的改进方向是：建立项目过程管理，是使规范化管理，保障项目的承诺；艳进行需求管理方面的工作，建立用户域软件项目之间的沟通，使项目真正反映用户的需求；建立各种软件项目几乎，如软件开发计划、软件质量保证计划、软件配置管理计划、软件测试计划、风险管理计划及过车改进计划等；积极开展软件质量保证活动（SQA）。

可重复级的改进方向是：不再按项目制定软件过程，而是总结各种项目的成功经验，使之规则化，把具体经验归纳为权组织的标准软件过程，把改进软件组织的整体软件过程能力的软件过程活动，作为软件开发组织的责任；确定全组织的标准软件过程，把软件工程及管理活动集成到一个稳固确定的软件过程中，从而可以跨项目改进软件过程效果，也可以作为软件过程剪裁的基础；建立软件工程过程小组（SPEG）长期承担评估域调整软件过程的任务，以适应未来软件项目的要求；积累数据，建立组织的软件过程库及软件过程相关的文档；加强培训。

已定义级的改进方向是：着手软件过程的定量分析，已达到定量地控制软件项目过程的效果；通过软件的质量管理达到软件质量的目标。

已管理级的改进方向是：防范缺陷，不仅在发现了问题能及时改进，而且应采取特定行动防止将来出现这类缺陷；主动进行技术改革管理、标识、选择和评价新技术，是有效的新技术能在开发组织中实施；进行过程变更管理，定义过程改进的目的，经常不断地进行过程改进。

优化级的改进目方向是：保持持续不断的软件过程改进。

（三）CMM 的内部结构

CMM 为软件过程能力的提高提供了一条改进的途径。CMM 由 5 个成熟度等级组成，每个成熟度等级有着各自的功能。除第一级外，CMM 的每一级按完全相同的内部结构构成的，如图 3。成熟度等级为顶层，不同的成熟度等级反映了软件组织的软件过程能力和该组织可能实现预期结果的程度。

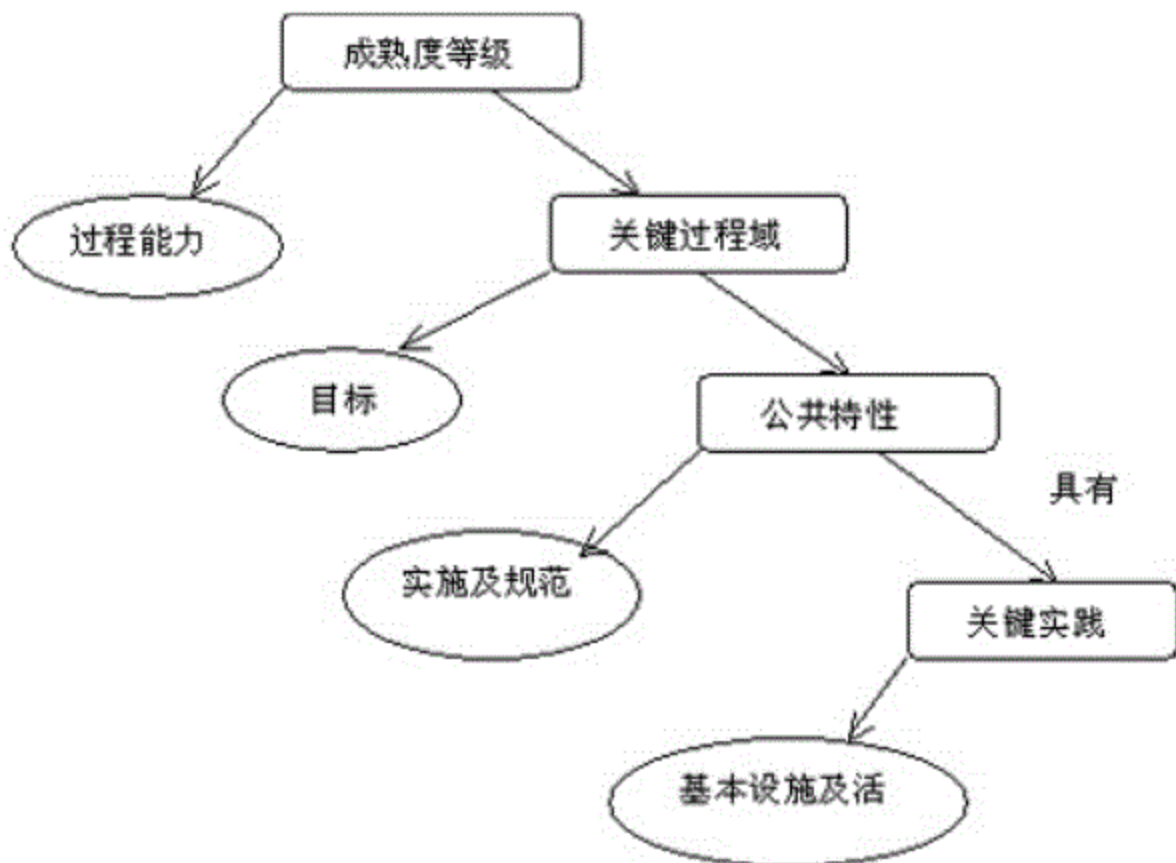


图3 CMM 的内部结构图

在 CMM 中，每个成熟度等级（第一级除外）规定了不同的关键过程域，一个软件组织如果希望达到某一个成熟度级别，就必须完全满足关键过程域所规定的要求，即满足关键古城域的目标。每一级的关键过程域的详细情况见表 1。

过程分类 等级	管理方面	组织方面	工程方面
优化级		技术改造管理 过程改造管理	缺陷防范
可管理级	定量完成过程		软件质量管理
已定义级	集成软件管理 组间协调	组织过程焦点 组织过程定义 培训程序	软件产品工程 同级评审
可重复级	需求管理 软件项目计划 软件项目跟踪域监控 软件转包合同管理 软件质量保证 软件配置管理		
初始级	无序过程		

表 1 关键过程域的分类

（四）软件过程评估和软件能力评价

软件过程评估所针对的是软件组织自身内部软件过程的改进问题，目的在于找出缺陷，提出改进方向。评估组以 CMM 模型为指引调查、鉴别软件过程中的问题，翻过来将这些问题与 CMM 关键实践活动所提出的指导一起用于确定组织的软件过程改进策略。

软件能力评价是对接受评价者在一定条件下、规定时间内能否完成特定项目的能力考核，即承担风险的系数大小。评价包括承包者是否有能力按计划开发软件产品，是否能按预算完成等。通过利用 CMM 模型确定评价结果后，就可以利用这些结果确定选择某一承包商的风险。也可以用来判断承包者的工作进程，推动他们改进软件过程。

CMM 为评估和评价提供了一个参考框架，指出了在评估和评价中通常采用的佛农步骤，如图 4 示。

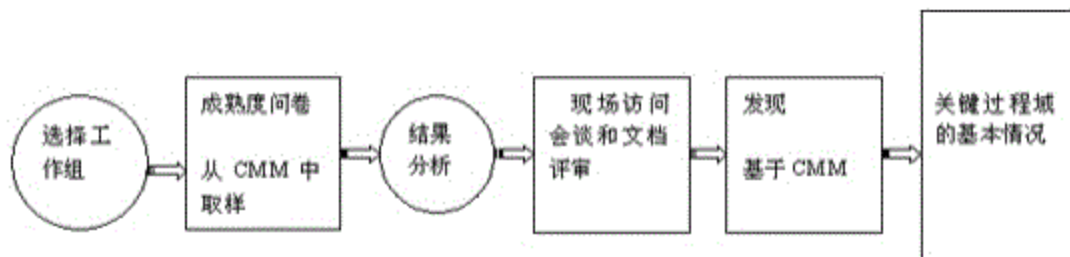


图 4 软件过程评估和软件能力评价的步骤

具体来说，评估过程是：选择一个工作组；完成问卷调查和取样工作；结果分析；现场访问；与 CMM 模型对照分析；依据关键过程域的基本情况列出评估提纲。以上步骤在软件过程评估和软件能力评价题勾勒很有参考价值的方法，但在具体操作时以下这些特点也值得考虑：

- ①、在现场访问和考察中，充分运用成熟度问卷和结果分析为依据。
- ②、以 CMM 模型作为现场调查的路线图。
- ③、利用 CMM 中的关键过程域定义软件过程中的优点和缺陷，从中发现差异。
- ④、对关键过程域目标是否备满足的实际情况出发，分析满意程度，写出书面报告。

尽管软件过程评估和软件能力评价有很多相似之处，但由于其目的和结果的不同，它们之间的差异也是必然存在的，如：

①、软件过程评估和软件能力评价在出发点和目标上的不同，使得会谈目的、调查范围、收集的信息和输出的表示方式上有着本质的不同。尤其在一些细节规范方面，评估和评价的方法有很大差异。

②、软件过程评估和软件能力评价的结果和结果所起的作用不同。因为两者的侧重点不一样，即使是对同一个应用项目，运用相同的方法，也不会得出相同的结果。

③、被评估和评价单位的态度对评估和评价活动的影响。评估在某种意义上被评估单位的态度较积极，而评价在某种意义上被评价单位的态度可能比较慎重。软件过程评估是在一个开放的、互相协作的环境中进行的，而软件能力评价往往是在有较大的阻力的环境中进行的。

（五）CMM 的组织保证

当人们面对 CMM 实施时，首先想到的就是人员的构成和各种小组的划分。它是实施 CMM 的组织保证，是一切活动的基础。CMM 在制定软件过程实施中本着尽量不和具体的组织机构和组织形式相联系的原则，为的是提供一个独立于具体企业而又有广泛指导意义的模型框架。但在实施各种软件关键实践中，不可避免地要涉及到角色和组织结构。所以为了使 CMM 能够使用

域各种级别和各种规模的企业，SEI 提出了一个相对抽象的组织结构，它与组织、项目、人员（角色）相关联，具有自己特定的术语，而且可能不同于其他组织所用的名词。例如基本概念中提到的主要的软件工作组的概念。

三、正确的态度看待 CMM

SEI 的 CMM 并不是软件开发的方法学，也不是产品模板，更不是过程法律。CMM 是过程改进的途径，是一套指南，帮助你通过持续的重复、测量和提炼，稳步创造与净化开发环境。CMM 的假定是：如果你实施一个不断重复、测量和提炼的大纲，作为环境改进的副产物，质量便会自然的提高。不要把 CMM 设想为一套规则，而应将它理解为一个学科，做事的一般方法。在这套指南下运作，你会发现这里有着广阔的空间，让你剪裁和塑造自己的大纲，以适应组织的特定要求。

CMM 不采用“用这种方法做这类事”的风格，它也不对由问题的 IT 组织提供快速的纠正方案。CMM 是一个指南针，指导你如何逃离暴风雪。CMM 是一个大纲，要求你对整个 IT 组织的有关部分，从高层领导到软件生产的第一次线工作者，都做出坚定的、长期的实施承诺。成熟的过程不可能在已也之间实现。

在如何解释 CMM 建议时，它允许极大的灵活性。CMM 意识到，IT 组织之间存在着很大的差别。他们的客户不同，使用的工具不同，人员智力和专业背景不同，从事的项目属于不同的类型，规模大小不同，要求也各不相同。因而，他们应当以自己的方式走向成熟。在一处活用的东西，在另一处未必适用。这一点非常重要，中国部分软件公司的前车之鉴也从某种程度上给了我们建议和经验教训，那就是，要灵活应用 CMM，不要幻想一夜就有成效。

小结

本文只是根据这两年的试题和自己的预测向广大系分考友提供一些 CMM 方面的知识。CMM 不是重点，但也有可能会考到一些知识，如基本概念等。在搜集资料 and 整理着篇文章时，遇到了一个矛盾，那就是：我要提供足够的资料以使读者不必花费金钱再去买一本书就可以复习有关 CMM 的知识，而同时又不能放太多的内容使读者浪费太多的时间在这上面。最后采取了一个折衷的办法，那就是尽量满足考试需求的情况下减少篇幅。在此声明，本文所涉及的内容只是本人的预测，并不是说考试范围不会超过本文的内容。所以有时间的朋友还是尽可能的扩大这方面知识面。希望这篇文章对你有帮助，谢谢。