

软件过程管理

如何加强软件过程管理，提高软件过程管理能力，做一个合格的软件项目经理？

****摘要：****软件过程是指用于软件开发以及维护的一系列活动、方法及实践、目的、资源、约束控制等一定的前提条件，多个活动组成的偏序集。工程化软件开发需要对软件开发整个过程进行有效的组织和管理，由此产生了一系列软件开发组织和管理方法，其主要目的是形成一种载体，用以积累和传递关于软件开发的经验教训。然而，由于软件开发的一些天然特性（比如复杂性和不可见性）的存在，使得描述软件开发过程的软件开发与组织方法也天然地带着一定的抽象性。由此带来了很多概念上的误导和实践中的争论，影响了上述目的的达成。软件过程决定了软件质量，高效率、高质量和低成本地开发软件，必须对于软件过程进行改善，对于软件过程管理进行加强。在此过程中，作为软件项目管理的负责人，软件项目成功的一切因素都在项目经理的工作范围内。软件项目经理参与整个项目开发的过程，利用他们的领导技能、技术知识和经验来指导的团队，制定可靠的执行计划以满足客户的需求。本文中，梳理概括了软件过程和软件过程管理若干核心概念，介绍软件组织与管理方法，使用PSP、TSP和CMM所提供的集成化、三维的软件改革框架。定义了软件中的项目经理，解释了该角色的典型工作要求，并提供了相关提高软件过程能力的方法。

****关键词：****软件工程；软件过程；软件项目管理；软件过程管理；CMM；PSP；TSP；RUP

Abstract: Software process refers to a series of activities, methods and practices, objectives, resources, constraint control and other certain preconditions used for software development and maintenance, and a partial order set composed of multiple activities. Engineering software development requires effective organization and management of the entire process of software development, resulting in a series of software development organization and management methods, the main purpose of which is to form a carrier to accumulate and transfer experience and lessons about software development. However, due to the existence of some natural characteristics of software development (such as complexity and invisibility), the software development and organization methods that describe the software development process are naturally abstract. This has brought about a lot of conceptual misleading and practical disputes, which affected the achievement of the above-mentioned goals. The software process determines the software quality. To develop software with high efficiency, high quality and low cost, the software process must be improved and the software process management must be strengthened. In this process, as the person in charge of software project management, all factors for the success of a software project are within the scope of the project manager's work. Software project managers participate in the entire project development process, and use their leadership skills, technical knowledge and experience to guide the team to formulate a reliable execution plan to meet customer needs. In this article, I summarized several core concepts of software process and software process management, introduced software organization and management methods, and used an integrated, three-dimensional software reform framework provided by PSP, TSP, and CMM. Defined the project manager in the software, explained the typical work requirements of the role, and provided relevant methods to improve the software process capability.

****Keywords:**** software engineering; software process; software project management; software process management; CMM; PSP; TSP; RUP

1 软件过程相关概念介绍

1.1 软件项目管理

软件项目管理被称为规划和带领项目团队的艺术和科学。顾名思义，其管理的对象是各类软件项目。具体而言，软件项目管理是应用方法、工具、技术以及人员能力来完成软件项目，实现项目目标的过程。整个软件开发过程中的目标识别、状态跟踪以及偏差纠正是软件项目管理的三大核心要素。完整的软件项目管理包含很多内容，例如成本和工作量的估算、计划和进度跟踪调整以及风险分析与控制等，都属于软件项目管理的范畴 (可以参考第 6 版 PMBOK 来了解项目管理知识领域)。尽管抽象概念上具有相似性 (例如，所有管理活动都应该包含目标识别、状态跟踪以及偏差纠正这三大核心要素)，软件项目管理的具体方法和实践会因为项目特征的差异呈现出不同的特点。软件项目管理需要借鉴一些本领域或者其他领域的经验教训，由此产生了一些用来描述这些经验和教训的概念，例如软件过程、生命周期模型等。下面分别加以介绍。

1.2 过程和软件过程

过程是指用于软件开发以及维护的一系列活动、方法及实践、目的、资源、约束控制等一定的前提条件，多个活动组成的偏序集。

软件过程是指软件生存周期所涉及的一系列相关软件工程活动的集合，其各个阶段实现了软件的需求定义与分析、设计、实现、测试、交付与维护。软件过程是在开发与构建系统时应遵循的步骤，过程由活动组成，活动则是由各阶段的任务组成，任务则起着将每一阶段的输入进行加工处理然后输出的作用。活动的执行可以是顺序的、重复的、并行的、嵌套的或者有条件地引发的。

好的软件项目管理离不开合理的软件过程，讨论过程改进的相关概念时，技术 (包括工具) 革新、人员培训以及流程优化等均属于软件过程改进的内容。因此，我们认为软件过程也应该包括上述 3 个方面在内。为了以示区别，我们可以将软件过程区分为狭义的过程和广义的过程：狭义的过程就是前文描述的一组有先后顺序的实践；广义的软件过程则应该包括技术、人员以及狭义过程这 3 部分，如下图所示。过程的作用则不仅仅是这个三角形中的一极，它更加关键的作用是连接技术和人员的粘合剂。只有将技术、人员以及狭义过程三者融合成一个整体，才是一个真正可以指导工作的软件过程。在某些文献中，会使用软件开发方法 或者软件开发过程等术语，其含义类似于狭义的软件过程。从这个意义上说，诸如净室 Cleanroom 方法、极限编程 XP (extreme programming) 方法、SCRUM方法、DSDM (dynamic systems development method) 方法、FDD (feature driven development) 方法、Gate 方法 等均为软件 (开发) 过程；而更一般地，敏捷软件过程 / 方法、轻量型过程 / 方法以及重型过程 / 方法等描述也是恰当的。一个软件过程既可以覆盖从需求到交付的完整生命周期，也可以仅仅包括某些特定开发阶段。例如，一个实现 (implementation) 过程就可能包括详细设计、编码、代码评审、编译以及单元测试等更为细小的开发步骤。甚至，为了确保代码评审工作的完成质量，我们也可以进一步定义、管理和改进代码评审过程。



图1 广义软件过程包括技术、人员以及狭义过程

软件过程可以分为软件工程过程、软件管理过程和软件支持过程三大类。其中工程过程指软件开发和生产的过 程，如需求分析、设计、编码、测试等过程；管理过程指对软件开发和生产过程进行管理的过程，如项目策划过程、跟踪监控过程、质量保证过程、配置管理过程等等；支持过程指对有效软件开发和生产进行支持的过程，如评审过程、培训过程、度量过程等等，这些过程不是相互孤立、彼此隔离的，它们都关系到软件产品的质量，必须将其科学、系统地组织成一个有效的运作体系，才能使组织所有与质量相关的活动有条不紊、高效地运行。

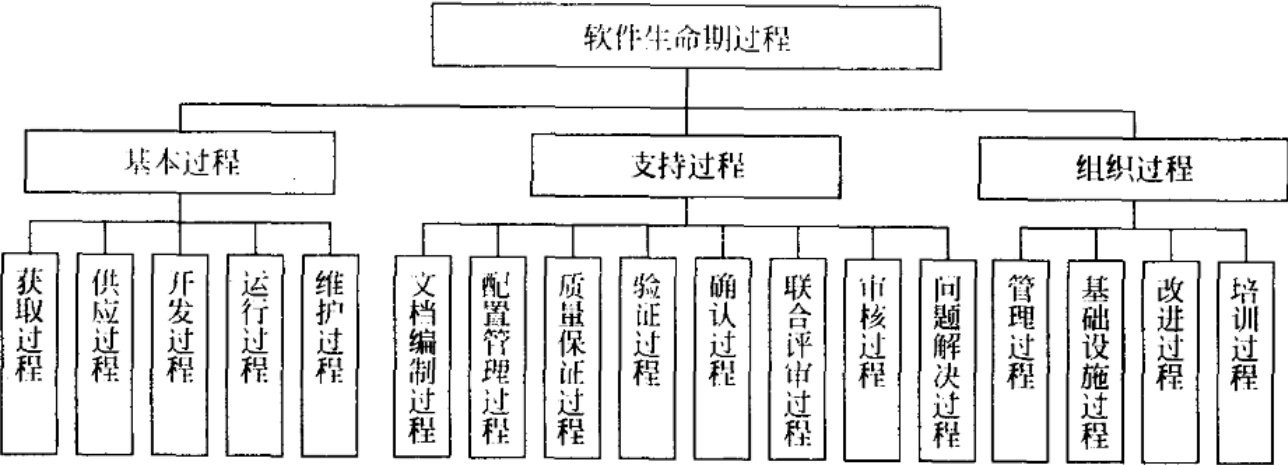


图2 IEC 12207软件生命周期过程标准

1.3 软件过程管理

软件过程管理是在软件生存周期内对质量、时间、费用三方面进行量化，通过质量管理、风险管理、工程管理和费用管理活动对这三方面进行分析和评价。一个有效的、可视的软件过程能够将人力资源、流程和实施方法结合成一个有机的整体，并能全面地展现软件过程的实际状态和性能，从而可以监督和控制软件过程的实现。对软件过程的监督、控制，实际孕育着一个管理的过程，这种管理的直接目的是为了让软件过程在开发效率、质量等方面有着更好的性能绩效。

如果将软件项目管理视作传统行业的产品生产管理的话，软件过程管理则应该是对生产该产品的流水线的设计、建设、维护、优化以及升级改造。软件过程管理一般包括了软件过程的建立、执行、监控、评估以及改进等活动。为了更好地开展软件过程管理，我们同样需要积累相关活动的经验教训，形成了若干可以参考的模型和方法，这其中最著名的软件过程管理参考模型之一可能就是能力成熟度模型 CMMI以及其后续的集成模型 CMMI。

1.4 软件过程模型

过程定义了方法使用的顺序、要求交付的文档、为保证质量和协调变化需要的管理每个开发组织都可以针对不同类型的软件项目，选取不同的工程过程，而且过程选取得是否合适，直接影响软件开发的效率与质量。过程模型的选择与组实施是需要开发经验做为基础的，主要有以下一些主要过程模型。

- (1) 编码-修正模型：只能应付于小的软件
- (2) 瀑布模型：比较适合需求相对稳定而又便于定义、开发有较大把握的系统
- (3) 演化模型：这是工程实践中大量使用的过程模型之一，又称快速原型法
- (4) 螺旋模型：将瀑布模型与演化模型结合起来，并且加入两种模型均略的风险分析，成为大型、高风险且复杂的系统常用的开发过程模型
- (5) 喷泉模型：这是来用面向对象方法开发系统的常用过程模型，其核心思想表现为阶段成果的高度复用和迭代，各阶段是连续的、无缝隙的而且又是相交的。因此，采用面向对象的开发不强调阶段的划分，各种活动可以灵活地迭代和交替
- (6) 迭代增量式开发模型：这种方法不是在项目结时一次性提交软件，而是分块逐次开发和提交。构造阶段由多次开发组成，每次开发都包含编码、测试和集成，所得产品满足项目需求的某一个子集，或提交给早期用

户，或纯粹是内部提交。每次迭代都包含了软件生命期的所有阶段

IS9000 与 CMM 模型（即软件组织能力成熟度模型）是国际上公认的软件质量保证的过程及其改进标准，它们肯定了软件过程对于软件质量的影响程度，好的过程意味着高质量的软件，而且好的软件过程还意味着用低的开销来生产高质量的软件

2 提高软件过程管理的方式

软件过程管理可照一些成熟的过程模型来进行，本文描述CMM过程模型和PSP/TSP软件过程结合的方式和基于RUP来提高软件过程管理的方式。

2.1 CMM、PSP、TSP结合进行软件过程管理

2.1.1 能力成熟度模型（CMM）

CMM(Capability Maturity Model for Software)中文译为“软件过程能力成熟度模型”。最早为美国 卡内基·梅隆大学的软件工程研究院(SEI)为满足美国联邦政府评估软件供应商能力的要求而开发的，之后SEI又结合实战经验不断对CMM进行完善。

SEI将CMM定义为：对于软件组织在定义、实现、度量、控制和改善其软件过程中各个发展阶段的描述。这个模型便于确定软件组织的现有过程能力和查找软件质量及过程改进方面最关键的问题，从而为选择过程改进战略提供指南。

CMM除了包括有效开发软件的作业程序外，还制订了五个循序渐进的质量等级(CMM1—CMM5)，分别为：初始级、可重复级、已定义级、已管理级和优化级，如图3所示。其中，CMM5是CMM认证的最高标准，可有效地帮助企业改进和优化管理，大大提高软件企业的开发水平和产品质量。

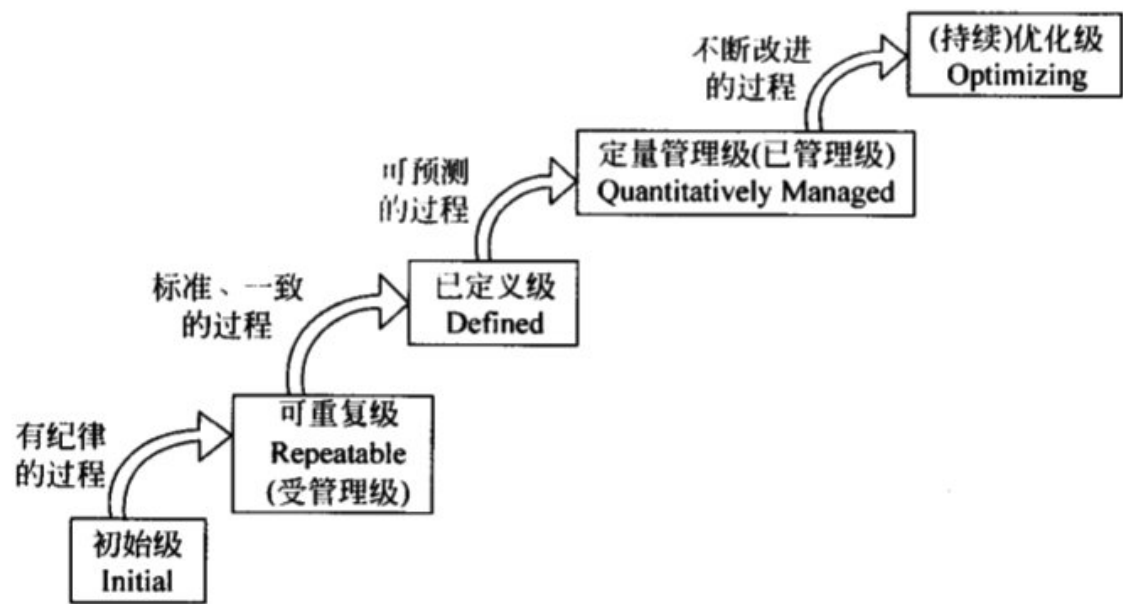


图3 CMM/CMMI 成熟度的5个等级

2.2 个体软件过程（PSP）

PSP (Personal Software Process) 是一种可用于控制、管理和改进个人工作方式的自我持续改进过程，是一个包括软件开发表格、指南和规程的结构化框架。美国Carnegie Mellon大学软件工程研究所 (CMU / SEI)的Watts S. Humphrey 带领团队于1995年开发并推出个体软件过程，在软件工程界引起了极大的轰动，是软件开发过程改进的里程碑，是由定向软件工程走向定量软件工程的一个标志。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能够说明个体软件过程的原则；帮助软件工程师做出准确的计划；确定软件工程师为改善产品质量要采取的步骤；建立度量个体软件过程改善的基准；确定过程的变化对软件工程师能力的影响。

个体软件过程注重个人的技能，能够指导软件工程师为保证自己的工作质量而采取相应措施，估计和规划自身的工作，度量和追踪个人的工作表现，管理自身的软件过程和软件产品质量。通过学习 PSP 的正规训练及实践，软件工程师们能够在他们参与的项目工作之中充分利用 PSP，从而保证了项目整体的进度和质量。

基于 CMM 中发现的实践，PSP 可以指导工程师，帮助他建立开发软件的一套结构化和规范的方法，在软件项目开发的总体成本中往往人员成本占 70%，所以工程师的技能与工作习惯很大程度决定了软件开发过程的结果。当软件开发人员在项目中面临各种各样的实际问题并寻求有效的解决问题方案时，就会更深刻地体会到 PSP 的作用。

2.1.3 群体软件过程 (TSP)

TSP (Team Software Process) 是由“软件质量之父 Watts. Humphrey 提出的一种采用广泛的团队过程。TSP 即群体软件过程，是为开发软件产品的开发团队提供指导，TSP 的侧重于帮助软件项目开发团队改善软件质量和生产率，并使其更好的满足成本及进度的目标。在群体软件过程中，指明了如何创建高效且具有自我管理能力的工程小组，工程人员怎样才能成为合格的项目团队成员，管理人员如何对群组提供指导和支持，如何保持良好的工程环境使项目组能充分发挥自己的水平等软件工程管理问题

Watts s. Humphrey 对群体软件过程的改革提出了一整套原则、策略和方法，将 CMM 要求实施的管理和 PSP 要求开发人员具有的技巧结合在一起，按时交付高质量的软件，把成本控制在预算的范围之内

群体软件过程注重团队的高效工作和软件产品交付能力，结合 PSP 的工程技能，指导软件工程师怎样将个体过程结合到群体软件过程中，并正确引导管理层如何支持和授权项目小组，坚持高质量的工作，依据数据进行项目的管理，确保生产出高质量的软件产品。TSP 的根本好处是它指导工程师在成本控制和大胆的进度计划下如何生产高质量的软件产品。

2.1.4 CMM、PSP、TSP结合

CMM 就是在软件界应用最为广泛的过程模型但是它仅提供了一个有力的框架，而未能明确实现过程所需要的具体知识和技能。国外很多软件企业为此推行了 PSP和TSP,这一方法体系已日渐成熟。PSP/TSP不仅是卡耐基 梅隆(SEI)等国际知名大学或软件学院中学生的必修课程,同时在各行业中也有广泛的应用。我国也有越来越多的企业实施了PSP/TSP来增强企业的竞争。

CMM 的本质是项目管理工程的一个部分。迄今为止，产业界和学术界公认 CMM 是当前最好的软件过程，然而它的成功与否和软件研发机构内部有关人员的积极参加和创造性活动密不可分。因为 CMM 中并未提供关于实现子过程域所需的具体知识和技能，所以进行个体软件过程 PSP 的研究与实践可以填补这一领域的空白，而且为基于个体软件过程和群体软件过程的优化提供了具体、有效的途径。群体软件过程 TSP 结合了 CMM 的管理方法和 PSP 的工程技能，建立、管理、授权并且指导项目小组在不超过计划成本的前提下，在承诺的期限范围内，不断生产并交付高质量的产品。从公布的 TSP 实验数据来看，结果是令人满意的。

CMM 在我国获得了各界关注，业内有过多次关于 CMM 的讨论，2000 年 6 月国务院颁发的《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和推动作用，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 年中国村电脑节上开设 CMM 专题论坛，吸引了众多业内人士参与。华为是典型的 CMM 成功案例，1999 年年初，华为技术公司研究管理部副总裁陈青开始了华为的 CMM 之路，到 2001 年 12 月，华为技术的印度研究所已成为中国第一个获得 CMM4 国际认证的软件研发机构。

群体软件过程（TSP）和个体软件过程（PSP）结合在一起，能够指导工程师做到：确保高质量交付软件；生产高安全性的软件产品；改进组织的软件项目过程管理；建立正确目标；定义团队角色；进行风险评估；订制团队计划；

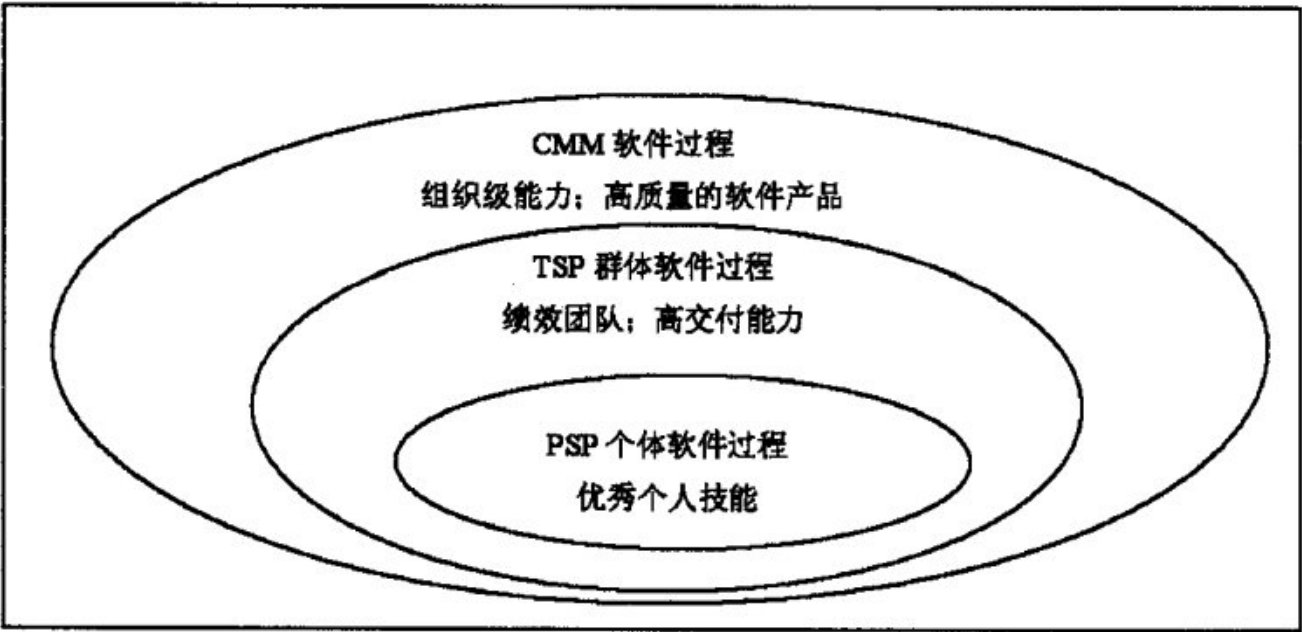


图4 PSP/TSP/CMM 关系图

CMM/CMMI和PSP/TSP的创始人都是 Humphrey,CMM/CMMI用于建立管理框架,设立过程改进的先后顺序,定义改进的内容等。PSP、TSP 和 CMM 为软件产业提供了一个集成化的、三维的软件过程改革框架，如图4所示。正确指导工程师开发和维护团队的工作。它表明如何建立一个自我导向的团队、具有强大凝聚力的团队及如何扮演一个有效团队成员的角色。它还能引领管理层 如何正确指导和支持这些团队,如何维护一个不断提高团队绩效的环境; PSP/TSP指导工程师如何才能完成高质量的工作；如何打造一个配合默契的团队；如何去实现 CMMI 所定义的目标等。他们相辅相成，互相补充，构成一个有机的整体。从软件企业角度来看，所有经验证明 PSP、TSP 能加速 CMMI 在企业范围内的实施，同时也是维持改进的需要。在众多世界知名企业开始实施 PSP、TSP 的情况下，中国的软件组织要提高自己的国际竞争力，PSP, TSP 是必由之路，不仅帮助提升了企业的对外形象和国际认知度，还能为企业带来更大的竞争优势。

2.2 基于RUP进行软件过程管理

统一软件开发过程（英语：Rational Unified Process，缩写为 RUP）是一种软件工程方法，为迭代式软件开发流程。RUP 中的开发活动是用例驱动的，即需求分析、设计、实现和测试等活动都是用例驱动的。RUP 中的开发活动是围绕体系结构展开的。

RUP 软件开发生命周期是一个二维的软件开发模型，RUP 中有 9 个核心工作流，这 9 个核心工作流如下。

- 业务建模 (business modeling) : 理解待开发系统所在的机构及其商业运作, 确保所有参与人员对待开发系统所在的机构有共同的认识, 评估待开发系统对所在机构的影响。
- 需求 (requirements) : 定义系统功能及用户界面, 使客户知道系统的功能, 使开发人员理解系统的需求, 为项目预算及计划提供基础。
- 分析与设计 (analysis & design) : 把需求分析的结果转化为分析与设计模型。
- 实现 (implementation) : 把设计模型转换为实现结果, 对开发的代码做单元测试, 将不同实现人员开发的模块集成为可执行系统。
- 测试 (test) : 检查各子系统的交互与集成, 验证所有需求是否均被正确实现, 对发现的软件质量上的缺陷进行归档, 对软件质量提出改进建议。
- 部署 (deployment) : 打包、分发、安装软件, 升级旧系统; 培训用户及销售人员, 并提供技术支持。
- 配置与变更管理 (configuration & change Management) : 跟踪并维护系统开发过程中产生的所有制品的完整性和一致性。
- 项目管理 (project management) : 为软件开发项目提供计划、人员分配、执行、监控等方面的指导, 为风险管理提供框架。
- 环境 (environment) : 为软件开发机构提供软件开发环境, 即提供过程管理和工具的支持。

RUP 把软件开发生命周期划分为多个循环 (cycle), 每个 cycle 生成产品的一个新的版本, 每个 cycle 依次由 4 个连续的阶段 (phase) 组成, 每个阶段完成确定的任务。这 4 个阶段如下。

- 初始 (inception) 阶段: 定义最终产品视图和业务模型, 并确定系统范围。
- 细化 (elaboration) 阶段: 设计及确定系统的体系结构, 制定工作计划及资源要求。
- 构造 (construction) 阶段: 构造产品并继续演进需求、体系结构、计划直至产品提交。
- 移交 (transition) 阶段: 把产品提交给用户使用。 每一个阶段都由一个或多个连续的迭代 (iteration) 组成。迭代并不是重复地做相同的事, 而是针对不同用例的细化和实现。每一个迭代都是一个完整的开发过程, 它需要项目经理根据当前迭代所处的阶段以及上次迭代的结果, 适当地对核心 workflow 中的行为进行裁剪。

在每个阶段结束前有一个里程碑 (milestone) 评估该阶段的工作。如果未能通过该里程碑的评估, 则决策者应该做出决定, 是取消该项目还是继续做该阶段的工作。

在 RUP 中, 是采用如图 5 所示的 “4 + 1” 视图模型来描述软件系统的体系结构。在 “4 + 1” 视图模型中, 分析人员和测试人员关心的是系统的行为, 因此会侧重于用例视图; 最终用户关心的是系统的功能, 因此会侧重于逻辑视图; 程序员关心的是系统的配置、装配等问题, 因此会侧重于实现视图; 系统集成人员关心的是系统的性能、可伸缩性、吞吐率等问题, 因此会侧重于进程视图; 系统工程师关心的是系统的发布、安装、拓扑结构等问题, 因此会侧重于部署视图。

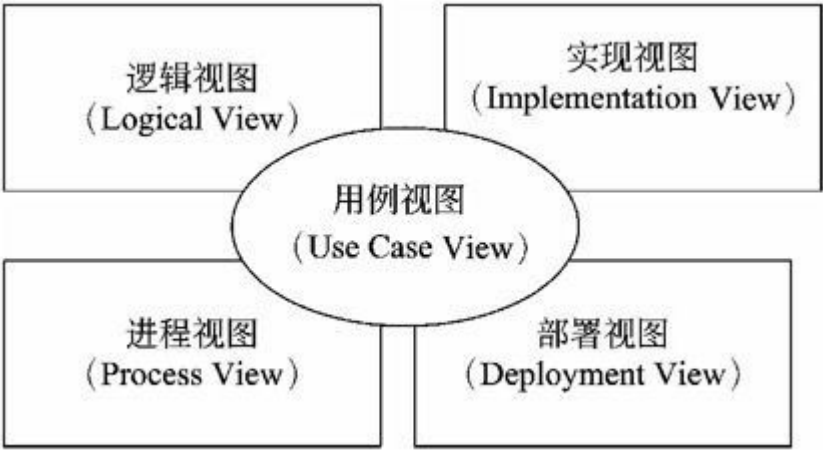


图5 RUP “4+1”视图模型

RUP 强调要采用迭代和增量的方式来开发软件，把整个项目开发分为多个迭代过程。在每次迭代中，只考虑系统的一部分需求，进行分析、设计、实现、测试和部署等过程，每次迭代是在已完成部分的基础上进行的，每次增加一些新的功能实现，以此进行下去，直至最后项目的完成。

RUP 是一个通用的过程模板，包含了很多关于开发指南、开发过程中产生的制品、开发过程中所涉及的各种角色的说明。RUP 可用于各种不同类型的软件系统、不同的应用领域、不同类型的开发机构、不同功能级别、不同规模的项目中。RUP 非常庞大，没有一个项目会使用 RUP 中的所有东西，针对具体的开发机构和项目，应用 RUP 时还要做裁剪，也就是要对 RUP 进行配置。

3 成为一名合格的软件项目经理

项目管理是一门运用系统科学的原理对项目建设进行规划、组织和控制的系统管理方法。它是一门知识密集、技术密集性的专业工作，涉及到管理水平、管理规范、人员素质组织形式等多方面的问题，因此对于进行项目管理的具体执行者—项目经理提出很高的要求，在软件项目集成开发过程尤其如此。

3.1 软件项目经理的重要性

作为软件项目管理的负责人，软件项目成功的一切因素都在项目经理的工作范围内，包括客户、环境、考核、激励等。传统的项目经理侧重技术，对大部分软件公司来说，软件项目经理首先必须是技术专家、行业专家，而后才是项目组织和管理者。

随着项目管理发展，现代软件项目经理不再是单纯技术完成者，他除需具备项目所涉及学科或专业的基础知识还要负责项目团队的组建，分配项目团队角色；需了解现代项目的知识体系，了解与项目相关的经济、法律方面知识。

简言之，项目经理为项目实施争取足够的人、财、物资源，合理配置所有资源，在规定工期和成本预算内，以可接受的质量完成项目的任务，具体到软件项目—就是实现软件功能

目前，对于国内软件企业而言，软件企业大部分是以对外承包项目的形式为生存和发展的途径。项目有大有小，项目的成败及效率就直接影响着公司运营成本和利润。一个优秀的软件项目经理，可以扭转不良项目局面，使之步正轨、实现盈利；不合格的软件项目经理会浪费人力、物力，甚至危及公司运营。因此，软件项目经理在软件项目开发过程中作用举足轻重，是软件项目的核心和焦点

3.2 软件项目经理的素质要求

软件项目经理的职责和工作性质决定其必须具备良好的知识结构、丰富的工程经验协调和组织能力以及良好的判断力。任何一种能力的欠缺都会影响项目开展，甚至导致项目失败。一个优秀的项目经理应具有以下四个重要品质

- (1) 解决问题。项目经理应该能准确地诊断出项目中存在的问题，提出解决方案，采激励项目组成员实现提出的方案。
- (2) 管理者的特性。项目经理必须自信，必须有统领全局的控制力。
- (3) 影响力。项目经理主要依靠的不是权力，而是用自己的人格魅力来领导项目组。
- (4) 理解人。项目经理必须能“识人”、“用人”，使得人尽其才，激发每个人成就感。

目前，国内软件企业有实践经验又有理论的项目经理少之又少，多数经理是从公司内部有一定的开发经验级程序员中选拔出来的，他们擅长技术，具备技术上解决问题的能力，但是没有足够的管理经验，在“识人”、“用人和管理者特性方面有待提高。

3.2 软件项目经理的实践探讨

成为一名合格软件项目经理，除明确软件项目经理的作用、职责、权利外，还应注意以下细节

- (1) 熟悉自己的工作环境、工作责任、工作职责，这是项目管理的前提。在新项目投标伊始，候选软件项目经理就应加入投标小组参加项目投标的全部工作，要确保熟悉项目的基本情况。一旦接手该项目时，保证能迅速上手，指导各方面工作。
- (2) 采用一种开放的、鼓励参与的组织文化围形式：要保证该项目的参与者不会因曾有过的失误而受到打击；保证参与讨论者不因其所言受到打压排挤保证大家对项目的讨论对事不对人，针对将来而不是过去，为开发人员营造一个宽松和鼓励性的氛围，可以减少和防止员工跳槽。
- (3) 软件项目经理应起到纽带作用，要注意协调技术、进度各方面关系。实施项目展开后，项目经理应定期召集技术人员沟通项目信息；时刻督促进度、及时调整实施方案，进行风险控制管理。
- (4) 软件项目经理肩负项目实施质量、成本、进度、安全等全部责任，如出现的超过自己权限的范围的事件，应当及时向上级有关部和人员汇报，请示处理方案或者取得自己处理的授权，切勿因隐瞒小问题而铸成大错。
- (5) 当项目发生业主违约或其它需要与客户联系的紧急情况时，要及时代表承包人向发包人提出要求和通知。项目经理按发包人认可的实施组织设计（或实施方案）、依据合同发出的指令，并要求组织实施。

参考文献

- [1] 荣国平,张贺,邵栋,王青.软件过程与管理方法综述.软件学报,2019,30(1):62-79
- [2]梅宏,陈锋,冯耀东,等. ABC:基于体系结构、面向构件的软件开发方法[J]. 软件学报,2003,14(4):721-732.
- [3] 朱少民, 左智编著. 软件过程管理 [M]. 北京：清华大学出版社， 2007.04.
- [4] 冯骏. 谈软件过程管理中实施的PSP和TSP[J]. 天津职业院校联合学报,2011,13(11):96-99.
DOI:10.3969/j.issn.1673-582X.2011.11.027.
- [5] 朱旭东. 软件过程与CMM[J]. 安徽大学学报（自然科学版）,2003,27(2):28-30. DOI:10.3969/j.issn.1000-2162.2003.02.007.

- [6] 范勇. 全面软件过程管理模型及应用[J]. 计算机工程与设计,2006,27(10):1764-1766.
DOI:10.3969/j.issn.1000-7024.2006.10.017.
- [7] 谢英辉,王如龙. 基于软件过程管理信息系统的软件过程信息化[J]. 科学技术与工程,2006,6(4):477-482.
DOI:10.3969/j.issn.1671-1815.2006.04.033.
- [8] 王海阳. 软件过程管理及其成本的平衡[J]. 计算机系统应用,2005(3):6-8. DOI:10.3969/j.issn.1003-3254.2005.03.002.
- [9] 夏雨,方柯. 基于CMMI的软件过程管理[J]. 河南科技,2014(11):247.
- [10] 李明树,王青. 基于过程控制的软件质量管理[J]. 电子学报,2002,30(z1):2032-2035.
DOI:10.3321/j.issn:0372-2112.2002.z1.029.
- [11] 王采玲. 软件项目经理重要性分析及实践探讨[J]. 科技资讯,2007(33):171. DOI:10.3969/j.issn.1672-3791.2007.33.139.
- [12] 陈娟秀. 合理的软件过程管理是软件质量的基础[J]. 情报探索,2008(3):47-49. DOI:10.3969/j.issn.1005-8095.2008.03.018.
- [13] 秦胜勇. 统一软件过程 RUP 的研究与实践 [J]. 电脑知识与技术, 2016,12 (4):99-100.
- [14] Bello Méndez, Y. & Sánchez, E. (2004). Project management for software process improvement. Paper presented at PMI® Global Congress 2004—EMEA, Prague, Czech Republic. Newtown Square, PA: Project Management Institute