

基于 CMM 模型的软件质量保障支撑平台框架^{*}

The CMM-Based Framework for Software Quality Assurance Platform

韩杰 顾庆 陈道蓄 谢立

(南京大学计算机软件新技术国家重点实验室 南京 210093)

Abstract Capability Maturity Model(CMM) is an evaluated model^[1] for software organizations' developing ability, and it can direct them to make process improvement. Here we create a CMM-based framework for software quality assurance platform, which focuses on the management of document flow and process flow in developing a software and maintaining its life cycle. It can effectively control the software process and enhance the developing efficiency, and hence, improve the resulting software quality.

Keywords Capability Maturity Model(CMM), Software Quality Assurance(SQA), Document flow, Software process

1 背景

二十一世纪是计算机世纪,软件产业将是新世纪的支柱产业。然而迄今为止,软件产品的生产效率和产品质量问题仍然严重阻碍着世界范围内的软件产业的发展,虽然人们不断地展开新方法和新技术的研究与应用,但仍未取得突破性的进展。在总结了数十年的经验教训后人们发现,软件开发组织的基本问题不在于新技术的采用,而在于软件的生产过程本身基本处于无序的状态,没有一套成熟的规章和标准可循,用以有效地管理软件生产过程。

为此,在美国国防部的资助下,卡耐基·梅隆大学(CMU)的软件工程研究所(SEI)自1987年开始,研制出一套软件生产过程成熟度的理论框架——能力成熟度模型(CMM),该框架同ISO9000-3系列一起已成为一整套评价和改进组织软件生产过程、提高软件生产能力和质量的理论和实践依据。CMM的有效性已为大量实践所证实,并已经成为对一个软件组织的生产能力和产品质量进行衡量的事实标准,同时,CMM本身也在不断地完善和发展。许多组织通过参加CMM资质认证,一方面促进了自身软件过程的改进,提高了软件产品的质量和生产率;另一方面,也有利于获得市场的认同,具有更强的竞争力。如美国空军、美国宇航局(NASA)要求其所有的软件提供商必须拥有CMM3级以上的等级。事实表明,CMM模型的成功运用可以

极大地提高一个软件组织乃至一个国家的整个软件行业的生产能力。

我国的软件产业起步于80年代初,经过20年的发展,已初具规模。但相比之下,我国软件组织开发不规范,文档不齐全,维护不完善,质量问题多,素质良莠不齐,基本不具备国际竞争能力,严重束缚了整个软件产业的发展,究其原因,我国软件组织没有完善的规范化管理标准和质量保障体系,软件开发过程很不成熟,甚至于部分组织根本没有意识到软件过程的重要性,因此,我们认为应大力宣传软件过程规范管理和软件质量保障意识,引进和推广CMM管理标准和质量保障体系,促进中国软件组织生产过程的自我完善,通过积极参与CMM认证,提高知名度和国际竞争能力,打开国际软件市场的大门,从而增强我国软件产业的国际接轨能力和国际竞争力,在占领国内市场的基础上鼓励出口,促进我国软件产业的可持续发展。

国内外为了提升软件组织的开发能力和效率,提高软件产品的质量,都在软件质量保障平台方面作了很多工作。美国的ImagoQA公司开发了一套针对用户需求的软件测试框架“Framework”来确保软件质量;数据设计系统公司(DDS)开发出一套进行软件配置管理的工具“PrimeCode”,它可以跨平台对软件产品整个开发周期进行有效的管理,并且超越了简单的源代码版本管理;国内的北大青鸟也进行了软件配置管理方面的研究,开发出JBCM系统^[4,5],它基于软件

^{*} 本项目得到863计划的支持,项目编号为863-306-ZD12-02-2,是“软件产业国际化示范工程”专题的子课题“基于CMM的软件质量保障协同框架”。

构件模型,支持大粒度的管理,可以支持多层次软件系统的开发与维护。

2 软件能力成熟度模型简介

软件能力成熟度模型(CMM 或 SW-CMM)是用来确定一个软件组织的软件开发过程成熟度以及指明如何提高该成熟度的参考模型框架^[1]。它描述了软件过程从无序到有序、从特殊到一般、从定性管理到定量管理、最终到达可动态优化的成熟过程,给出了该过程中五个成熟阶段的基本特征和应遵循的原则、采取的行动,以帮助软件组织改进其软件过程,软件组织首先判断自己属于哪一个成熟阶段,然后再根据自身的实际情况决定应该重点采取哪一些活动来更有效地改进自己的软件过程。

2.1 基本概念

· 软件过程:开发和维护软件及其相关产品(包括项目计划、设计文档、程序代码、测试用例和用户手册等)的一系列活动,包括“工程活动”和“管理活动”两方面。

· 软件过程能力与效能:通过遵循其软件过程,软件开发活动能够达到预期结果的程度称之为软件过程能力;而软件开发活动实际得到的结果称之为软件过程效能。

· 软件过程成熟度:软件过程被明确有效定义、秩序、管理、测量和控制的程度。当一个组织达到了一定的软件过程成熟级别后,它将通过制定策略、建立标准等使它的软件过程制度化,而制度化又促使组织通过建立基础设施等来支持相关的方法、实践和过程,从而使之可以持续并维持一个良性循环。

2.2 CMM 的基本内容

软件能力成熟度模型(SW-CMM)共有五个成熟度级别,从低到高分别为初始级(Initial)、可重复级(Repeatable)、已定义级(Defined)、定量管理级(Maturing)和持续优化级(Optimizing)。除了初始级以外每一级中都由若干的关键过程域(KPA:Key Process Area)组成,关键过程域中包括五类公共特征(Common Feature),每一类公共特征又包括若干个关键实施(Key Practice)。可由图1看出CMM的结构示意图。

CMM模型的成熟度理论主要涉及对软件组织和各类资源的管理;以及对软件工程过程的定义管理和如何度量、管理、改进这些过程;同时还包含对软件工程过程中使用的开发工具和技术的管理。

2.3 CMM 的主要作用

CMM模型通过对软件组织文档流、过程流的有效管理,可以指导其如何控制软件产品的开发和维护

过程,以及如何向成熟的软件工程体系演化,并形成一套良性循环的管理文化。CMM模型在促进软件开发组织的软件能力成熟度不断提高的同时,可用以指导软件组织的质量保障工作,从而大幅提高其软件质量水平。

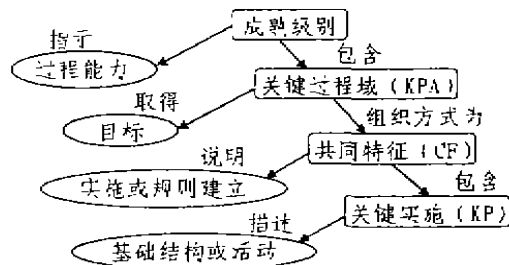


图1 软件能力成熟度模型的结构示意图

3 软件质量保障支撑平台框架

该框架主要基于CMM模型,兼顾ISO9000-3国际标准,通过对软件组织的文档流及过程流进行有效的控制和管理,提升软件组织开发软件系统的能力及效率,从而大幅提高其软件产品的质量^[2]。框架主要由两部分构成,首先是软件组织基于标准规范和多年经验建立的标准文档库、阶段模块库和软件过程库,其次是围绕由这几个库引申出来的,针对软件产品开发过程的管理及监控。

3.1 定义部分

定义部分是按照CMM模型等规范来建立软件组织的标准文档库、阶段模块库和软件过程库^[1],并且可以对其进行创建、修改、删除等操作,其本身也可以从完成的项目中吸取经验教训,进行动态优化。

3.1.1 标准文档库 主要用于组织和维护各种文档模板和完整的模板样例,在软件开发过程中,文档代表各阶段的中间产品,如需求文档、计划文档、设计文档、代码文档等,甚至最终的软件产品也可以看作文档,包括软件文档、帮助文档以及评测文档等。对应不同类型的软件组织,每一种文档类型都可以有多个文档模板和模板样例。用户对文档流进行操作时,标准文档库负责提供相关文档模板和样例以供选择、参考。用户也可以生成新的文档模板或者将已经完成的文档作为样例存入文档库中。

3.1.2 阶段模块库 事先将完成好的标准阶段模块存入库中,用户可以直接根据需要调出这些阶段模块;选择合适的模块后再略加修改便可以生成自己的阶段模块。阶段模块对应软件开发(生命)周期的一个阶段,如需求分析阶段、软件设计阶段和计划制定阶段等,不同的软件生命周期模型会有不同的阶段模式,

即使对一些共有的阶段,不同的模型也会有不同的内容,因此在选择阶段模块时首先要确定欲采用的软件周期模型,再根据组织的不同状况选择合适的阶段模块,用户可以随时扩充和裁剪阶段模块库以适应其需求。

3.1.3 软件过程库 存放针对不同周期模型的软件过程描述。可以有两种类型:一是面向企业的标准软件过程;二是针对具体项目的定义软件过程。用户可以在周期库中查看或抽取已有的过程定义和描述;或者制作新的软件过程将其存入周期库中;也可以从库中删除已过时和无用的过程描述。

3.1.4 定义部分功能结构图 本框架定义部分的功能结构图如图2。软件组织综合考虑系统及外部需求,从软件生命周期库中选择合适的生命周期模型,生成项目生命周期;从文档库和阶段库中取出需要的单元模块,构造软件组织标准软件过程,再针对具体项目裁剪组织标准软件过程,与项目生命周期生成项目定义软件过程,经执行部分,生成软件产品,并把中间经验教训、各种文档反馈回各个库中。

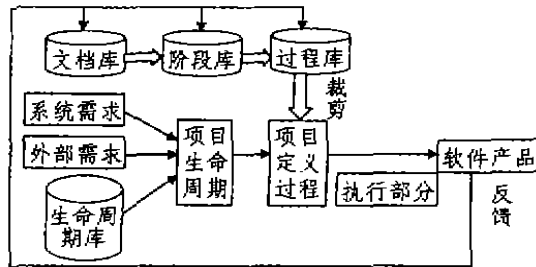


图2 软件质量保障框架定义部分功能结构图

3.2 执行(管理)部分

3.2.1 功能结构图 软件质量保障支撑平台的执行部分可以有如下的功能结构:

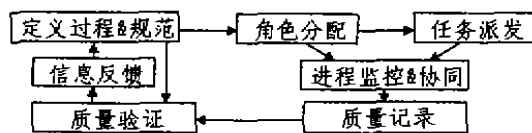


图3 软件质量保障框架执行(管理)部分功能结构图

- 定义过程 & 规范:由定义部分生成的项目定义过程和CMM质量标准或ISO 9000-3规范。过程 & 规范的选择可由计算机辅助完成。

- 角色分配:根据定义过程指定责任,分配给相关人员或部门承担;该部分主要由人工完成。

- 任务派发:分解活动,分派给相关小组完成。任务的分解可以由计算机辅助完成;其他可由人工完成。

- 进程监控 & 协同:按文档流、 workflow 协同管理

和指导软件的开发过程。该部分工作可由计算机辅助完成。

- 质量记录:记录每一个活动的输入、执行和输出。该工作可由平台自动完成。

- 质量验证:比较质量记录和制定的质量标准,监督过程的完成情况,了解完成效果。该工作可由计算机辅助完成。

- 信息反馈:根据执行结果判断定义过程和规范的合理性,为过程的修改提供依据。该工作可由计算机辅助完成。

保障支撑平台的执行管理部分主要有监督和验证两方面的职能^[7]。监督可以帮助企业按照制定的过程和质量标准开发软件,对软件开发的进度和中间产品有一个良好的维护和控制,保证最终软件产品的质量。验证帮助企业了解质量记录的完整性以及同标准的一致程度,同时可以产生必要的反馈、验证所订标准的合理性和完善程度。

3.2.2 执行部分运行结构图 如图4所示。按照某一决策过程为具体项目选定执行过程 & 规范后,经过派发将任务分解到相应的工作组。工作组在协同框架下合作完成软件的开发任务,其间涉及角色分配,过程的质量记录以及质量验证等。选定过程 & 规范的执行结果可以为进一步的过程定义和修改工作提供反馈。

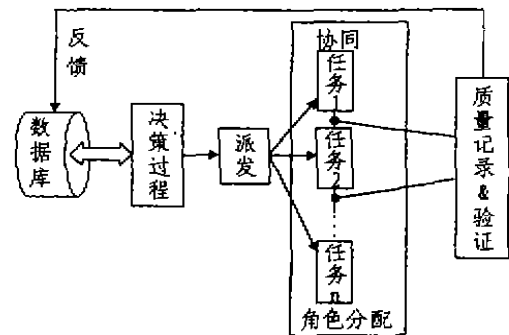


图4 软件质量保障框架运行结构图

3.3 框架实现

本框架采用 Client/Server 结构,使用 TCP/IP 协议建立联系,支持协同开发。标准文档库、阶段模块库,软件过程库建在服务器端,采用 Lotus Domino 进行数据库的管理以及文档流的协调、管理。用户在客户端进行软件产品以及各种文档的开发、维护。为了保持各个库中的相关资源的一致性,需要采用类似 MS Source-safe 的 Check in/Check out 机制。图5为该框架的总体实现结构图。

带标记信号量——一种新型同步与互斥机制

Tagged Semaphores: A New Mechanism for Synchronization and Mutual Exclusion

徐宝文

(东南大学计算机科学与工程系 南京 210096)

Abstract This paper introduces a new kind of semaphores——tagged semaphores. Tagged semaphores are the generalization of standard semaphores and have more powerful capability than the latter.

Keywords Concurrent programming, Semaphore, Tagged semaphore, Mutual exclusion, Synchronization

1 问题的引出

信号量机制是 E. W. Dijkstra 最先提出来的^[1],是并发程序设计中一种基本的、重要的同步与互斥原语,也是研究并发程序行为的重要机制^[2,3]。其描述能力强、机理简单、易于掌握,能方便地描述处理绝大多数并发程序设计的同步与互斥问题(如共享变量的保护)。信号量原语自问世二十余年来,已广泛地应用于许多操作系统与程序设计语言的研究与设计中,成了它们不可分割的一部分。信号量虽然不为一些语言设计者所欣赏,但它仍然得到了很广泛的应用,是研究并发程序同步与互斥的必不可少的工具。尽管如此,也有一些问题用信号量描述起来比较困难或比较复杂。下面是一个典型的入厕问题:

设某地有一公共厕所,它有 N 个蹲位,但没有将

男女蹲位分隔开。因此,当厕所里有人在入厕时,即使还有空位异性也不能进去(可用一块牌子来标示)。但是,只要 N 个蹲坑未被全部占满,与已在入厕者同性别的人就可以进去;而当厕所中无人入厕时,任何人都可以进去。

这一问题虽然比较特殊,但也有一定的普遍性,这类例子也比较多。例如,如果一旅馆每个客房可住多个旅客,那么在为旅客安排客房时就会遇到这种互斥问题。如果一饭店比较大,可同时分批接待若干个代表团或会议就餐,那么在用餐标准相同的情况下,服务员在安排餐桌时可采用如下策略:对每一张餐桌在无人就座时,哪个人先入座就把这张餐桌安排给该人所属的代表团或会议,其他人不得入座(即使餐桌没有坐满)。只有当该桌人都就餐完毕退出以后,其他尚未就餐人员才能再按代表团或会议入座就餐。再如,一排球场的

徐宝文 教授,博士生导师,主要从事程序设计语言,软件工程,并行程序设计等方面的教学与科研工作。

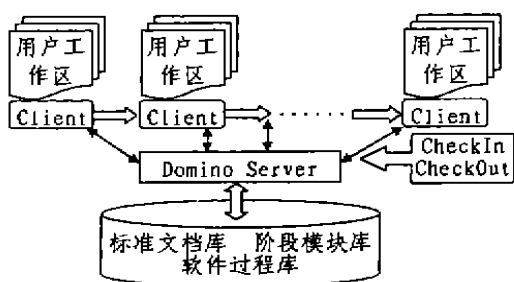


图5 软件质量保障框架总体结构图

参考文献

- 1 Paulk M C, et al. Key Practice of the Capability Maturity

Model, Version 1.1: [CMU/SEI-93-TR-24]. Feb. 1993

2 Barbacci M, et al. Quality Attributes. Dec. 1995

3 Paulk M C. the Rational Planning of (Software) Projects In: the 1st World Congress for SQ, section 4, June 1995

4 张路,李欣,梅宏,杨美清. 基于复用的软件开发过程中的配置管理. 北京大学计算机科学技术系

5 Feiler P H. Software Configuration Management: Advance in Software Development Enviroments. Mar. 1990

6 Goldenson D R, Herbsleb J D. After the Appraisal: A Systematic Survey of Process Improvement, its Benefits, and Factors that Influence Success. Aug. 1995

7 Paulk M C. Effective CMM-Based Process Improvement. In: Proc. of the 6th Intl. Conf. on SQ, Oct. 1996. 226~237

8 谢冰,张路,钟林辉. 软件配置管理系统 JBCM. 北大软件工程研究所