

基于 CMM 的 CPMS 平台的角色定义工具的设计与实现^{*}

谈宁韡 许 平 顾 庆 陈道蓄

(南京大学软件新技术国家重点实验室 南京210093)

摘 要 CMM 已成为软件研究领域与软件企业普遍关注的对象。但传统软件过程支持系统并非专门针对 CMM 而设计的,因此它们在支持企业实施 CMM 的能力方面存在一定的欠缺,并不能良好地支持企业 CMM 的实施。为此本文在基于 CMM 软件过程保障平台系统 CPMS 中提出并实现了基于 CMM 的新型软件角色定义工具,它能有效地对 CMM 二级、三级实施良好的支撑,从而使企业达到实现 CMM 的要求。

关键词 定义工具, CMM, 软件过程

The Design and Realization of the Role-Definition Tool of CMM Based CPMS Platform

TAN Ning-Wei XU Ping GU Qing CHEN Dao-Xu

(The State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093)

Abstract CMM is now of great importance to the software research institutions and software companies. Though traditional software process support systems are not specially designed by CMM, they have no ability to support the actualization of CMM. Based on CPMS (CMM software Process Management System), a new software-role-define tool is presented and comes into being. The tool can effectively support CMM class 2 and 3, so that can support the using companies to satisfy the CMM request.

Keywords Definition tool, CMM, Software process

1 前言

对软件质量的需求一直是不言而喻的,如何在约定的预算与期限里开发出可靠并可使用的软件对许多软件企业来讲是一项艰巨的任务^[1]。经过软件界多年研究表明,软件过程的有效管理和不断改善才是解决问题的根本所在^[2,3]。

目前国际上关于软件过程标准和模型的研究较有代表性的工作为 CMM、ISO9000/ISO9001 和 ISO15504,按照各自侧重点均给出了软件开发过程的一套标准。现在使用最为广泛、最为业界接受的是 CMM (Capability Maturity Model),由 CMU 软件工程研究所 SEI 组织开发的软件过程模型,侧重于大中型软件企事业组织中有关软件过程的宏观管理和项目。目前在软件过程支持工具与环境的研究上,出现了许多成熟的产品与研究系统^[7~10]。国内的同行也对此作了大量的工作^[11~13]。这些系统具备一定的软件过程支持能力,但其出发点都不是专门针对 CMM 而设计的,并不能完全支持 CMM 所涉及到的庞杂的软件工程内容,从而缺乏独立支持企业实施 CMM 的能力,因此我们自行设计了 CPMS 系统 (CMM-based software Process Management & control System),它基于工作流的基本技术,并能直接对 CMM 二级(可重复级)、三级(已定义级)的实施进行良好的支持,从而克服传统软件过程支持系统在这方面的不足,最终达到支持软件企业实现 CMM 的要求。

针对 CPMS 系统自身的需求——对软件过程中人员的有效管理——本文设计了一种角色定义机制,对企业的组织模型进行符合平台要求的描述,以满足 CPMS 系统中执行机构任务时对人员成分有效管理的需求,同时介绍了角色定义

模块的设计与实现。

2 CMM 及机构结构

2.1 CMM 简介

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。其描述了有效的软件过程单元的框架,为软件机构描述了从混乱的、不成熟的软件过程向成熟的、有纪律的软件过程改进的一条途径^[4]。CMM 包括了5个成熟度级别,每个级别都包含了一系列的 KPA,指出了该成熟度级别的基本特征和应遵循的原则、采取的行动,整个软件过程改进的各个级别间是不可跳跃的。CMM 指导软件机构控制开发和维护软件的过程,并发展成为具有优秀的软件工程和管理的机构文化,目的在于引导软件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、采取的行动机构的软件过程改进,最终提高软件过程能力。

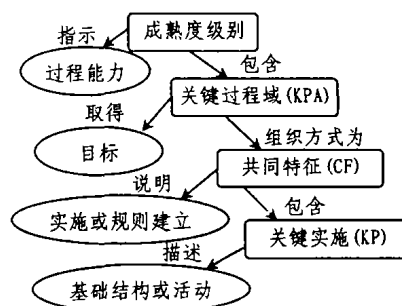


图1 软件能力成熟度模型(CMM)结构图

^{*}基金项目:国家863高技术项目“基于 CMM 的软件质量保障平台及应用”(编号:2001AA113090)。谈宁韡 硕士研究生,研究方向:工作流系统,软件过程管理。许 平 硕士研究生,研究方向:网络视频点播,P2P。顾 庆 博士,副教授,研究方向:分布式计算,工作流系统。陈道蓄 博士生导师,研究方向:分布式计算与并行处理。

2.2 CMM 的机构结构

尽管 CMM 一直努力独立于任何具体机构结构及模型,但是 CMM 实践却无法脱离于具体的机构结构,有必要在机构结构和任务方面给予专门的说明。

2.2.1 机构结构的概念 首先,有必要对一些专有概念进行解释^[5]:

任务:是指由一个或多个人承担已定义职责的单位,任务的基本要素是任务中已定义的人员;

机构:是指公司或者其它实体中的一个单位,它从整体上管理许多项目,实际使用“机构”时是指实施估价或者过程改进工作的范围;

项目:机构承担的具体任务,该任务要求对特定产品进行开发和维护,这需要机构中的各部门合作共同完成;

机构结构:前述定义下的“机构”所具有的具体的结构,通常是人员的组织结构,以支持需要独立性的活动,在软件过程中,这些人员的参与是不可缺少的;

软件过程:是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。

对于具体的 CMM 实践都有相对应的机构结构,在软件过程和机构结构中,最基本的元素是人员,人员是一个成功软件项目中最重要因素,在一个具体的机构结构中参与软件过程(及每一个软件项目)的人员可以分为^[5]:

负责人:在其责任范围内对实施任务和活动的人员提供技术、管理指导和控制,其职能包括职责范围内的计划、组织、指导和控制工作;

上级负责人:是机构中足够高层的负责人,主要关注该机构的长期活力,而不是短期计划和契约性质的事务与压力。通常一个上级负责人负责多个项目,并提供和保护该机构软件过程改进所需的长期资源;

项目负责人:对整个项目负完全责任,是指导、控制、管理和规范某个软件或软/硬件系统建设的人,项目负责人是最终对客户负责的人;

项目软件负责人:对一个项目的软件活动负完全责任,控制一个项目的软件资源,按照软件约定与项目负责

人打交道；

一线软件负责人:在由软件工程师和其他相关成员组成的机构中(如一个部门和项目小组),一线软件负责人负责完成人事和技术活动的直接管理工作(包括提供技术指导、管理人员和报酬);

软件任务主管:负责某项具体任务的技术小组的领导工作,主管技术工作,并负责向该任务的工作人员提供技术指导,软件任务主管和其他工作人员一样向一线软件负责人汇报;

工作人员(staff)、软件工程人员和成员(individual): CMM 使用了几个不同的名词用于区分 CMM 不同实践中负责不同技术工作(任务)的成员。

工作人员就是成员,包括负责完成一项指定工作的任务主管,但不是负责人。

软件工程师就是软件技术人员(如分析员、程序员和工程师),包括软件任务主管,他们执行项目的软件开发和维护,但不是负责人。

在关键实践中使用的术语“成员”的概念与其使用的上下文有关。

在具体的任务或者机构中,任务和个人并不是一一对应的关系,其间关系可以是多多映射关系。

2.2.2 企业的机构结构 在一个具体的机构中,人员通常以小组的形式组成负责一组任务或者活动的集合,小组在前述定义下的人员之上构成了企业的机构结构的基本单元。小组是如下几个概念的综合:指定的兼职人员;来自不同部门的几个兼职人员;几个专职人员;专职作的一个或者多个部门。这些统一起来都可以称为小组。一个机构承担的具体项目,都是由多个小组共同完成整个项目的完整过程的。

一个完整的软件过程,通常由软件工程组、软件相关组(包括软件质量保证组、软件配置管理组和软件工程过程组)、系统工程组、系统测试组、软件配置管理组和培训组共同完成。小组之间的关系是一种相互依赖的关系,软件过程的每一方面都被分离出来由确定的小组来完成,小组与小组之间由软件过程串接起来,其作用不能相互替换。小组与小组的结构关系如图2所示。

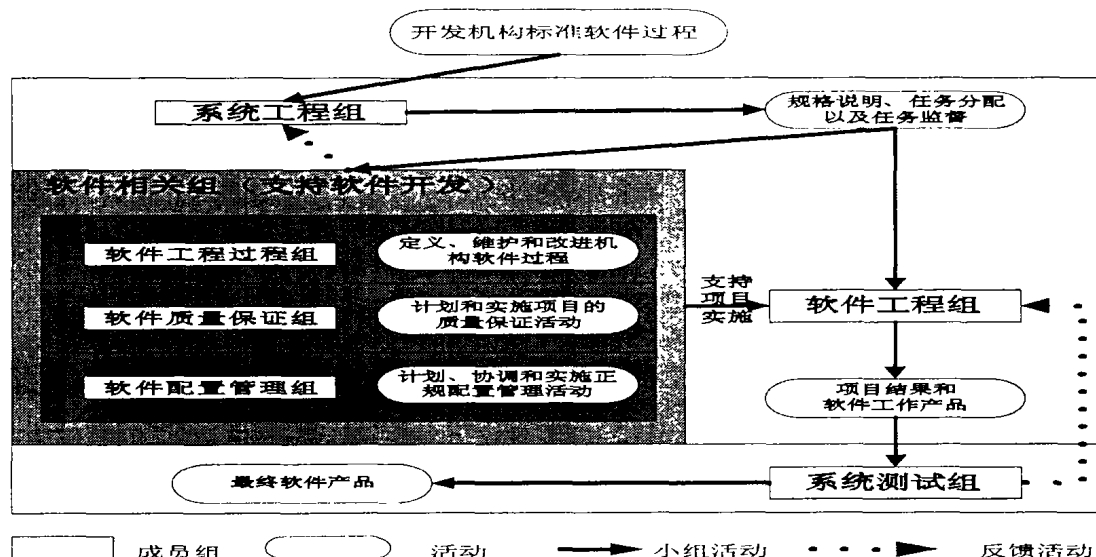


图2 小组的关系结构图

培训组并没有在图中列出,其作用是协调和安排机构的培训活动,为整个机构提供满足要求的机构成员。

软件机构的组织结构,出于软件企业本身的特点,存在两个特点:

其一,软件机构的成员,不管在企业内部还是企业之间,流动性极强。

其二,具体软件过程的执行,在启动相应的软件过程活动时,要求对成员的权限有明确的定义,这对于软件过程的顺利执行有着非常重要的意义。

在机构结构中,引入具体的成员分类,以小组的形式组织相应的人员,具体的软件过程执行时只涉及人员的类型,并不涉及具体的人员信息。这样可以最大限度地避免由于人员的变动带来的软件过程执行上的困难,保证了软件机构的正常的人员演化。成员本身的分类,是引入成员“权限”的基础,处于不同类别的成员,在软件过程的定义阶段可以赋予不同类型的权限,确保整个软件过程管理的一致性和有序性,最终保证软件过程的顺利执行。

3 CPMS 平台组织模型

CMM 的 CPMS 平台在机构结构方面对于组织模型有自身的要求,企业已有的组织结构方式也不可能改变,基于此,CPMS 平台必须有自己的组织模型,以达到对 CMM 的充分支持,在不改变企业自身组织模型的前提下提供企业足够的“柔性”以满足企业实际使用中可能的各种情况。

针对软件机构对成员的分类,在 CPMS 平台中引入了“角色”概念,“角色”是一种抽象的概念,在软件过程中,指的是在软件过程中担当某种任务,启动某些子过程的某一类人员的总称。对应于前面的机构结构,角色实际上是成员分类的一种形式上的替换,比如,项目负责人等等。“角色”都有自己的属性定义,通过属性相互区分,通过定制“角色”的属性,可以方便地定义、修改和取消“角色”的权限等属性。

引入角色概念后,机构结构的小组由相关的“角色”构成,在定义、执行和修改软件过程时使用相应的角色替换具体的成员,通过成员和角色间的关联映射实现成员执行相应的软件过程,避免由于人员的变动引起的软件过程整体的变动。成员在软件过程中由任务所决定的自身的权限属性,也可由成员与角色的映射关系而自由变动。成员与角色的映射关系随成员自身的属性的变更可以是一对多、多对多或者是多对一的关系,这种映射的方式极大地增强了平台的适应性。

“角色”应用于平台活动的各个方面,包括预定义过程、过程元素、标准过程,引入“角色”使得过程裁剪时更灵活和方便。对应于现实的软件企业,成员的属性——工作状态、地址、所处部门、职务等等都是处于一种动态的改变中的,引入“角色”后,同一成员在过程中所扮演的角色可以随时变换,体现出机构组织模型中成员自身的演化。“角色”的作用就是将具体的人与过程相隔离,实现具体个人和项目过程间的弱耦合^[6]。

在平台定义阶段,可以将软件过程的每一个活动的执行者确定为某一类“角色”,即关联的一个集合;同时成员定义部分将企业的成员继承到平台中来,成员通过自身的属性与其它成员相区别。通过成员与角色的关联映射,用户可以方便地将任何成员与其可以担当的角色对应起来,在过程模型的执行中,用户可以用“角色”为桥梁,再根据成员与角色的关联映射,动态地为活动配置具体的执行人员。

通过上述描述,在本平台中使用的组织模型具有如下特点:

能够在不改变软件企业自身组织结构的前提下实现对平台需求的支持,满足了 CMM 中对于人员实行有效管理的需

求;

引入的“角色”概念使得企业具有足够的柔性以适应不断变更的成员构成,实现了机构结构的具体化,将机构结构由抽象转为具体化,保证了软件过程的顺利执行;

通过角色定义机制,成员与任务相关的权限可以方便地得以实现,成员的自身演化可以在过程定义阶段得到体现,对成员的管理也变得十分容易。

4 CPMS 角色定义工具

软件企业在进行软件工程时,其软件人员总是以特定的角色开展他的工作。在 CMM 实践中,对人员的组织与安排是非常细致的,根据要实施的项目(Project)以及要参照的软件过程框架(SPF)^[3],机构须针对不同项目配置许多小组,而在不同项目之间,小组成员允许交叉。图3是角色定义工具的结构图。

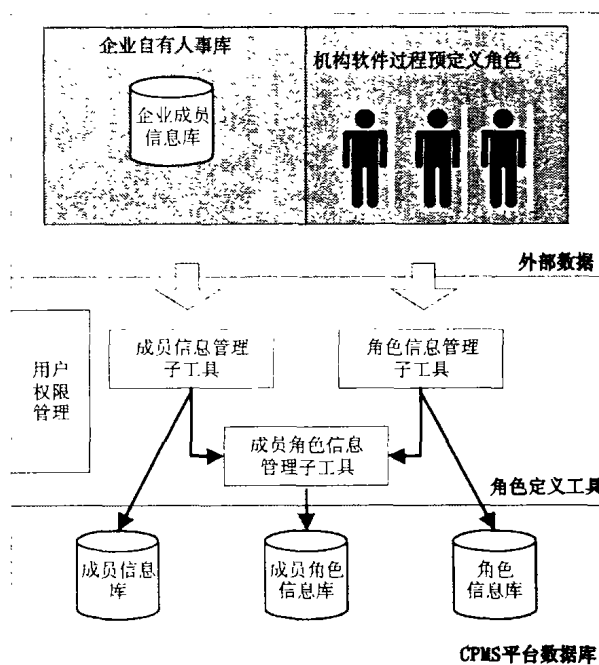


图3 角色定义工具

机构在启动一个项目时,必须完成 CMM 角色与机构成员的映射。因此,角色定义管理工具需要完成的工作是:1. 企业成员信息的管理:工具由企业自身的成员信息库提取所需的原始资料,通过对原始数据的剪裁和加工获得规范的成员信息并存入平台自身的成员库中;2. CMM 角色信息管理:建立与查询 CMM 角色名表;3. 成员角色信息管理:配置各个项目参与成员与 CMM 角色的对应关系,建立成员与角色间的映射关系;4. 用户权限管理:对不同属性的成员配置不同的权限,避免对角色管理工具使用的随意性。

CPMS 系统中,软件企业成员均以特定的角色进行软件活动的相应执行,而系统也是根据成员所处角色的权限向其展示软件过程中的相应部分。这样使得不同角色的用户对系统流程有着不同的视图,从而有利于系统安全与保密。

成员信息直接由机构本身的自有人事库中继承,其作用是对参与具体软件过程实施的成员信息进行剪裁和加工以获得符合平台规范的成员信息。

图4是整个角色管理工具的界面,在其上的成员管理子工具及角色管理子工具中可分别进行成员信息以及角色信息的录入,为角色定义机制提供必要的成员、角色信息支持。成员

角色信息子工具则为用户提供了编辑和删除成员与角色关联映射信息的执行工具,这种关联映射实现了具体成员与过程

中角色的对应。同时工具的整体预览模块则提供了全局的预览和具体信息的编辑手段。

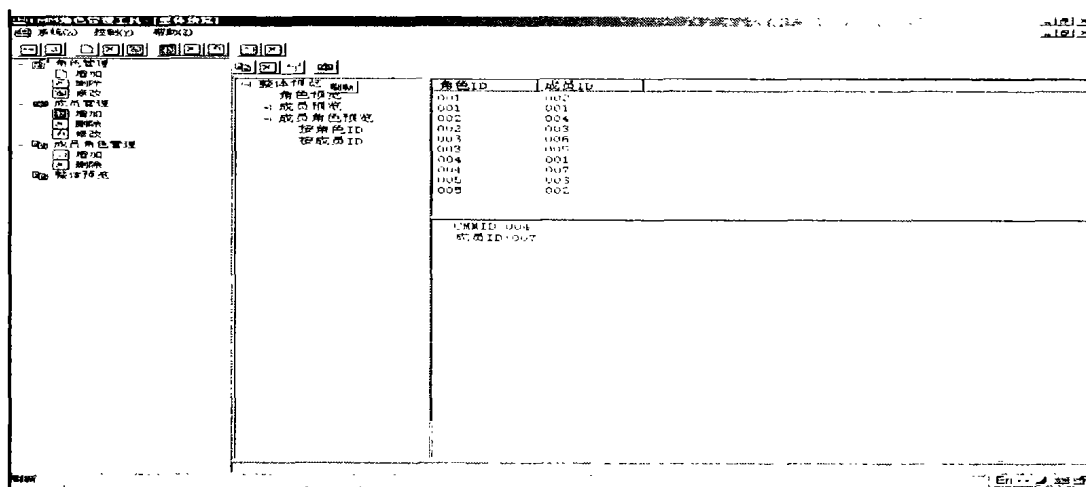


图4 角色管理工具

与传统的软件过程支持定义工具^[7~10]支持的成员定义部分相比,CPMS 平台定义工具对应的角色定义工具的特点在于:

1. 提供了一套图形化的编辑工具,可方便地向用户提供角色定义、剪裁功能。并且定义好后,便能支持软件过程的解释执行。

2. 强化了 CMM 中角色概念,并为此设计了完整的支持与监控工具。

3. 针对系统中的不同人员角色,向其提供系统的不同视图,从而有利于系统信息的安全与保密。

4. 引入的“角色”概念使得软件过程具有极大的柔性,以适应于不同的企业组织。

结束语 CMM 中的机构结构是为了确保企业顺利实行 CMM 标准、规范软件开发过程的一个重要的管理基础。组织建模是对机构结构进行科学管理和分析,实现符合 CMM 标准的机构结构的重要途径,也是企业实施 CMM 的先决条件。为此,文章在已有的 CPMS 平台的基础上提出了一套结合企业已有的组织形式,同时适合于 CMM 标准的机构结构组织模型。通过具体成员与角色的映射关系,实现了机构结构模型与 CMM 软件过程的松散耦合。基于该模型,现在已经初步开发出相应的平台子工具——CMM 角色管理工具,用于对企业进行适合于 CMM 的、全面的机构结构定义。

目前 CPMS 系统已通过专家验收,在今后的研究方向是更具柔性和灵活性的软件企业的机构结构,重点是构建基于对象的软件企业组织模型,以及对 CMM4 级中关于人员管理的进一步支持等等。

参考文献

1 Gillies A. Software Quality Theory and management. CHAPMAN & HALL

2 Paulk M C, Weber C V. Key Practice of the Capability Maturity ModelSM Version 1.1: [Technical Report CMU/SEI93-TR-025 ESC-TR-93-178]

3 Report of the Defense Science Board Task Force on Military Software. Office of the Under Secretary of Defense for Acquisition, Washington, D. C. , Sep. 1987

4 Paulk M, et al. The Capability Maturity Model for Software, Version2B Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213-3890, 1997

5 CMU/SEI. The Capability Maturity Model : Guidelines for Improving the Software Process

6 Olson T G, Reizer N R, Over J W. A Software Process Framework for the SEI Capability Maturity Model. CMU/SEI-94-HB-01, Sep. 1994

7 Estublier J, Belkhatir N. Process-centered SEE and Adele. In: Proc. of the 1992 Intl. Work-shop on Computer-Aided Software Engineering CASE '92, IEEE Computer Society Press, 1992. 156~165

8 Bolder G, Taylor R. Endeavors: A Process System Integration Infrastructure. In: Proc. of the 4th Intl. Conf. on the Software Process, IEEE Computer Society Press, 1996. 76~85

9 Fernstrom C. Process WEAVER: Adding Process Support to UNIX. In: Proc. of the Second Intl. Conf. on the Software Process, IEEE Computer Society Press, 1993. 12~26

10 Bandineli S, Fuggetta A, Ghezzi C. Software Process Model Evolution in the SPADE environment. IEEE Transaction on Software Engineering, 1993, 19(12)

11 杨芙清,等. 软件复用与软件构件技术. 电子学报, 1999, 27(2): 68~75

12 梅宏,等. 青鸟构件库的构件度量. 软件学报, 2000, 11(5): 634~641

13 <http://www.sei.pku.edu.cn/research/intro/jb.htm>