

软件配置管理在软件开发平台中的应用

裴树军, 陈德运, 陈晓雪

(哈尔滨理工大学 计算机科学与技术学院, 黑龙江 哈尔滨 150080)

摘要:介绍了软件组织设计及实施软件配置管理的方案,讨论了软件配置管理的基本概念,提出了一种基于 CMMI 实施配置管理的流程,重点讨论了配置标识、版本控制、变更控制等实践活动.然后,提出了一种集成软件配置管理功能的软件开发平台.通过在现有开发平台中引入文档管理,代码管理等模块,对软件的开发过程进行监督与控制,使软件开发过程满足 CMMI 二级的配置管理规范,实现了提高开发质量和效率的目的.

关键词:能力成熟度模型;配置管理;软件过程改进;软件开发平台

中图分类号: TP311.52 **文献标志码:** A **文章编号:** 1007-2683(2010)01-0028-05

The Implement of SCM in Software Development Platform

PEI Shu-jun, CHEN De-yun, CHEN Xiao-xue

(School of Computer Science and Technology, Harbin University of Sicebee of Technology Harbin 150080, China)

Abstract: This paper introduces the design and implementing scheme of SCM that one software organization may adopt, discusses the basic concepts of SCM, putting forward the implement of configuration management based on CMMI and putting the stress on the practices of configuration identifier, version control and change control. A software development platform integrated software configuration management capabilities. By the introduction of document manage and code management module is also presented, so that the whole process of software development can manage by them, so that the whole process of software development can manage by the configuration management norm to achieve the purpose of improvement the quality efficiency of development.

Key words: CMMI; software configuration management; software process improvement; software development platform

0 引言

随着软件开发平台的不断推广和广泛应用,现代软件的开发,尤其是行业软件的开发过程,早已脱离了早期手工作坊式的开发过程,转向了工业化,标准化的发展方向.如何能低成本,高质量,按时生产出软件产品,成为软件企业的追求目标.其规范软件企业开发过程中的管理和控制的 CMMI 标准目前正在国内外软件企业越来越多的关注.

CMMI(Capability Maturity Model Integration For Software,软件能力成熟度模型集成)是由美国卡内基梅隆大学的软件工程研究所(SEI)在 SW-CMM(Capability Maturity Model For Software 软件成熟度模型)的基础上研发并在全世界推广实施的一种软件能力成熟度评估标准,主要用于指导软件开发过程的改进和进行软件开发能力的评估.

CMMI 共有 5 个级别,除了初始级,每个成熟度级别都包含若干个关键过程域(Key Process Area KPA).CMMI 中的受管理级(二级)包含 7 个 KPA,

收稿日期: 2008-11-21

作者简介:裴树军(1969—),男,副教授.

陈德运(1962—),男,教授,博士生导师,chendeyun@hrbust.edu.cn.

其中软件配置管理 (Software Configuration Management, SCM) 就是其中的一个 KPA^[1-2], 同时也是 CMM 二级中的一个 KPA. 本文分析研究了 CMMI 二级中的配置管理关键过程域, 提出了一种基于 CMMI 实施配置管理的流程和实施方案, 并设计了一种嵌入软件配置管理功能的软件开发平台, 对配置管理及软件开发过程的改善效果进行了评价.

1 软件配置管理过程 KPA

随着软件规模的不断增加, 中间产品不断增多, 使得管理起来十分困难. 配置管理是软件开发环境管理的核心. 如果不做好配置管理工作, 经常会导致以下问题:

- 1) 同时更新: 当两个或更多的程序员独立开发同一软件时, 一个人所做的变更很容易损害他人的工作.
- 2) 共享代码: 当共享代码中的一个错误得以修复时, 常常不能让共享代码的所有人知道.
- 3) 通用代码: 在大型系统中, 若通用软件的功能已修改, 所有使用通用代码的人都有必要知道. 如果缺乏有效的代码管理, 就无法保证找到并提醒每个使用者.
- 4) 版本冲突: 很多大型软件是以增量式发布的方式开发的, 可能会同时发布多个活动版本, 参与错误修复和软件增强的人员很多, 这样很容易产生冲突和混乱.

上述问题如果解决不妥就会造成混乱, 而处理这些混乱又要花费更多的时间和精力. 由此可见, 做好配置管理工作是十分重要的.

软件配置管理 (Software Configuration Management, SCM) 过程域就是建立和维护在项目的整个生命周期中软件项目产品的完整性, 其中包含了对改变的控制和所有能影响到改变的软件因素的管理. 实施 SCM, 就是要求遵守技术和行政的纪律, 确定配置项的功能特性和物理特性, 形成文档, 控制这些特性的变更、记录和报告变更的处理和实施状态, 验证与所规定的需求的一致性.

2 软件配置管理过程

SCM 是 CMMI 中的一个基础构件, 在这个 KPA 中虽然包含了业界普遍认可的、行之有效的最佳实践. 然而, 在该模型中, 并没有给予如何实施配置管理的指导, 只告诉实施人员在配置管理过程中应该做什么以及一些简单的方法. 这给采用该模型进行

软件过程改进、实施配置管理的组织带来了一定的难度. 基于此, 本文中提出了基于 CMMI 实施配置管理的一种行之有效的方法. 由于本软件平台的限制, 本文主要从软件配置管理过程的 SCM 流程、版本控制和变更控制方面进行了研究.

2.1 软件配置管理的基本概念^[3-4]

1) 配置: 有关技术文档中陈述的或产品中已实现的硬件或软件的功能和物理特性.

2) 软件配置项 (Software Configuration Item, SCI): 为配置管理指定的硬件、软件或两者的一个聚合体, 在配置管理过程中视为一个单一实体.

3) 基线: 一种通过正式评审和认可的规格说明或产品, 此后将其作为进一步开发的基础, 只有通过正式的变更控制过程才能变更.

4) 版本: 是某一配置项的已标识了的实例.

2.2 软件配置管理流程

结合本软件开发平台, 建立了一套新的配置管理流程, 详情如图 1 所示. 各个流程部分的角色分配如下. 项目的配置管理工作由项目经理负责监督, 并处理配置管理过程中产生的问题. 项目组成员负责协助配置管理工作. 配置管理员根据项目计划, 制定《配置管理计划》^[6]. 在实施过程中始终遵循项目配置管理计划执行项目的配置管理并进行配置审计. 填写并维护《配置变更及跟踪表》和《配置状态报告》, 按照计划进行配置库的备份. 项目结束后资产管理员负责的组织资产收集, 维护组织的资产库. 软件质量保证人员负责定期检查配置管理过程, 审计配置库的执行情况.

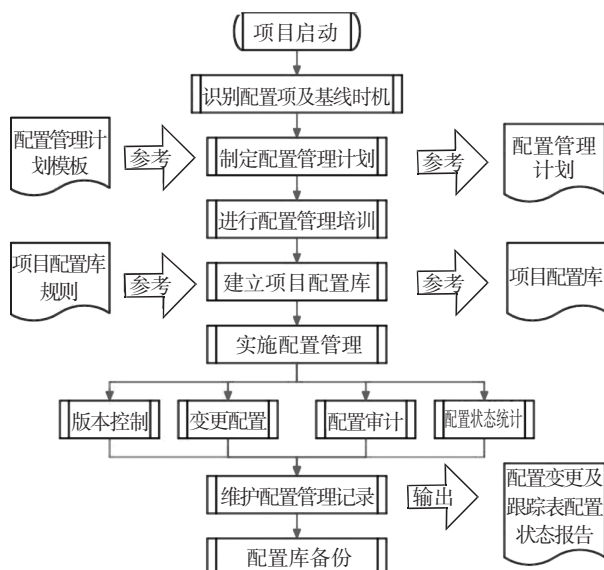


图1 软件配置管理流程

2.3 版本控制

版本控制是软件配置管理的核心,其目的是按照一定的规则保存配置项的所有版本,避免发生版本丢失或混淆等现象,保证产品的可追溯性.

配置项的状态有4种“编辑”、“发布”、“调试”和“修改”.其状态变更如图2所示.

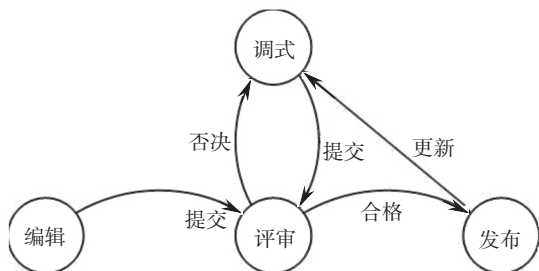


图2 版本控制配置项状态

配置项最初建立时其状态为“编辑”,在此状态下,由开发人员对其进行编辑;编辑完成后的配置项经过提交进入“评审”状态;若通过评审,则其状态变为“正式发布”,否则转入“调试”状态,进行修改;发布后的配置项若需修改,其状态也可变为“调试”,当配置项修改完并重新通过评审时,其状态又变为“正式发布”,如此循环.

2.4 变更控制

变更控制是通过创建产品基线,在产品的整个生存周期中控制它的发布和变更.其目的是建立一套控制软件修改的机制,保证生产符合质量标准的软件和每个版本的软件包含所有必须的元素及工作,使其在同一版本中的各元素可以正常工作,以确定在变更控制过程中控制什么、如何控制、谁控制变更、何时接收变更、批准和检验,变更控制过程.变更控制的流程如图3所示.

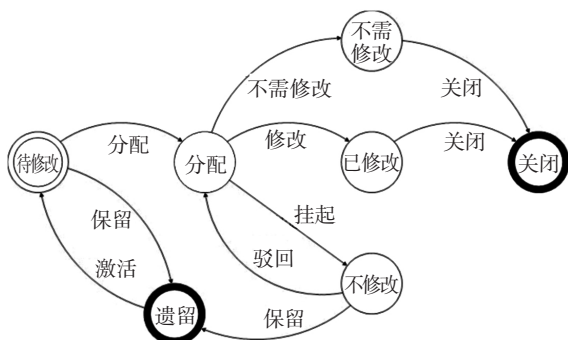


图3 变更控制流程

从图3中可以看出,需要变更的软件元素首先处于“待修改”状态,由QA人员将其分配给相

关人员进行修改,修改完成后再经QA人员批准后将置为“关闭”.待分配的变更要求若无法修改,则可将其设为“不修改”状态,由QA人员将其置为“遗留”;若认定为不需要修改,则也由QA人员将其关闭.

在实施CMMI过程中,SCM是最需要软件工具支撑的KPA.因此,本文将CMMI体系结构中的软件配置管理过程域与软件开发平台的设计有机结合起来,提出了一种集成软件配置管理过程域的软件开发平台.

3 集成软件配置管理过程域的软件开发平台的设计

平台在设计阶段将CMMI二级中的软件配置管理过程域对软件开发过程中的各种要求考虑进来,目的在于在平台的使用,即软件开发的全过程都能够处于软件开发平台的监督管理之下,用以引导企业实施CMMI,并帮助开发团队自觉遵守CMMI开发规范,使开发过程可高质量地进行.

3.1 开发平台的整体结构

从结构图4中可以看出,平台大体上可分为开发与控制两大部分,开发部分用于对应用程序的设计与开发,控制部分用于对整个开发过程中所有的配置项进行管理,使开发满足CMMI关于配置管理过程域的要求.

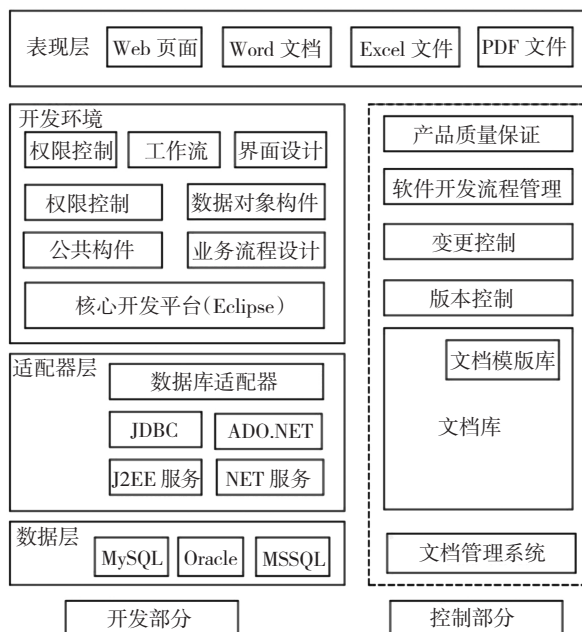


图4 软件开发平台结构图

该软件开发平台基于跨平台的 Java 语言,通过利用 Eclipse 的插件技术^[6],在 Eclipse 开发平台上进行构建.平台在整体上分为三个层次:持久层,开发环境层与表现层.持久层通过提供数据库适配器的 DAO 控件,实现面向对象的数据访问以及对像的持久化;开发环境层集成了开发过程中常用的公共构件,数据对象构件,工作流以及方法学的功能,用于对开发过程的支持,表现层是平台开发界面的展现,为提高开发速度,降低开发难度,平台引入了图形化开发界面,开发过程中常用的组件,控件以及数据对象,均可以通过图形化的拖动,缩放等编辑操作进行设计.平台以 X ml 作为系统数据总线,并可应用在不同的操作系统上,来开发不同的应用系统.

3.2 开发平台对 SCM 资源的管理

本文前半部分主要研究了 SCM 在平台实施时的流程和具体版本的变更控制,那么在平台设计时如何把它们包括在内,并符合 CMM 要求是本文要研究的另外一个方面.在平台的设计中,引入如下功能模块,以确保满足软件的开发 CMMI 体系的要求.

1) 对 SCM 文档资源的管理. SCM 的文档资源包括:①SCM 的方针政策,描述 SCM 的目标、方法、途径,由 CM 经理编制;②SCM 过程,是整个文件体系的核心,将 SCM 标准,包括执行活动在内的所有共同特性的关键实践予以文件化和制度化,由 CM 经理组织人员编制;③配置管理规范 and 适合目标机构的配置管理基本章程,为具体执行 SCM 活动提供作业规范或模板,一般由从事这项具体工作的人员编制,由 CM 经理审核,经项目经理批准.

平台提供了上述文档资源的模版及书写手册,以及一个文档书写工具,以供相关文档的编写;同时,平台拥有一个独立的文档库,项目开发过程中的全部文档,都通过这个文档库,保存于文件服务器中.

2) 对 SCM 代码资源的管理.根据 SCM 的目标和过程,以及具体应用时的组织规模、软件开发方式和特点.选用 Rational Clear Case 用于对代码资源的管理.同时,引入了 ClearCase 插件将 ClearCase 集成到开发环境中. ClearCase 将软件产品开发源代码保存于数据库中,开发团队的每一名成员可在开发环境中将产品的每一个模块下载到本地进行修改,待修改完成后,再将上传的新版本并入开发流中,进行

统一管理.

通过 ClearCase 软件系统及平台中的 ClearCase 插件,QA 人员可以管理多个版本的软件开发,包括跟踪和运行软件构建,个别程序或整体的各种版本的交付等.并使开发人员按小组有效地工作,允许他们精细地协调,彼此共享工作,并隔离他们之间的不稳定修改.

4 配置管理对软件开发过程的改善

为了对配置管理对软件开发过程的改善效果进行科学的评价,将在集成了软件配置管理过程域的软件开发平台下的开发过程与在引入配置管理之前传统软件开发平台下的开发过程的数据进行比较.比较的内容包括程序开发人员在开发阶段平均每天开发代码量与测试阶段缺陷修正的速度两个方面.

4.1 程序开发人员日均完成代码行数

我们采集了项目组三名程序开发人员在不同时期的日均开发代码行数,用以评价引入配置管理对项目开发效率的贡献.从表 1 中的数据可以看出,随着软件开发过程的不断规范化,程序开发人员的工作效率会显著提高.

表 1 程序开发人员日均完成代码行数

程序员	引入配置管理前行数	引入配置管理后行数	将引入配置管理前集成入开发平台后行数
A	59	72	78
B	44	60	68
C	58	70	78
平均	53.7	67.3	74.7

4.2 缺陷修正速度

缺陷是指单元测试中以及集成测试和系统测试中发现的问题,在本文中,采用产品遗留问题加权 and 来描述软件系统中的缺陷存在情况^[7]:

产品遗留问题加权 and (DI) =

$$(\text{致命问题数} \times 10 + \text{严重问题数} \times 3 + \text{一般问题数} \times 1 + \text{提示问题数} \times 0.5) \times S_n$$

其中 S_n 为产品版本的复杂度.对于开发一个相同规模的软件系统,在集成了软件配置管理过程域的软件开发平台下的开发过程与在引入配置管理之前在传统软件开发平台下的开发过程的 DI 值及其变化情况如图 2 表所示.

表 2 DI 值及其变化情况

天数	引入配置管理前	引入配置管理后	将引入配置管理前集成入开发平台后
0	615	407	298.5
10	400.5	225	156
20	225.5	168	75
30	127	78	19
40	95	50	9
50	70	32	5
60	63.5	12	1

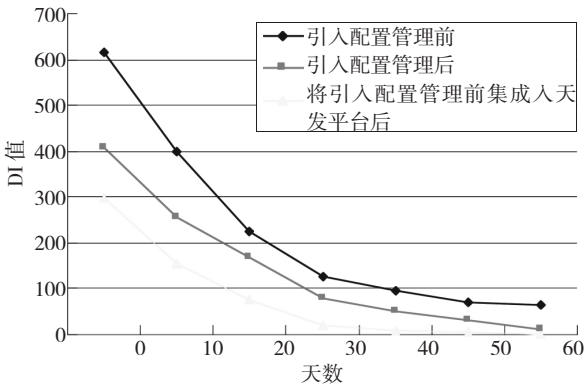


图 5 DI 值及其变化情况

由以上数据可以看出,随着配置管理过程域的引入和在开发平台中的集成,及由此带来的软件开发过程的科学化与规范化,使软件开发的质量与效率都有了较大提高.

5 结 语

本文探讨了软件配置管理过程,提出了适合本

平台的配置管理流程、版本控制、变更控制以及文件体系的建立等活动,设计了集成 SCM 过程的软件开发平台,最后从应用本平台实施具体项目开发过程中抽取所需数据,并采用产品遗留问题加权和方法进行了科学验证.可见本平台在集成 SCM 后可以对开发过程加以规范的管理与控制,极大地提高了开发的质量与进度,确保了开发团队能够及时可靠地完成复杂的开发任务.

参 考 文 献:

[1] Mark C. Paulk, Charles V. Weber, SUZANNE M. GARCIA, et al, Marilyn Bush. Key Practices of the Capability Maturity ModelSM [R]. Version 1.1. Technical Report CMU/SEI-93-TR-025 ESC-TR-93-178. 1993.

[2] WILKIE F G, MCFALL D, MCCAFFERY F. An Evaluation of CMMI Process Areas for Small to Medium sized Software Development Organizations, 2005, 3:315-320.

[3] 于宏霞,陈 凯,白英彩. 基线技术在软件配置管理过程中的应用 [J]. 计算机应用与软件, 2006, 2: 43-45.

[4] 梁成才,金 红,马闰娟. CMM 二级 KPA 软件配置管理的设计及实施方案 [J]. 计算机工程, 2003, 9(1): 259-277.

[5] 王 勇,张发勇,周顺平. CMM 软件配置管理的理论与实践 [J]. 计算机工程与应用, 2004: 102-104.

[6] 米天胜. 基于开放式插件体系结构的 Eclipse 平台及其应用 [J]. 中国管理信息化, 2007, 12: 20-24.

[7] 张 艳,刘 辉. CMM 中配置管理的变更控制 [J]. 湖北工业大学学报, 2005, 20(2): 74-76.

(编辑:付长缨)