

软件配置管理在电力信息平台开发中的应用^①

夏 慧, 孔 震, 钱亚康

(国网电力科学研究院 信息通信技术分公司, 南京 211100)

摘 要: 软件配置管理有利于对整个开发团队进行管理。介绍了软件配置管理定义以及配置管理工具、电力信息 PI3000 平台开发中的版本控制流程、变更控制流程, 设计并实现了基于 XML 定义的代码签出签入工具, 该工具使配置管理工作变得快速准确, 并与缺陷管理系统配合使用。实际应用表明该工具有利于 PI3000 平台开发的配置管理工作有效地执行。

关键词: CMMI; 软件配置管理; 版本控制; 变更控制; XML

Application of Software Configuration Management to the Development of Power Information Platform

XIA Hui, KONG Zhen, QIAN Ya-Kang

(Information and Communicate Technology Company, State Grid Electric Power Research Institute, Nanjing 211100, China)

Abstract: Software Configuration Management is useful to the management of whole development team. This paper introduces definition of software configuration management and tools of software configuration management. This paper also introduces version control process and change control process of the development of PI3000 platform. A tool of codes check out and check in based on XML is designed and implemented. Configuration management is becoming fast and exact using this tool. This tool is used with defect management system. The results of application show that this tool is useful to the work of configuration management in the development of PI3000 platform.

Key words: CMMI; software configuration management; version control; change control; XML

1 引言

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。

2 软件配置管理概述

2.1 软件配置管理的定义

软件配置管理简称 SCM (Software Configuration Management), 是 CMMI2 级中的关键过程域。首先确认在开发过程中所有会产生变动的软件产品的关键元

① 收稿时间:2011-07-13;收到修改稿时间:2011-08-17

素(如代码、设计文档)为配置项,并对其在接下来的所有变化进行相应管理,保留有益的变更,控制无益的变更。SCM 最根本、最简洁的定义就是用于管理软件的变更的过程域,它可以有效控制软件开发过程的混乱程度,从而保证开发方向的正确性和产品质量的可靠性。

2.2 软件配置管理概述

配置管理过程是软件生命周期中的重要过程元素之一,在 CMMI 中,配置管理有三个特定的目标:建立基线,跟踪和控制变更,建立完整性。建立基线有三个特定实践:标识要被列入配置管理之下的配置项,建立一个配置管理和变更管理系统,创建和发布基线;一旦使用一个配置管理系统建立了基线,所有的变更都要在第二个目标两个特定实践中被跟踪和控制,首先跟踪变更请求,然后跟踪配置项的内容变更;第三个目标有两个特定实践:保持配置项的记录;执行确保基线正确的配置审计。

2.3 配置管理工具

目前,常用配置管理工具有 Starteam、PVCS、ClearCase (CC)、Visual SourceSafe (VSS)、Concurrent Versions System (CVS)。

Starteam 提供了真正的协同工作,使得开发环境内外的所有的功能范畴和团队能够积极参与共同的活动,是一个集合了版本控制、构建管理和缺陷跟踪系统为一体的软件。

PVCS 能够实现源代码、可执行文件、应用文件、图形文件和文档的版本管理;它能安全地支持软件并行开发,对多个软件版本的变更进行有效的控制管理。PVCS 不仅具备版本管理、建立管理、构造管理、文体追踪这些基本的功能模块,还融合了需求管理、需求变更管理技术,并支持工作流程。

CC 是 ROSE 构件的一部分,主要应用于复杂的产品发放、分布式团队合作、并行的开发和维护任务。解决了对当前多种产品的开发和维护,保证产品版本的精确,重建先前发布的产品,加强开发政策的统一和对特殊版本需求的处理等等软件开发中的尖锐问题。主要功能有版本控制、工作空间管理、建立管理和过程控制等。

VSS 将所有的项目源文件以特有的方式存入数据库。开发组的成员不能对该数据库中的文件进行直接的修改,而是由该版本管理器将该项目的源程序或是

子项目的源程序拷贝到各个成员自己的工作目录下进行调试和修改,然后将修改后的项目文件作 Check in 提交给 VSS,由它进行综合更新。主要功能有组内协调、版本跟踪、归档和恢复、项目之间的差异比较、历史信息过滤等。

CVS 的基本工作思路是在一台服务器上建立一个仓库,仓库里可以存放许多不同项目的源程序。由仓库管理员统一管理这些源程序。这样,就好像只有一个人在修改文件一样。无限制的版本管理检出(check out)的模式避免了通常的因为排它检出模式而引起的人工冲突,可用于各种平台。CVS 的主要功能有版本管理、完善的冲突解决方案、代码权限的管理、支持方便的版本发布和分支功能、代码集中的配置和调整代码。

3 PI3000平台开发中的配置管理

3.1 配置管理角色职责

PI3000 平台是由十多个复杂子系统组成的电力业务基础软件平台。在开发过程中,使用的配置管理工具为 Visual SourceSafe。VSS 集成在开发工具 Eclipse 中,可以很好地体现出 VSS 的易用性和强大功能。设计、开发过程中的所有文档、源程序、数据库脚本文件和补丁都作为配置管理项。

为保证 PI3000 平台开发的配置管理流程正常运转,我们设计了项目经理、配置控制委员会、配置管理员、系统集成员、开发人员五种角色。项目经理负责批准、发布配置管理计划。配置控制委员会负责制定常用策略、审核变更申请。配置管理员负责软件配置管理工具的日常管理与维护。系统集成员负责构建系统、发布系统。开发人员根据配置管理要求完成开发任务。

3.2 版本控制和变更控制

3.2.1 版本控制

版本控制是全面实行软件配置管理的基础,可以保证软件技术状态的一致性^[8]。实际上,对版本的控制就是对版本的各种操作控制,包括签入签出控制、版本的分支和合并、版本的历史记录和版本的发行。如图 1 所示,PI3000 平台配置库分为开发库、Release 库和产品库。开发人员平时修改的代码直接签入开发库。项目经理制定版本发布计划后,由系统集成员从开发库下载版本断面,形成测试分支,同时形成

Release 库。从形成测试分支到正式发布期间，测试发现的缺陷，开发人员在 Release 库上修改，并同时归并到开发库中。当正式测试发布完成后，Release 库的代码形成产品库，标记为 1.0 版本。当产品正式发布后，对于现场发现的紧急需要处理的缺陷在产品库上进行，并同时归并到开发库中。当缺陷修改完成后，发布 1.0SP1，同时标识为 1.0.1 版本，产品库缺陷发布以此类推。当项目经理制定下一个版本发布计划时，重复以上过程。

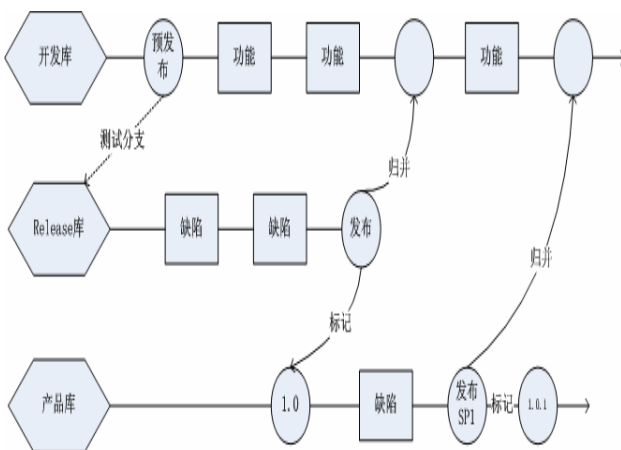


图 1 版本控制示意图

3.2.2 变更控制

当 Release 库和产品库上需要修正产品缺陷时，为保证库代码的相对稳定以及产品发布的严格控制，需要对上述配置库进行权限控制。开发人员只能对开发库具有读写权限，对 Release 库和产品库的修改统一由配置管理员进行。当开发人员需要修改 Release 库或产品库上的代码时，需要按照图 2 所示的流程进行。

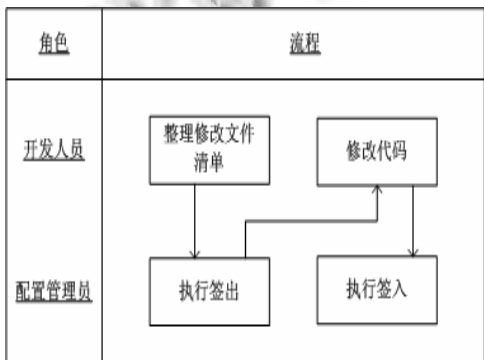


图 2 代码签出签入流程图

在图 2 中，开发人员整理修改文件清单，将清单交给配置管理员执行签出，配置管理员将签出的代码交给开发人员进行修改，最后由配置管理员将修改完的代码签入。可以看出，所有代码签出的相关工作由配置管理员进行。当修改缺陷较多时，配置管理员的工作将是一个瓶颈，会影响开发人员的工作。为此，我们设计并实现了基于 XML 定义的代码签出签入工具。

3.3 基于 XML 定义的代码签出签入工具

3.3.1 界面原型

界面设计如图 3 所示。代码签出时，输入问题号，点击按钮“签出代码”，即可将代码签出并压缩放到路径“D:\PI3000 for Java Release”中。代码签入时，输入问题号，点击按钮“签入代码”，即可将开发人员提交的代码签入 VSS 库中。

	本地工作路径	登陆用户名	登陆密码
Release库	D:\PI3000 for Java Release	xiahui	*
产品库	D:\PI3000 for Java Product	xiahui	*
问题号	19202		
	签出代码		签入代码

问题19202的签出申请文件19202-1yt.xml的代码已签出并压缩。

图 3 签出签入工具界面原型

3.3.2 签出处理流程

签出处理按照“得到连接 VSS 用户名和密码->得到需要处理的问题->处理问题->持久化用户名和密码”的流程进行，处理问题的逻辑如下：

```
If（问题是 Release 库问题）
{
    连接 Release 库；
    查询该问题下有几个签出申请文件 files；
    Foreach（file in files）
    {
        删除本地目录文件；
        下载签出申请文件；
        Foreach（file1 in 签出申请文件中的修改文件）
        {
```

```
If (file1 已经签出)
    更新提示框, 提示该文件已经签出;
Else
    签出文件;
}
压缩签出文件;
记录需要处理的增加和删除文件;
}}
Else
{
    按照 Release 库流程处理产品库问题;
}
```

3.3.3 签入处理流程

签入处理流程跟签出处理类似, 按照“得到连接 VSS 用户名和密码->得到需要处理的问题->处理问题->持久化用户名和密码”的流程进行, 处理问题的逻辑如下:

```
If (问题是 Release 库问题)
{
    连接 Release 库;
    查询该问题下有几个签入申请文件 files;
    Foreach (file in files)
    {
        删除本地目录文件;
        下载并解压待签入的代码文件;
        下载签入申请文件;
        Foreach (file1 in 签入申请文件中的修改文件)
        {
            If (file1 未签出)
                更新提示框, 提示该文件未签出;
            Else
            {
                if (待签入文件不存在)
                    更新提示框, 提示代码未提交;
                else
                {
                    签入文件;
                    if (版本没有发生变化)
                        更新提示框, 提示代码未改变;
                    else
                        打版本号标签;
                }
            }
        }
    }
}
```

```
}}}}
```

```
Else
```

```
{
```

```
    按照 Release 库流程处理产品库问题;
```

```
}
```

3.3.4 工具特点

基于 XML 定义的代码签出签入工具大大减轻了配置管理员的工作, 代码签出签入的工作处理快速并且准确, 除此之外, 该工具还具有以下特点。

与缺陷管理系统配合使用 代码的签出签入和缺陷管理系统中的问题号关联, 如果不输入问题号, 可以查询出缺陷管理系统中的所有待处理的问题进行处理。

自动区分 Release 库和产品库 产品库缺陷是当版本发布后现场发现的问题, 可在缺陷中进行标记, 该工具则可以自动区分 Release 库和产品库。

智能提示 在处理过程中, 遇到非正常情况时, 能够给出提示。如签出代码时, 代码已被其它问题签出; 签入代码时, 代码未提交或未发生改变等等。

3.4 应用情况

基于 XML 定义的代码签出签入工具至 2009 年 10 月应用于 PI3000 平台开发的配置管理以来, 为 PI3000 版本发布、补丁发布节约了大量的时间。假设一次发布问题数 i 个, 申请签出次数 j 次, 修改文件 k 个, 处理一个文件需要 0.5 分钟, 则投入时间按公式 计算。以本次 2011.6.24 发布补丁计算, 本次发布问题数 31 个, 申请签出次数 60 次, 修改文件 236 个, 按公式计算, 手工方式耗时 22 倍多。且修改文件越多, 节约成本越高。从准确性来看, 工具方式更准确, 不易出错。

4 结语

做好软件配置管理有利于对整个开发团队进行管理, 基于 XML 定义的代码签出签入工具有利于软件版本控制和变更控制很好地执行。该工具在自动生成配置报告、自动生成本次发布修改的文件清单方面还有待完善, 以进一步减轻配置管理员的工作。

总之, 软件本身是思维逻辑和数据的固化, 而思维逻辑和数据是无形的, 软件配置管理就是对这些无形的思维逻辑和数据进行变更记录, 使之可以度量、统计分析和管理的。通过配置管理的有效执行, PI3000

(下转第 100 页)

分配的联合优化有利于改善系统性能。

本文是基于单中继协作通信进行研究的,而现实网络是复杂的多中继协作通信网络。如何从多个中继中选择一个最优中继来协助某用户完成通信从而提高系统性能,是目前值得深入研究的问题。而本文的结论从一定意义上可以作为最优中继的优先选择准则之一,即系统完成功率分配后,可以通过计算中继的最优位置,来缩小中继的选择区域以确定优先选择对象。因此,研究多中继协作通信系统中的中继选择策略是下一步的研究方向。

参考文献

- 1 Gesbert D, Shafi M, Shiu DS, Smith PJ, et al. From theory to practice: an overview of MIMO space-time coded wireless systems. *IEEE Journal on Selected Areas in Communications*, 2003,4,21(3):281-285.
- 2 Nosratinia A, Hunter TE, Hedayat A. Cooperative communication in wireless networks. *IEEE Communication Magazine*, 2004,11,42(10):74-80.
- 3 彭木根,王文博.协同无线通信原理与应用.北京:机械工业出版社,2009.
- 4 Zhao Y, Adve R, Lim TJ. Improving amplify-and-forward relay networks: optimal power allocation versus selection. *IEEE Trans. on Wire Communications*, 2007,8,6(8):3114-3123.
- 5 Zhang Q, Zhang JM, Shao CJ, et al. Power allocation for regenerative relay channel with Rayleigh fading. *Vehicular Technology Conference*, 2004,1,2:1167-1171.
- 6 Meng Y, Jing L, Sadiadpour H. Amplify-forward and decode-forward: the impact of location and capacity contour. *Military Communications Conference*, 2005,11,3:1609-1615.
- 7 Craig JW. A new, simple and exact result for calculating the probability of error for two-dimensional signal constellations. *Military Communications Conference*, 1991,11,4(7):571-574.
- 8 Proakis J. *Digital Communications*. New York: McGraw-Hill, 2001.

(上接第8页)

平台开发项目基本实现了一致性、完整性和可追溯性。

参考文献

- 1 杨丽红,李志蜀,袁晓玲,肖静.CMMI2级中配置管理过程域的研究和实施. *计算机应用*, 2005,12:404-406.
- 2 柳永坡,刘凌辉,晏海华,金茂忠,刘雪梅.白盒测试中源代码变更管理方法的研究与实现. *计算机工程与应用*, 2009,45(6):85-88.
- 3 金兰.基于CMMI的软件配置管理研究. *软件导刊*, 2009,8(11):42-45.
- 4 王珍英.配置管理在软件项目管理中的应用. *计算机系统应用*, 2008,17(6):101-104.
- 5 聂华北,张艺超.软件配置管理工具综述. *计算机系统应用*, 2008,17(7):125-128.
- 6 Hass AMJ. *Configuration Management Principles and Practice*. Wesley A, a Pearson Education Company, 2003. 245-247.
- 7 裴树军,陈德运,陈晓雪.软件配置管理在软件开发平台中的应用. *哈尔滨理工大学学报*, 2010,15(1):28-32.
- 8 张艳,刘辉.CMM中配置管理的变更控制. *湖北工业大学学报*, 2005,20(2):74-76.
- 9 尼米智.软件配置管理入门. *山西财经大学学报*, 2004,26(2):222.
- 10 彭俊.在软件开发中如何部署VSS进行版本控制. *铜业工程*, 2009,2:61-63.
- 11 Extensible Markup Language(XML) <http://www.w3.org/XML/>, 2005.
- 12 阿莉萍,赵文耘,张志.可变粒度的软件配置管理工具的研究. *计算机工程*, 2003,29(5):86-88.
- 13 徐勉.解析本土化软件配置管理. [2007-4-14]. <http://www.51testing.com/html/81/n-7681.html>
- 14 姜平利,傅育熙.软件配置管理及实现. *计算机工程*, 2004, (12):45-47.
- 15 张萍,易东云.软件配置信息设计实现方法. *计算机系统应用*, 2003,12(2):60-62.