

基于 ClearCase 的软件配置管理与持续集成

姜 文,刘立康

(西安电子科技大学 通信工程学院,陕西 西安 710071)

摘 要:软件配置管理是整个软件开发生命周期中一个核心的管理过程。ClearCase 是业界非常优秀的软件配置管理工具,主要应用于复杂环境下的软件产品开发和维护工作,适合大型软件团队使用。ClearCase 可以实现并行开发,提高软件开发效率。文中叙述了配置管理工具 ClearCase 的特点;介绍了 ClearCase 软件配置管理的基本概念和相关角色;详细介绍了配置管理的工作流程,主要包括权限管理、ClearCase 控制管理下的开发工作、建立编码基线、构建内部转测试版本、变更管理、版本管理等内容;详细介绍了基于 ClearCase 配置管理的持续集成;最后介绍了一个典型工作案例。工作实践表明,采用该配置管理方法有助于提高软件质量和工作效率,也便于管理工程师及时了解工作进度和解决存在的问题。

关键词:版本对象库;配置管理;视图;流;基线;持续集成

中图分类号:TP311.56

文献标识码:A

文章编号:1673-629X(2016)01-0010-08

doi:10.3969/j.issn.1673-629X.2016.01.003

Software Configuration Management and Continuous Integration Based on ClearCase

JIANG Wen, LIU Li-kang

(School of Telecommunication Engineering, Xidian University, Xi'an 710071, China)

Abstract: Software configuration management is a core management process in the entire software life cycle. The ClearCase, a configuration management tool, is mainly used for the software product development and maintenance in complex environment, which is suitable for large software team. ClearCase can realize the parallel development, improving the development efficiency. The characteristics of ClearCase was described, the basic concept of the configuration management and the related roles was introduced, the work flow of it was discussed in detail, mainly including rights management, the development under the ClearCase, the code base establishment, internal test version construction, version management, etc. Introduce in detail configuration management and continuous integration based on the ClearCase. Finally give a typical case. Practice shows that using this method helps to improve the quality of software configuration management, also facilitate managers understand the work progress and solve the problems in time.

Key words: object library of version; configuration management; view; flow; baseline; continuous integration

0 引 言

软件配置管理在软件项目开发过程中起着至关重要的作用。ClearCase 是业界非常优秀的软件配置管理工具。采用 ClearCase 进行版本控制可以实现并行开发,提高软件开发效率。ClearCase 采用两种应用方式:UCM ClearCase 和 Base ClearCase。UCM ClearCase 提供了更加完善的功能,主要应用在大型软件项目的开发。Base ClearCase 易于掌握,适用于中小型软件项目的开发。

持续集成是完全的自动化构建过程。它的特点是

代码提交频率高,构建频率也高,实现快速反馈。从而可快速修复代码中的缺陷,提高代码质量;加强工程师沟通,明确工作进度,给项目管理提供了很好的保证。

文中主要介绍的是 UCM ClearCase,详细介绍了基于 ClearCase 的软件配置管理和持续集成,最后介绍了一个典型工作案例。

1 ClearCase 的特点

1.1 特点描述

软件配置管理工具 ClearCase 主要用于 Windows

收稿日期:2015-04-28

修回日期:2015-08-05

网络出版时间:2016-01-04

基金项目:国家部委基础科研计划;国防预研基金项目(A1120110007)

作者简介:姜 文(1986-),女,工程师,硕士,CCF 会员,研究方向为图像处理和模式识别、数据库应用、软件工程等;刘立康,副教授,研究方向为数字通信、图像传输与处理、图像分析与图像识别等。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20160104.1505.022.html>

和 Unix 开发环境。ClearCase 主要有以下特点和功能:

(1) 版本控制 (Version Control)。

ClearCase 可以对软件开发环境中的各种对象类型进行版本控制,其中包括源代码、二进制文件、可执行文件、文档、目录内容、测试相关文件、编译器等;可以自动追踪文件和目录的变更情况。ClearCase 具有分支和归并功能,从而支持并行开发。版本的组织是通过图形化的版本树来体现的,可以通过版本树查阅配置项任一版本的情况。

(2) 工作空间管理 (Workspace Management)。

ClearCase 提供两种视图。动态视图 (Dynamic view) 提供网络用户快速取得软件代码和文档资料。静态视图 (Snapshot view) 提供用户下载软件代码和相关资料,支持局部构建和离线作业。

具有快照 (snapshot) 和分支 (branch) 功能,使版本控制的功能进一步增强。快照是项目中多个配置项各自的当前版本的集合,能方便地恢复配置项早期版本,支持批量检出、检入等操作。用户可以创建独立的开发路径,进行并行开发和多版本开发。

(3) 构建管理 (Build Management)。

ClearCase 可以选择构建的文件版本,记录软件构建过程的相关信息;可以重建任何以往的版本;可以并发执行多个构建脚本来完成软件版本构建。自动建立 system build 清单,与一般常用的 nmake 及 GNU make 兼容。

(4) 过程控制 (Process Control)。

在软件开发过程中,ClearCase 可以通过权限管理对开发工程师授予不同的权限来控制对文件或目录的各种操作。可以自动生成常规日志,监控软件的变化情况。可以提供开发工程师开发活动的定制安排,使软件开发过程的管理趋于自动化。

1.2 搭建配置管理的软、硬件环境

ClearCase 配置管理的硬件环境主要是由服务器和客户端组成,采用 C/S 网络结构。

1.2.1 服务器端安装配置

服务器端由 VOB/View 服务器、注册 (Registry) 服务器、许可证 (License) 服务器和 ClearQuest^[1-3] 数据库服务器组成。若使用 ClearCaseWEB 客户端,则还需安装 ClearCaseWEB 服务器,通常 ClearCaseWEB 服务器和 View 服务器会配置在同一台机器上。

由于 ClearCase 利用 Windows 的域来进行用户的权限管理,首先在一台服务器上预安装 ClearCase,新建一个域。在 ClearCase 预安装完成后,会自动创建一个具有超级权限的管理员组 (ClearCase 组,CC 组) 和有超级权限的用户 clearcase albd,clearcase albd 属于 CC 组。CC 组的成员称为 CC 管理员。

以 CC 管理员身份安装各种服务器软件。对于有多个大项目的团队来说,通常配置多台 ClearCase 服务器。CC 管理员可以在 VOB 服务器创建项目 PVOB,根据项目需要,可以建立多个 VOB。创建新项目后可以将相关的源文件导入 VOB。

1.2.2 客户端安装配置

(1) 将本机加入 ClearCase 所在域;

(2) 将登入本机的域用户加入本机管理员组,使该用户拥有本机管理员权限;

(3) 该用户在本机上以默认设置完成 ClearCase 客户端安装。

客户端是提供给开发工程师的终端平台。

2 ClearCase 配置管理的基本概念

2.1 ClearCase 基本术语

(1) 元素^[4-5]。

元素由版本控制下的文件和目录组成。文件元素是可以储存在文件系统中的任何文件。目录元素中包含着文件元素及其他目录元素。

在 ClearCase 中没有提到软件配置管理中常用的概念配置项 (Configuration Item, CI),配置项是软件生存期内,能相对独立开发的一个程序实体或文档。

配置项和元素之间的关系:元素是软件配置项的载体,配置项是元素存储的内容。

(2) 版本 (Version)。

版本是指元素的某一特定时刻存储的内容。元素的一个检入 (Check in) 产生一个新的“版本”,元素的版本号增加 1。元素记录了它所代表的文件或目录的所有版本。

(3) 工作空间。

工作空间称为“视图 (view)”。ClearCase 提供了两种视图。静态视图将文件下载到视图中,通过数据库跟踪下载到工作目录中的文件版本。动态视图通过虚拟文件系统提供对于元素版本的存取操作,不需要将元素拷贝到本地目录中。

(4) 版本对象库 (Version Object Base, VOB)。

版本对象库简称版本库,VOB 是软件配置管理 (SCM) 系统的核心,用来存储文件、目录和元数据的永久数据存储池。

ClearCase 有两种 VOB,项目 PVOB (Project VOB, PVOB) 和标准 VOB。PVOB 用来组织和管理正在开发的项目,一个 PVOB 可以有多个 VOB。VOB 用于正在开发的系统。

通常逻辑上版本库可以分为开发库 (动态库、程序员库、工作库)、受控库 (静态库、主库)、产品库 (软件仓库) 3 种。

(1)开发库:存储开发过程中的源代码、可执行代码、数据和相关的技术文档,为开发工程师的工作提供支持。

(2)受控库:存放处于受控状态的基线库和其他软件产品。

(3)产品库:项目交付用户的软件产品和相关的运行环境。

2.2 UCM 术语

(1)项目(Project)。
UCM 项目^[6]是开发团队为执行开发活动而进行代码或其他配置项修改的协作环境。ClearCase 中基本 UCM 项目由一个共享工作区和多个私人工作区组成。

(2)构件(Component)。
构件通过一个根目录来定义。将开发团队进行开发的文件和目录组合在一起就形成了构件。开发团队以构件为单位进行开发、集成和发布。一个 VOB 可以包含一个或多个构件。

(3)活动(Activity)。
活动用于跟踪完成一项开发任务的工作,它记录了开发工程师为完成或交付某个开发任务而创建或修改的文件集合(变更集)。

(4)基线(Baseline)。
构件的一个版本就是一个基线。基线用于标识构件中各个元素的某个版本。它代表了构件在项目开发某一特定阶段的一个版本。

基线是下一步开发的出发点(见表 1)。基线的变更受到严格控制。在一个软件开发过程中,上一个基线加上新的内容形成下一个基线。

表 1 软件开发过程中的主要阶段基线及相应的软件配置项列表

阶段	阶段基线	配置项
设计	需求基线	开发计划文档、需求文档、测试计划
	概要设计基线	系统设计、系统测试文档
	详细设计基线	程序结构详细设计、数据库设计、单元测试
实现	编码基线	源代码、可执行程序、相关数据
	测试基线	测试报告、测试用例
	产品基线	对外发布的产品

设计阶段基线,通常以文档的形式存在。实现阶段基线,包括文档、代码和可执行程序。

(5)工作区和流。
工作区是与变更相关的开发区域。而在 UCM 中工作区则由视图和流组成。

流是一种 ClearCase 对象,UCM 定义了两种流,分别是开发流(Development Stream)和集成流(Integration

Stream)。开发流主要是那些涉及单个开发工程师的工作的流;而集成流则是那些涉及对一个项目中所有开发工程师的活动进行合并的流。

每个项目通常有一个集成流和多个开发流。对于大型项目可以增加测试流和基线流,见表 2。

表 2 流与工作区

流	工作区
开发流	小组内部共享工作区
	个人工作区
集成流	共享工作区
测试流	测试工作区
基线流	受控工作区

3 软件开发过程中参与配置管理的角色

参与配置管理的相关角色:项目经理(PM)、软件系统工程师(SE)、配置管理员(CMO)、持续集成工程师(CIE)、软件开发工程师(DEV)、软件测试工程师、质量保障工程师(QA)、资料工程师。

其中,项目经理、软件开发经理、软件测试经理、系统工程师、配置管理员、质量保障工程师组成配置控制委员会(Change Control Board, CCB)。

CCB 的职责:审核各种变更申请;建立、更改基线的设置;定制访问控制;负责指导配置管理的各项活动,为项目经理的决策提供建议。

4 软件开发过程中配置管理的工作流程

软件项目开发可以分为三个阶段:计划、开发和产品维护。对于配置管理来说,后两个阶段所涉及的活动基本相同,可以合并为开发和维护阶段。

4.1 UCM 工作流程

ClearCase 提供 UCM^[6]和 Base 两种软件开发模式。Base ClearCase 是基于文件的管理方式,UCM 方式则主要采用 Activity 方式进行项目管理。UCM 在 Base ClearCase 的基础上提供了高层抽象,简单易用。

ClearCase UCM 模式是一种基于 Activity 的开发过程,项目经理负责创建项目。一个项目通常包括公共区、集成测试区和多个私有工作区。项目经理负责维护项目公共区域,集成测试工作人员在集成测试区工作,开发工程师在私有工作区独自进行工作。

UCM 工作流程包括三个软件循环开发过程:项目管理循环、开发循环和集成测试循环,如图 1 所示。

4.2 编制配置管理计划

配置管理计划主要描述配置管理的组织机构、职责和接口、版本库的结构、配置管理环境的建立、访问权限设置、基线的建立、各个里程碑处所提交的工作产

品及提交时间、变更控制及配置状态统计等内容。

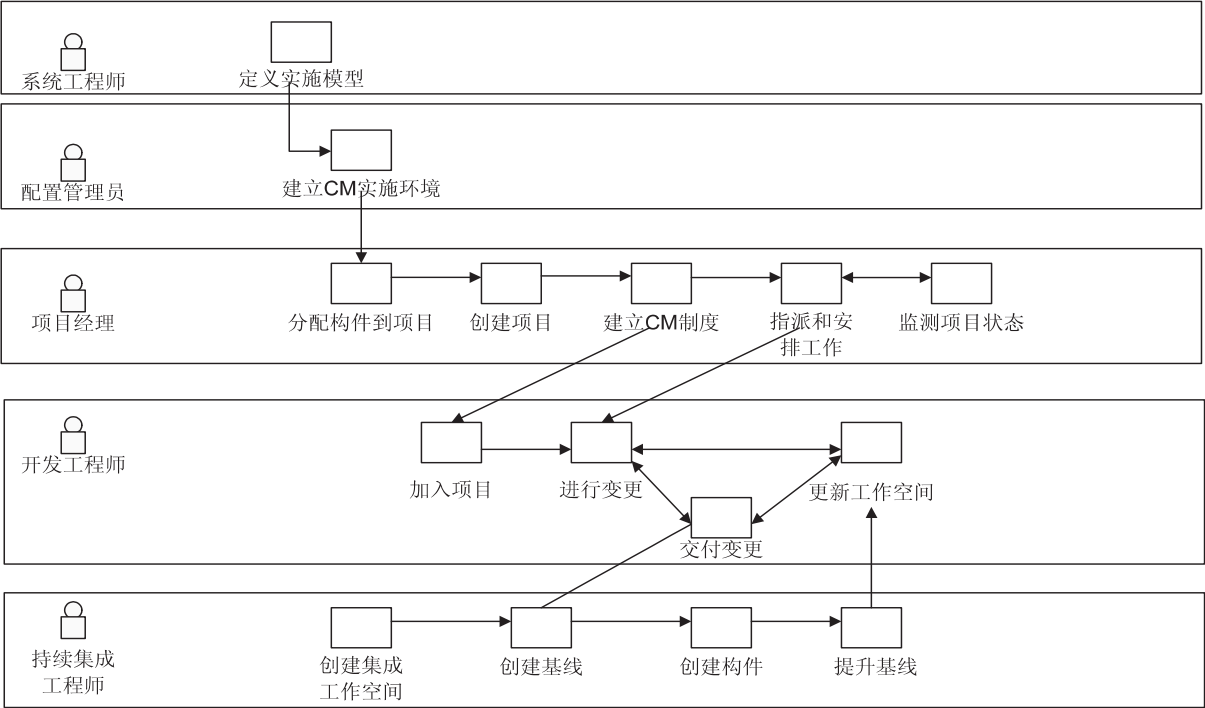


图 1 UCM 工作流程图

4.3 权限管理

ClearCase 本身不具有管理用户的功能,而是通过 Windows 系统中的域来管理用户。与 Unix 文件管理系统类似,每个 VOB 里的目录和文件的权限控制都基于 owner、group 和 other,访问模式分为 r(读)、w(写)、x(执行)。

ClearCase 安装完成后,会自动创建一个具有超级权限的用户组 CC 组。配置管理员属于该组,对 ClearCase 管理具有最高的权限。ClearCase 按组来设置权限,每个用户可以属于几个组,但只能属于一个主组。对于一个 ClearCase 对象,其 owner 和配置管理员有最高权限。配置管理员负责项目组的权限管理,见表 3。

4.4 ClearCase 控制管理下的开发工作

配置管理员首先建立一个 PVOB,然后在这个 PVOB 下面创建子 VOB。项目经理将源文件导入 VOB,通知、指导开发工程师加入创建的项目。开发工程师在客户端的 ClearCase Explorer 中,用 Join Project 来加入到这个 UCM 项目。每个开发工程师创建自己的工作区(开发流和开发视图)。

在客户端 ClearCase Explorer 选择“MyActivities”,开发工程师可以看到自己要处理的 Activity。开发工程师针对被分配的活动(工作任务)进行工作。

开发工程师每天主要工作如下:

根据集成流提供的新基线更新(Update)工作区代码。

检出(Check out)自己需要的文件开始工作。提供新的源代码或对原来的代码进行改进,同时对代码进行单元测试,静态走码检查,冲突处理和本地构建工作。现在一些开发工程师采用测试驱动开发的方式编写代码。

表 3 软件项目开发过程中的权限管理

组	成员	管理对象	权限
CC 管理组	配置管理员	拥有最高权限,管理整个 VOB	r、w、x
	项目经理和 其他管理人员	各类文档 管理文档	r r、w
开发组	开发组长	管理文档写的权限	r
		本组负责的子系	r
	开发工程师	自己负责的模块	r、w、x
		对自己负责的模块	r、w、x
测试组	测试组长	开发文档	r
		所有开发文档、管理目录	r
	测试工程师	测试目录	r、w、x
		自己负责的测试模块	r
集成组	持续集成工程师	测试目录下的测试用例和文件	r、w、x
		集成和发布目录	r、w、x
		其相关他目录	r

最后将评审过的代码检入(Check in)到集成流。开发工程师向 VOB 提交代码时,要添加注释、说明、CR 单号、修改原因等,以保证可追溯。

4.5 开库和锁库

配置管理员负责每天版本库的开库和锁库,开库和锁库主要是针对集成流。每天开库将集成流相应的

权限赋予项目组成员。开发工程师完成代码提交后,锁库取消项目组成员的相应权限,仅保留配置管理员和持续集成工程师的权限,以便持续集成工程师完成集成构建工作。

4.6 持续集成建立编码基线

配置管理员锁库后,持续集成工程师从集成流检出新提交的代码。启动 CI 服务器对新提交的代码结合原有项目资源开始构建工作。采用增量构建,快速创建一个新版本;自动执行一系列测试(单元测试、静态测试和一些耗时的代码审查任务等)。自动执行冒烟测试(在集成测试环境中自动执行核心的测试用例),来验证新提交的代码有没有对项目造成破坏。若构建成功,建立新的编码基线,通过电子邮件通知项目经理和开发工程师。若构建失败,不更新基线,通过电子邮件通知项目经理和相关的开发工程师对提交的代码进行及时修改。

4.7 构建内部转测试版本建立测试基线

通常选择每周执行一次构建内部转测试版本。持续集成工程师从集成流取得新提交的代码后,执行全构建任务。全构建任务执行时间较长,有些工作还需要手动执行。启动 CI 服务器执行所有的构建任务,通常包括环境清理、更新代码、编译、测试、代码审查、打包、部署、性能测试等。生成的软件版本放在测试流中并标记版本号,提交转测试流程。由测试经理组织进行版本验证;测试工程师进行各种测试,包括功能测试、性能测试等,测试中发现问题,知会项目经理和开发工程师及时处理。完成内部转测试版本测试,就建立了新的测试基线,同时也建立了新的编码基线(Tested),和每日构建建立的基线相比由内部转测试版本建立的基线等级要高一些。配置管理工程师执行基线化的操作,并发布基线建立的报告。

4.8 变更管理

变更管理是软件配置管理的三大核心之一。ClearCase 通过与 ClearQuest 集成提供变更请求管理和缺陷跟踪管理。

ClearCase 同 ClearQuest 的集成是通过 ClearCase 的版本库(VOB)关联 ClearQuest 的数据库来实现。ClearQuest 可以捕获和跟踪各种类型的变更请求(例如产品缺陷、功能增强、文档更新等)。ClearQuest 提供最佳的变更请求管理解决方案。这使得开发团队可以更容易管理各种类型的变更。UCM 模式中的变更和缺陷管理是基于活动来完成的,整个工作流程由 ClearQuest 提供。

4.8.1 变更请求管理

在软件研发过程中会遇到 4 类变更请求:需求变更、计划变更、设计变更以及工程变更。

变更管理流程:

以上提到的 4 类变更请求,都采用一个通用的变更管理流程。

主要由以下 6 个步骤组成:

(1)提交:首先提交变更申请,在申请中详细描述变更的范围、原因和依据,变更的优先级,变更的影响,建议的变更方案和替换方案。

(2)评估:CCB 收到变更请求之后,对请求进行评估和分析,识别和评估受影响的所有配置项,确定修改方案、修改人、修改版本、修改时间、验证方式和验证人。必要时邀请相关受影响的工程师参与评审。

(3)决策:接受的变更请求,由配置管理员负责授权。

(4)实现:修改人执行变更并验证。

(5)验证:之后,由 CCB 指定的验证人对变更进行验证。

(6)完成:验证通过后,CCB 确保修改验证合格的配置项合入版本库,然后关闭变更请求并发布。

ClearCase UCM 通过 ClearQuest 对变更活动进行提交、评估、决策、实现、验证和完成,从而实现变更的自动化管理。运用 ClearQuest 可以方便地跟踪、管理相关的变更请求,充分掌握变更的现状。

4.8.2 缺陷跟踪管理

ClearQuest 对缺陷处理过程进行自动跟踪,从 BUG 提交到关闭,记录了 BUG 所有的改变历史。项目经理可以及时查询各种 BUG 的状况,可以在 ClearQuest 中创建缺陷处理的活动,把任务分配给开发工程师处理。开发工程师在自己的开发视图中,通过 My-Activities 菜单选项可以看到分派给自己的任务,根据测试人员对缺陷的描述,修复相应缺陷。项目经理可以查看缺陷处理的状况,跟踪缺陷修复的过程。

4.9 配置状态发布

配置状态发布是建立与维护配置项的记录;为相关工程师提供准确的配置信息;通过记录和报告变更请求的状态,为产品的质量、进度、趋势等提供数据。

配置状态发布的主要内容包括:配置项状态、基线发布情况、版本发布情况、变更管理情况、版本库以及权限情况。

4.10 版本管理

版本是产品或者解决方案在不同时间段的特性集合,一个产品或解决方案可以有多个版本。

4.10.1 版本管理流程

版本包括内部转测试版本(Tested)以及对外发布版本(Released),项目开发任务完成后,测试流中最终的版本成为对外发布版本(Released),提供给客户使用。

版本管理的过程如下:

(1)项目经理将制定评审版本计划,包括内部转测试版本以及对外发布版本。

(2)开发工程师和资料工程师完成源代码开发和文档编写。

(3)持续集成工程师对代码进行每日构建工作,每周完成内部转测试版本构建。

(4)提交转测试版本,由测试经理组织进行版本测试和验证。

(5)开发活动完成之后,由配置管理员准备最终的产品版本,以及版本文档、产品文档、版本光盘包等;提供给测试经理,组织进行最后的版本测试和验证。

(6)测试和验证通过之后,配置管理员提交对外发布版本,由项目经理组织各领域的代表评审,确认版本符合发布要求。

(7)进行正式的版本发布流程,版本最终传递给客户。

版本的管理流程如图2所示。

4.10.2 配置管理的移交

配置管理的移交活动通常发生在产品开发结束的GA点(即产品由开发状态转为维护状态之后)。配置管理员需要完成以下工作:

(1)整理、归档数据、代码(包括可执行文件)和文档;

(2)归档或删除项目配置库;

(3)移交产品配置库。

5 基于 ClearCase 配置管理的持续集成

现在软件开发通常采用持续集成(CI)^[7-14]技术。CI服务器可以自动检查动态库的变化,设定时间区间,定时自动完成集成构建工作,并以邮件的形式向整个开发团队反馈集成构建结果。几个小时就可以完成一次集成构建,一次构建通常可能包含编译、测试、审查和部署,以及其他一些事情。本节以ICP-CI服务器为例介绍持续集成。

5.1 安装持续集成工具

(1)根据开发项目的代码规模、编译环境、编译工具来选择CI服务器的CPU、内存等硬件配置和操作系统。一套CI服务器很可能同时承担多个软件产品的持续集成工作。

(2)ICP-CI服务器系统有两种常见的部署形式,分别是单机式和分布式。

单机式指的是整套CI服务器系统由一台服务器组成,该服务器自主完成持续集成任务,这种系统适用于代码量小于一百万行的开发团队。

分布式指的是整套CI服务器系统由一台主控服

务器和多台代理服务器组成,将持续集成任务下发到各个代理服务器上,任务完成之后将任务结果和日志文件发回到主控服务器,这种方式适用于代码量大于百万行的开发团队。

(3)安装持续集成工具ICP-CI,根据编译环境的不同持续集成工具ICP-CI可以分为不同的版本。在Windows环境下,主控服务器部署工具为ICP-CI-Windows-Master,代理服务器部署ICP-CI-Windows-Agent。在Linux环境下,主控服务器部署工具为ICP-CI-Linux-Master,代理服务器部署ICP-CI-Linux-Agent。持续集成工具ICP-CI是跨平台的,主控服务器和代理服务器可以选择不同的操作系统。通常主控服务器选择Windows环境。

(4)ICP-CI工具在安装的过程中需要解压安装包安装。对于ICP-CI-Windows-Master,需要进入ICP-CI-Windows-Master下的master目录中,执行批处理启动MySQL数据库之后,启动ICP-CI的网页版页面。对于ICP-CI-Windows-Agent和ICP-CI-Linux-Agent将安装包解压之后,需要将安装ICP-CI-Windows-Master的服务器IP地址配置到agent目录下conf文件夹对应的配置文件中,再执行批处理脚本启动代理服务器的CI进程。

5.2 安装 ClearCase 客户端工具和创建代码视图

CI服务器需要安装ClearCase客户端工具。完成ClearCase客户端工具的安装之后,就可以根据配置管理员提供的产品代码动态视图和视图规则,使用ClearCase客户端工具创建视图,从版本库下载代码到代理服务器的指定路径下。

5.3 搭建构建工程

打开ICP-CI网页的页面,创建产品的构建工程,使用ANT脚本完成出包脚本。构建工程创建成功之后,分别配置更新代码、静态检查、进程编译、出包、用例测试等步骤。

5.3.1 代码更新

在ClearCase版本库锁库之后,将开发工程师提交到ClearCase版本库的所有代码通过命令行工具中的更新命令“Cleartool update”更新到代理服务器的视图中。

5.3.2 静态检查

对于产品的代码是使用C/C++来编码的,采用的静态检查工具为Pclint。根据产品的模块编译分成不同的*.Int文件,用模块名用来命名*.Int文件。*.Int文件中需要配置该模块的相关代码路径。在配置对应模块的Pclint任务时,将*.Int文件配置进去。

5.3.3 编译

各模块的编译脚本和编译类型需要在ICP-CI的

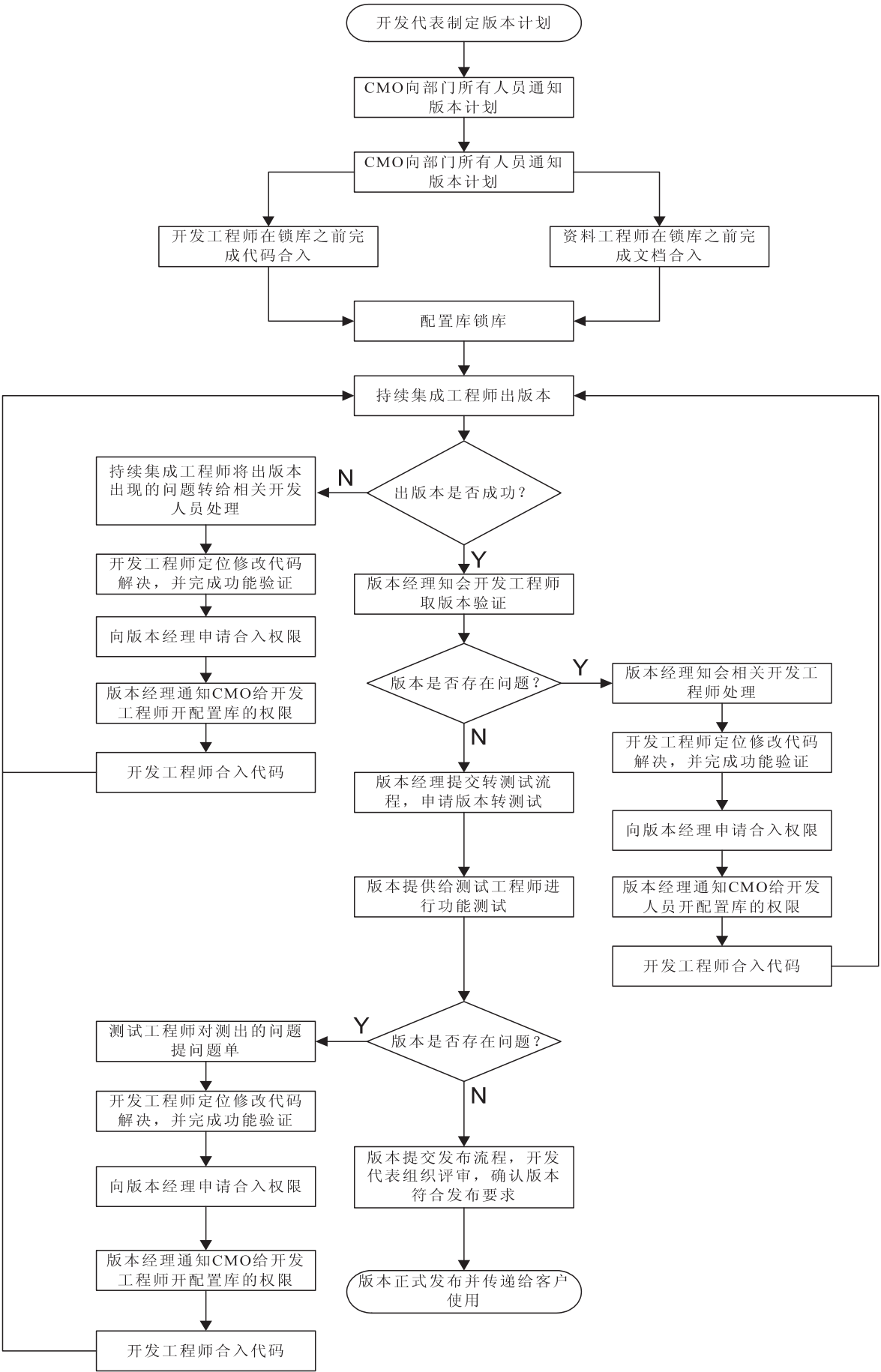


图 2 版本的管理流程

任务页面上完成配置。产品代码在编译过程中可以查找开发工程师在编码过程中没有关注到的编译错误。编译为下一步的出包步骤生成进程文件。

5.3.4 出 包

出包任务在编译任务完成之后,根据全部编译任务提供的进程文件,在 ICP-CI 的任务页面上完成对出包的 ANT 脚本配置,生成版本包。将版本包部署到相应的目录下。

5.3.5 测试用例

在 ICP-CI 页面完成对测试用例任务的配置。在测试用例环境下对版本包进行自动化测试,完成版本包的初步测试。

6 典型案例

某公司有一个软、硬件结合的大型软件开发项目,采用的 ClearCase 版本是 7.0.1。持续集成主控服务器操作系统为 Windows 2003 Server,代理服务器操作系统为 Windows 7。参与持续集成的服务器共 20 台,1 台主控服务器和 19 台代理服务器。采用批处理脚本和 ANT 脚本编程完成软件的持续集成构建工作(完成进程编译、静态检查和版本出包等任务)。工作实践表明,采用基于 ClearCase 的配置管理和持续集成有助于提高软件质量,也便于项目经理和开发工程师及时了解工作进度和解决存在的问题。

7 结束语

长期的工作实践表明,配置管理在软件开发过程中占有重要地位。采用持续集成进行自动化构建可以使配置管理工作更有效率,有助于减少开发中埋藏 BUG 的风险,从而提高软件质量;同时也便于项目经理和开发工程师及时了解工作进度和解决存在的问题。应用软件的配置管理工作做好了,将很大程度上提高软件质量,降低软件开发成本,推动软件产业的健

康发展。

参考文献:

[1] 刘文红. CMMI 项目管理实践[M]. 北京:清华大学出版社,2013.

[2] 刘江华,王 立,马 玲,等. 软件开发过程与配置管理:基于 Rational 的敏捷方案设计与应用[M]. 北京:电子工业出版社,2011.

[3] 袁肃蓉,王 萍,黄万民,等. 基于 ClearCase 的软件配置管理环境的规划和实施[J]. 海南大学学报:自然科学版,2009,27(1):54-59.

[4] Bellagio D, Milligan T. Software configuration management strategies and IBM rational ClearCase: a practical introduction [M]. 2nd ed. USA: IBM Press, 2005.

[5] Lee K A. IBM rational ClearCase, ant and CruiseControl[M]. USA: IBM Press, 2006.

[6] Tykal J. Best practices for using composite baselines in UCM [R]. [s. l.]; PrenticeHall, 2004.

[7] McConnell S. Daily build and smoke test[J]. IEEE Software, 1996, 13(4):143-144.

[8] 罗时飞. 敏捷持续集成(CruiseControl 版): 高效研发之道[M]. 北京:电子工业出版社,2008.

[9] 韩万江,姜立新. 软件项目管理案例教程[M]. 第 2 版. 北京:机械工业出版社,2009.

[10] Duvall P M, Matyas S, Glover A. 持续集成软件质量改进和风险降低之道[M]. 北京:电子工业出版社,2012.

[11] 李 进. 某公司软件持续集成改进的分析设计及实施[D]. 北京:北京邮电大学,2012.

[12] 徐仕成,杨邦荣. 基于 CruiseControl 的持续集成实现方案[J]. 计算机与数字工程,2007,35(4):169-171.

[13] 吴 奕. 软件配置管理工具在大型网站开发中的应用[D]. 上海:复旦大学,2011.

[14] 杨宏英,林长松. 基于 IBM Rational ClearCase 的配置管理及应用:中国软件行业协会[C]//CSSPI2006 第五届中国系统与软件过程改进年会论文集. 北京:出版地不详,2006:156-162.

(上接第 9 页)

[4] Alur R, Dill D L. A theory of timed automata[J]. Theoretical Computer Science, 1994, 126(2):183-235.

[5] Spivey J M. The Z notation: a reference manual[M]. UK: Prentice Hall International Ltd, 1992.

[6] 戎 玫,张广泉. 模型检测新技术研究[J]. 计算机科学, 2003, 30(5):102-104.

[7] 朱维军,刘保罗,周清雷. 时间自动机与信号自动机的互模拟算法[J]. 华南理工大学学报:自然科学版,2008,36(5):38-42.

[8] 李 娜,姚从军. 互模拟的一些基本性质[J]. 云南师范大学学报:哲学社会科学版,2010,42(5):68-73.

[9] Alur R. Techniques for automatic verification of real-time sys-

tems[D]. Stanford: Stanford University, 1991.


[10] Alur R, Courcoubetis C, Dill D. Model-checking for real-time systems[C]//Proceedings of fifth annual IEEE symposium on logic in computer science. [s. l.]; IEEE, 1990:414-425.

[11] 钱俊彦,赵岭忠,古天龙. 一种基于时间自动机的时钟等价性优化方法[J]. 计算机工程,2005,31(18):71-73.

[12] 狄杨思. 形式规范的自动验证算法的研究[D]. 南京:南京航空航天大学,2012.

[13] Cao Z, Wang H. Hybrid ZIA and its approximated refinement relation[C]//Proceedings of the 6th international conference on evaluation of novel approaches to software engineering. Beijing, China; [s. n.], 2011:260-265.

基于 ClearCase 的软件配置管理与持续集成

作者：[姜文](#)，[刘立康](#)，[JIANG Wen](#)，[LIU Li-kang](#)
作者单位：[西安电子科技大学 通信工程学院, 陕西 西安, 710071](#)
刊名：[计算机技术与发展](#)
英文刊名：
年，卷(期)：2016(1)

引用本文格式：[姜文](#). [刘立康](#). [JIANG Wen](#). [LIU Li-kang](#) [基于 ClearCase 的软件配置管理与持续集成](#)[期刊论文]-[计算机技术与发展](#) 2016(1)