

# 软件配置管理中的基线问题研究

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:**基线是软件配置管理中的一个重要概念。基线有多种不同的定义,是一个比较模糊的工程概念。每种基线定义只反映了基线内涵的某个侧面,需要分析研究各种基线定义之间的区别和内在联系。文中结合配置管理的基本概念,分别归纳并给出了五种不同的基线定义;叙述了基线的层级、软件项目的阶段基线、实现阶段基线的水平等级和粒度;之后介绍了基线管理的过程和基线管理在软件开发中的优点。通过叙述各种基线的相关概念,理清了各种基线概念之间的关系和内在联系。集中介绍了基线相关的技术问题,有助于更好地应用和管理基线。工作实践表明,深入理解基线概念的内涵有助于掌握各种软件配置管理工具的区别和共同之处,从而在各种环境下做好软件开发的配置管理工作。

**关键词:**配置管理;配置项;版本;版本库;基线

中图分类号:TP391.41

文献标识码:A

文章编号:1673-629X(2016)06-0006-05

doi:10.3969/j.issn.1673-629X.2016.06.002

## Research on Baseline in Software Configuration Management

JIANG Wen, LIU Li-kang

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

**Abstract:** Baseline is an important concept in software configuration management. There are many different definitions of baseline, which is a vague concept of engineering. Each baseline definition reflects the connotation of the baseline from a profile, only needing to analyze the difference and inner link for various definitions between the baseline. In this paper, combined with the basic concept of configuration management, five different baseline definitions are summarized respectively. It describes the baseline level, baseline phase of software project, and baseline levels and granularity in implementation phase. Then it introduces the baseline management process and the advantages for baseline management in software development. Through a variety of related concepts of the baseline, the relationship between the concept of baseline and inner link is clarified. It focuses on the baseline related technical problems, which contributes to application and management of the baseline. Practice shows that to understand the connotation of the concept of baseline in depth helps to grasp the difference and common of software configuration management tools, doing a good job of software configuration management in a variety of circumstances.

**Key words:** configuration management; configuration items; version; repository; baseline

## 0 引言

软件配置管理是对软件配置项的管理。为了使软件配置项能够随项目的进展有序、准确的变化,为项目提供有效的参考,引入了“基线(Baseline)”的概念。在一个开发阶段结束或一组功能开发完成后,要对相应的配置项进行基线化,形成各类基线。

基线是软件配置管理<sup>[1-9]</sup>中的一个重要的基本概念。基线有多种不同的定义,也是一个比较模糊的工程概念。文中介绍了配置管理的基本概念,分别归纳

并给出了五种不同的基线定义;叙述了基线的层级、软件项目的阶段基线、实现阶段基线的水平等级和粒度;介绍了基线管理的过程和基线管理的优点。各种基线定义都反映基线内涵的某个侧面,文中厘清了各种基线定义的区别和内在联系。

## 1 配置管理的基本概念

### 1.1 配置项(Configuration Item)

配置项是配置管理的基本单位,可以分为三个主

收稿日期:2015-09-05

修回日期:2015-12-09

网络出版时间:2016-05-25

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,硕士研究生,CCF 会员,研究方向为图像处理与分析、文字信息分析处理、数据库应用和软件工程;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.20160525.1706.024.html>

要类别:计算机程序(源代码和可执行程序)、文档和数据。配置项包含了软件开发过程中产生的各种信息。

配置项的识别是配置管理活动的基础,也是制定配置管理计划的重要内容。

### 1.2 元素( Element)

在配置管理工具 ClearCase 中,元素是版本控制下的文件和目录。文件元素可以是文件系统中的任意文件。目录元素可以包含文件元素及其他目录元素。

配置项和元素之间的关系:元素是软件配置项的载体,配置项以元素的形式保存在计算机中。配置项是元素存储的内容。

### 1.3 版本( Version)

版本亦称配置标识,是实例化了的配置项。在软件开发过程中,配置项的版本随项目进展而不断变化,会被赋予不同的版本号。

版本的演变一般有两种方式:串行演变和并行演变。版本的这两种演变形式通常结合在一起,形成更为普通的带分支的版本树。版本树反映了项目开发演变的历史。每个元素记录了它所代表的文件或目录的版本。

### 1.4 分支( Branches)

分支描述了元素的树状版本结构;每个元素都有一个主分支,代表开发的主线,可以有多个独立开发线的子分支,子分支还可以再有子分支。

### 1.5 版本库( VOB)

版本对象库( Version Object Base, VOB)简称版本库( VOB),也称为配置管理库。版本库是软件配置管理系统的核心,用来存储文件、目录和元数据的数据存储库。版本库以目录树结构形式的文件系统来保存信息。它存储了版本控制下所有的文件、目录和元数据等。

通常逻辑上版本库可以分为开发库—DL( 动态库、程序员库、工作库)、受控库—CL( 静态库、主库)、产品库—PL。版本库的实际物理存储结构可根据软件项目具体情况来确定。受控库和产品库是由配置管理员管理,所有的配置项经评审通过后才能入库。

版本库的结构见表 1。

### 1.6 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。

### 1.7 软件项目

简单地说,项目就是实现某个目标的工作任务。一个软件项目要素包括软件开发过程、软件开发结果、

软件开发资源和软件项目的客户。一个软件项目可以有多个版本库。

表 1 版本库的结构

库结构	目录	配置项
DL	开发文档	项目计划、需求说明书、概要设计、详细设计、数据库设计
	源代码	软件源代码、相关数据
	测试	测试计划、测试用例、测试报告
	其他产品私人工作区	项目开发过程中的其他软件产品 Web 远程开发人员的工作区
CL	基线库	设计阶段基线
		编码基线
		测试基线
	其他库	其他受控的软件产品
PL	用户文档	用户手册、系统管理员手册、培训资料
	产品	发布的软件产品
	产品环境	产品环境相关的软件产品

### 1.8 流( Stream)

流是一种 ClearCase 对象,UCM 定义了两种流:开发流( Development Stream)和集成流( Integration Stream)。开发流主要是单个开发工程师的工作流;集成流则是对项目中所有开发工程师的活动进行合并的流。

基线和流的相互关系:基线由流产生,流使用基线作为其配置。

## 2 基线( Baseline) 的定义

基线<sup>[10-12]</sup>是软件配置管理中的一个重要的基本概念,有多种定义,大致分为如下几类。

### 2.1 基于产品

这种定义方式比较宽泛,不仅仅适用于软件配置管理。

IEEE 的基线定义:已经通过正式评审和批准的某规约或产品,它可以作为进一步开发的基础,并且只能通过正式的变更控制过程进行变更。

基线定义 1:基线是某个规约或者产品,具有三个特点:

- (1)通过正式评审并得到认可;
- (2)下一步工作的基础;
- (3)基线变更需要通过严格的变更管理过程。

### 2.2 基于配置项

基线定义 2:在软件配置管理过程中,一个配置项或一组配置项在不同时间点,通过正式评审而进入受控的一种状态,就是基线。

一些配置项成为相对稳定的逻辑实体,这个过程称为“基线化”。基线建立之后,该基线的内容被“冻结”,不能随意修改。基线是下一步开发的出发点和

参考点。对基线进行修改将严格按照变更管理的过程进行。作为配置管理的基础,基线保证了后续开发活动所需信息的稳定性和一致性。

2.3 基于版本

基线定义 3:基线是项目文档或者源代码等文件的一个稳定的版本。基线有名称、版本、标识符、日期等属性。它是进一步开发的基础,其变更必须通过正式的变更程序。

如果某一产品版本被定为基线,那么它就被冻结,要想变更基线必须建立一个新的版本。

2.4 基于里程碑 (Milestone)

里程碑就是开发过程中的“阶段”。“阶段”强调的是过程,而“里程碑”则强调过程的终点和终点的标识。

基线定义 4:基线是软件开发过程中的一个里程碑,其标志是一些软件配置项的交付,并且这些配置项通过技术审核获得认可。这些配置项纳入基线管理,作为后续工作的标准,标志着项目可以进入下一个过程。

2.5 UMC 构件的基线和复合基线

在 ClearCase UCM<sup>[11-17]</sup> 中,一个构件表示一组元素,基线表示构件中的一组版本。基线确定了一个构件中的每个元素的一个版本,如图 1 所示。

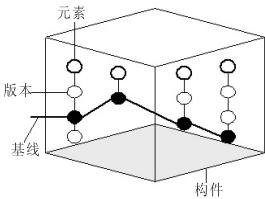


图 1 基线

基线定义 5:基线是一个构件在某一特定时刻一组元素的集合,包含了该时刻所选取元素版本的集合,记录了软件开发过程中完成的工作。

UCM 工程项目只有一个构件,该构件的基线就是该工程项目的基线。一般来说,工程项目可能有多个构件,工程项目基线包含每个构件的特定基线。

复合基线:如果工程项目中构件较多,采用复合基线进行管理。复合基线选取每个构件中的特定基线来表示整个工程项目的基线。

基线的定义是一个实用的工程定义,以上五种定义有着内在的联系,反映了基线内涵的某个侧面。在一些书中通常将几种定义糅合在一起作为基线的定义。

3 基线的层级

基线分为三个层级(项目基线、构件基线、单项基

线),如表 2 所示。

表 2 基线的层级

层级	类型	载体	使用者
1	项目基线	版本库的受控库	项目组
2	构件基线	构件中全体元素特定版本的集合	开发团队
3	单项基线	元素的版本	开发人员

(1)项目基线是项目组使用的基线,保存在版本库的受控库 (CL) 中。一个软件项目可以有多个版本库。每个版本库可能由多个构件组成,但至少包含一个构件。

(2)构件基线是供开发该构件的团队使用。

(3)单项基线是指对单个配置项建立的基线,供开发该配置项的开发人员使用。

4 项目阶段基线

软件开发过程中的主要阶段基线如表 3 所示。

表 3 软件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	计划基线	开发计划
	需求基线	需求说明书
	概要设计基线	系统架构设计、模块设计
	详细设计基线	程序实现的详细设计、数据库设计
实现	编码基线	代码、单元测试报告、可执行程序、数据
	测试基线	集成测试报告、系统测试报告、测试用例
	产品基线 (发布基线)	对外发布的产品

计划基线、需求基线、概要设计基线和详细设计基线是设计阶段基线,以文档的形式存在。编码基线、测试基线和产品基线是实现阶段基线,包括文档、代码和可执行程序。常见的产品版本和补丁发布,对应产品基线。

对于需求、设计、编码、测试来说,需求是设计的上游,设计是编码的上游,编码是测试的上游。如果上、下游之间采用基线的方式,也就是明确需求基线、设计基线、代码基线、测试基线,不仅能够为下游环节提供稳定的工作基础,而且分别记录需求、设计、编码工作产生的缺陷情况,能够明确各自的职责。

5 实现阶段基线的水平等级和粒度

5.1 实现阶段基线的水平等级

每个基线有一个相应的级别,通过级别划分,可以大致了解基线的状况。ClearCase 中实现阶段基线的等级分为 5 级 (Rejected、Initial、Built、Tested、Released)。

(1)Rejected:没有通过测试,或者已经被新基线取代的基线。

(2) Initial:软件开发初始阶段确定的基线。

(3) Builted:集成构建生成的基线,通常采用增量构建。

(4) Tested:进行全量构建生成内部转测试版本,提交测试组进行全面的测试工作,通过后产生测试 (Tested) 基线。

(5) Released:代码开发工作完成后,测试 (Tested) 基线通过全面测试验证、评审后升级为发布基线。

注:

(1) 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。

(2) 增量构建:仅对新提交的代码进行编译,利用原有项目资源快速构建一个新的版本。构建成功后生成新的基线(增量基线)。

(3) 全量构建:对全部代码进行编译、链接,构建生成新的基线(全量基线)。

5.2 基线的粒度

现在软件开发通常采用持续集成 (Continuous Integration) 技术。持续集成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。几个小时就可以建立一条新的基线。构建的频率越高产生的基线粒度越小。

频繁集成构建有利于及时发现和解决各种问题,加快了代码流动,减少了任务间的等待时间;但是加大了提交代码和测试的工作量,需要采取一种折中的选择。

通常采用每日构建生成 Builted 基线,每周构建生成测试 (Tested) 基线。

6 基线管理

软件开发过程中与软件基线管理相关的工作人员如下: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)、资料开发工程师。

其中,项目经理、质量保障工程师、配置管理员、开发组长、测试组长、系统工程师组成了产品配置控制委员会 (Change Control Board, CCB)。

与基线相关的配置管理活动如下所述。

6.1 基线的建立

设计阶段,软件系统工程师完成软件配置项的设计工作,经过评审后成为设计基线。

在实现阶段,持续集成工程师从动态库检出代码,生成新的代码基线。

配置管理员负责将新的基线加入受控库中,进行管理和维护。

6.2 基线标识和发布

基线由配置项组成,标识基线首先应该标识配置项。

配置项 (CI) 应分配唯一标识。标识规则可参考以下示例:

CI 标识=项目标识/CI 类型/CMI 名称/CI 版本

例如:Prj1/DOC/用户手册/1.00

基线标识包括项目名称、项目标识、基线名称、版本、发布时间和所包含的配置项清单。

基线经过 CCB 审批后就可以发布(基线发布表可参考图 2)。

项目名称		项目标识	
基线名称		发布时间	
基线版本		前向基线版本	
基线组成列表			
配置项名称		标识	版本
审批意见	审批人: 审批时间:		
相关方签字			

图 2 基线发布表单示例

配置管理员负责基线的标识和发布工作。基线只有经过审核,确认其是正确、有效、一致时,才能正式发布。基线发布后开始正式使用。

6.3 基线变更

基线变更是对基线的配置项的更改,必须通过严格的变更步骤,在配置管理系统的变更控制下进行。

主要由以下 4 个步骤组成:

(1) 提交变更申请:基线变更的范围、变更的优先级、变更的影响、变更方案。

(2) 变更评估:CCB 对基线变更请求进行评估、分析和评审。

(3) 变更审批:若接受变更请求,则确定变更方案。在项目经理的指导下,由配置管理员负责授权。

(4) 变更实现:执行基线变更并验证;验证通过后,合入受控库中的基线库;关闭变更请求并发布。

配置管理员对基线变更进行审核,将审核结果和当前配置情况提交项目经理批准。批准后,配置管理员执行基线化的操作,并发布基线建立的报告。

7 基线管理的优点

软件开发过程中实施基线管理主要有如下优点:

(1) 提供统一规范和同步开发工作:基线为后继的开发工作统一标准规范,确保团队开发工程师工作保持同步。

(2) 及时处理各种问题:采用持续集成技术,选择合适的基线粒度,有助于项目经理及时了解工作进度



和存在的问题,及时处理。

(3)配置项关联性:通过项目基线确保配置项之间正确的关联状态。保证设计满足需求方案、产品的实现符合设计要求。

(4)版本差异分离:新项目可以从基线的版本序列中选择合适的版本开始工作,与原项目(主要分支上)进行的变更实现隔离。

(5)变更可回退性:如果变更有问题,可采用取消变更的方式,回退至稳定版本。

(6)问题可追踪性:通过基线版本序列可以追踪产品的各种问题,可以重新建立基于某个版本的配置,可以重现软件开发过程中的软件缺陷和各种问题。

## 8 结束语

各种软件配置管理工具有不同的设计理念,但都采用基线作为配置管理的基础。基线管理能够明确各个开发阶段的职责,使上游阶段为下游阶段提供稳定的工作基础;为软件产品的维护和新版本开发提供了良好的基础。各种基线定义都反映基线内涵的某个侧面,需要研究各种基线定义的不同和内在联系。文中厘清了各种基线相关概念之间的关系,有助于更好地应用和管理基线。工作实践表明,深入理解基线概念的内涵有助于掌握各种软件配置管理工具的区别和共同之处,从而在各种环境下做好软件开发的配置管理工作。

### 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] 王安生. 软件工程化[M]. 北京:清华大学出版社,2014.
- [3] 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.

- [4] 董越. 未雨绸缪—理解软件配置管理[M]. 第2版. 北京:电子工业出版社,2012.
- [5] 董越. 软件集成策略—如何有效率地提升质量[M]. 北京:电子工业出版社,2013.
- [6] 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.
- [7] 相玉娟. 基于变更管理的持续集成研究与应用[D]. 合肥:合肥工业大学,2009.
- [8] 刘海,张卫民,任开军. 一种网格环境下的软件配置管理系统[J]. 计算机工程与科学 2007, 29(8) 105-107.
- [9] 陈帆,尹黎明,肖卫军,等. 软件配置管理中的分支管理与协作机制研究[J]. 计算机工程与科学,2005, 27(1) 70-73.
- [10] 周念. 基于 SVN 的软件配置管理的应用研究[D]. 武汉:武汉理工大学,2013.
- [11] Aiello B, Sachs L. 配置管理最佳实践[M]. 北京:人民邮电出版社,2013.
- [12] Bellagio D E, Milligan T. Software configuration management strategies and IBM rational clearcase: a practical introduction [M]. 2nd ed. USA: IBM Press, 2005.
- [13] 彭德良. 基于 RUP 的软件配置管理[D]. 西安:西北工业大学,2007.
- [14] Tykal J. Best practices for using composite baselines in UCM [R]. [s. l. ]: Prentice Hall, 2004.
- [15] Collins-Sussman B, Fitzpatrick B W, Pilato C M. Version control with subversion for subversion 1.5 [M]. [s. l. ]: [s. n. ], 2005.
- [16] Koskela J. Software configuration management in agile methods [R]. [s. l. ]: [s. n. ], 2003.
- [17] Lee K A. IBM rational clearcase, ant and cruisecontrol [M]. USA: IBM Press, 2006.