



软件配置管理 及其工具应用

◎ 黄军 刘晓梅 熊勇 编著



人民邮电出版社
POSTS & TELECOMMUNICATIONS PRESS

TP311.5

52



软件配置管理 及其工具应用

◎ 黄军 刘晓梅 熊勇 编著



石化 S178096G

人民邮电出版社

图书在版编目 (CIP) 数据

软件配置管理及其工具应用 / 黄军, 刘晓梅, 熊勇编著. —北京: 人民邮电出版社, 2002.12
ISBN 7-115-10777-7

I. 软… II. ①黄… ②刘… ③熊… III. 软件—配置—管理 IV. TP31

中国版本图书馆 CIP 数据核字 (2002) 第 090345 号

内 容 提 要

本书围绕软件配置管理和配置管理工具的应用而编写, 详尽地介绍了软件配置管理基础知识和相关的软件配置管理工具。全书分为 9 章和附录, 内容包括软件配置管理技术的基础知识、版本控制工具 Visual SourceSafe 基础和应用、软件配置管理工具 ClearCase 基础、ClearCase 客户端应用、ClearCase 服务器应用、其他配置管理工具、版本合并工具 Beyond Compare 应用、缺陷和时间记录表。

本书注重实用性, 希望读者通过对本书的学习, 了解并熟练掌握软件配置管理技术和软件配置管理工具的应用。本书既可作为初级和中级程序员的自学指导书, 也可作为软件开发人员和项目经理的学习参考书, 适用于从事项目管理的广大技术人员以及作为相关院校师生的学习参考。

软件配置管理及其工具应用

◆ 编 著 黄 军 刘晓梅 熊 勇
责任编辑 马 嘉

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
读者热线 010-67132692
北京汉魂图文设计有限公司制作
北京密云春雷印刷厂印刷
新华书店总店北京发行所经销

◆ 开本: 787×1092 1/16
印张: 13.5
字数: 324 千字 2002 年 12 月第 1 版
印数: 1-5 000 册 2002 年 12 月北京第 1 次印刷

ISBN 7-115-10777-7/TP · 3156

定价: 25.00 元

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

前言

伴随着软件团队人员的增加,软件版本的不断变化,开发时间的紧迫以及多平台开发环境的采用,软件开发面临越来越多的问题,其中包括对当前多种产品的开发和维护、保证产品版本的精确、重建已经发布的产品、加强开发政策的统一和对特殊版本需求的处理等,解决这些问题的唯一途径是加强管理,而软件开发管理的核心是软件配置管理。

软件配置管理是贯穿整个软件生存周期的一项技术。它的主要功能是控制软件生存周期中软件的变化,减小各种改变所造成的影响,确保软件产品的质量。软件配置管理是人们对软件更改进行控制和管理的过程中,经过长期的探索,总结以往的经验教训而逐步形成的一套科学管理规范,是针对软件更改的一个关键的软件支持过程。正确应用软件配置管理是开发高质量软件所不可缺少的。软件配置管理的过程是软件开发过程中质量管理的精髓。

要想有效地实施软件配置管理,除了培养软件开发者的管理意识外,更重要的是使用优秀的软件配置管理工具。

目前软件配置管理工具可以分为以下3个级别。

第1个级别为简单的版本控制工具,是入门级的工具。例如 CVS、Visual SourceSafe 等。

第2个级别为项目级配置管理工具,适合管理中小型的项目。例如 PVCS、MKS 等。

第3个级别为企业级配置管理工具,具有强大的过程管理功能。例如 CCC Harvest、ClearCase 等。

软件配置管理贯穿于整个软件生存周期,所以对于软件配置管理工具的使用也会“始于项目开发之初,终于产品淘汰之时”。要想通过软件配置管理工具实施软件配置管理,首先要深刻了解配置管理的实质,然后将配置管理的管理方法以制度化实施,在允许的情况下最大限度地调整工作结构,制定合理的软件配置管理方案,根据自身的需要和人员素质,最后选购合适的软件配置管理工具。

本书共分为9章,下面简略地介绍各章的内容。

第1章首先介绍了软件配置管理的历史、概念、模式、作用、重要性,然后介绍了软件配置管理过程、基线更改控制流程、软件配置管理工具、软件配置管理计划,最后介绍了制定软件配置管理方案及实施配置管理的好处。

第2章介绍软件配置管理实践中常用的规程和流程——配置管理规程、更改控制规程、产品开发文档控制规程、软件配置管理计划规范、基线化流程、缺陷跟踪流程。

第1章和第2章是对基础知识的介绍,基础知识介绍之后,本书接下来介绍软件配置管理工具的应用。

第3章首先介绍了 Visual SourceSafe 特性、工作原理,然后介绍了 SourceSafe 安装、SourceSafe 数据库,最后介绍了 Visual SourceSafe 在项目中的应用,用 Client/Server 管理模型构建的软件开发管理的基本框架。

第4章首先介绍 ClearCase 的重要概念、ClearCase 的安装,然后介绍了 ClearCase 原理、主要功能、特性、ClearTool 工具,最后介绍了 ClearCase 的优势及配置方案。

第5章首先介绍了用 ClearCase 管理软件开发项目，然后介绍了 ClearCase 的 License 机制和 ClearCase 使用技巧，最后介绍了软件并行开发流程。

第6章首先介绍 ClearCase 系统管理员和发布工程师，然后介绍管理 VOB 数据库，最后介绍 ClearCase 管理控制台和 ClearCase 系统规划。

第7章介绍了 CCC/Harvest、Vertical Sky Solution、Source Integrity、RCS、CVS 等软件配置管理工具。

第8章介绍了版本合并工具 Beyond Compare 的使用，并重点介绍了 Beyond Compare 提供的文件夹比较、文件比较功能。

第9章介绍缺陷记录表、时间记录表。缺陷记录表记录用户什么阶段引入、发现的缺陷，缺陷的数量、类型；时间记录表用时间描述了开发过程。

本书的附录中列出了 CMM 等级 2 中的软件配置管理、配置控制组和配置管理组在软件开发过程中的职责、软件配置管理计划示例、软件修改控制记录表、将 Windows NT Server 4 安装为主域控制器。

本书的光盘中附带了 ClearCase 软件包（用户需要购买授权）和 Beyond Compare 软件。

限于篇幅，本书中对一些相关基础内容未作详细地解释。

本书编写者希望读者通过对本书的学习，了解并熟练掌握软件配置管理技术和软件配置管理工具的应用，因此本书注重实用性。

本书部分图形用 Visio 2002 绘制。

本书 Visual SourceSafe 和 ClearCase 软件的应用基于 Microsoft Windows 操作系统。

本书参考了大量的国内外资料，在这里衷心感谢这些作者对我们的帮助，同时衷心感谢 Rational 公司提供的软件包。最后感谢王申康教授和孔繁胜教授对我们写作的帮助和指导。

限于作者水平和时间仓促，书中难免有不足和疏忽之处，恳请各位同仁和读者多多批评指正。

作者

目 录

第 1 章 软件配置管理概述	1
1.1 软件配置管理的历史	1
1.2 软件配置管理的基础知识	2
1.2.1 软件配置管理术语	2
1.2.2 软件配置管理的定义	11
1.2.3 软件配置管理的模式	15
1.2.4 软件配置管理的功能领域	15
1.2.5 软件配置管理的作用	16
1.2.6 软件配置管理的重要性	17
1.2.7 基于软件配置管理的开发团队成员	17
1.3 软件配置管理过程	18
1.3.1 配置标识	18
1.3.2 版本管理	20
1.3.3 变更控制	21
1.3.4 配置审核	22
1.3.5 配置状态记录	22
1.4 基线更改控制流程	24
1.5 软件配置管理系统	27
1.5.1 软件配置管理系统的概念	27
1.5.2 软件配置管理系统的功能	28
1.6 软件配置管理工具	30
1.6.1 软件配置管理工具的发展	30
1.6.2 软件配置管理工具如何体现其系统概念	31
1.6.3 软件配置管理工具的主要功能	32
1.6.4 软件配置管理工具及其提供商评估	34
1.6.5 商用的软件配置管理工具	35
1.6.6 软件配置管理工具的简单应用	37
1.7 软件配置管理计划	39
1.8 构造和实施软件配置管理的解决方案	40
1.8.1 组建配置管理方案构造小组	40
1.8.2 对目标机构进行了解、评估	41
1.8.3 制订实施计划	42
1.8.4 定义配置管理流程	43

1.8.5 试验项目的实施	44
1.8.6 全面实施	45
1.9 实施配置管理的好处	45
第 2 章 软件配置管理的规程	47
2.1 配置管理规程	47
2.2 更改控制规程	49
2.3 产品开发文档控制规程	51
2.4 软件配置管理计划规范	53
2.5 基线化流程	58
2.6 缺陷跟踪流程	59
第 3 章 版本控制工具 Visual SourceSafe	65
3.1 Visual SourceSafe 的功能和特性	65
3.2 Visual SourceSafe 的重要概念	66
3.3 SourceSafe 的安装	67
3.3.1 SourceSafe 服务器的安装	67
3.3.2 SourceSafe 客户端的安装	68
3.3.3 SourceSafe 的重要文件	68
3.4 SourceSafe 数据库的配置和管理	70
3.4.1 创建 SourceSafe 数据库	70
3.4.2 打开 SourceSafe 数据库	71
3.4.3 将用户添加给 Visual SourceSafe 数据库	72
3.4.4 分析数据库	73
3.4.5 针对远程 SourceSafe 数据库设置许可权限	74
3.5 用 SourceSafe 组织软件开发项目	75
3.5.1 将项目添加给 SourceSafe 数据库	75
3.5.2 建立个人工作空间	78
3.5.3 基本开发步骤	79
3.5.4 基本的版本管理方法	82
3.6 软件版本管理的基本框架	85
3.6.1 Client/Server 模型	86
3.6.2 实现技术与方法	87
3.6.3 管理过程中注意的问题	88
第 4 章 ClearCase 基础	89
4.1 ClearCase 的重要概念	89
4.2 ClearCase 的安装	93
4.2.1 环境及软、硬件要求	93
4.2.2 ClearCase 服务器的安装	94

4.2.3	ClearCase 客户端的安装.....	96
4.2.4	删除 ClearCase.....	98
4.3	ClearCase 原理	98
4.4	ClearCase 的主要功能	98
4.4.1	版本控制	99
4.4.2	工作空间管理	100
4.4.3	建立管理	100
4.4.4	过程控制	100
4.5	ClearCase 特性	101
4.6	Cleartool 工具	103
4.6.1	基础知识	103
4.6.2	常用的命令	104
4.7	ClearCase 的优势	107
4.8	配置方案	108
第 5 章	ClearCase 客户端应用	109
5.1	用 ClearCase 管理软件开发项目	109
5.1.1	创建个人工作空间	109
5.1.2	基本开发步骤	112
5.1.3	并行开发	118
5.1.4	基本的版本管理	121
5.2	ClearCase Details 工具	124
5.3	ClearCase 的 License 机制	125
5.4	ClearCase 使用技巧	126
5.5	软件并行开发流程	131
5.6	UCM	132
第 6 章	ClearCase 服务器应用	133
6.1	ClearCase 系统管理员和发布工程师	133
6.2	创建 ClearCase 用户账号和用户组	134
6.3	管理 ClearCase 网络	134
6.4	管理 VOB 数据库	135
6.4.1	创建 VOB	136
6.4.2	数据输入	136
6.4.3	VOB 和视图的存储目录	138
6.4.4	VOB 数据库备份	139
6.4.5	VOB 数据库恢复	140
6.4.6	VOB 保护模式	140
6.4.7	VOB 权限设置	142
6.4.8	VOB 的加锁	143

6.5 ClearCase 的 Registry 机制	145
6.6 ClearCase 管理控制台	146
6.7 ClearCase 系统规划	148
第 7 章 其他软件配置管理工具	151
7.1 CCC/Harvest	151
7.1.1 过程驱动的配置和变更管理	152
7.1.2 全面的变更和配置管理	152
7.1.3 提供多种的开发模式	153
7.1.4 灵活易用性	154
7.1.5 CCC/Harvest 支持的平台	154
7.2 Vertical Sky Solution	155
7.2.1 简介	156
7.2.2 技术特点	156
7.2.3 系统要求和集成性	157
7.3 Source Integrity	158
7.3.1 全面的软件配置管理功能	159
7.3.2 有效的变更管理	161
7.3.3 通过 Web 访问软件配置管理和变更管理	161
7.3.4 支持平台	161
7.4 RCS	161
7.5 CVS	162
第 8 章 版本合并工具 Beyond Compare	164
8.1 Session 设置	164
8.2 目录比较功能	166
8.3 文件比较功能	172
8.4 快捷键	176
第 9 章 缺陷、时间记录表	178
9.1 缺陷记录表	178
9.2 时间记录表	180
附录	182
附录 A CMM 等级 2 中的软件配置管理	182
附录 A.1 执行约定	182
附录 A.2 执行能力	183
附录 A.3 软件配置管理所执行的顶层活动	184
附录 A.4 测量和分析	186
附录 A.5 验证实施	186

附录 B 配置控制组在软件开发过程中的职责	187
附录 C 配置管理组在软件开发过程中的职责	188
附录 D 软件配置管理计划示例	190
附录 D.1 引言	190
附录 D.2 管理	191
附录 D.3 软件配置管理活动	193
附录 D.4 工具、技术和方法	194
附录 D.5 媒体控制	194
附录 D.6 对供货单位的控制	195
附录 D.7 记录的收集、维护和保存	195
附录 D.8 配置管理报表	196
附录 E 软件修改控制记录表	199
附录 F 将 Windows NT Server 4.0 安装为主域控制器	202
参考文献	205
参考站点	206

第1章 软件配置管理概述

在软件开发过程中，伴随着开发工作的进展会产生许多信息，例如可行性分析、需求分析说明、总体设计说明、详细设计说明、编码设计说明、源代码、可执行码、用户手册、测试计划、测试用例、测试结果、在线帮助等技术文档，以及合同、计划、会议记录、报告等管理文档。另一方面，在软件开发过程中出现变更是不可避免的。面对如此庞大且不断变动的信息集合，如何使其有序、高效地存放、查找和利用成为软件工程项目十分突出的问题。软件配置管理正是为了解决这个问题而提出的，它为软件开发提供了一套管理办法和活动原则，成为贯穿软件开发始终的重要质量保证活动。

本章介绍软件配置管理的基础知识和使用方法。

1.1 软件配置管理的历史

要了解一个陌生的事物，最好的办法就是追本溯源，从它的产生及演变的历史入手，这样会得到一个比较全面的认识。

配置管理的概念最早来源于制造业，尤其是在美国的国防工业中。当时开发的产品规模比较小，复杂程度也不高，在整个生产过程中所有的开发、设计和变更常常是由一个人或一组人来完成。当产品的复杂性增加时，不可能再由一个人或一组人来完成产品的设计和生；更重要的是，设计在不断地变化；此外，这些产品的开发可能经历几年，由不同的人负责进行。当控制权从一个人交给另一个人时，因为没有正规的方法要求将文档、设计和在开发期间所做的修改合理地记录下来，很可能丢失一些相关的信息。

1962年，美国空军在进行喷气飞机设计时，为解决控制和通信问题，制定和发布了一个AFSCM375-1标准用于配置管理，这是第一个用于配置管理的标准。美国军用标准（Military Standard，简称MIL-S）和国防部（Department Of Defense Standards，简称DOD-STD）的很多其他标准都是遵循这一标准而制定的。

伴随着计算机的普及，软件开发工业的重要性不断增强。软件系统的应用使工作更加轻松，因为计算机做了很多以前必须手工进行的工作。人们逐渐习惯于自动化系统带来的方便，并且开始对计算机有越来越多的要求。伴随着更多的软件供应商带来的新的更好的软件产品，软件开发组织开发的计算机程序在规模上越来越大，结构也越来越复杂，技术越来越先进。与此同时开发组织的规模和性质也发生了变化。很多项目有成千的开发人员，他们来自于不同的国家，有不同的文化和社会背景，庞大的系统可以划分为若干子系统，由分布在世界上各个地域的不同的公司分别开发完成。

计算机程序的复杂性使系统更加难于管理，项目开发组人员逐渐增多，并且越来越分散，不可能再依靠几个人控制和管理开发过程。而且开发人员之间的信息沟通、经验共享等问题

也开始在项目中出现。美国国防部和几个国际性组织包括 IEEE (Institute of Electrical and Electronics Engineers, 美国电气和电子工程师协会)、ANSI (American National Standards Institute, 美国国家标准协会) 和 ISO (International Standards Organization, 国际标准化组织) 开始着手解决软件开发过程中配置管理问题, 最终形成了他们自己的标准。其中最为普遍使用的是 ANSI/IEEE 标准。现在软件配置管理已经被大多数软件组织采纳和接受, 并作为开发的准则。市场上有很多工具和软件包, 能够帮助实现软件配置管理过程的自动化, 使软件配置管理的使用更加容易。

1.2 软件配置管理的基础知识

要实施软件配置管理, 首先要深刻了解软件配置管理的实质, 为此, 本节阐述软件配置管理的概念、作用和相关内容。

1.2.1 软件配置管理术语

配置管理 (Configuration Management) 是指用于控制系统一系列变化的学科, 通过一系列技术、方法和手段来维护产品的历史, 鉴别和定位产品独有的版本, 并在产品的开发和发布阶段控制变化, 通过有序管理和减少重复性工作, 保证生产的质量和效率。

不同于配置管理, 软件配置管理以计算机为载体 (不论工具和产品), 不光维护产品的状态、历史纪录, 同样还支持存储、恢复和产品制造。软件配置管理是软件工程学科中涉及概念较多的一项内容, 为了便于说明, 下面给出一些软件配置管理相关术语 (主要是软件配置管理计划规范 GB/T 12505-90) 的定义和说明。

(1) 项目委托单位 (Project Entrust Organization)

项目委托单位指为产品开发提供资金, 通常也是 (但有时也未必) 确定产品需求的单位或个人。

(2) 项目承办单位 (Project Undertaking Organization)

项目承办单位指为项目委托单位开发、购置或选用软件产品的单位或个人。

(3) 软件开发单位 (Software Development Organization)

软件开发单位是指直接或间接受项目委托单位委托而直接负责开发软件的单位或个人。

(4) 用户 (User)

用户指实际使用软件来完成某项计算、控制或数据处理等任务的单位或个人。

(5) 软件 (Software)

软件指计算机程序及其有关的数据和文档, 也包括固化了的程序。

IEEE 给出的定义为: 计算机程序、方法、规则、相关的文档资料以及在计算机上运行时所必须的数据。

由此可见, 软件已不再只是一个程序和一本使用手册, 而是包括大量的程序、文档和数据。

(6) 软件对象 (Software Object)

软件对象是在项目进展过程中产生的、可由软件配置管理加以控制的任何实体。每个软

件对象都具有一个唯一的标识符、一个包含实际信息的对象实体、一组用于描述其自身特性的属性与关系,以及用于与其他对象进行关系操作与消息传递的机制。

软件对象按其生成方式可分为源对象(Source Object)与派生对象(Derived Object),按其内部结构形式可分为原子对象(Atomic Object)与复合对象(Composite Object),按照软件开发的不同时刻(状态)可分为可变对象(Mutable Object)与不可变对象(Immutable Object)。

(7) 配置(Configuration)

配置指在配置管理中,软件或硬件所具有的(即在技术文档中所陈述的或产品所实现的)那些功能特性和物理特性。

(8) 重要软件(Critical Software)

重要软件指其故障会影响到人身安全,会导致重大经济损失或社会损失的软件。

(9) 软件生存周期(Software Life Cycle)

软件生存周期指从对软件系统提出应用需求开始,经过开发,产生出一个满足需求的计算机软件系统,然后投入运行,直至该软件系统“退役”为止。其间经历系统分析与软件定义、软件开发以及系统的运行与维护等3个阶段。其中软件开发阶段一般又分成需求分析、概要设计、详细设计、编码与单元测试、组装与集成测试、系统测试以及安装与验收等6个阶段。

(10) 软件开发库(Software Development Library)

软件开发库指在软件生存周期的某一个阶段期间,存放与该阶段软件开发工作有关的计算机可读信息和人工可读信息的库。

(11) 配置项(Configuration Item)

配置项指一个配置中的实体,它满足一项最终使用功能,并能在给定的参考点上单独标识。在 GB/T 11457-1995《软件工程术语》中给出配置项另外一个定义:为了配置管理目的而作为一个单位来看待的硬件和/或软件成分,满足最终应用功能并被指明用于配置管理的硬件/软件,或它们的集合体。

软件配置管理的对象是软件配置项,它们是在软件工程过程中产生的信息项。按照 ISO 9000-3 的规定,软件配置项可以是:

- ☐ 与合同、过程、计划和产品有关的文档和数据;
- ☐ 源代码、目标代码和可执行代码;
- ☐ 相关产品,包括软件工具、库内的可复用软件、外购软件及用户提供的软件。

组成上述信息的所有项目构成了一个软件配置,而其中的每一项便称为一个软件配置项,这是配置管理的基本单位。在软件开发过程中,最早的软件配置项是系统软件规格说明书,随着软件开发过程的不断深入,软件配置项也迅速增加。

软件配置项基本可划分为两种类别:

☐ **软件基准**——经过正式评审和认可的一组软件配置项(文档和其他软件产品),它们作为下一步的软件开发工作的基础,并且只有通过正式的变更控制规程才能被更改。例如设计报告是编码工作的基础,设计报告可作为软件基准。

☐ **非基准配置项**——没有正式评审或认可的一组软件配置项。

还可以从下面角度划分软件配置项。

一个软件系统划分为几个配置项要由项目经理所确定的开发策略决定。读者可以参考由

NASA 给出的软件配置项划分原则（《NASA Software Configuration Management Guidebook》，1995），每个软件或某集合符合如下条件之一，可视为一个软件配置项：

- ☐ 该软件集合是独立设计、实现和测试的；
- ☐ 该软件集合对总体性能是关键的，或存在高风险的，或关系到系统安全性；
- ☐ 该软件集合极为复杂，涉及高新技术，或有严格的性能要求；
- ☐ 该软件集合与其他现有软件项目或由其他机构提供的软件之间有直接接口；
- ☐ 预计该软件集合在成为可运行软件之后会有比常规更多的修改；
- ☐ 该软件集合包括了某个特定范围的所有功能，如应用软件、操作系统等；
- ☐ 该软件集合安装在与系统其他部分不同的计算机平台上；
- ☐ 该软件集合被设计成可重复使用的。

下面介绍软件配置项的命名/编号。

软件的每个组件/部件的标识必须唯一，以便于用该标识符来跟踪和报告软件配置项的状态。通常，对每一个软件配置项要赋予一个标识名称或符号，软件配置项的各部分又在该标识符下附上描述符。NASA-CM-GDBK 给出的一个标识例子是：组成航天飞机飞行软件的软件配置项可标识为 FS（Flight Software for a Spacecraft）；而该飞行软件的组成部分，例如飞行执行程序可标识为 FS_EX，表示它是 FS 软件配置项的第二层次组件；该执行程序的各元件（子程序）可编号为 FS_EX_001、FS_EX_002 等。

因此，可以根据“型号代号_分系统/设备配置代号_所处研制阶段代号_软件产品分类编号_配置项编号”原则来对各软件配置项及其组件、子程序及相关描述文档进行命名、编号。

最后介绍一下软件配置项的状态。

在软件的整个生存周期中，一般包括制定计划、分析评估、设计、测试、运行/维护等状态。相应地可以把软件配置管理项分为设计态、测试态、受控态和运行态 4 种状态，其中受控态即指软件配置管理项处于配置管理状态。这 4 种状态相互之间的联系具有方向性。软件配置管理项的 4 种状态如图 1.1 所示。沿图 1.1 所示实线箭头所指方向的状态变化是允许的，反之则是不允许的，虚线表示为了验证或检测某些功能或性能而重新执行相应测试，正常情况下不沿虚线方向变化。

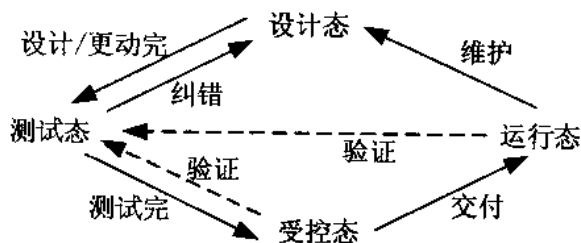


图 1.1 软件配置管理项的 4 种状态

（12）软件受控库（Software Controlled Library）

软件受控库指在软件生存周期的某一个阶段结束时，存放作为阶段产品而释放的、与软件开发工作有关的计算机可读信息和人工可读信息的库。软件配置管理就是对软件受控库中的各软件项进行管理，因此软件受控库也叫做软件配置管理库。

对库中的软件配置项的状态变更要实施正规的、严格的控制，包括软件配置管理项的检

入、检出，以及对其功能的可追踪性和各软件配置管理项的一致性的审查。软件受控库可在项目组、子系统、分系统上分别设立，由专人维护。

(13) 软件产品库 (Software Product Library)

软件产品库指在软件生存周期的系统测试阶段结束后，存放最终产品、交付给用户运行或在现场安装的软件的库。软件产品库可在分系统、系统层上分别设立并维护。如果软件产品库中的产品需要更改，则应将此产品在产品库中加锁，提取软件受控库中相应的软件配置项，履行受控库中的更改控制手续，直至更改完成并确认其能完成指定功能和性能后，方可再次存入软件产品库。

(14) 配置管理 (Configuration Management)

配置管理是对以下各项在运用技术上和行政上的管理和监视的一门学科。对一个配置项的功能特性和物理特性进行标识并写成文档；对这些特性的更改进行控制；对更改处理过程和实施状态进行记录和报告；以及对是否符合规定需求进行验证。

(15) 接口控制 (Interface Control)

接口控制指描述由一个或多个部门提供的，两个或两个以上的配置项接口的所有功能特性和物理特性的过程。在这些功能特性和物理特性实现之前，要确保对它们所做的修改已经过评审和批准。

(16) 版本 (Version)

版本是某一配置项的已标识了的实例 (Instance)。也可以说，不可变的源对象经质量检查合格后所形成的新的相对稳定的格局 (配置) 称为软件版本。

每个软件对象可具有一个版本组，它们彼此间具有特定的关系，这种关系用以描述其演变情况，通常软件对象的版本组呈树形结构。

(17) 版本控制 (Version Control)

版本控制就是管理在整个软件生存周期中建立起来的某一配置项的不同版本。

在软件工程过程中所涉及的软件对象都要加以标识。在对象成为基线 (基线的概念将在下面进行介绍) 以前可能要做多次变更，在成为基线之后也可能需要频繁地变更。这样对于每一配置对象可以建立一个演变图，这个演变图记叙了对象的变更历史，如图 1.2 所示。以图 1.2 为例，配置对象 1.0 经过修改成为对象 1.1，又经历了小的修改和变更，产生了版本 1.1.1 和 1.1.2。接着对版本 1.1 做了一次更新，产生对象 1.2，又持续演变生成了 1.3 和 1.4 版本。同时对对象 1.2 做了一次较大的修改，引出一条新的演变路径：版本 2.0 和版本 2.1。目前已经有许多工具 (例如 ClearCase) 可以辅助这种标识工作。

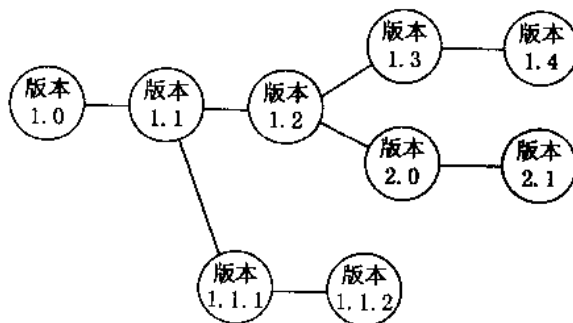


图 1.2 版本的演变图

(18) 释放 (Release)

释放指在软件生存周期的各个阶段结束时, 由该阶段向下阶段提交该阶段产品的过程。它也指将系统测试阶段结束时所获得的最终产品向用户提交的过程, 这个过程也叫做交付 (Delivery)。

(19) 基线 (Baseline)

基线指一个配置项在其生存周期的某一特定时间, 被正式标明、固定并经正式批准的版本。也可以说, 基线是软件生存期中各开发阶段末尾的特定点, 又称里程碑。只有由正式的技术评审而得到的软件配置项协议和软件配置的正式文本才能成为基线。它的作用是使各阶段工作的划分更加明确化, 使本来连续的工作在这些点上断开, 以便于检验和肯定阶段成果。例如, 明确规定不允许跨越基线修改另一阶段的文档。如图 1.3 所示, 是软件开发各阶段的基线。

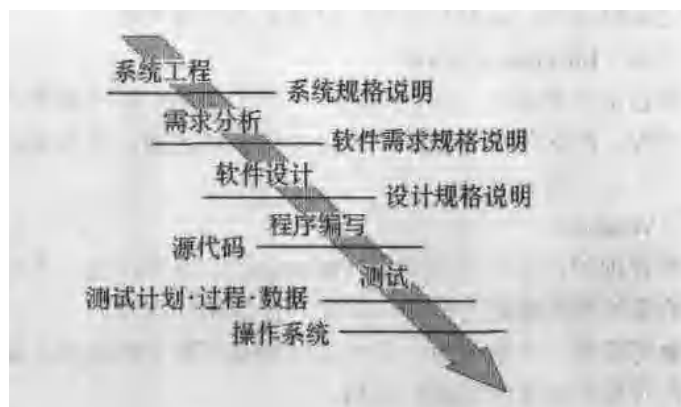


图 1.3 软件开发各阶段的基线

一个软件配置项一旦成为基线, 就把它存放到项目数据库 (也称项目信息库或软件仓库) 中。当一位软件组织成员想要对基线配置项进行修改时, 就把它从项目数据库中复制到该工程师的专用工作空间中, 如图 1.4 所示。图中把一个标号为 B 的配置项从项目数据库复制到工程师的专用工作空间 (例如 ClearCase 的视图) 中。这个活动记录在一个记事文件中。工程师可以在 B' (B 的副本) 上完成要求的变更, 然后用 B' 来更新 B。有些系统中把这个基线配置项锁定, 在变更完成、评审和批准之前, 不许对它进行任何操作。

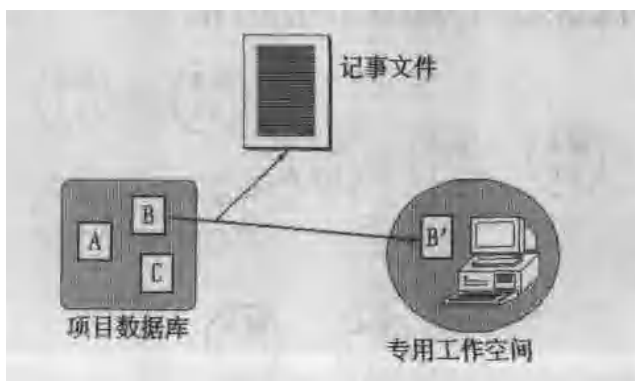


图 1.4 软件开发各阶段的基线

总之,基线是软件配置管理的一个重要概念。从某种意义上讲,它是在软件开发过程中为进行质量控制而引入的,它是开发进度表上的一个参考点与度量点,是后续开发的稳定基础。基线的形成实际上就是对某些配置进行冻结。

(20) 软件配置 (Software Configuration)

软件配置指一个软件产品在软件生存周期各个阶段所产生的各种形式(机器可读或人工可读)和各种版本的文档、程序及其数据的集合。该集合中的每一个元素称为该软件产品软件配置中的一个配置项。

软件工程过程的输出信息有3种:计算机程序,描述计算机程序的文档(包括技术文档和用户文档),数据结构。在软件工程过程中产生的所有的信息项(文档、报告、程序、表格、数据)就构成了软件配置。软件配置是软件开发进展到某一时刻时产生的全部信息所形成的一种格局,它反映并描述了软件开发阶段的状况。

软件配置的具体形态可分为以下两种。

❑ 不可直接执行的材料。例如书写的文档、程序清单、测试数据、测试结果等。

❑ 可直接执行的材料。例如目标代码、数据库信息等。它们可由计算机处理,存储于某种存储介质上。

(21) 功能基线 (Functional Baseline)

功能基线指在系统分析与软件定义阶段结束时,在经过正式评审和批准的系统设计规格说明书中对开发系统的规格说明;或是指在经过项目委托单位和项目承办单位双方签字同意的协议书或合同中,所规定的对开发软件系统的规格说明;或是由下级申请并经上级同意或直接由上级下达的项目任务书中所规定的对开发软件系统的规格说明。功能基线是最初批准的功能配置标识。

(22) 分配基线 (Allocated Baseline)

分配基线指在软件需求分析阶段结束时,经过正式评审和批准的软件需求的规格说明。分配基线是最初批准的分配配置标识。

(23) 产品基线 (Product Baseline)

产品基线指在软件组装与系统测试阶段结束时,经过正式评审和批准的有关软件产品的全部配置项的规格说明。产品基线是最初批准的产品配置标识。

(24) 基线配置管理 (Baseline Configuration Management)

基线配置管理指建立经正式评审和认可,并作为进一步开发工作的基础的基线的过程。某些(如软件设计和代码)软件工作产品应该有在预先确定点上建立的基线,并且应该对这些项施加严格的更改控制过程。当与顾客打交道时,这些基线提供控制和稳定性。

(25) 基线管理 (Baseline Management)

基线管理是指在配置管理中,运用技术上和行政上的管理来指定一些文档和更改这些文档,这些文档在某些特定时刻正式标识和建立起基线。

(26) 软件基线审计 (Software Baseline Audit)

软件基线审计是指对于软件基线库的结构、内容和设施的考查,以便查证基线是否符合描述基线的文档。

(27) 软件基线库 (Software Baseline Library)

软件基线库是指存储配置项及相连记录的仓库。