中国移动通信
CHINA MOBILE

配置管理培训

移动改变生活

目录

1.配置管理基础知识

配置管理误区

- 误区一：版本控制=软件配置管理
  - 版本控制只是配置管理最基本的层次和功能
- 误区二:编码水平最差=配置管理员
  - 国外公司一般都由有丰富编程经验的人担任软件配置管理人员，有的时候配置管理部分的工作职责直接由开发经理担任
- 误区三：采用配置管理工具=有效的配置管理
  - 一是规范的软件开发流程
  - 二是合格的配置管理参与人员(配置管理员、开发人员、项目经理等)

配置管理作用

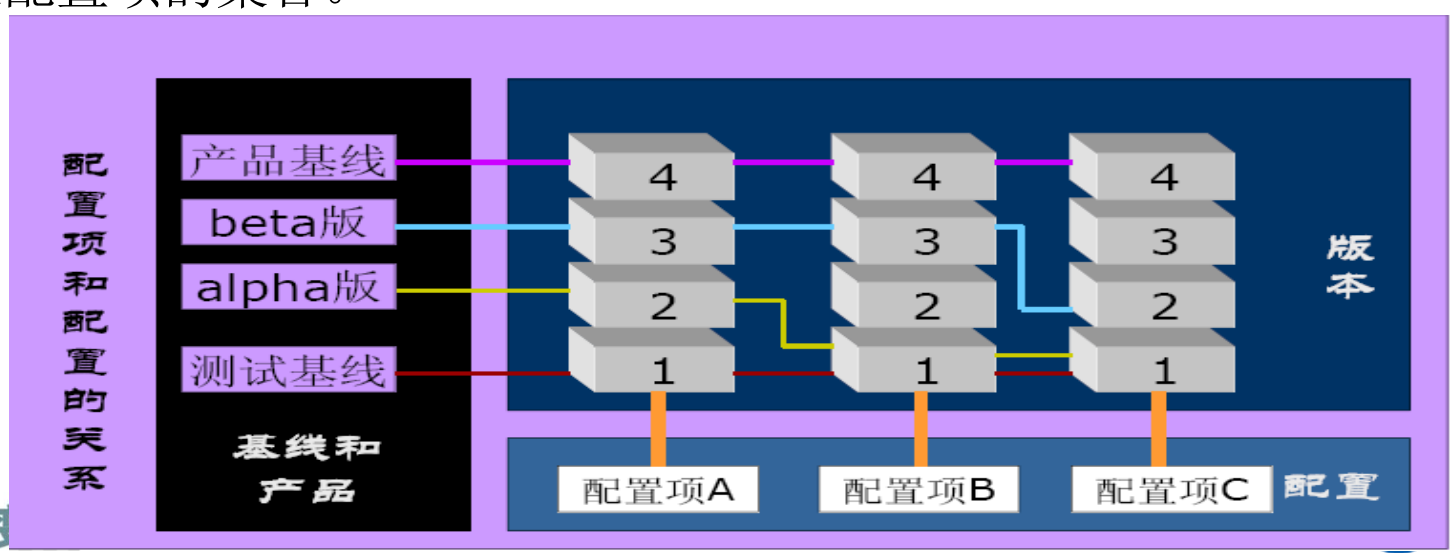
配置管理作用：

- **节约费用**
  - 缩短开发周期
- **有利于知识库的建立**
  - 代码对象库
  - 业务及经验库
- **规范管理**
  - 量化工作量考核
  - 规范测试
  - 加强协调与沟通

概述－软件配置管理

配置管理概述：

- **配置** —— 配置由部件表和部件分解图组成。部件分解图定义了基线中包含的所有要素以及如何将它们安装在一起。
- **软件配置项** —— 为了配置管理的目的而作为一个单位来看待的软件要素的集合。
- **软件配置** —— 软件产品在生存期各个阶段的不同形式（记录特定信息的不同媒体）和不同版本的程序、文档及相关数据的集合，或者说是配置项的集合。



概述—软件配置管理

W.Babich 的解释

- 软件配置管理能协调软件开发，使混乱减少到最小。软件配置管理是一种标识、组织和控制修改的技术，目的是最有效的提高生产率。

GB/T 11457 :1995《软件工程术语》国家标准

- A.表示和确定系统中配置项的过程，在系统整个生存期内控制这些配置项的投放和更动，记录并报告配置的状态和更动要求，验证配置项的完整性和正确性。
- B.对下列工作进行技术和行动指导与监督的一套规范：
  - 对配置项的功能特性和物理特性进行标识和文件编制工作；
  - 控制这些特性的更动情况；
  - 记录并报告这些更动进行的处理和实现的状态。

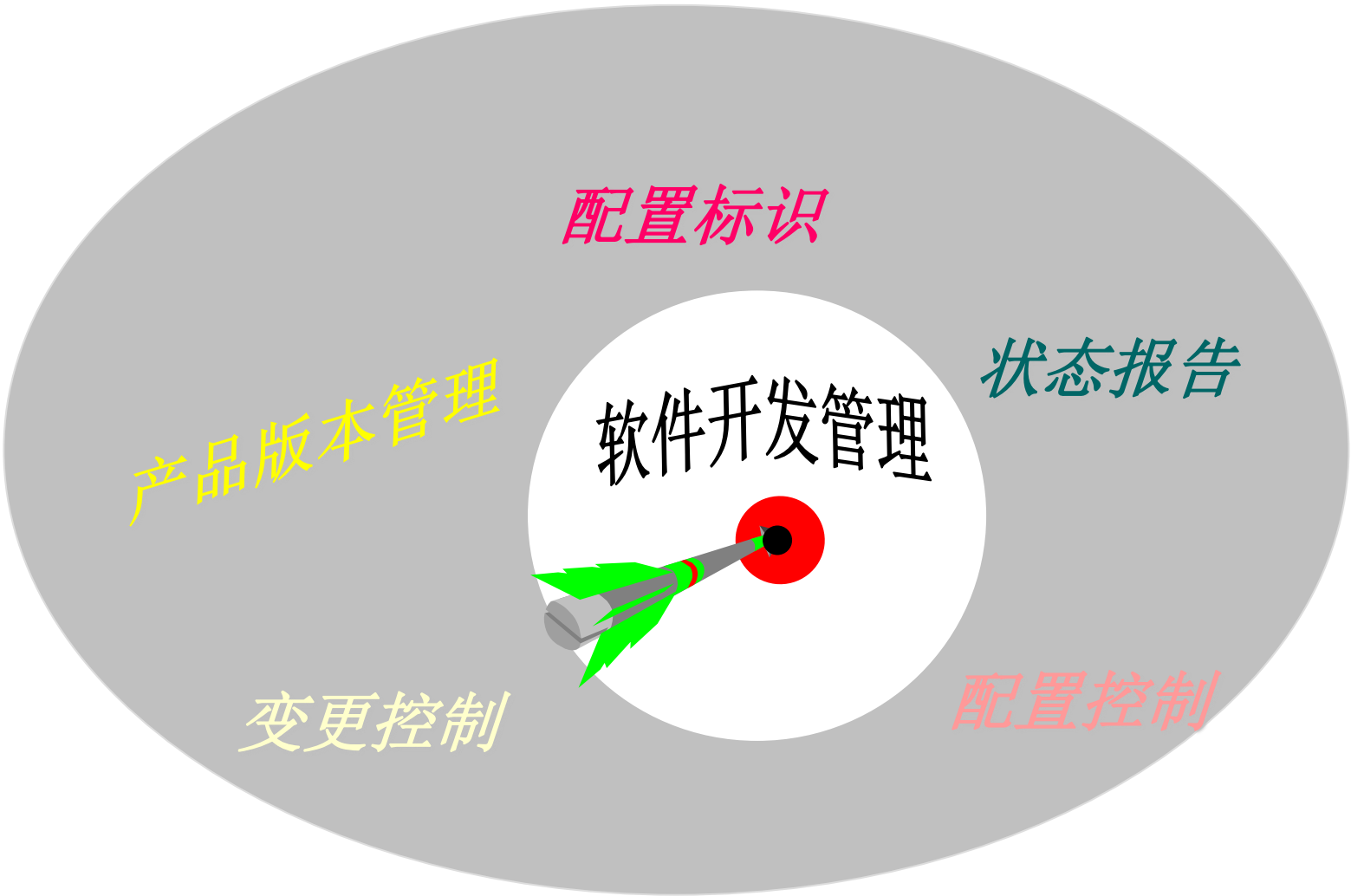
概述—范围

类别	特征	举例
定义	需求分析及定义阶段完成后得到的工作产品	需求规格说明书、项目开发计划、设计标准或设计准则、验收测试计划
设计	设计阶段结束后得到的产品	系统设计规格说明、程序规格说明、数据库设计、编码标准、用户界面标准、测试标准、系统测试计划、用户手册
编码	编码及单元测试后得到的工作产品	源代码、目标码、单元测试数据及单元测试结果
测试	系统测试完成后的工作产品	系统测试数据、系统测试结果、操作手册、安装手册
维护	进入维护阶段以后产生的工作产品	以上任何需要变更的软件配置项
环境	软件开发环境、软件维护环境	编译器、操作系统、编辑器、数据库管理系统、开发工具、测试工具项目管理工具、文档编辑工具

概述—过程

活 动	任 务	解 释
1 实施过程	开发配置管理计划	计划描述：配置活动、这些活动的规程、进度、配置管理组织及与其他组织的关系 计划应形成文件
2 配置标识	制定标识规则	以控制软件项及其版本 标识内容包括：基线文档、版本基准号、其他
3 配置控制	标志并记录变更申请、分析与评价变更、批准申请 实现、验证和发行已变更的软件项 审核跟踪变更、控制并审核受控软件项	跟踪变更原因、变更授权 以保证重要功能的安全或保密
4 配置状态报告	编制管理记录和状态报告	表明受控项（包括基线）的状态和历史 状态报告应包括变更号、最新版本、发行标识、版本号及各种版本比较
5 配置审计	确定和保证软件项的功能完整性、物理完整性	
6 发行管理和交付	有效控制软件产品和文档的发行和交付 在产品的生存期内保存代码、文挡的主拷贝	包括重要的安全或保密功能的代码和文档应按组织的方针处理、储存、包装和交付

概述—配置管理任务



配置标识—要求

配置标识要求

- 唯一性
- 可追溯性
- 反映产品的结构
- 支持工具

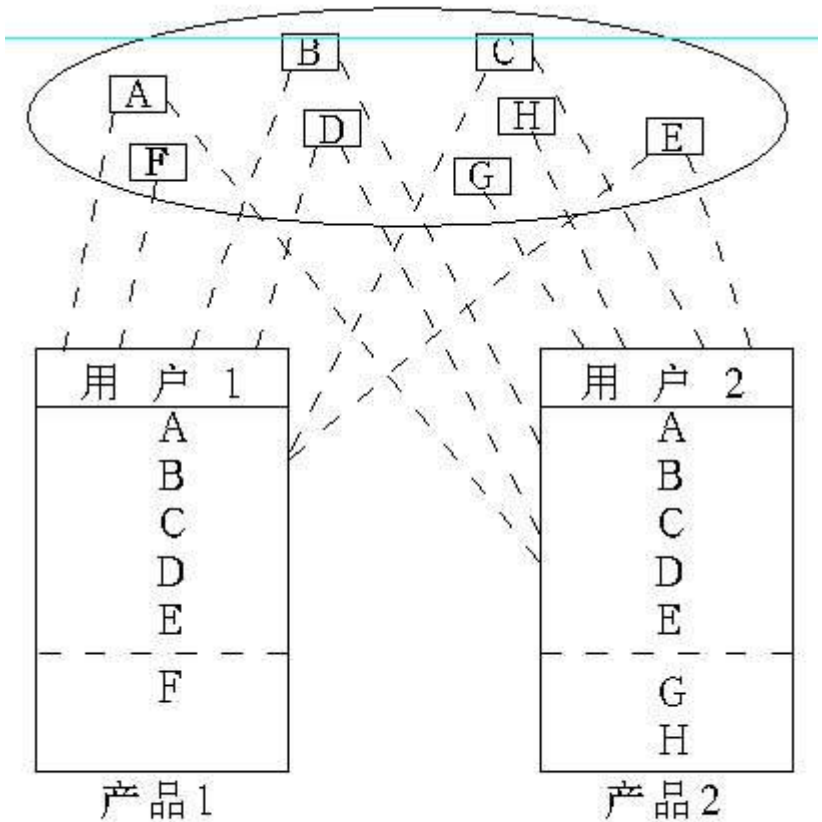
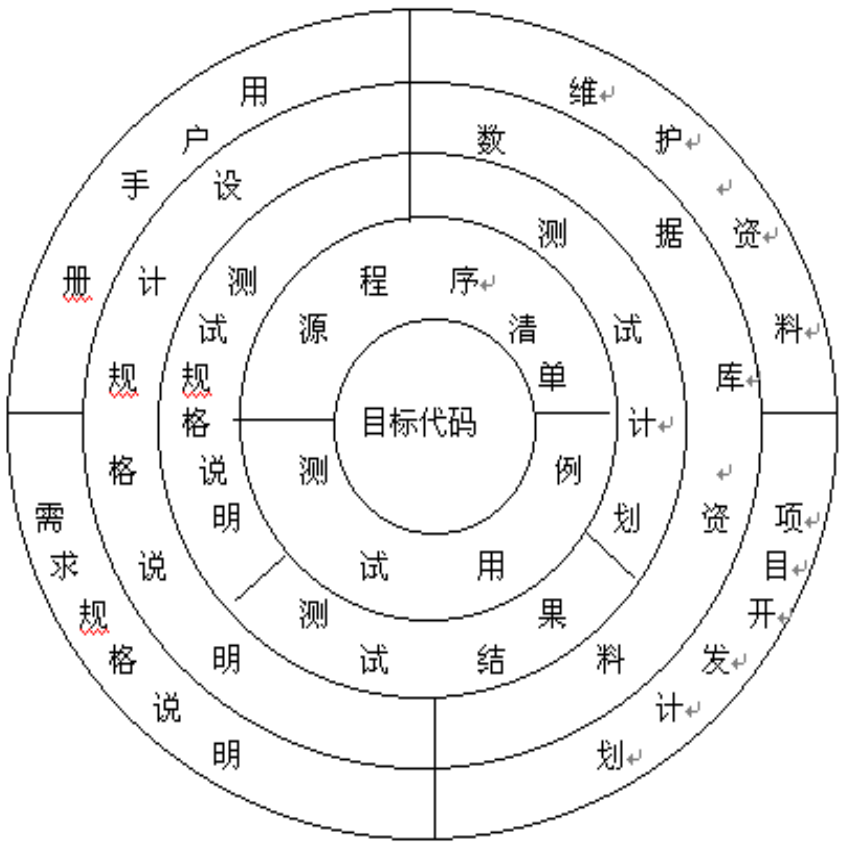


图3　两个产品具有不同的配置

配置标识—对象

配置项：就是配置管理的对象，简单来讲它符合以下任意一个特点：

- 1、它会被两个或两个以上的项目成员共同使用。
- 2、它会随着项目的开展而发生变化。
- 3、对项目重要的工作产品。
- 4、一些工作产品之间的关系非常紧密，一个变化其他的就会受到影响。
- 5、配置项本身的变化可以使用“版本管理”对其进行控制



配置标识－例子1

例1: 文档的标识

标识规则:

VVV-WWW-BBB-XXX.NN.MM

含义:

VVV － 系统识别符
WWW － 分系统识别符
BBB － 文档类别
XXX － 版本标识
NN － 发布号
MM － 修订标识

控制—变更管理

- 软件变更的不可避免性
- 软件变更的复杂性
  - 软件配置项数量大
  - 版本多
  - 变更的迁延性
  - 人员沟通协调
- 变更管理的任务
  - 分析变更
  - 记录和追踪变更
  - 采取措施保证变更在受控状态下进行

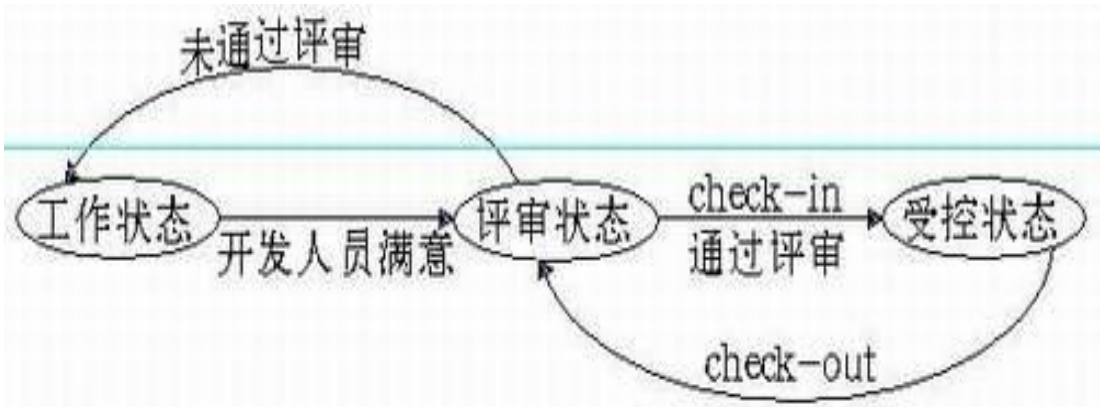


图4　配置项的状态变化

控制—基线定义

基线：软件文档或源码(或其它产出物)的一个稳定版本,它是进一步开发的基础。

- 基线反映分配给配置项的标识号及其相应的实体。
- 一组已经分配了唯一标识号的需求、设计、源代码文卷以及相应的可执行代码、构造文卷、和用户文档（相关的实体），可以认为是一个基线。
- 基线是项目储存库中每个工件版本在特定时期的一个“快照”。它提供一个正式标准，随后的工作基于此标准，并且只有经过授权后才能变更这个标准。
- 对基线的更改必须遵循变更控制规程。
- 交付给外部顾客的基线一般称为"放行"基线，内部使用的基线一般称为"构造"基线。

控制—基线定义

基线作用：重现性、可追踪性和报告。

- 重现性是指及时返回并重新生成软件系统给定发布版的能力，或者是在项目中的早些时候重新生成开发环境的能力。
- 可追踪性建立项目工件之间的前后继承关系。其目的在于确保设计满足要求、代码实施设计以及用正确代码编译可执行文件。
- 报告来源于一个基线内容同另一个基线内容的比较。基线比较有助于调试并生成发布说明。

控制—基线的概念

基线建立时机：一般可以在里程碑完成前（或每次迭代结束时）

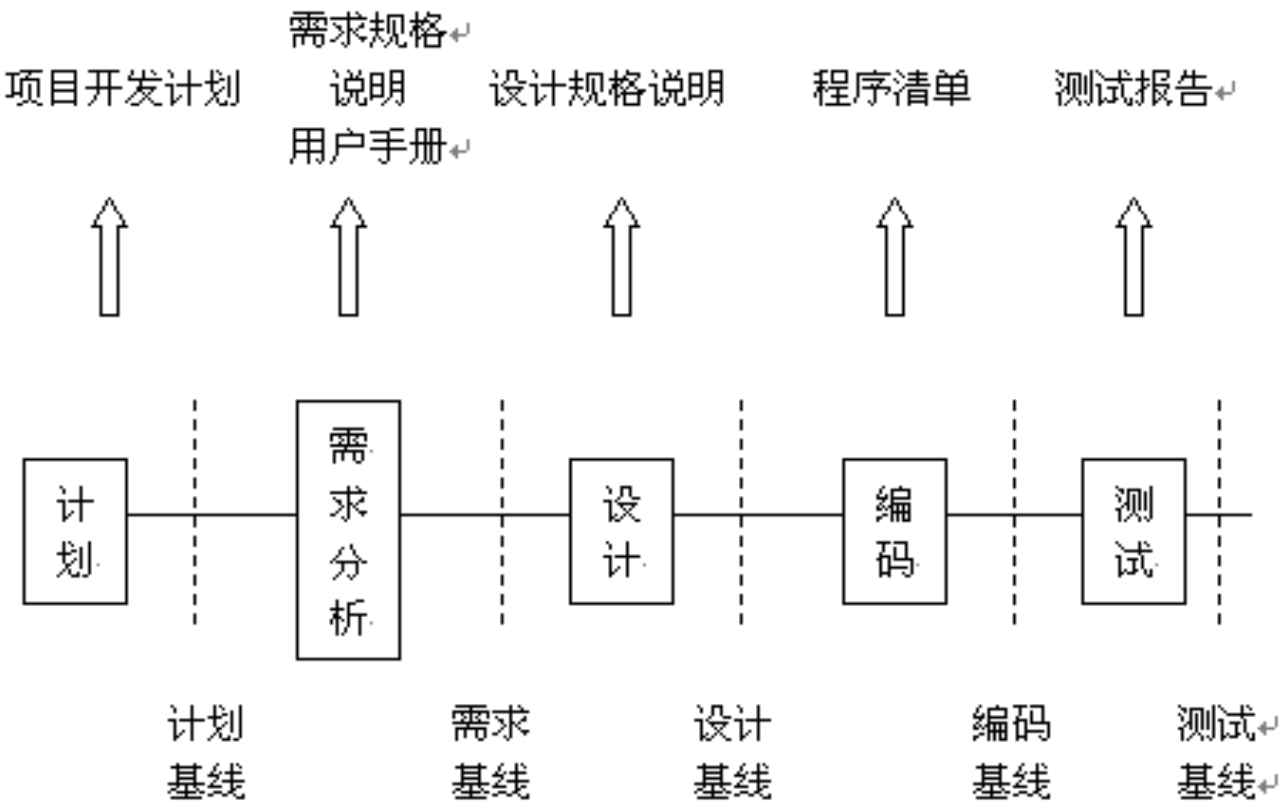


图 9　软件配置基线

控制—变更控制

- 对基线提出的变更必须经过一定层次的评审
- 建立更改控制层
- 必须确定和理解提出的变更对经费、进度、软件开发和生产造成的影响
- 必要时，必须获得配置控制委员会的批准、并配置主要管理人员和项目组成员
- 必须正确实施被批准的变更
- 一旦变更被批准，必须通知所有受影响的部门

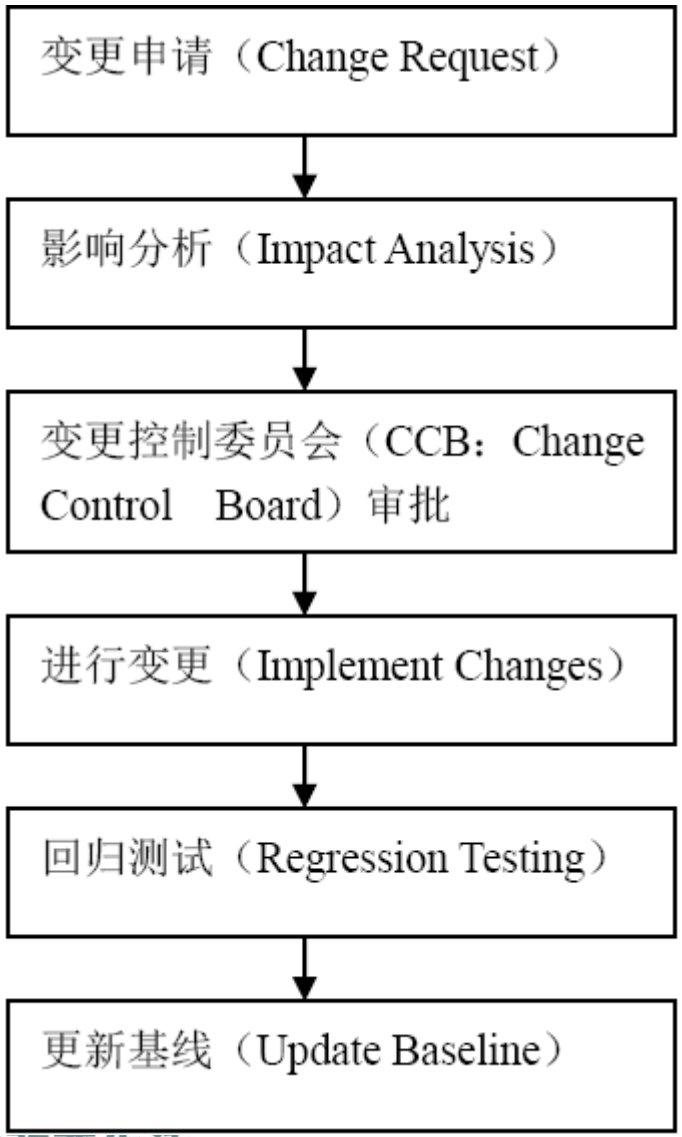
控制—配置库

- 作用
  - 记录与配置相关的所有信息
  - 利用库中的信息可评价变更的后果
  - 可利用库中的信息查询
- 三类库
  - 开发库
  - 受控库
  - 产品库

控制—人员组织

- 变更控制委员会(Change Control Board)也称为配置控制委员会(Configuration Control Board)
- 变更授权人(Chang control authority,CCA)
- 配置经理
- 配置管理员

控制—流程



- 变更申请不限于需求的改变
- 也可能来源于开发中的缺陷
- 所有变更都需要经过一个变更申请过程

控制—变更请求表

项目名		变更请求标识	

变更请求：　　变更请求人 _____　日期 _____

变更理由 _____

变更描述 _____

影响范围 _____

变更优先性考虑 _____

评估变更工作量 _____

分析与评估　　分析人 _____　日期 _____

分析与评估意见 _____

审批　　CCB 负责人 _____　日期 _____

CCB 审查意见 _____

变更实施　　实施负责人 _____　日期 _____

变更实施情况 _____

质量保证审查　　QA 负责人 _____　日期 _____

审查意见 _____

配置管理审查　　CM 负责人 _____　日期 _____

审查意见 _____

变更申请考滤的属性：

- 规模大小
- 是否有备选方案
- 变更复杂度
- 严重性
- 进度
- 成本
- 测试

控制—版本

- **版本**
  - 一个基线或一个软件配置项特殊的事例。
- **软件版本**
  - 为满足不同用户的不同使用要求，如适用于不同运行环境或不同平台的系列产品；
  - 软件产品投入使用以后，经过一段时间运行提出了变更的要求，需要做较大的修正或纠错，增强功能或提高性能。

控制—并行开发

- **版本控制通过分支和合并为并行开发提供支持**
  - 并行开发允许不同的项目在同一时间使用相同的原文件；
  - 并行开发隔离了那些永远不被共享的工作；
  - 并行开发允许工程师即使某一条开发线被冻结了，仍可以沿着一个分支继续开发。

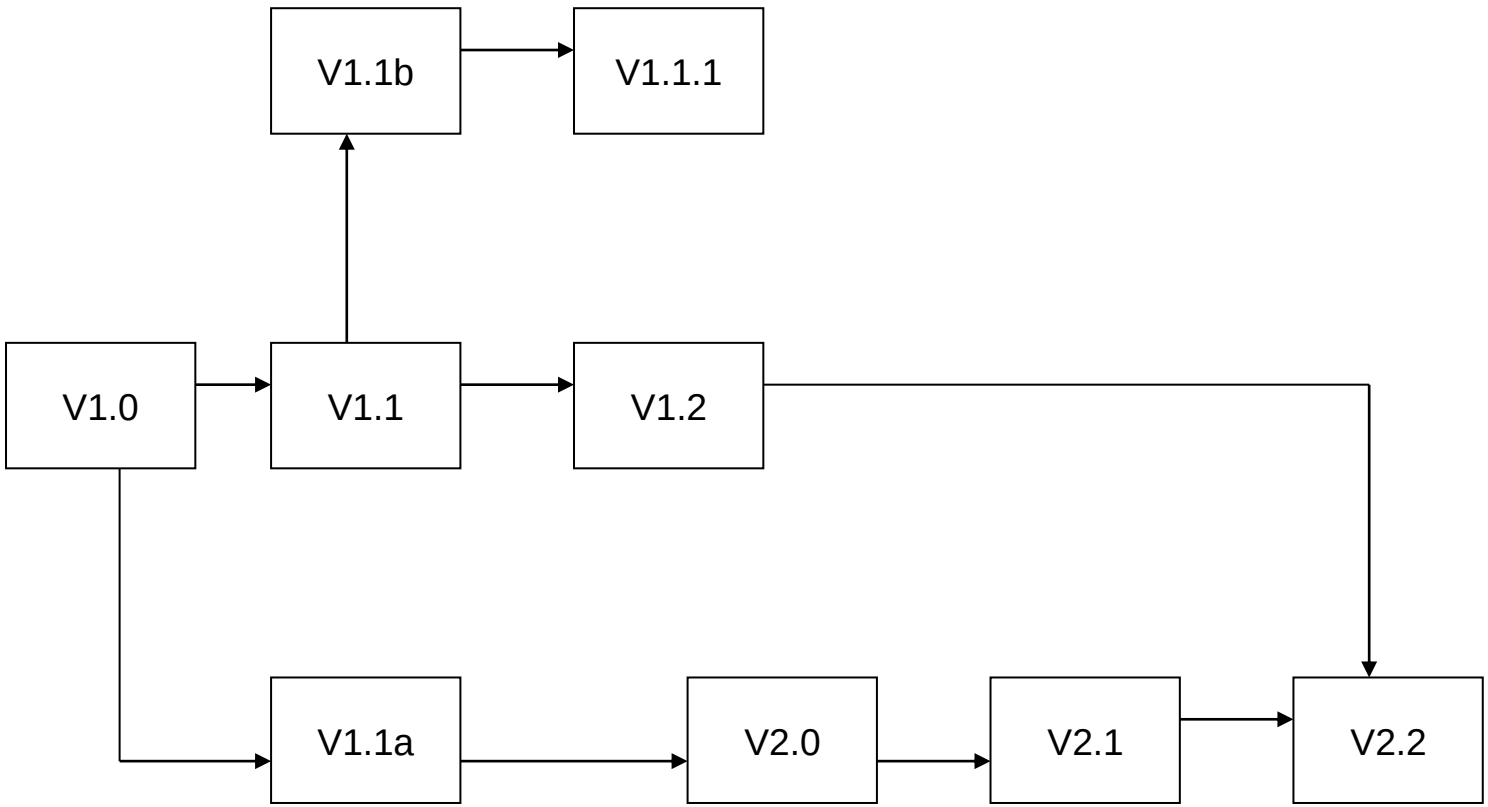
控制—版本控制

- 分支—是软件配置项同时沿着两个或多个分支展开，新版本独立地添加到各自的分支中。
- 合并—有选择地将分支或其它基线中对源文件所做的修改与主分支中相应的源文件对应起来的过程。
- 文件比较—用来比较两个或多个分支或基线中具有相同名字的文件，并识别这些不同的文件。

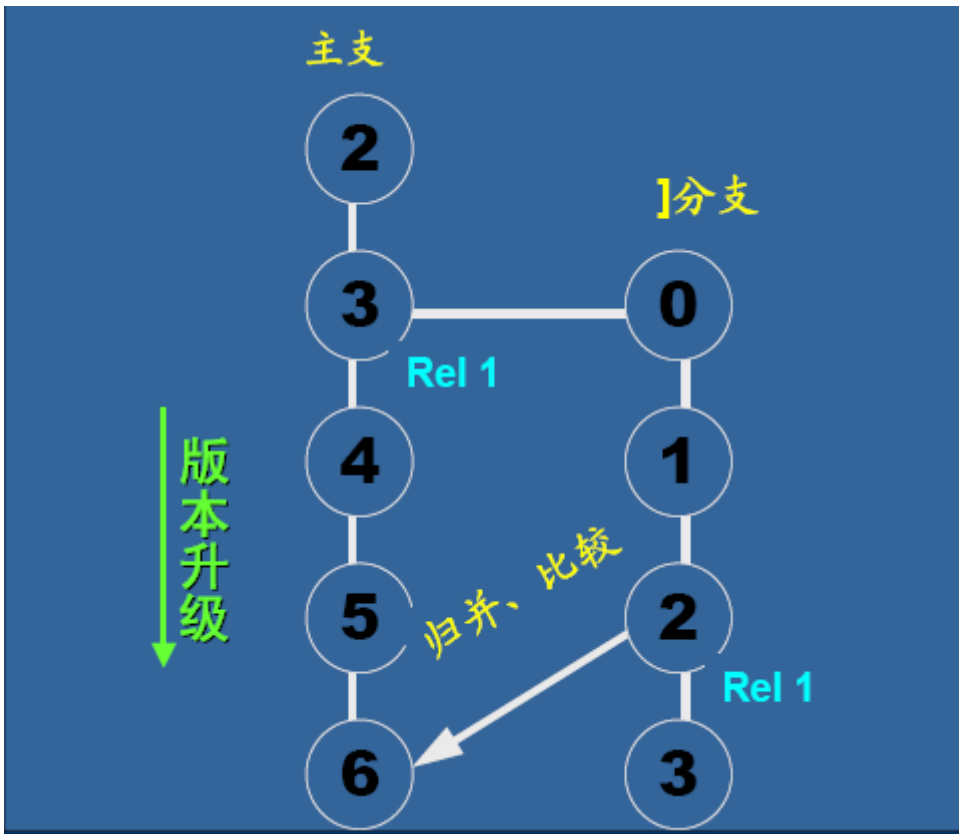
控制—版本管理

- 标识
  - 号码版本标识
  - 符号版本标识
- 版本管理工具

控制—例子



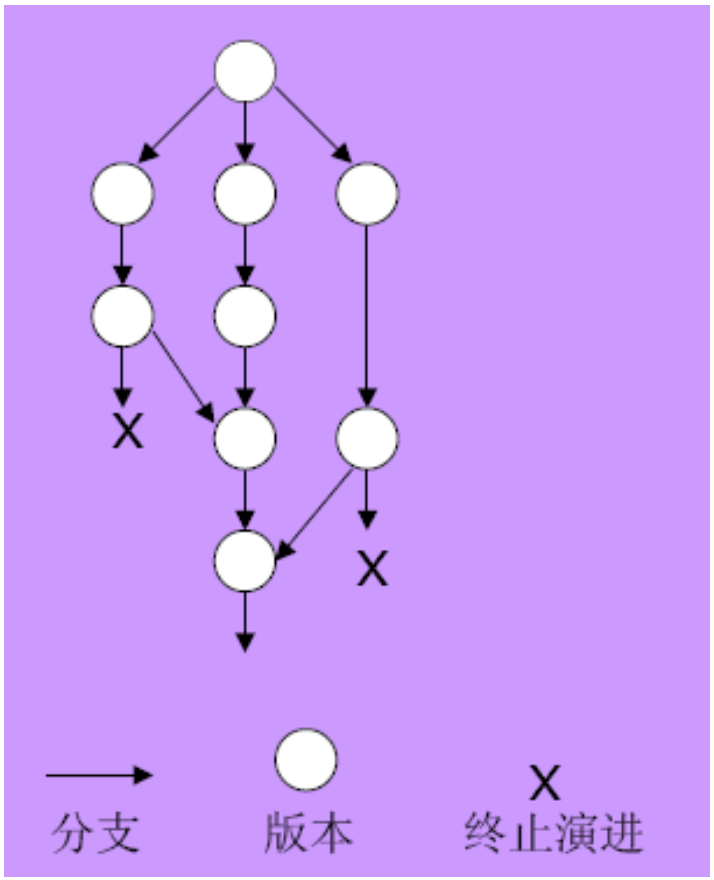
分支与合并



分支作用

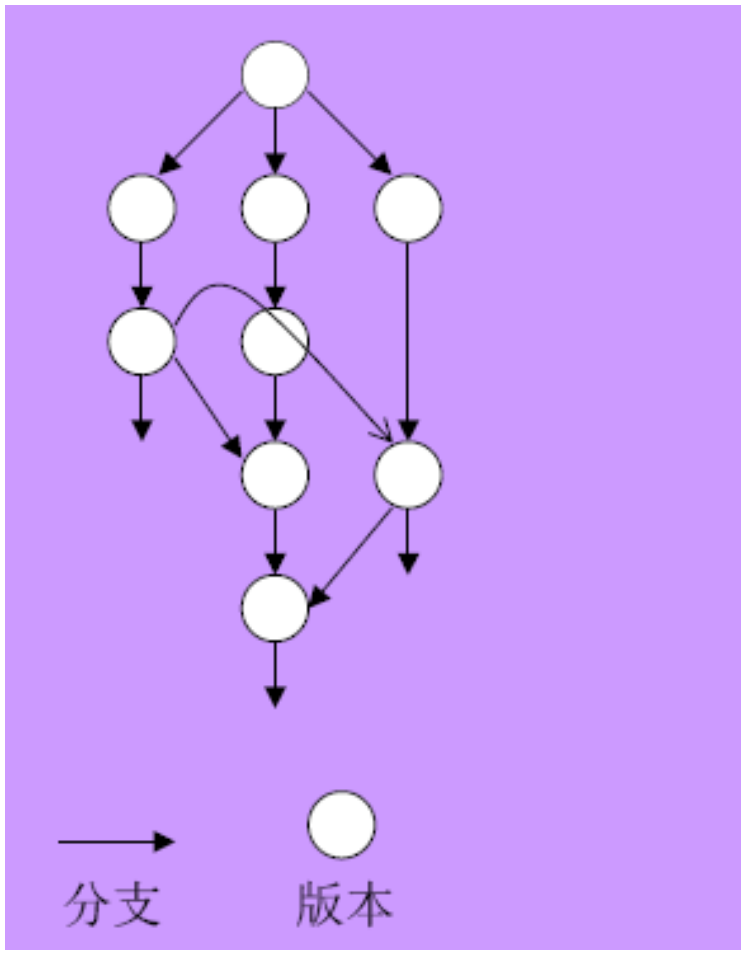
- 缺陷修正
- 多版本
- 并行开发

修正分支



- 子分支完成合并后，不再演进
- 所有合并必须合并到主分支
- 适用于对软件缺陷的修正工作管理

多版本分支（核心版片）



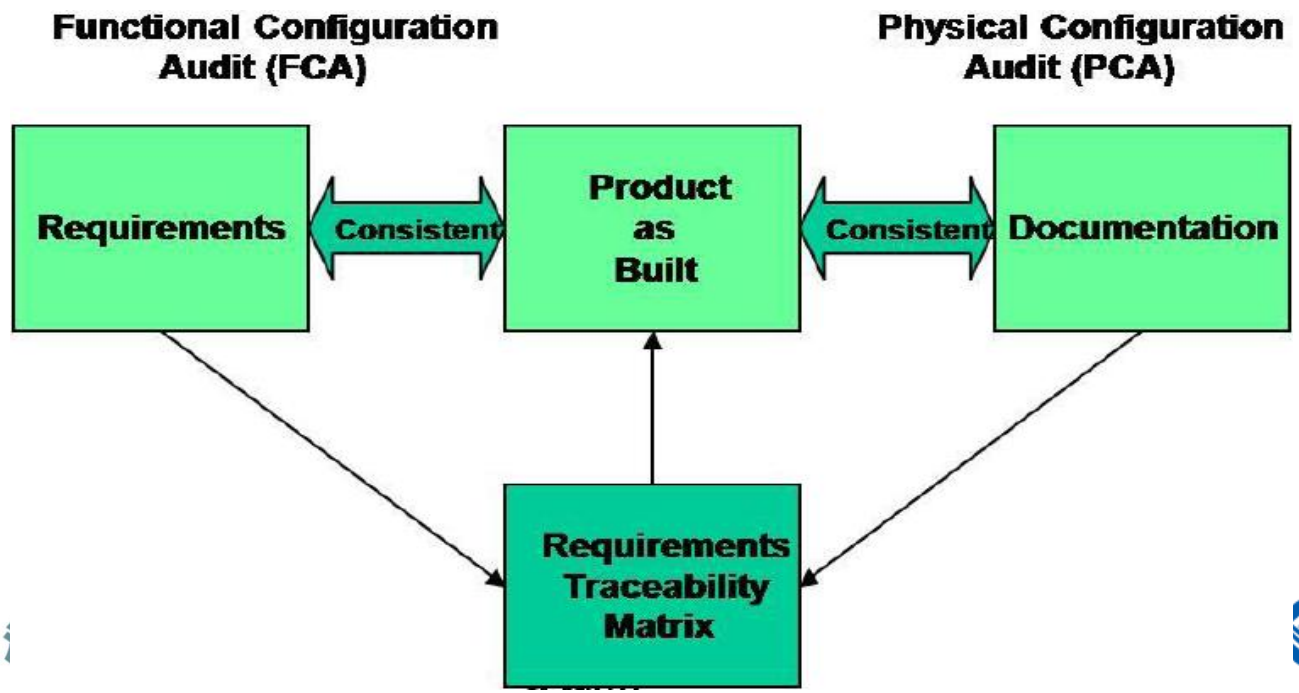
- 各个分支均可进行合并
- 但是所有的变化均要反映到主分支
- 适用于具有核心版本的产品
- 围绕核心版本推进各个分支的演进
- 一个子分支的演进可以通过主分支传递给其他子分支

配置审计—概念

- **配置审计**
  - 配置审计验证产品是否根据需求和合同来建立。对于存储配置项及相关记录的软件基线库的结构、内容和设施进行检验，其目的在于验证基线是否符合描述基线的文档。
- **验证包括：**
  - 对于产品的功能和性能的需求作比较，可参考需求跟踪矩阵
  - 配置项的处理是否有背离初始的规格说明或已批准的变更请求的现象；
  - 配置标识的准则是否得到了遵循；
  - 变更控制规程是否以遵循，变更记录是否可供使用
  - 是否保持了可追溯性。

审计—主要工作

- **功能配置审计**
  - 验证配置项的实际功效是与其软件需求一致的。
- **物理配置审计**
  - 确认每一个配置项符合预期的物理特性（版本、存放位置等），产品和相应的文件（如用户手册、操作指南、版本说明等）是否相符）。



审计—主要作用

- **确保软件配置管理的有效性，不允许出现任何混乱现象。例如：**
  - 防止出现向用户提交了不适合的产品，如交付了不适当版本的用户手册；
  - 发现不完善的实现，如开发出不符合初始规格说明或未按变更请求实施变更；
  - 找出各配置项间不匹配或不相容的现象；
  - 确认配置项已在所要求质量控制审查之后作为基线入库保存；
  - 确认记录和文档保持着可追溯性。

审计—工作步骤

- 由项目经理决定何时进行配置审计工作
- 质量保证组或软件组的配置管理组指定该项目的配置审计人员
- 项目经理和配置审计员决定审核范围。
- 配置审计员准备配置审计检查单
- 配置审计员安排时间审核文档和记录，审核活动可能涉及到：

项目范围	配置项的检入（check-in）及检出（check_out）
评审记录	配置项的变更历史
测试记录	文件的命名
变更请求	版本的编号

- 配置审计远在审核中发现不符合现象，并作记录。
- 由项目经理负责消除不符合现象。
- 配置审计员验证所有发现的不符合现象确已得到解决。

报告—配置状态

- 任务
  - 有效的记录和报告管理配置所需要的信息。
- 目的
  - 及时、准确的给出软件配置项的当前状况，供相关人员了解，以加强配置管理工作。

报告—信息

- 配置项的当前标识
- 已交付软件的配置
- 变更请求或问题报告的状态
- 已获准变更的状态

状态报告

- *发生了什么（What）?*
- *为什么要发生(Why)?*
- *谁做的(Who)?*
- *什么时候发生的(When)?*
- *在哪儿改变的(Where)?*

2. 配置管理体系构建

2.1 配置管理组织

软件配置管理实施级别	具体内容	履行角色	涉众/受益者
项目级	● 将配置管理任务纳入到项目总体计划之中 ● 对代码打基线，作为新开发的稳定基础 ● 建立适当的分支策略与结构 ● 编译代码 ● 创建带交付物的发布包 ● 参加 CCB 会议	项目中的兼职或专职配置管理员	项目经理及项目组成员
应用级（产品级）	● 评估应用（产品）级别的配置管理需求 ● 选择最适合该应用的配置管理技术	专职配置管理员、配置管理经理	产品经理
	● 定义应用级别的配置管理计划（独立地），如果在组织级别不存在的话 ● 为应用建立配置管理环境 ● 为应用建立配置管理流程（检出、检入、构建、发布、变更控制、问题管理等等） ● 执行配置管理培训		
组织级	● 确定配置管理知识域 ● 评估组织级别的配置管理风险 ● 评估配置管理的支持和赞助者 ● 定义 SCM 预算 ● 建立配置管理的针对性架构 ● 定义组织级别的配置管理计划 ● 定义通用配置管理术语	配置管理组，包括组长，发布经理等等	组织管理层

2. 配置管理体系构建

2.2 配置管理涉及到角色

- 项目经理（Project Manager，PM）：
  - 项目经理是整个软件研发活动的负责人，他根据软件配置控制委员会的建议批准配置管理的各项活动并控制它们的进程。
- 配置控制委员会（Configuration Control Board，CCB）：
  - 定制开发子系统；
  - 定制访问控制；
  - 制定常用策略；
  - 建立、更改基线的设置，审核变更申请；
  - 根据配置管理员的报告决定相应的对策。

2. 配置管理体系构建

2.2 配置管理涉及到角色

- 配置管理员（Configuration Management Officer，CMO）：

- 根据配置管理计划执行各项管理任务，定期向CCB提交报告，告，并列席CCB的例会。其具体职责为以下几项：
- 软件配置管理工具的日常管理与维护；
- 提交配置管理计划；
- 各配置项的管理与维护；
- 执行版本控制和变更控制方案；
- 完成配置审计并提交报告；
- 对开发人员进行相关的培训；
- 识别软件开发过程中存在的问题并拟就解决方案。

2. 配置管理体系构建

2.2 配置管理涉及到角色

- 系统集成员（System Integration Officer，SIO）：

  - 系统集成员负责生成和管理项目的内部和外部发布版本，具体职责为以下几项：
  - 集成修改；
  - 构建系统；
  - 完成对版本的日常维护；
  - 建立外部发布版本。

- 开发人员（Developer，DEV）：

  - 开发人员的职责就是根据组织内确定的软件配置管理计划和相关规定，按照软件配置管理工具的使用模型来完成开发任务。

2. 配置管理体系构建

2.3 配置管理基本流程

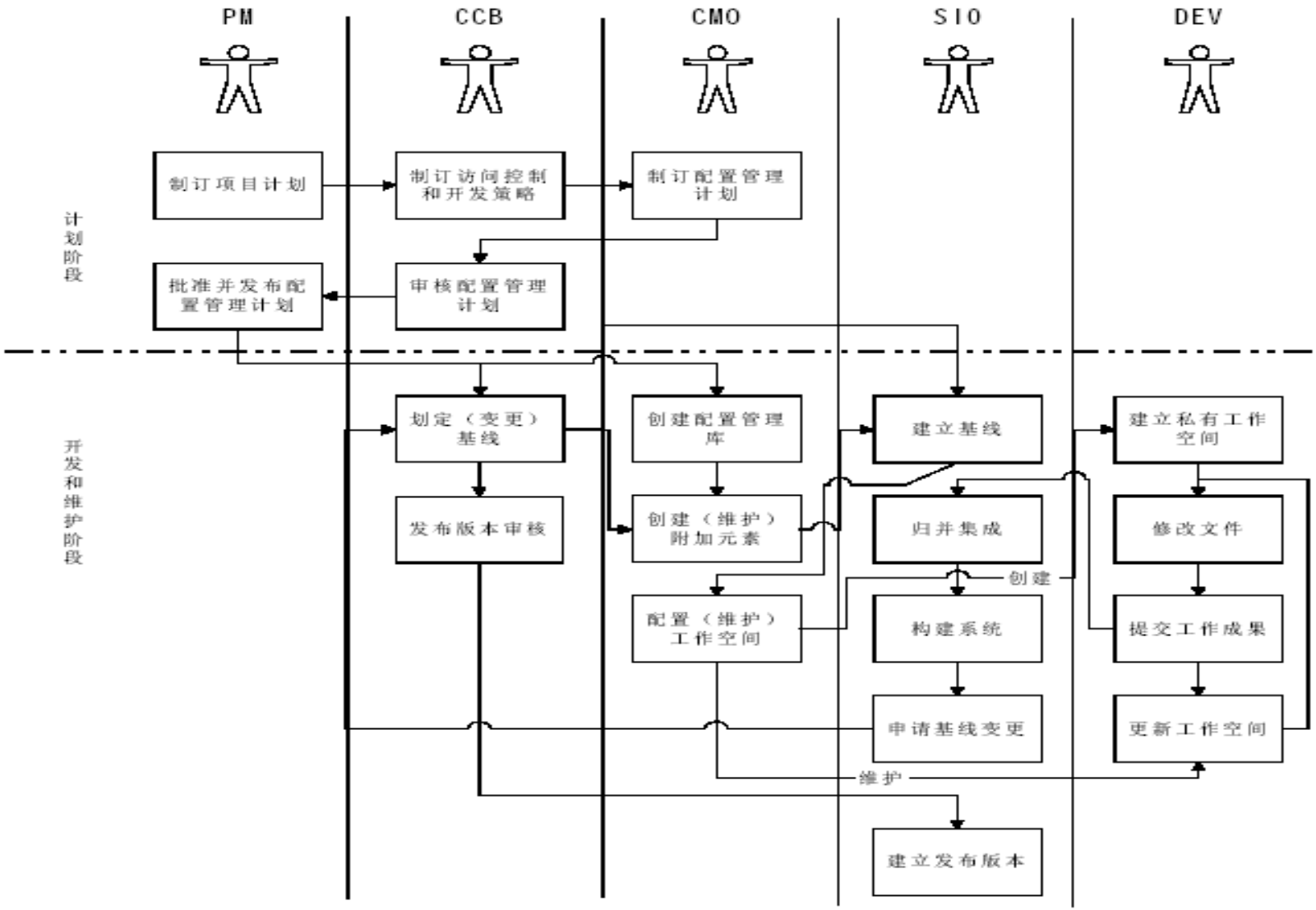
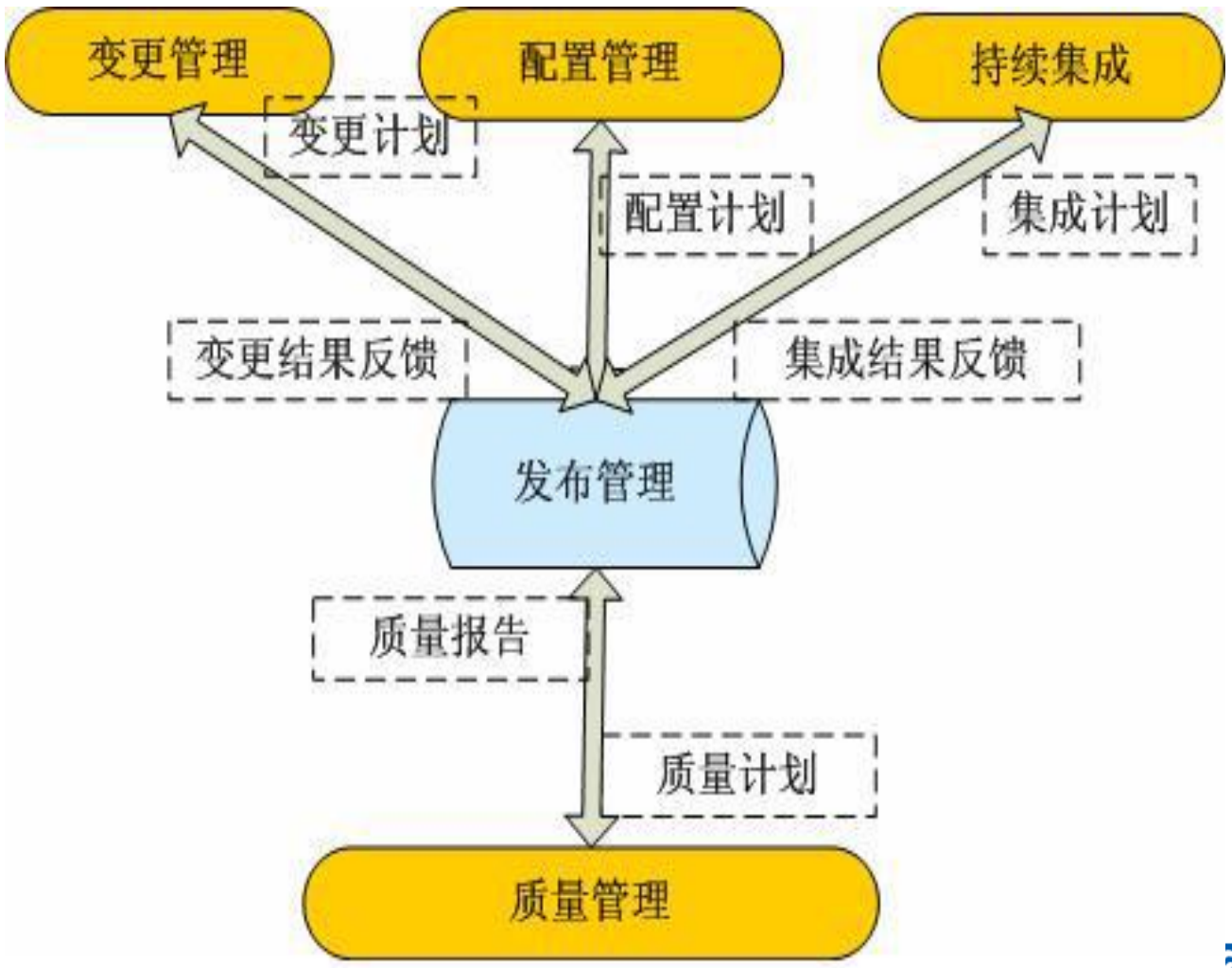


图 1. 软件配置管理基本流程

2. 配置管理体系构建

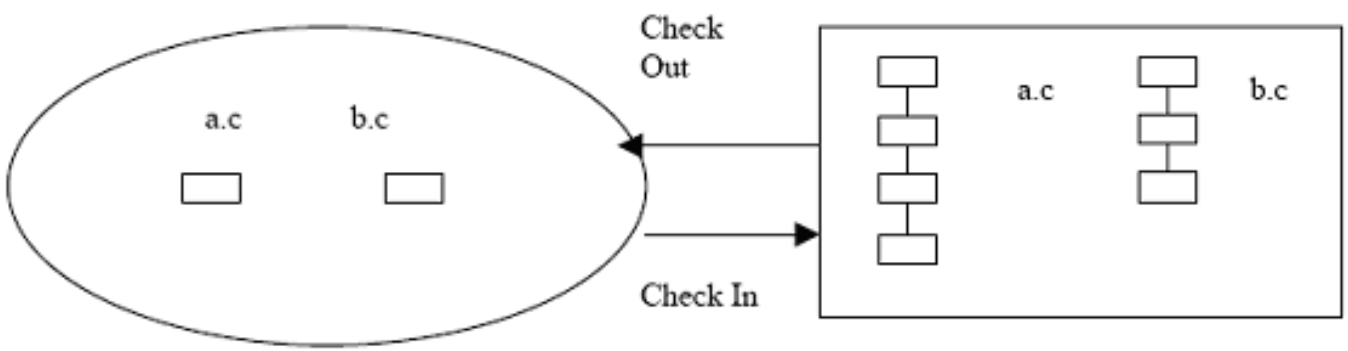
2.4 发布管理与配置管理关系



3.配置管理商业模型

◆ 3.1 CICO模型

- CICO模型主要关注的是单个文件的版本控制。图显示了一个支持CICO模型的CM系统的工作过程。用户利用库和文件系统来进行工作。
- 文件被版本化并存储到库中，新版本的产生是由库工具控制的。然而， 文件在库中不是可以直接存取的，用户必须去检出（即Check Out）一个文件的版本到工作空间中以便读取它的内容。
- 更改后的文件可以被检入库中（即Check in）,产生文件的一个新版本。
- 代表工 具： SCCS和CVS



3.配置管理商业模型

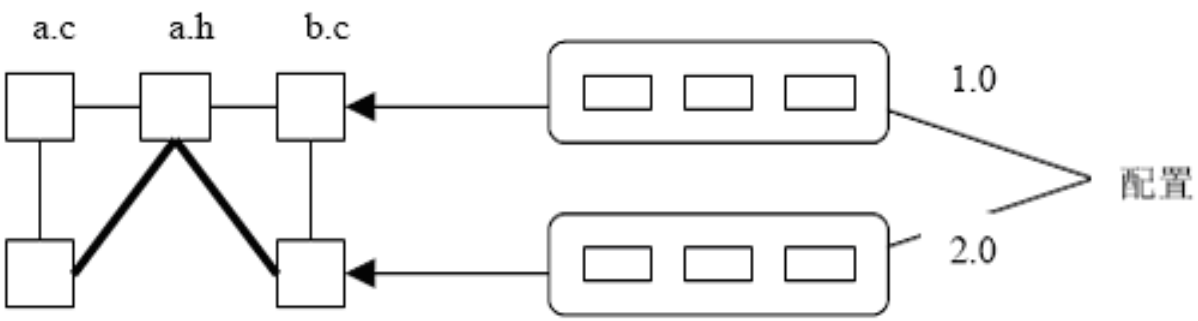
◆ 3.2 组织模型

- 组织模型由CICO模型自然导出，建立于构件版本图的基础之上，同时依赖于存储库和工作空间的概念，可以通过对构件加锁进行并发控制。
- 组织模型的重点是在CM系统支撑下加强了对创建配置、对有关的历史信息的管理和使用他们作为工作环境的支持。
- 代表工 具： CC

3.配置管理商业模型

◆ 3.3 长事务模型

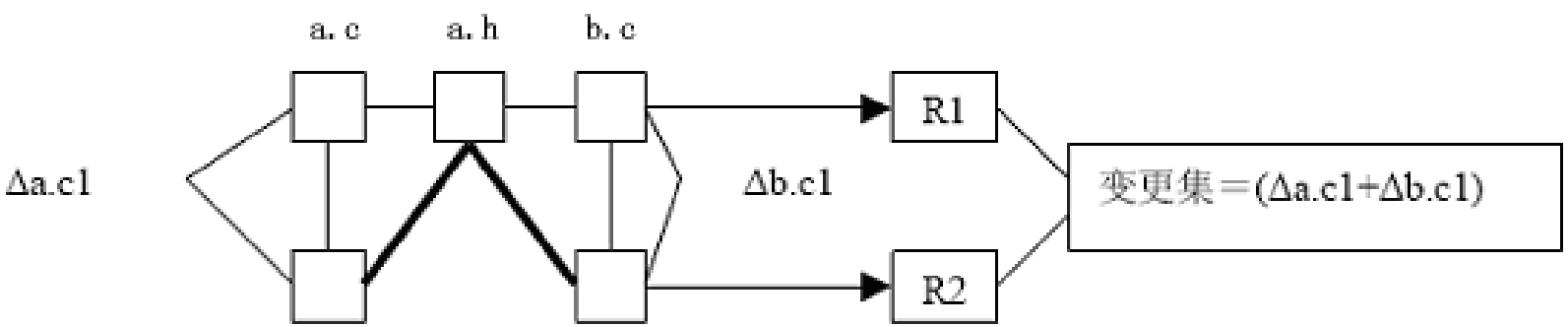
- 长事务模型主要支持包括一系列原子变更的全系统演变和由团队开发员对系统变更的协调。
- 开发员主要操作配置而非单独的构件。
- 长事务由两个概念组成：工作空间和并发控制方案。
- 工作空间来源于存储库或一个封闭工作空间中的一个固定配置。
- 代表工 具： NSE



3.配置管理商业模型

◆ 3.4 变更集模型

- 主要集中于对系统配置的逻辑变更的支持。
- 在这个模型中引入的变更集表示组成逻辑变更的对不同构件修改的集合，它是创建变更的活动完成后对逻辑变更的记录。
- 支持这个模型的CM系统用户可以直接操作变更集。在变更集模型中，配置可描述为由基线和一组变更集组成。
- 代表工具：UCM和SABLIME



4.常用配置管理工具—VSS

4.1 Visual SourceSafe

- Visual SourceSafe是Microsoft公司推出的配置管理工具，是Visual Studio的套件之一。SourceSafe是国内最流行的配置管理工具，用户量绝对是第一位。

VSS的简单工作原理

- 将项目所有的源文件（包括各种文件类型）以特有的方式存入数据库
- 客户端将程序拷贝到各自的工作目录下进行调试修改，然后进行Checkin到服务器，进行综合更新
- 支持多个项目之间文件快速高效的共享
- 每个成员对项目文件所作的修改将被记录到数据库中;
- VSS可以很容易地与Microsoft Access、 Visual Basic、 Visual C++、Visual FoxPro和其他的开发工具集成在一起

4.常用配置管理工具—VSS

- VSS的优点
  - “简单易用，一学就会”
- SourceSafe的主要局限性
  - 只能在Windows下运行，不能在Unix, Linux下运行。SourceSafe不支持异构环境下的配置管理，对用户而言是个麻烦事。这不是技术问题，是微软公司产品战略决定的
  - 适合于局域网内的用户群，不适合于通过Internet连接的用户群，因为SourceSafe是通过“共享目录”方式存储文件的

4.常用配置管理工具—CVS

4.2 Concurrent Version System

- CVS 是 Concurrent Version System（并行版本系统）的缩写，它是著名的开放源代码的配置管理工具
- CVS的官方网站是http://www.cvshome.org/
- 官方提供的是CVS服务器和命令行程序，但是官方并不提供交互式的客户端软件
- 许多软件机构根据CVS官方提供的编程接口开发了各色各样的CVS客户端软件，最有名的当推Windows环境的CVS客户端软件——WinCVS
- WinCVS是免费的，但是并不开放源代码

4.常用配置管理工具—CVS

◆ 4.2 CVS与VSS比较

- SourceSafe有的功能CVS全都有，CVS支持并发的版本管理，SourceSafe没有并发功能。CVS服务器的功能和性能都比SourceSafe高出一筹

- CVS服务器是用Java编写的，可以在任何操作系统和网络环境下运行。CVS深受Unix和Linux 的用户喜爱。Borland公司的JBuilder提供了CVS的插件，Java程序员可以在JBuilder集成环境中使用CVS进行版本控制

- CVS服务器有自己专用的数据库，文件存储并不采用SourceSafe的“共享目录”方式，所以不受限于局域网，信息安全性很好

4.常用配置管理工具—CVS

4.2 CVS缺点

- CVS的主要缺点在于客户端软件，真可谓五花八门、良莠不齐
- Unix和Linux 的软件高手可以直接使用CVS命令行程序，而Windows用户通常使用WinCVS
- 安装和使用WinCVS显然比SourceSafe麻烦不少，这是令人比较遗憾的

4.常用配置管理工具—CC

◆ 4.3 ClearCase

- Rational公司的ClearCase是软件行业公认的功能最强大、价格最昂贵的配置管理软件
- ClearCase主要应用于复杂产品的并行开发、发布和维护，其功能划分为四个范畴：
- 版本控制、
- 工作空间管理（Workspace Management）、 构造管理（Build Management）、
- 过程控制（Process Control）。
- ClearCase通过TCP/IP来连接客户端和服务器。
- 另外，ClearCase拥有的浮动License可以跨越UNIX和Windows NT平台被共享。

4.常用配置管理工具—CC

4.3 ClearCase

- ClearCase的功能比CVS、SourceSafe强大得多，但是其用户量却远不如CVS、SourceSafe的多。主要原因是：
- ClearCase价格昂贵，如果没有批量折扣的话，每个License大约5000美元。对于中国用户而言，这无疑是天价
- 用户只有经过几天的培训后（费用同样很昂贵），才能正常使用ClearCase。如果不参加培训的话，用户基本上不可能无师自通
- 选择配置管理工具应当综合考虑价格、易用性和功能因素，而不是购买最先进的工具。令人满意的工具通常是价格低廉、简便易用、功能恰好够用

4.常用配置管理工具—SVN

4.4 Subversion

- Subversion版本控制系统，它可以管理各个时刻的文件和目录
- Subversion将文件存放在repository库中。这个仓库非常类似于一个普通的文件服务器，只是它还可以记录文件和目录曾经做过的每次变更
- 可把版本控制系统比作一种“时间机器”
- Subversion的仓库可以通过网络来访问，允许不同的用户在不同的计算机上使用
- Subversion是一个自由的、开放源码的版本控制系统
- Subversion只是版本控制系统，不是软件配置管理系统（SCM）

5.配置管理人员素质

- 个人素质
  - 良好职业道德
  - 沟通技巧
  - 细心
  - 有耐心
- 专业知识
  - 项目管理的知识
  - 软件配置管理的专业知识
  - IT背景知识,对系统（操作系统、网络、数据库等方面）比较熟悉

谢 谢!