

Innovative Solutions for Insurance

CC-CQ 配置管理实践

-- eBaoTech的CM改进经验分享

安卫宾



November 14, 2006

Copyright © 2006 eBaoTech Corporation

Agenda

- 背景
- 为什么选择CC-CQ?
- 基于CC-CQ配置管理改进
- 几种典型的UCM使用模式
- 改进情况总结

背景 – eBaoTech

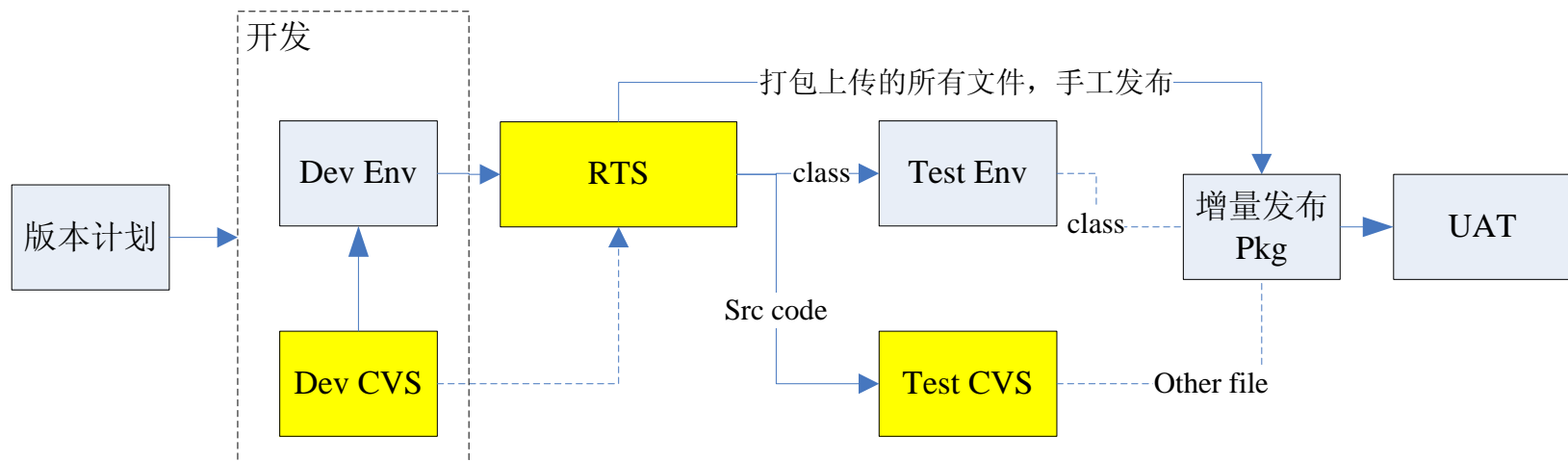
- 易保网络技术有限公司是世界领先的保险应用软件提供商，致力于为产、寿险公司提供基于Java的新一代保险核心业务应用软件解决方案。
- 易保的客户包括来自亚洲、欧洲的众多知名保险公司。和传统的软件开发商相比，易保的软件产品在产品创新、提高效率、降低成本及缩短实施时间方面具有明显的优势，因而获得了客户的广泛赞誉。

eBaoTech

Innovation Solutions for Insurance

背景 – CM in eBaoTech

- CM工具及存在的问题
 - 使用CVS和RTS工具进行配置和发布管理
 - 由于RTS工具的流程控制和可靠性的问题，导致
 - 源码和环境存在不一致性的风险
 - 开发团队成员的代码与任务的匹配性管理困难，无法跟踪到节点
 - 版本发布出错频率较高



Agenda

- 背景
- 为什么选择CC-CQ?
- 基于CC-CQ配置管理改进
- 几种典型的UCM使用模式
- 改进情况总结

为什么选择CC-CQ - 开发过程存在的问题

现状

• 人力资源成本

- 客户化产品需要大量的维护人员，人力成本高
- 基于复制的客户化产品，阻碍了开发成果的共享

• 项目进度管理

- 对变更和缺陷进度的监控较弱，无法准确监控项目进度
- 对开发人员的工作量和进度的监控较弱

• 项目质量管理

- 开发版本、测试版本和发布版本存在不一致性的风险
- 无法跟踪需求、变更或缺陷到代码版本

• 软件资产的安全性

- 由于复本过多，权限管理难度较大
- 人员流动可能造成资产流失风险

• 主要目标

- 降低易保软件开发和管理成本
- 提高易保软件产品的质量
- 提高易保项目监控和管理水平
- 确保易保软件资产安全

为什么选择CC-CQ? – 基于CC/CQ的解决方案

• 人力资源成本

- 核心产品特性通过分支和归并实现客户化之间的共享，降低产品开发和维护成本

(存在问题:

1. 对配置管理的管理员要求较高，需要经过系统培训)

• 项目进度管理

- 通过CC-CQ统一变更管理平台，基于任务驱动项目开发过程
- 实时监控项目进度及开发人员工作量

(存在问题:

1. 需求和变更流程、报表查询需要定制)

CC-CQ解决方案

• 主要目标

- 降低易保软件开发和管理成本
- 提高易保软件产品的质量
- 提高易保项目监控和管理水平
- 确保易保软件资产安全

为什么选择CC-CQ? – 基于CC-CQ的解决方案

CC-CQ解决方案

• 项目质量管理

- 通过基线和分支管理，避免开发、测试、发布多套代码存在
- 通过CC-CQ统一变更管理，需求、变更或缺陷到代码版本和代码行之间关联，实现严格变更控制，而且便于实现Peer Review
- 有效贯彻开发策略的实现
(存在问题：
1. 对配置管理的管理员要求较高，需要经过系统培训)

• 主要目标

- 降低易保软件开发和管理成本
- 提高易保软件产品的质量
- 提高易保项目监控和管理水平
- 确保易保软件资产安全

• 软件资产的安全性

- 集中管理各分支的代码权限
- 统一权限管理
(存在问题：
1. 对配置管理的管理员要求较高)

Agenda

- 背景
- 为什么选择CC-CQ?
- 基于CC-CQ配置管理改进
- 几种典型的UCM使用模式
- 改进情况总结

基于CC-CQ的配置管理改进 - Overview

CM流程 的制定

- 配置管理指导书
- 版本管理流程
- 变更管理流程

CC-CQ环境 的建立

- 规划和搭建CC-CQ环境，CQ变更流程的二次开发
- CC-CQ预安装的制作和发布
- 完成VOB的规划和建立

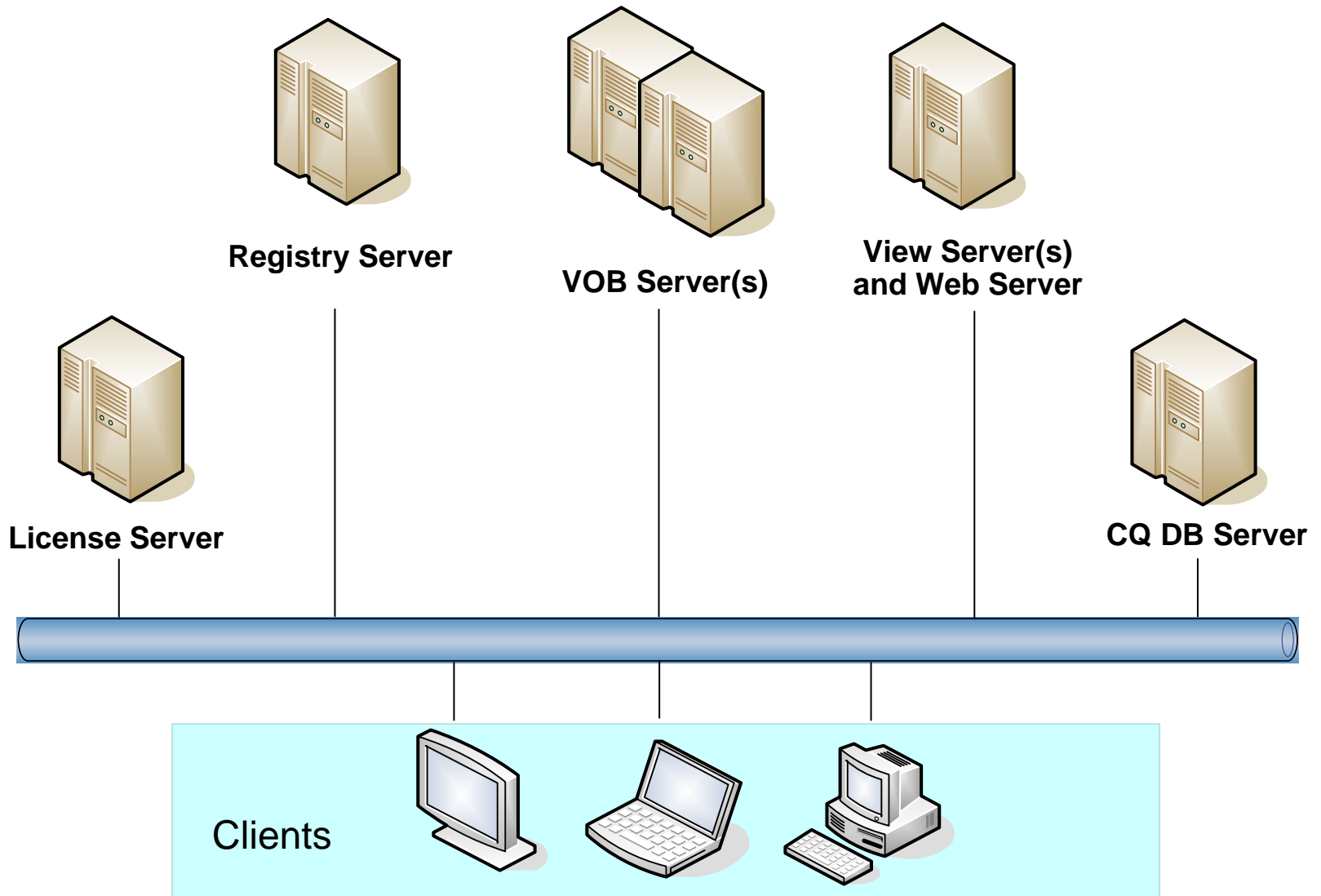
源码/发布脚本 整理和导入

- 整理完整的源代码和发布脚本，形成基线版本导入CC
- 完成权限和项目设置，形成可用的项目环境

培训及 Support

- 对整个开发团队进行配置管理流程及CC-CQ工具的培训
- 现场指导开发团队使用，解决问题

CC-CQ平台 – 典型的环境



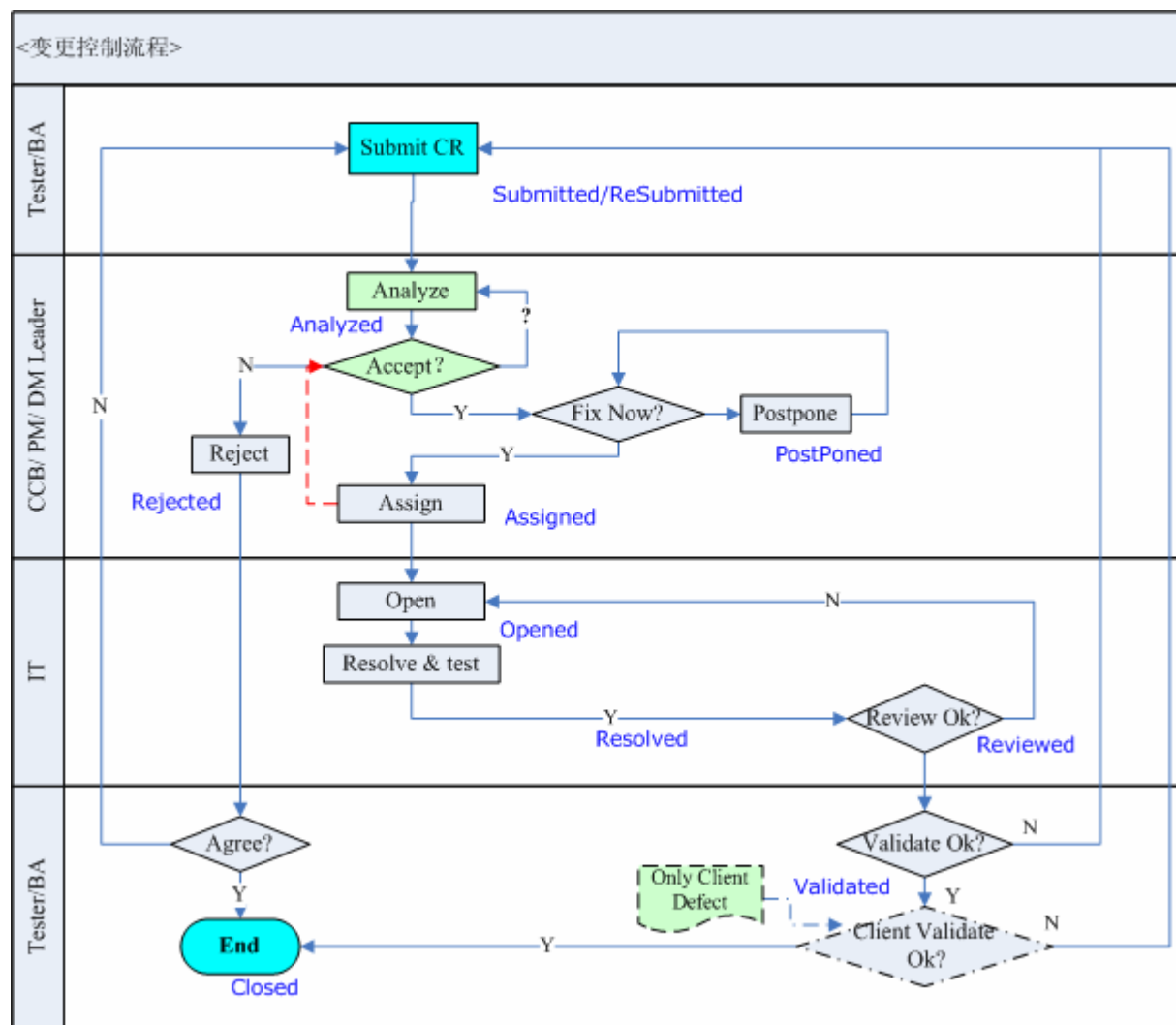
CC-CQ平台 – 变更管理流程的定制

- 变更控制流程

- 通过变更控制流程的控制，保证了只有Assign状态的CR，项目成员才能在CC中修改
- 只有Resolved状态的CR，才能Deliver到集成流

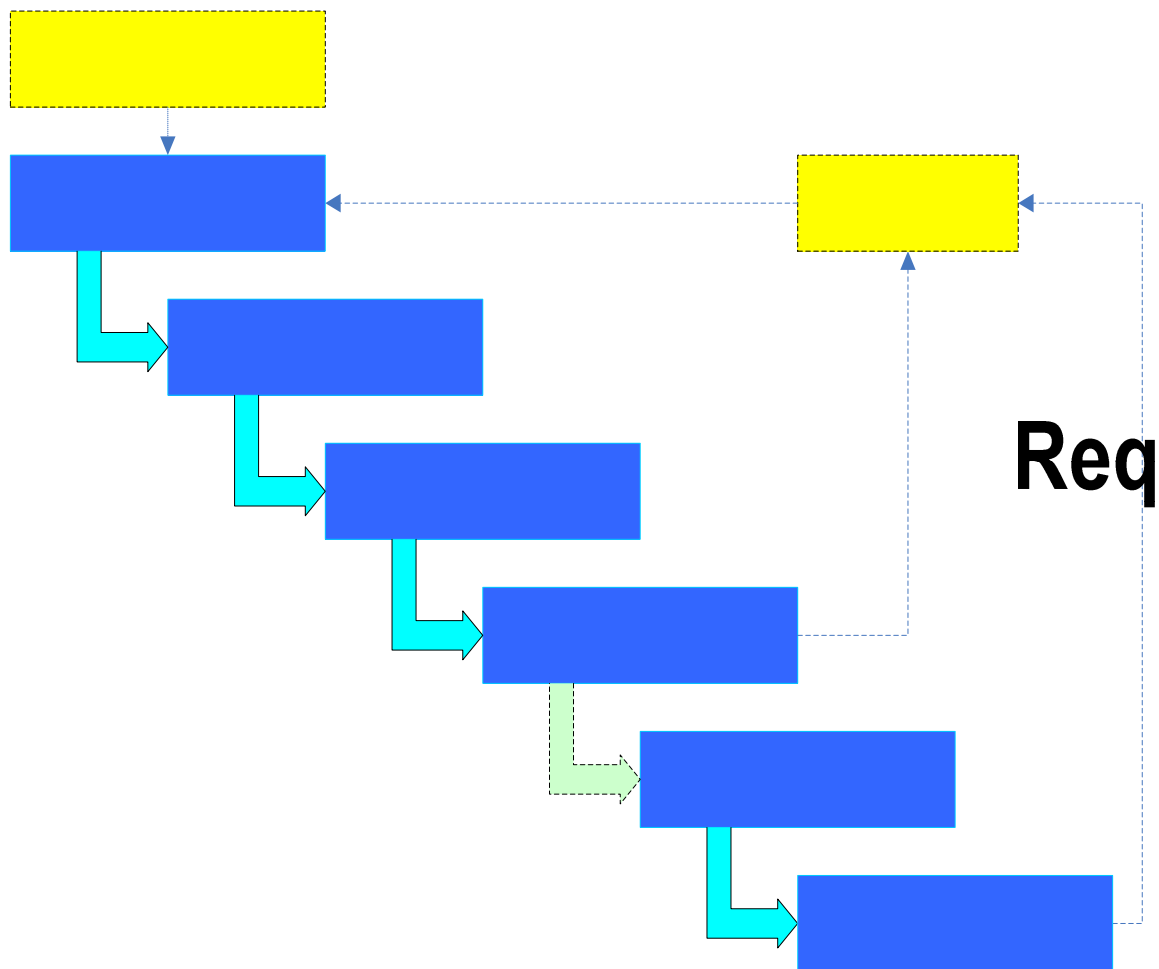
- CQ定制的一些建议

- 流程统一，账号统一
- 集中管理
- 增强可配置性
- 实现与其他项目管理工具的集成



CC-CQ平台 – 版本管理流程的定制

- 版本管理的流程，将版本纳入全生命周期的管理
 - 版本计划
 - 版本的构建和发布
 - 版本测试
 - 版本外发等
- 版本管理的流程与变更控制流程集成在一起使用
- 版本自动构建的工具与CC-CQ集成，实现全量及增量的发布



Agenda

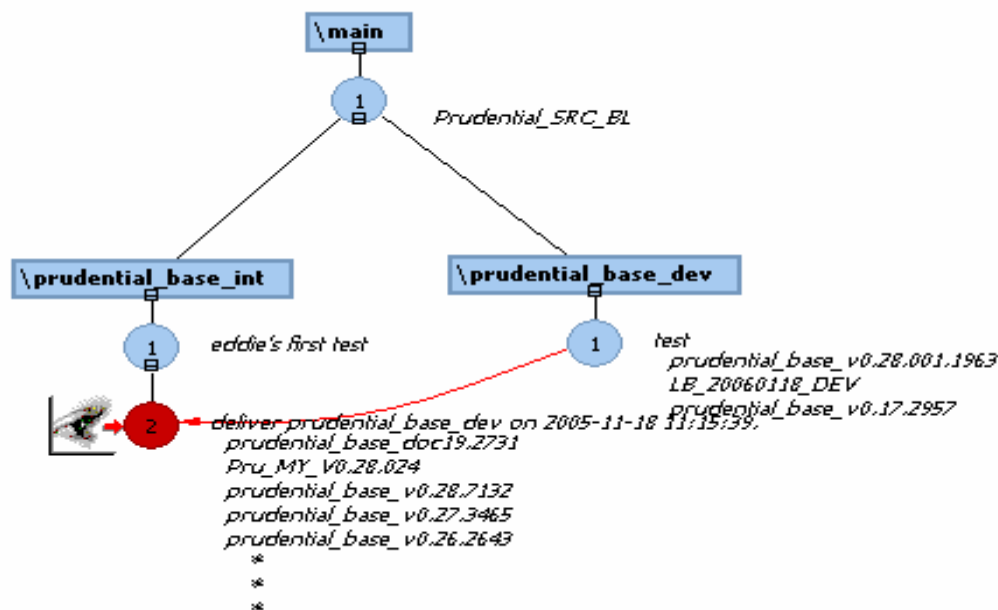
- 背景
- 为什么选择CC-CQ?
- 基于CC-CQ配置管理改进
- 几种典型的UCM使用模式
- 改进情况总结

CC-CQ的使用方式- UCM

- UCM 是在Base CC的基础上对开发过程进行了封装，采用UCM可以使项目更快，更有效的使用CC进行管理，而不需要掌握Base ClearCase的细节.
- 主要特点
 - 封装的开发流程，提供比较固定的使用模式
 - CC-CQ集成使用，严格的由CQ中的变更驱动CC中任何配置项的修改
 - 易于管理：通过流封装了分支及分支上的活动，使管理活动的粒度从文件级扩展为Activity级

CC-CQ的使用方式- UCM开发模型

- 默认模型
 - 一个集成流
 - 每个开发人员使用私有开发流
 - 开发完成后，开发人员将活动 Deliver 集成流
 - 基线生成后，开发人员Rebase自己的开发流，同步其他开发人员的修改
- 单共用开发分支模型
 - 所有的开发人员共用一个开发分支
 - Deliver的权限受控
- 多共用开发分支模型
 - 按照不同的开发小组，隔离开发活动
 - Deliver的权限受控



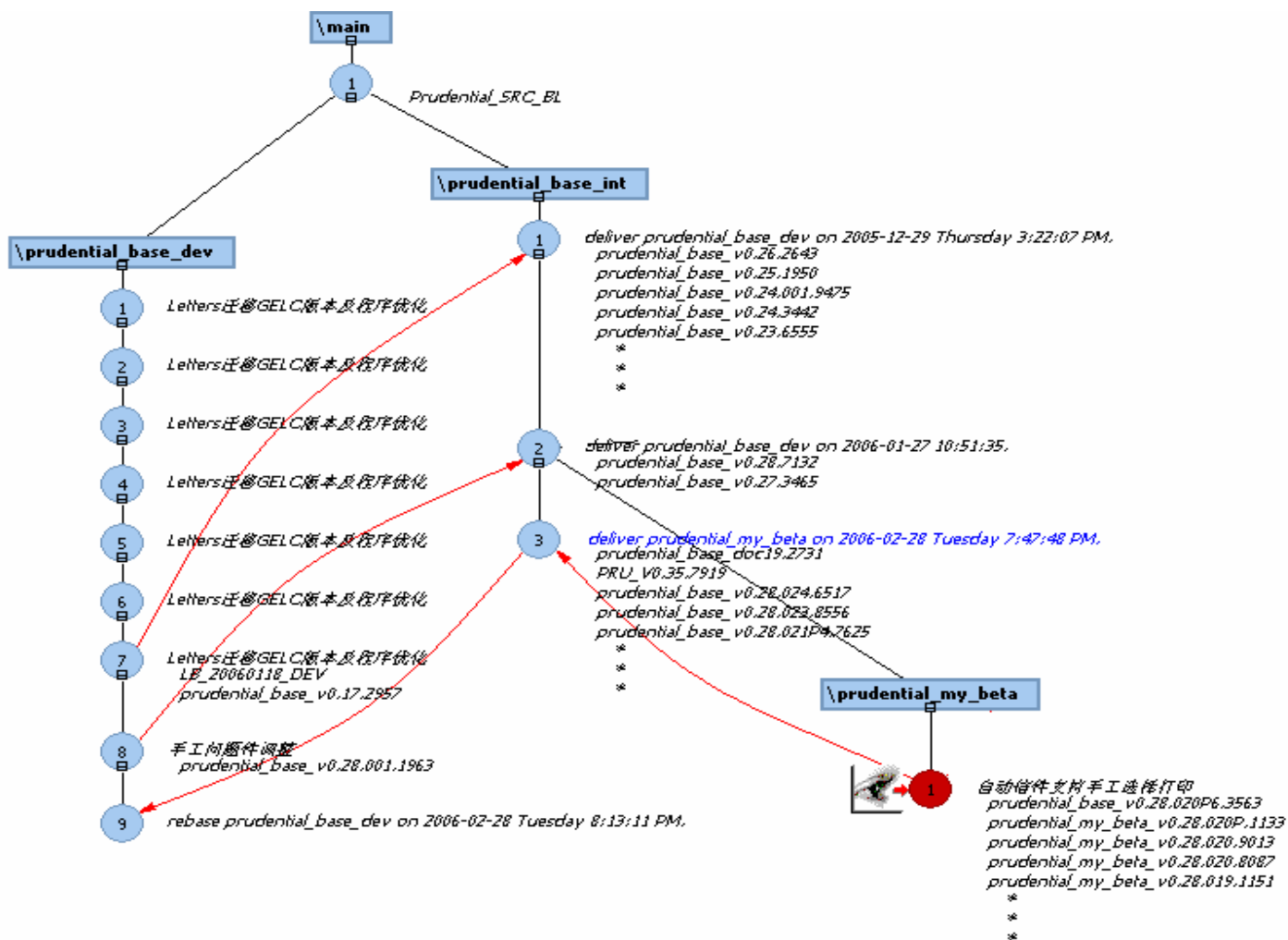
CC-CQ的使用方式- UCM开发模型

- 开发与Bug修复并行

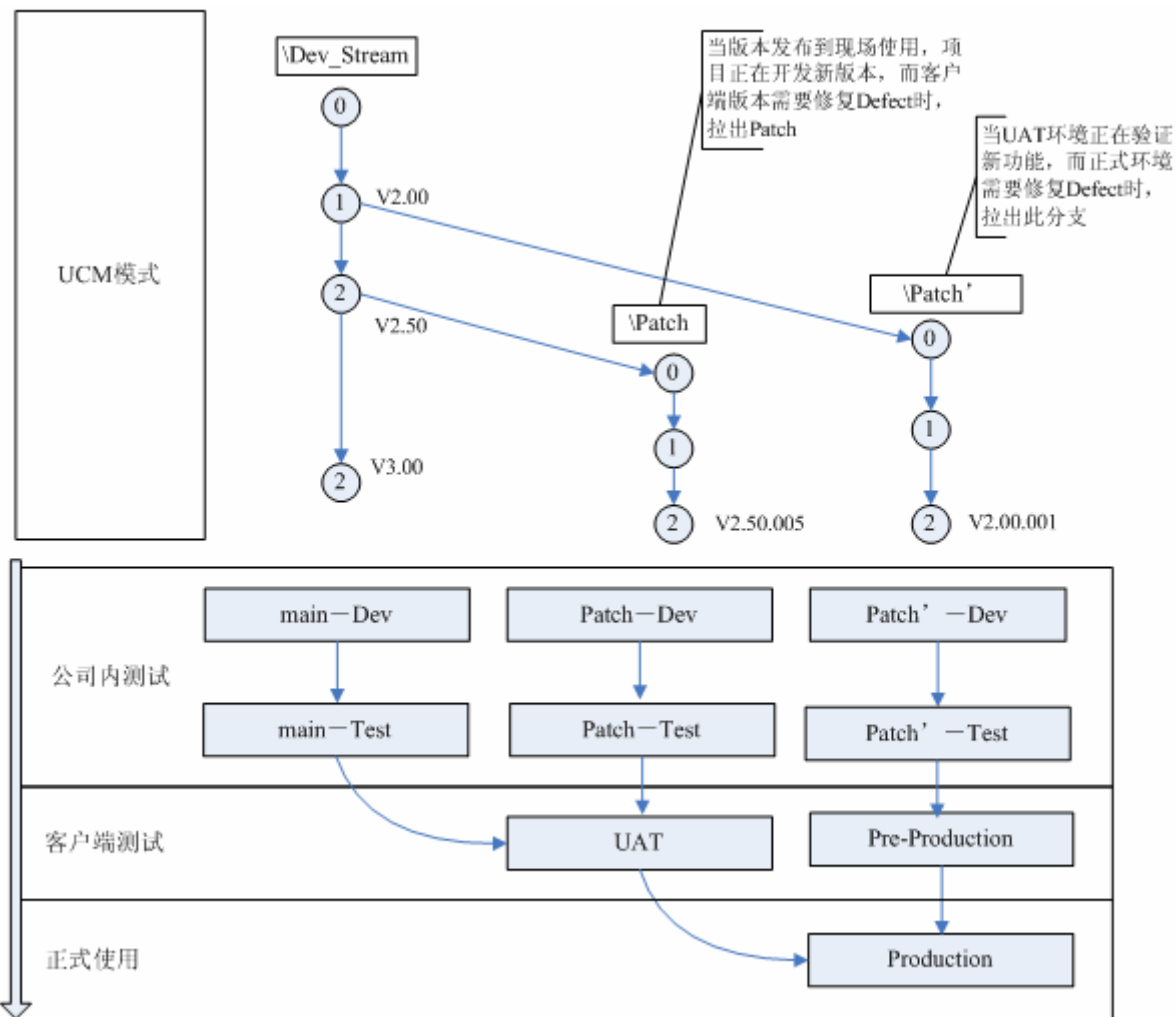
- 集成流
- 2个开发流
 - 新功能
 - Bug修复

- 缺点:

- Bug修复周期必须很短



CC-CQ的使用方式- UCM开发模型与环境对应



Agenda

- 背景
- 为什么选择CC-CQ?
- 基于CC-CQ配置管理改进
- 几种典型的UCM使用模式
- 改进情况总结

改进情况总结

- 在试点项目实施CC-CQ之后，在如下方面取得了比较好的效果
 - 版本计划得到有效的跟踪和管理
 - 项目开发效率有明显的提高
 - 项目的源码和环境的一致性得到保证
 - 项目不同环境对应的代码实现在分支间方便的归并，节约了大量的人力
 - 项目版本的质量改善明显
- 该项目组被评选为公司的Quality Star，而CM团队也因成功支持该项目，被评选为公司的Innovation Star.

***Thank You
&
Discussion !***