

# 构建可靠的自动化发布体系

葛隼@携程  
junge@ctrip.com



## 苦逼的开发



# CONTENTS

一、Coding之外的瓶颈

二、建立面向交付的运维体系

三、快速迭代中的质量管理

四、构建可靠的自动化发布

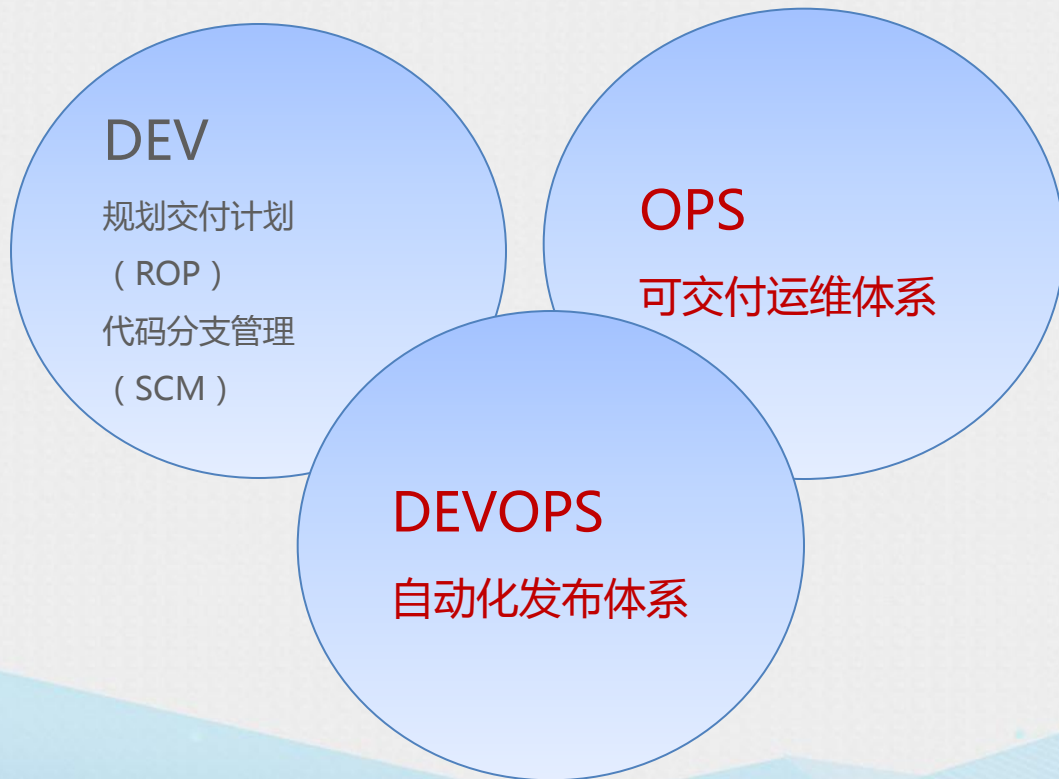
### ■ 面临的挑战

1. **团队协同**：13+SBU事业部，1900+人团队协作交付
2. **迭代压力**：每周800+项目或变更，如何构建可靠的测试/开发环境
3. **项目间应用相互依赖**：不能管理依赖版本就会出现混乱
4. **排障需要透明**：开发要求更多生产权限，运营要求限制开发权限
5. **发布过程保障**：平滑的用户访问，99.99%网站可用性（一秒价值¥ 5000）
6. **上线压力**：半夜的上线过程是每个程序员的煎熬和噩梦

互联网公司的特征：**快**

**产品变化快，交付要更快！**

### ■ 我们的结构





### 开发和运维的鸿沟

#### DEV

- 忙于解决需求和Bug
- 不愿意关心环境问题
- 生产排障，像跨栏
- 上线问题，项目延期

#### OPS

- 生产可用性99.99%
- 标准化运营流程
- 生产故障，只能升级事件
- 开发需求太多，满足不了

### ■ 打破鸿沟，关键看理念

1. DEV：提前规划项目，预支OPS资源
2. OPS：规划权利边界，透明运维体系，标准资源，被动转主动
3. DEVOPS：管理代码分支，环境构建，预控环境和发布风险

### ■ 关键点

1. 应用须支持裸机发布
2. 可伸缩的云服务器资源体系（标准化IDC资源）
3. 环境变量管理（应用配置文件\容器配置文件\Pool依赖关系）
4. 自动化测试、性能测试体系
5. 透明的服务器监控体系（Zabbix\XMon）



### 代码、配置和环境分离，自动构建环境

容器配置

站点和环境配置工具  
Domain Model

服务器

应用包

**Release Branch**  
APP Code  
ConfigProfile

ConfGen



一键部署

Environment

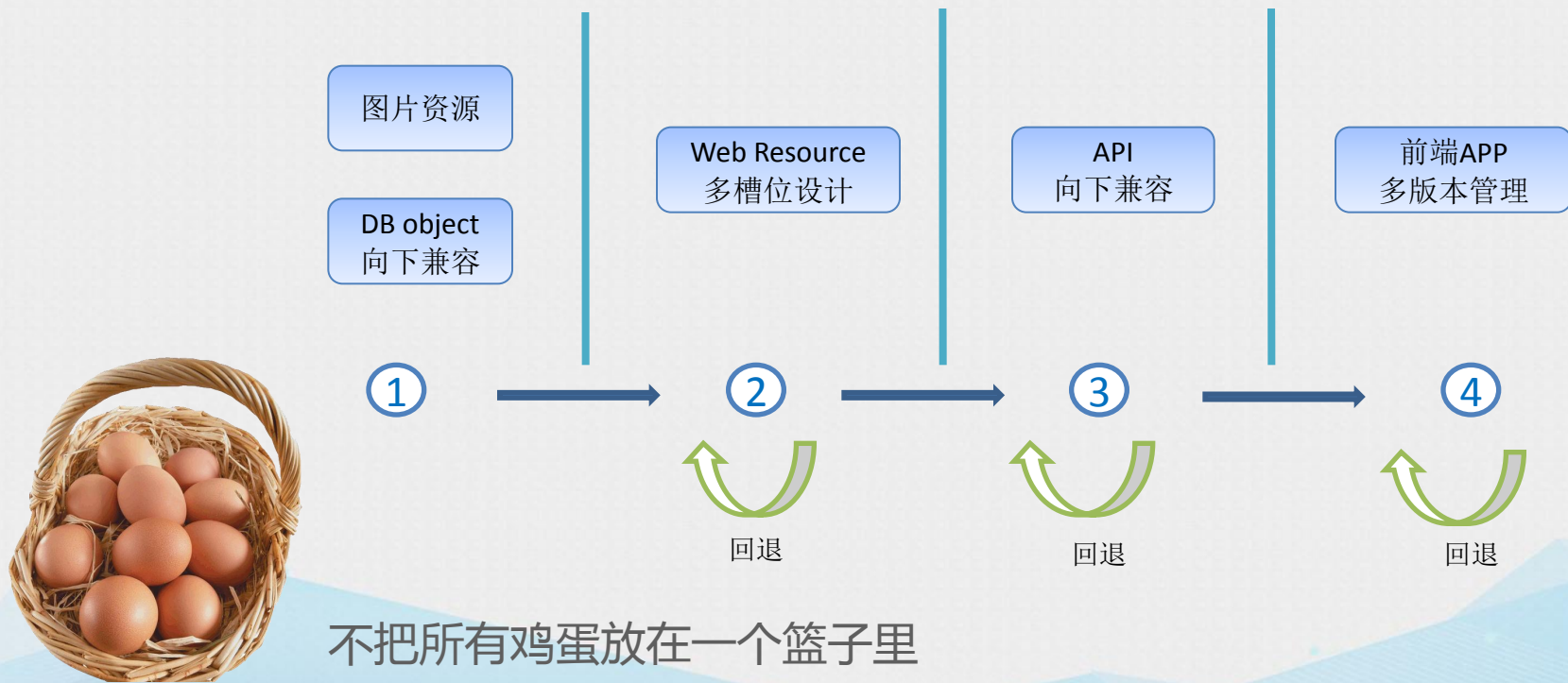
### 总结

- 打破思想鸿沟，规划和透明资源
- 标准化环境配置，管理环境变量
- DEVOPS工具管理过程

#### ■ 迭代是产品的生命线，但质量必须被控制

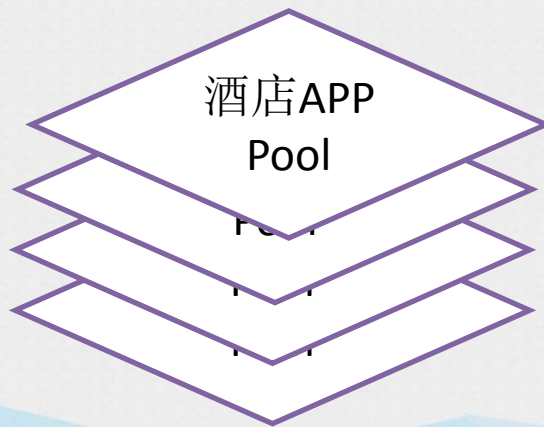
1. 合理资源发布顺序
2. Pool关系解耦
3. 生产预测试服务器群
4. 精细化Rolling过程
5. 完整的监控体系
6. 回退的指标基线

#### 合理资源发布和回退顺序



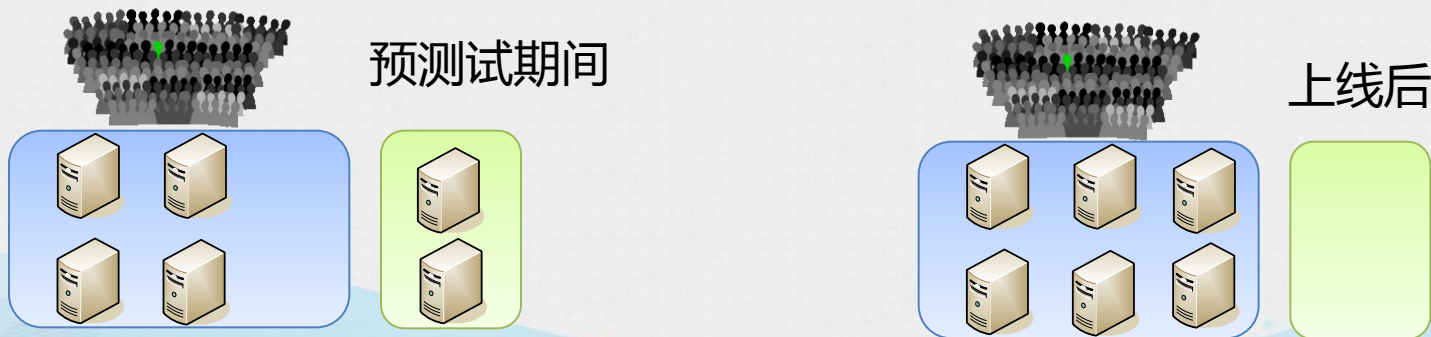
#### Pool关系解耦

- 规划Pool层面关系 ( SBU/API/APP )
- 构造稳定的Pool顺序
- Pool应用按机器同进同出



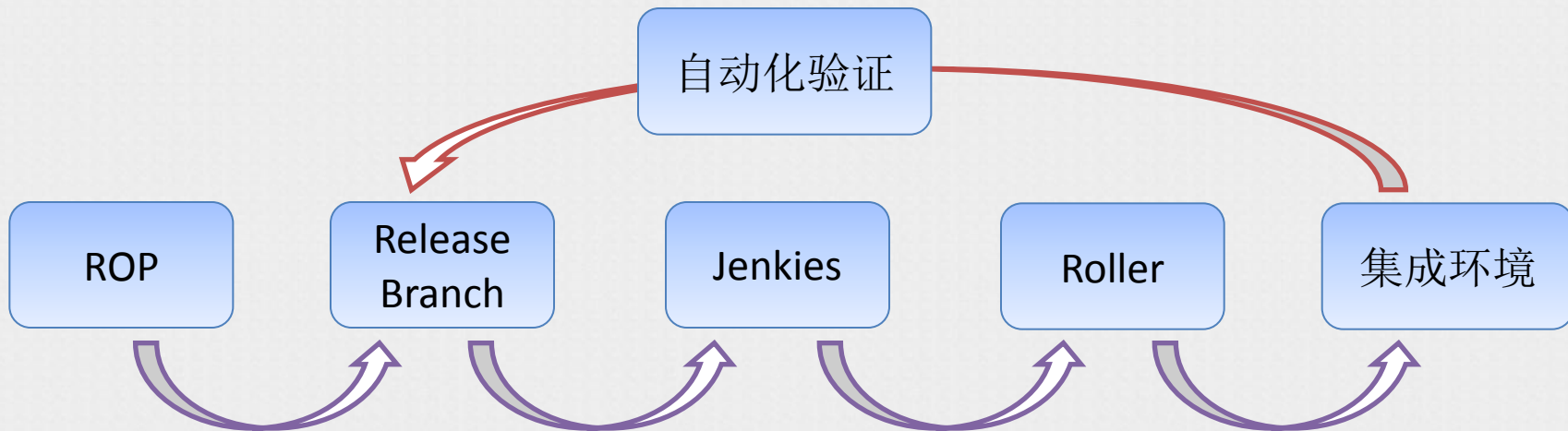
#### 生产预测试服务器群

- 代码提前更新到新版本
- 预测试服务器外网不能进入，内网可访问
- 尝试用预测试服务接入生产流量，并通过工具收集反馈





### 编译即发布



#### 精细化控制Rolling过程



## 完整的监控体系

1. 服务器性能指标
2. 容器异常指标
3. 应用健康指标
4. 数据库负载指标
5. ATP

集群内机器性能指标

总服务器数:12

健康:11

告警:1

异常:0

服务器级

应用级

Rolling面板

Rolling监控

回退历史

192.168.1.10

10.8.89.120

100 %

0 %

0 %

0

62

0

0

0

Get-IISLOG

10.8.9

10.8.89.119

100 %

0 %

0 %

0

1

0

0

0

Get-IISLOG

10.8.9

10.8.89.118

100 %

0 %

0 %

0

86

0

0

0

Get-IISLOG

10.8.9

名称: FlightIntl.Aggregator.Search.CR.20140218\_CR178369\_enginews.flight.Search\_2014/2/18 22:25:36

状态: Completed

查看详细指标

10.8.9

IP

正常

500

400

500

200

400

300

10.8.74.140

100 %

0 %

0 %

0

26

0

0

0

0

Get-IISLOG

10.8.9

10.8.89.117

100 %

0 %

0 %

0

18

0

0

0

Get-IISLOG

10.8.9

10.8.89.116

100 %

0 %

0 %

0

86

0

0

0

Get-IISLOG

10.8.9

10.8.89.115

100 %

0 %

0 %

0

62

0

0

0

Get-IISLOG

10.8.9

10.8.89.114

100 %

0 %

0 %

0

70

0

0

0

Get-IISLOG

192.168.1.10

名称: FlightIntl.Aggregator.Search.CR.20140218\_CR178369\_enginews.flight.Search\_2014/2/18 22:21:20

状态: Completed

查看详细指标

10.8.74.140

IP

正常

500

400

500

200

400

300

10.8.74.140

100 %

0 %

0 %

0

8

0

0

0

0

Get-IISLOG

10.8.89.113

100 %

0 %

0 %

0

12

0

0

0

0

Get-IISLOG

10.8.89.112

100 %

0 %

0 %

0

91

0

0

0

0

Get-IISLOG

10.8.89.111

100 %

0 %

0 %

0

61

0

0

0

0

Get-IISLOG

10.8.89.110

100 %

0 %

0 %

0

73

0

0

0

0

Get-IISLOG

#### ■ 回退指标

1. 可量化指标：容器异常指标、服务器性能指标、应用健康指标
2. 触发硬指标立即触发回退
3. 软指标触发预警和停止动作
4. 紧急回退和常规回退

#### ■ 总结

1. 发布质量需要DEVOPS过程来保证
2. 发布质量需要指标来评估
3. 回退不是失败，而是避免更大的错误

### ■ 名词说明

AR : Auto Roller 自动化配置工具

ROP : Roll Out Plan 开发团队发布的配置工具

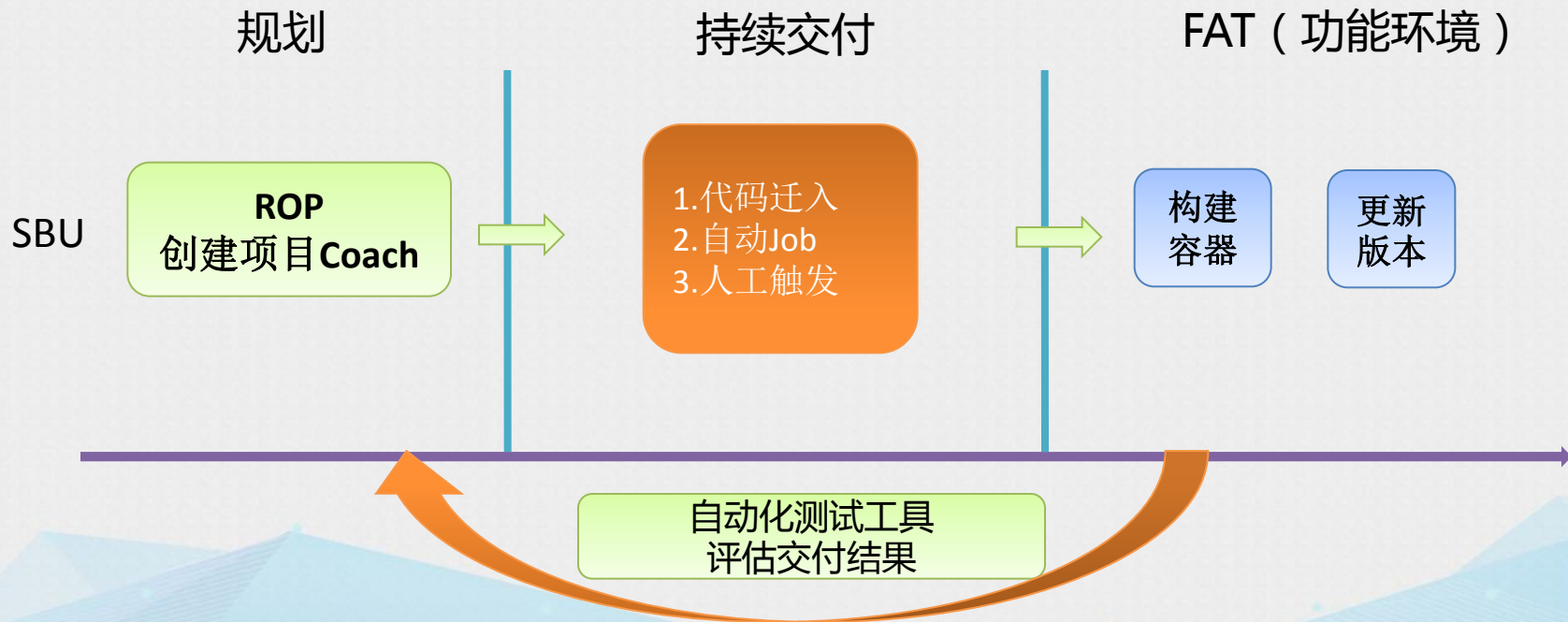
Roller Monitor : 跟踪发布过程 , 收集各项指标 , 评估发布质量

### ■ Train和Coach :

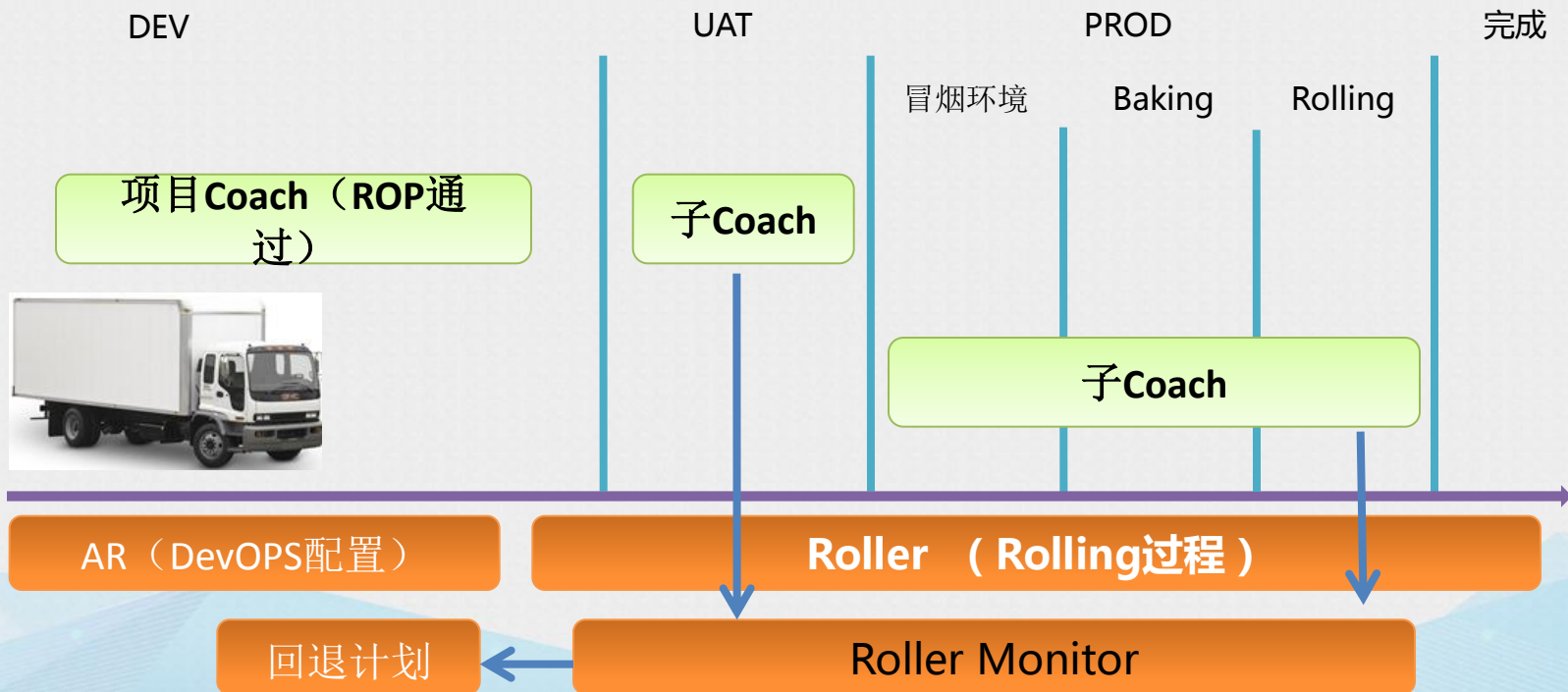
1. 项目Coach : 由1个或多个应用组成
2. Pool Coach : 指部署同一服务器一组的应用集合 ( Pool )
3. 多个Coach组成一个Train
4. SBU管理所属的Train , 小迭代Coach , 大迭代Train



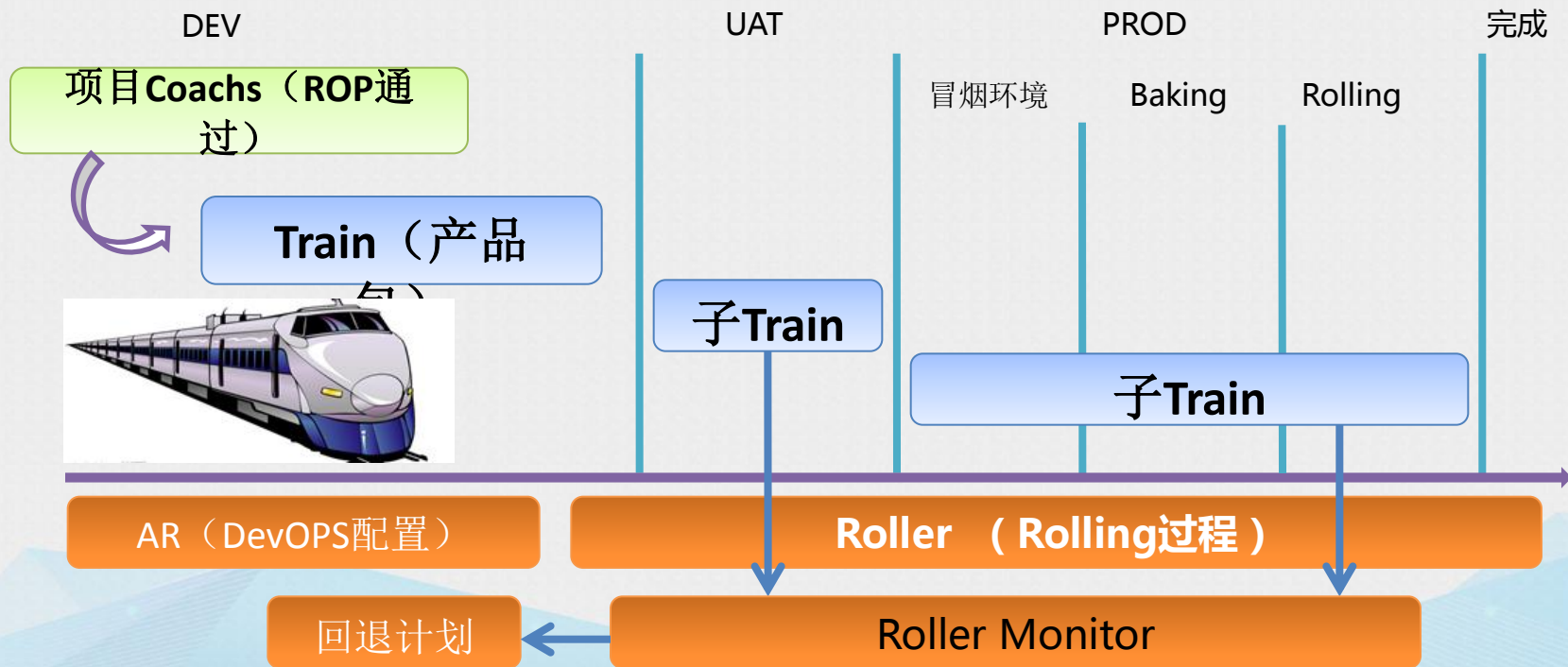
### ■ 测试环境发布：按项目交付，持续集成



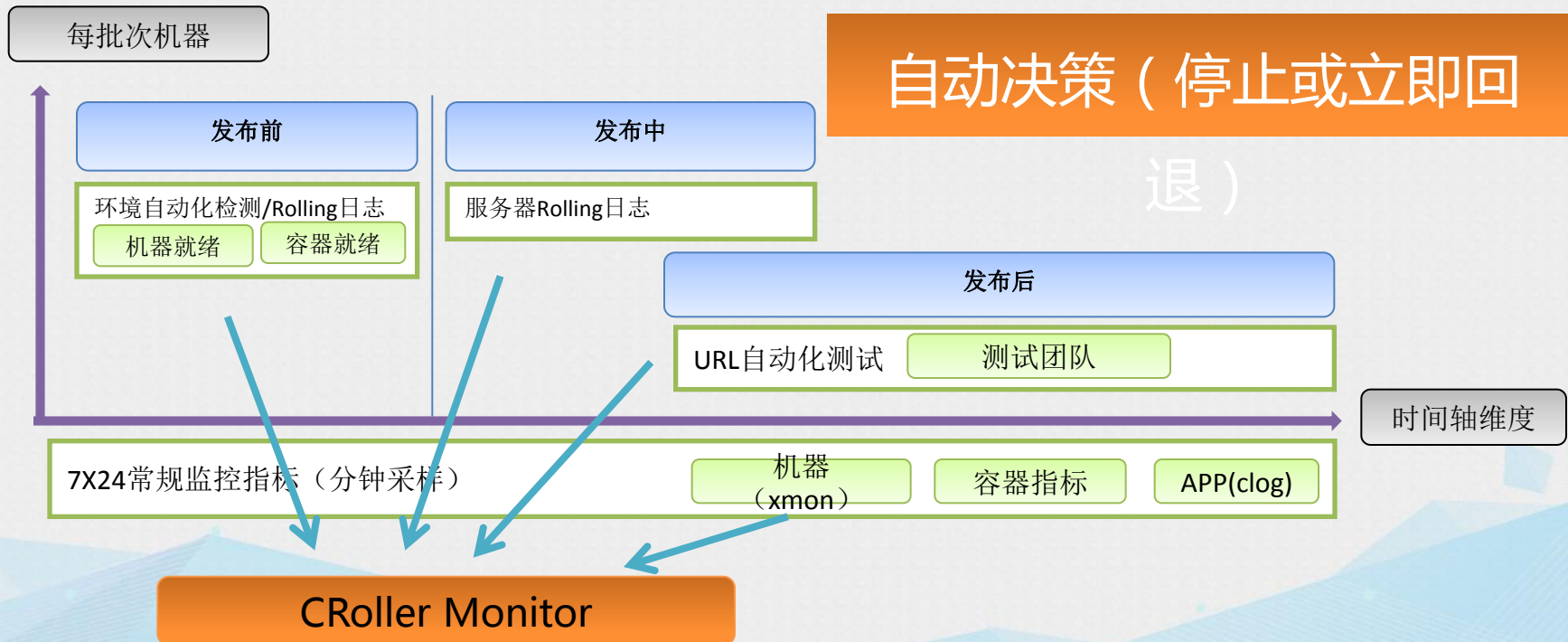
### 生产紧急发布



### ■ 基于Pool集群的生产发布，最大可能减少生产影响



### CRoller Monitor 自动监测每批次交付版本质量



### 可靠的回退

- 工具自动回退（可量化指标和基线）
- 不确定问题（先暂停，再决定继续或回退）
- 手工回退（紧急和常规）

选择回滚方案

**紧急回滚：**  
1.需要加载缓存的应用，请慎重使用，此操作会导致整个集群重启；  
2.快速回退会导致某些DB或SOA服务负载短期飙升；  
3.对所有参与Rolling服务器，每批次不做延时处理，回退速度快，风险高

☐ 我已明确回滚风险

紧急回滚

**普通回滚：**  
对所有参与Rolling服务器，默认每批4分钟切换处理(拉入、拉出各2分钟)，回退稳定可靠

普通回滚

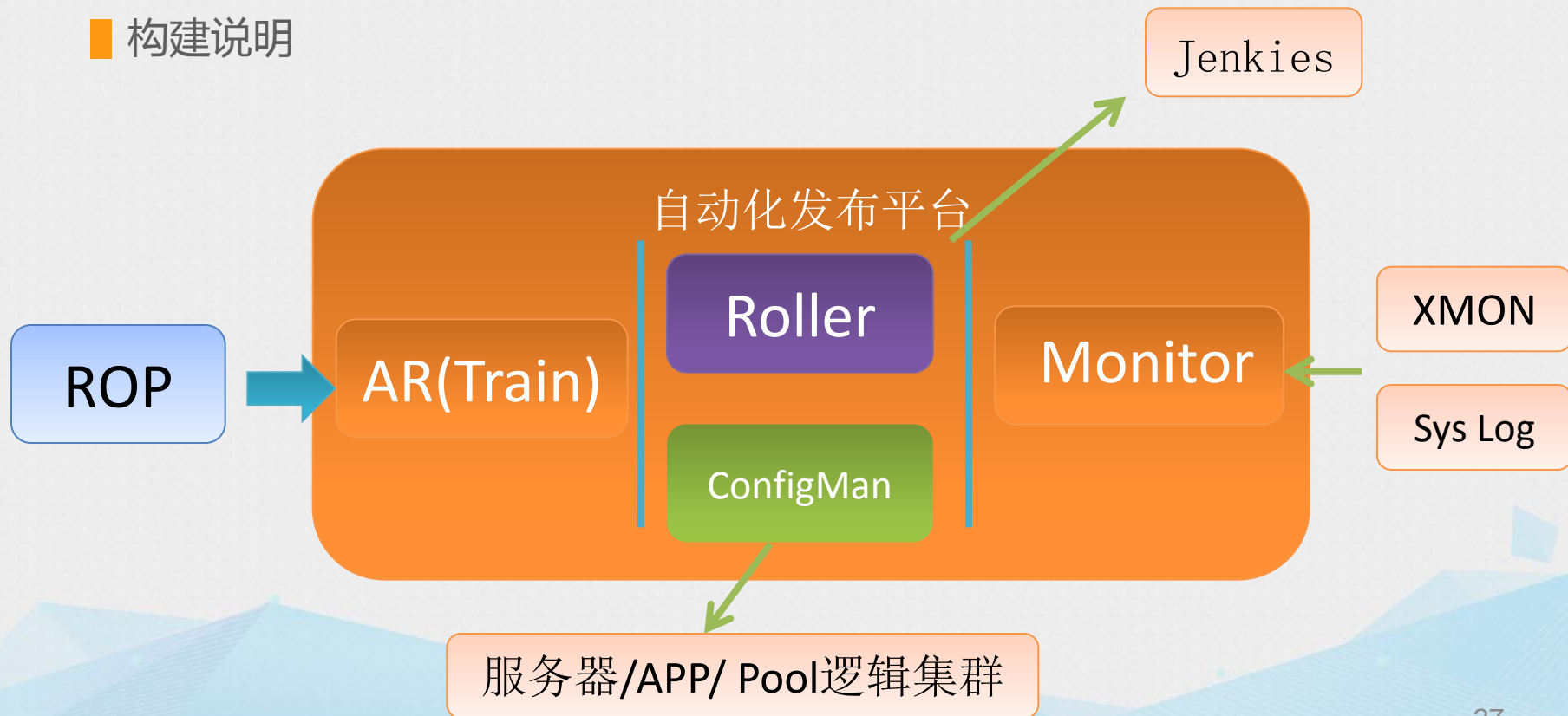
### ■ 总结

- **快            牺牲质量**

- 不打没准备的仗（提前规划计划和回退方案）
- 不把所有鸡蛋放在一个篮子
- 分解过程，改进体系，控制影响范围
- 量化指标，监控所有关键细节
- 工具胆大、心细，自动化控制过程



### ■ 构建说明





谢谢欣赏!  
2014. 04. 24