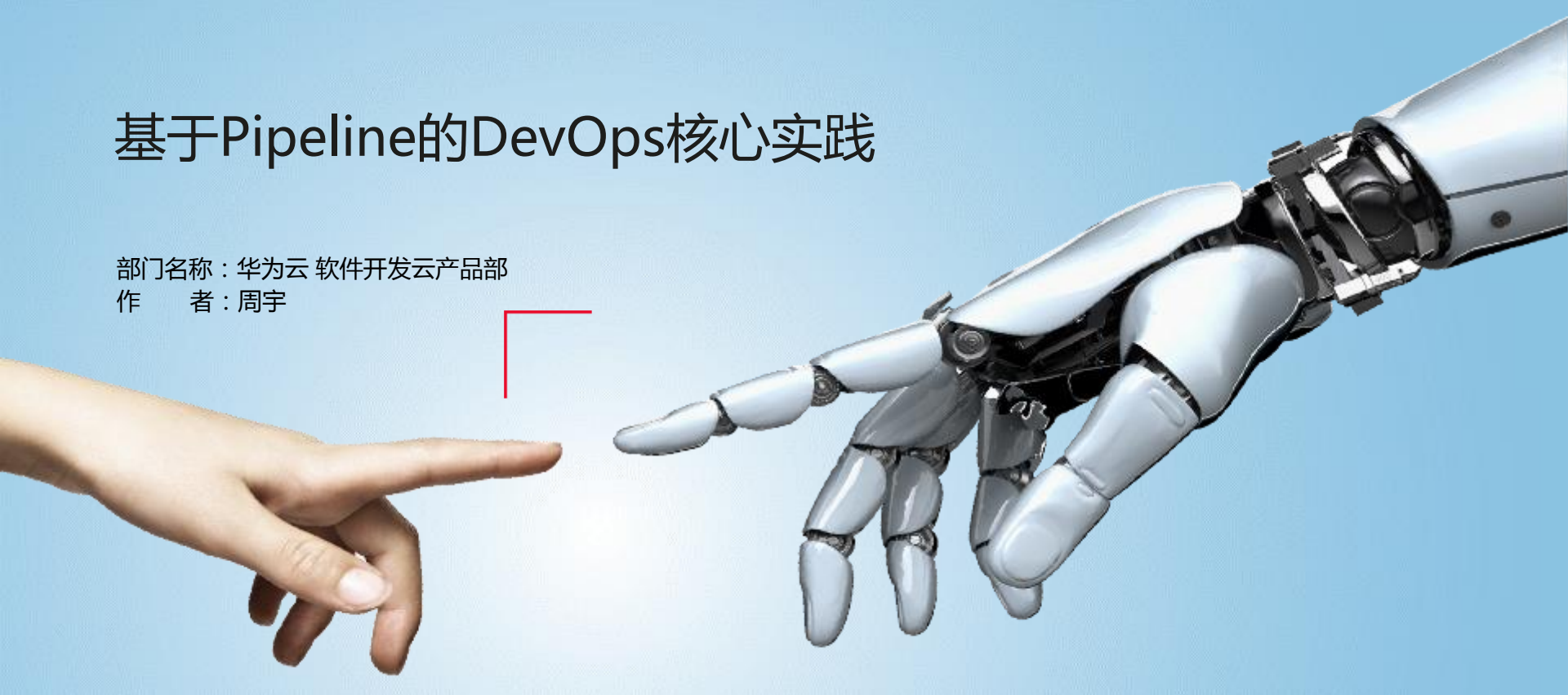


# 基于Pipeline的DevOps核心实践

部门名称：华为云 软件开发云产品部  
作者：周宇

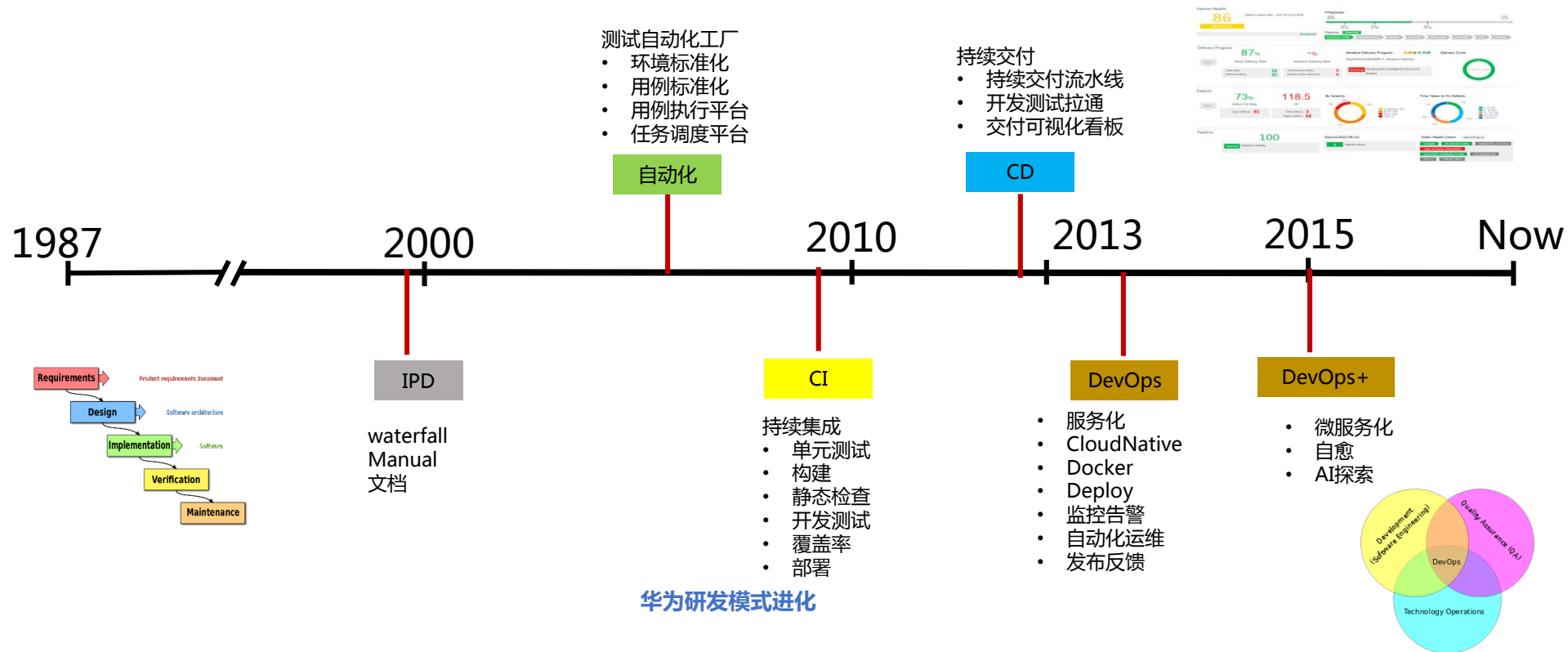


# 目录

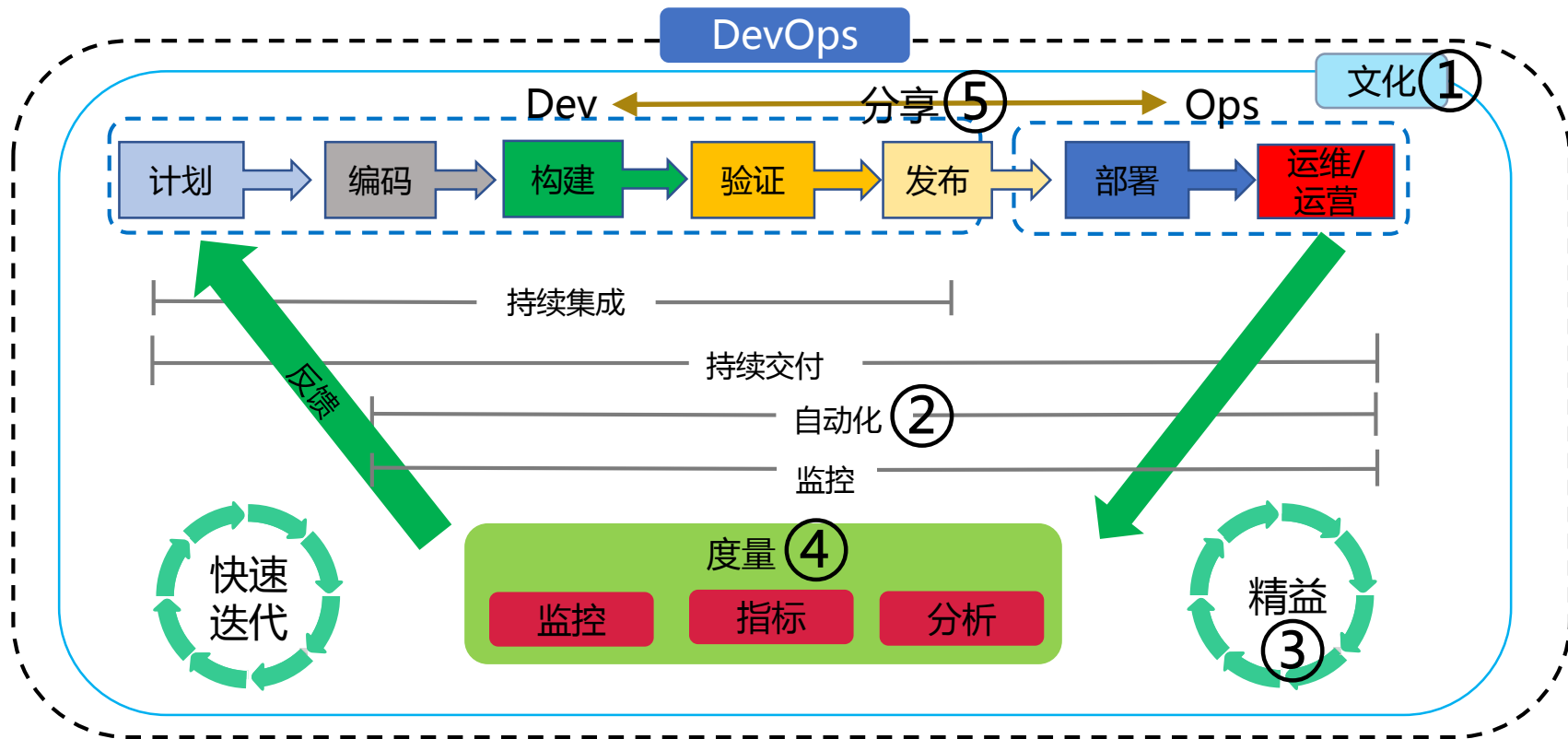
---

1. DevOps在华为
2. 不同研发模式下流水线的应用与思考
3. 流水线关键技术
4. 快速交付实战演练

# DevOps在华为



# DevOps CALMS全景图



# DevOps关键实践

DevOps实践的关键：团队协作 + 自动化能力 + 架构 + 基础设施

## 团队协作

全功能产品团队

领域特性团队

精益开发  
特性看板

构建、部署、验证  
自动化

灰度发布  
金丝雀发布

自动化持续监控

特性发布闭环

流水线

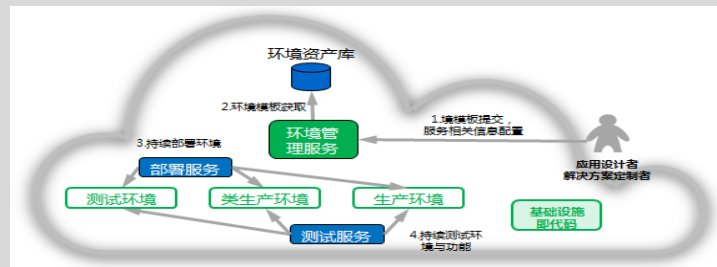
## 架构

服务化/微服务架构:



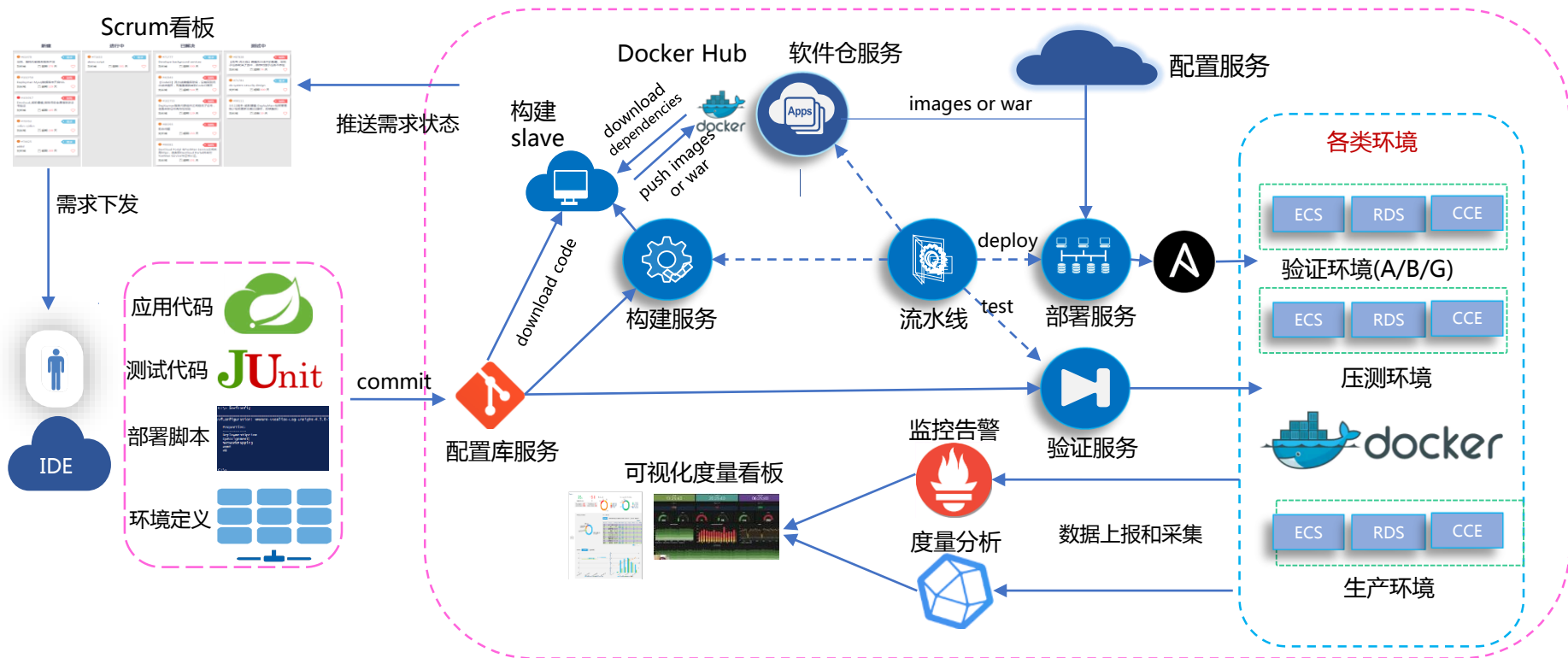
## 环境

基础设施即代码：



# DevOps工具链

**DevOps工具链**: 从需求下发、到代码提交与编译, 测试与验证到部署与运维, 打通软件交付的完整路径, 提供软件研发端到端支持



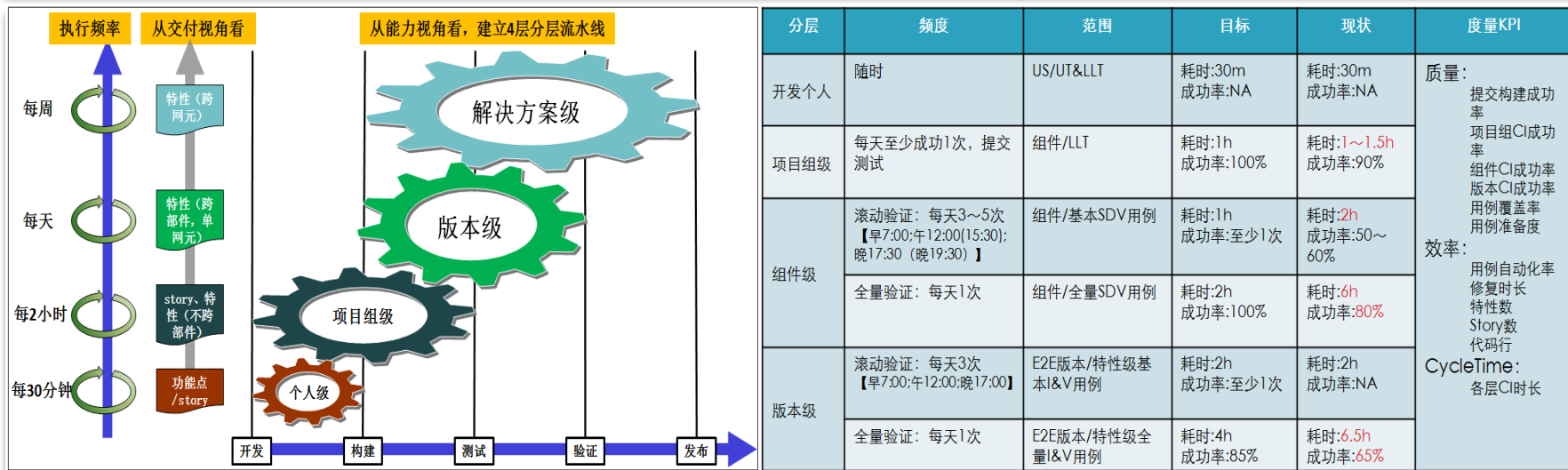
# 目录

---

1. DevOps在华为
2. 流水线支撑的研发模式
3. 流水线关键技术
4. 流水线演示

# 一、大规模开发：分层分级持续交付模式

通过分层分级降低软件交付复杂度；便于分层分级验收；便于分层设置交付周期，建立合理的交付节奏



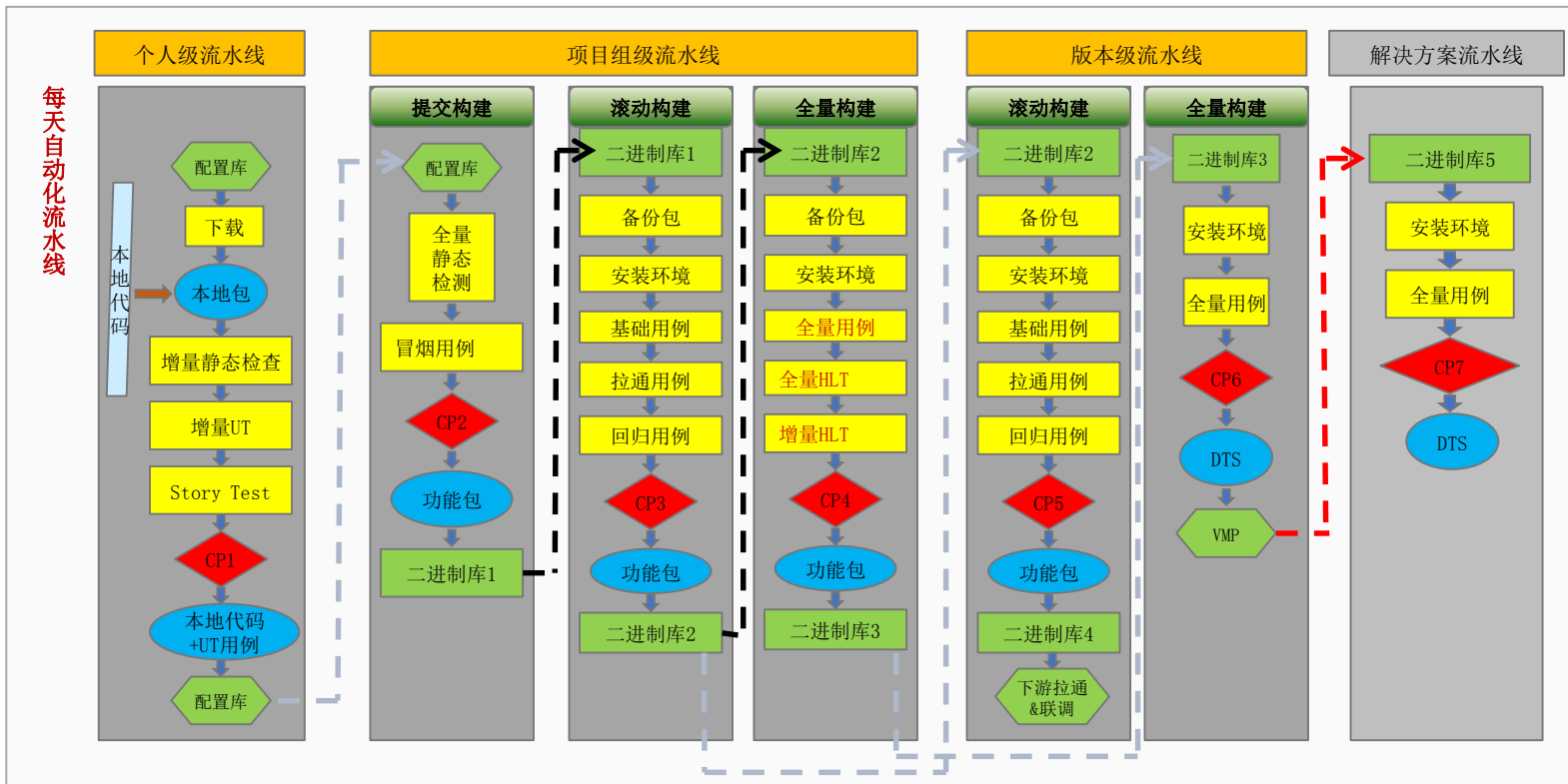
建立分层分级交付需要建立并完善开发与集成能力，包括：

- 1、技术上，设计解耦、开发与集成的依赖与顺序，交付界面比较清晰
- 2、组织上，开发、测试、I&V围绕分层交付构建协同关系
- 3、管理上，项目管理（关注交付顺序、耦合/依赖关系）与版本开发分支管理



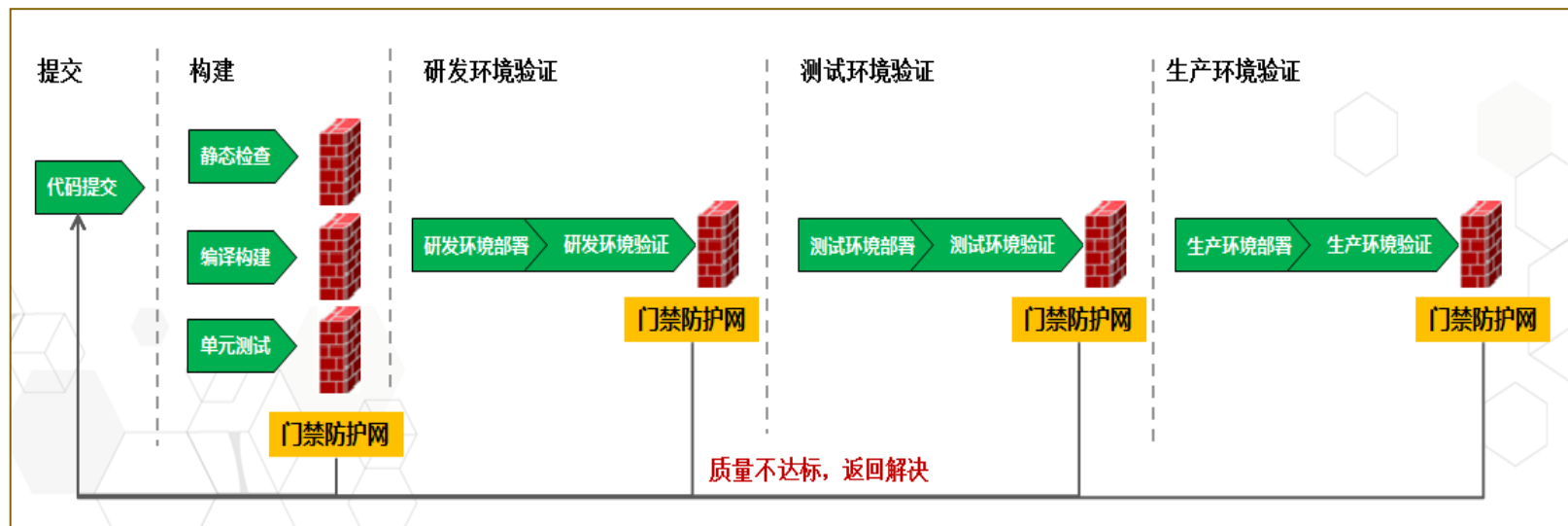
# 分层分级流水线应用

根据四层交付模型，建立代码到版本交付的自动化流水线，实现价值流高速流转



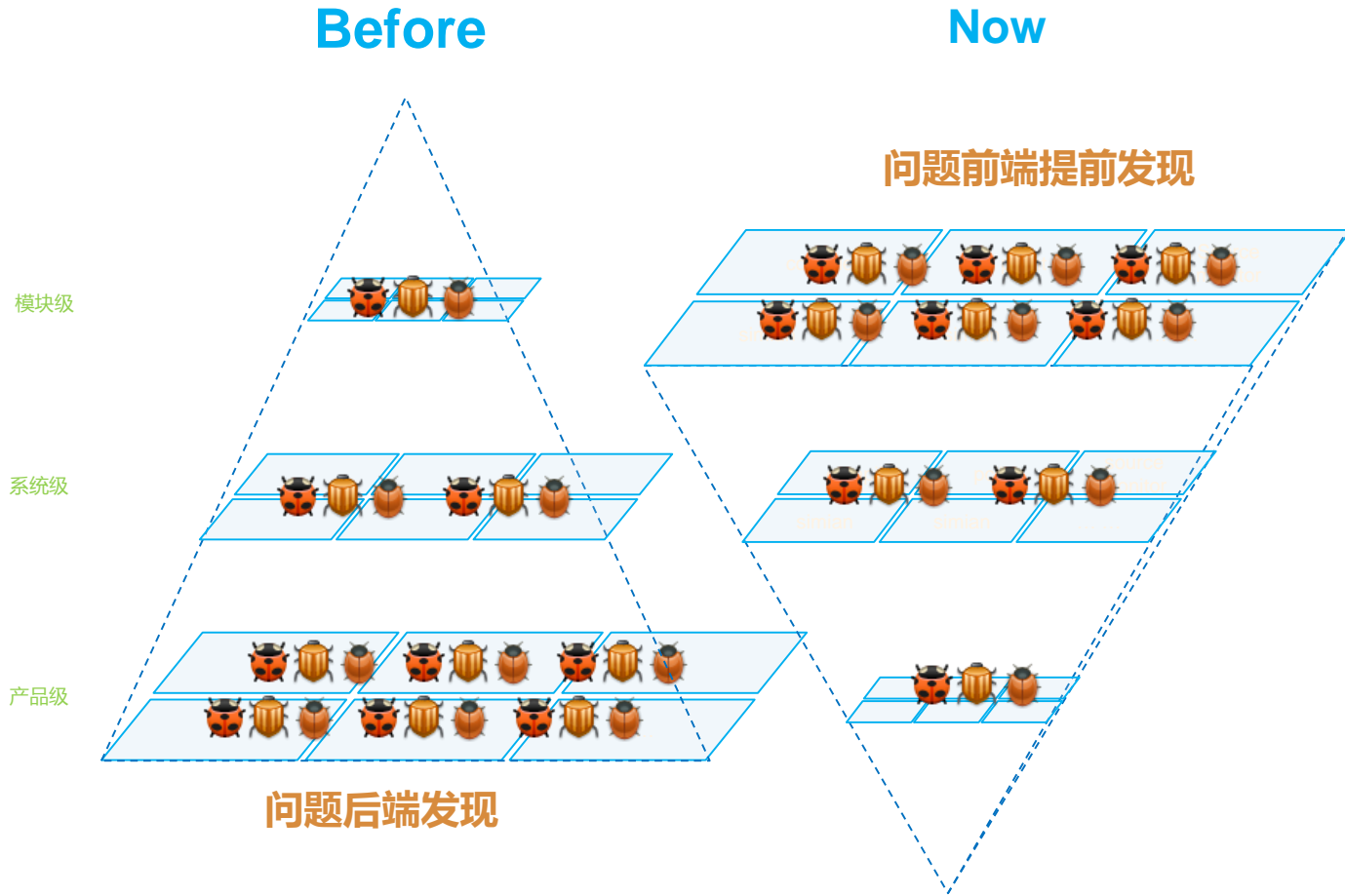
# 单流水线中分段门禁与质量看护

开发过程中问题不能及早发现，不断向后端积累，造成交付质量差，问题解决代价高



流水线执行过程中，在交付的各环境通过质量门禁自动发现问题，阻止不满足要求的交付继续向下游执行

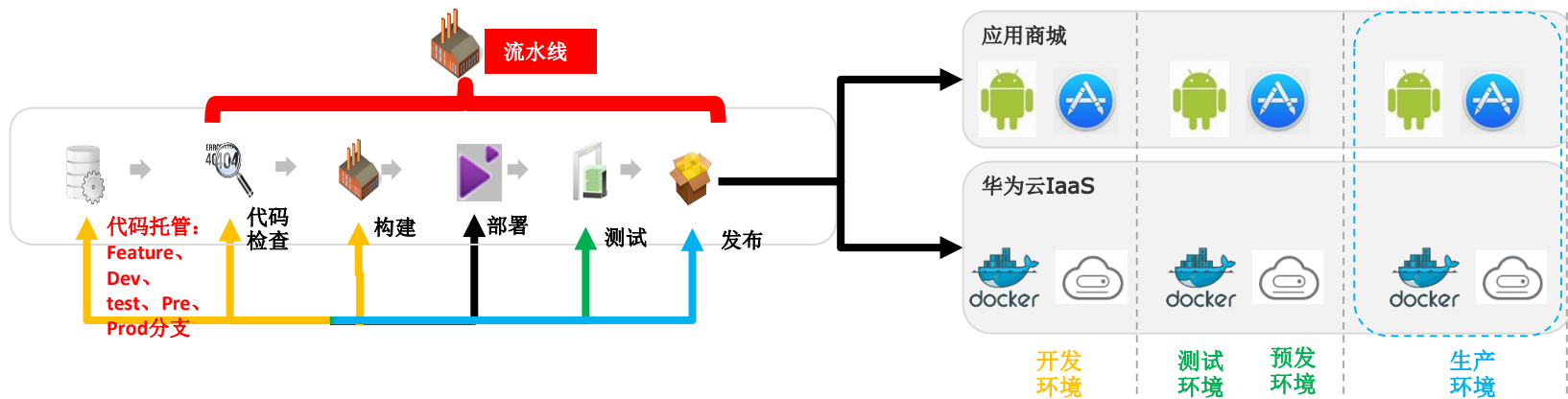
# 多级流水线层层防护，缺陷发现前移



# 分层分级持续交付模式痛点与思考

1. 流水线模板，帮助用户快速搭建第一条流水线
2. 流水线复制功能，帮助其他成员快速创建对应层级流水线
3. 流水线级联触发&调度，实现个人、项目、版本、解决方案级流水线逐级自动触发

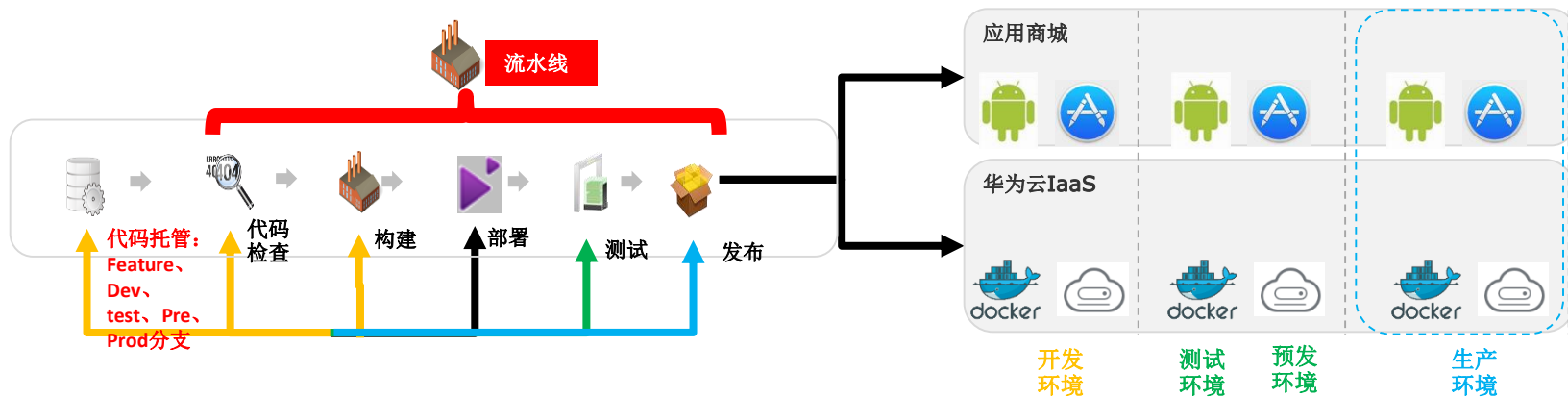
## 二、微服务开发：微服务独立发布上线



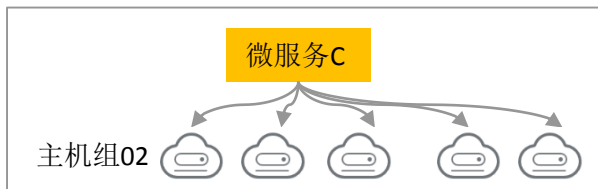
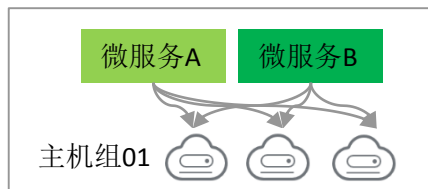
**开发人员：**作为全栈开发攻城狮，开发人员执行流水线实施CD持续交付，将master分支分别部署到开发、测试、预发、生产环境，并执行自动化测试等质量保障活动，确保产品质量。

- 当前：此处需要**产品经理或者版本经理进行人工审核，确认通过才发布。**
- 后续：流水线承载的质量保障手段，能够完全保障质量的情况下，取消人工审核，实施全自动化上线。

# 不中断业务升级：流水线实现灰度发布

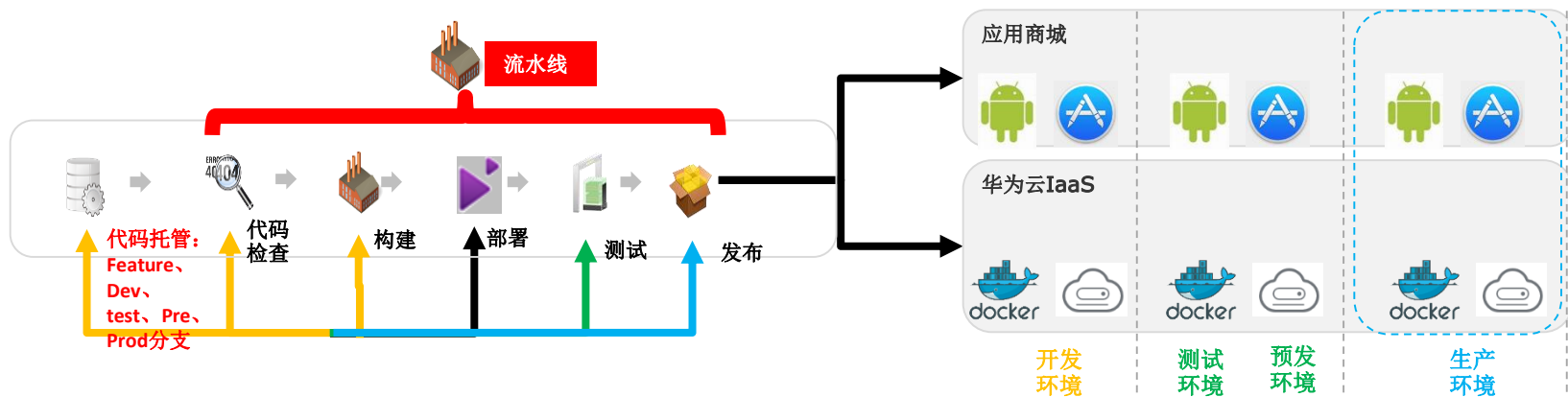


xx科技：1、微服务数量总计31个微服务；2、生产环境60+虚机；3、服务自动弹性扩容



.....

# 升级失败：流水线自动回滚



1、开发人员：执行流水线实施CI持续构建，将develop分支部署到开发环境进行联调。

2、测试人员：执行流水线实施CD持续交付，将master分支部署到功能、性能、可靠性、安全性等环境进行多轮测试。

A、运维人员不会重新构建，而是拿到软件包直接开始部署

3、运维人员：执行流水线实施CD持续交付，将测试人员确认质量合格的“软件包”部署到生产环境。

【注】：运维人员通常不重新构建包，而是从拿到通过测试的包直接进行部署！

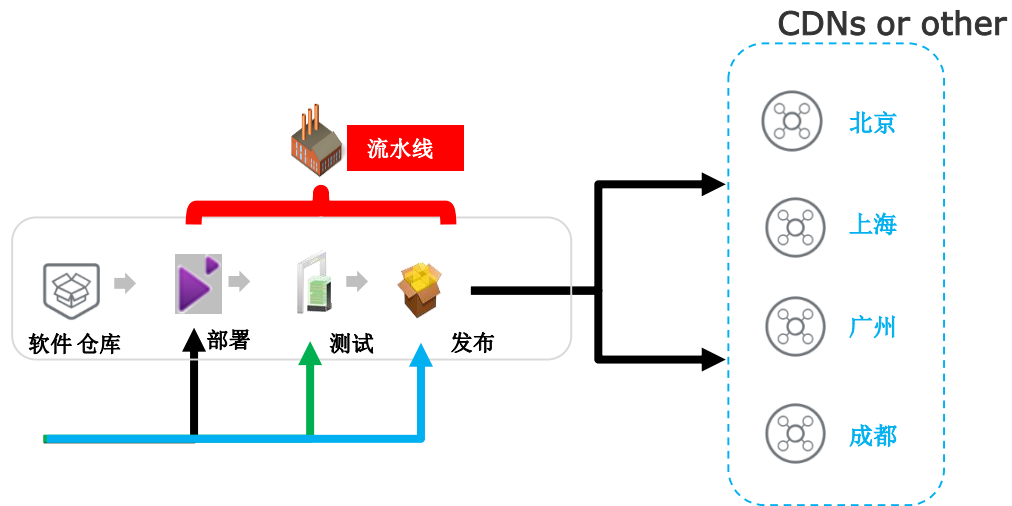
B、此处需产品经理或项目经理进行人工审核，确认通过才发布

# 微服务开发模式下对流水线的思考

1. 容器化部署
2. 微服务多版本标签，对软件包版本管理的强化(版本号自增长)
3. 微服务架构设计
4. 上线人工审核
5. 灰度发布
6. A/B测试
7. 重点关注在线测试比重加大



### 三、静态资源部署：部署流水线实现多region自动更新



## 部署流水线：

1. 针对静态资源软件包实施更新、部署
2. 多region部署支持

# 目录

---

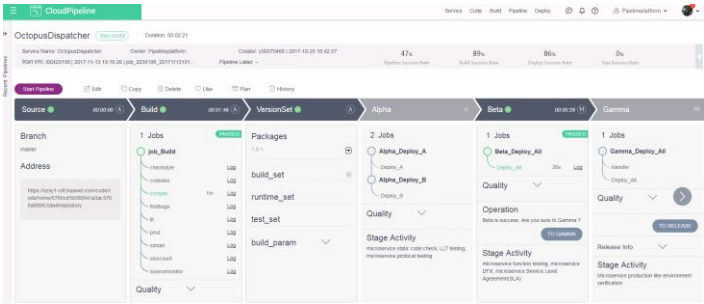
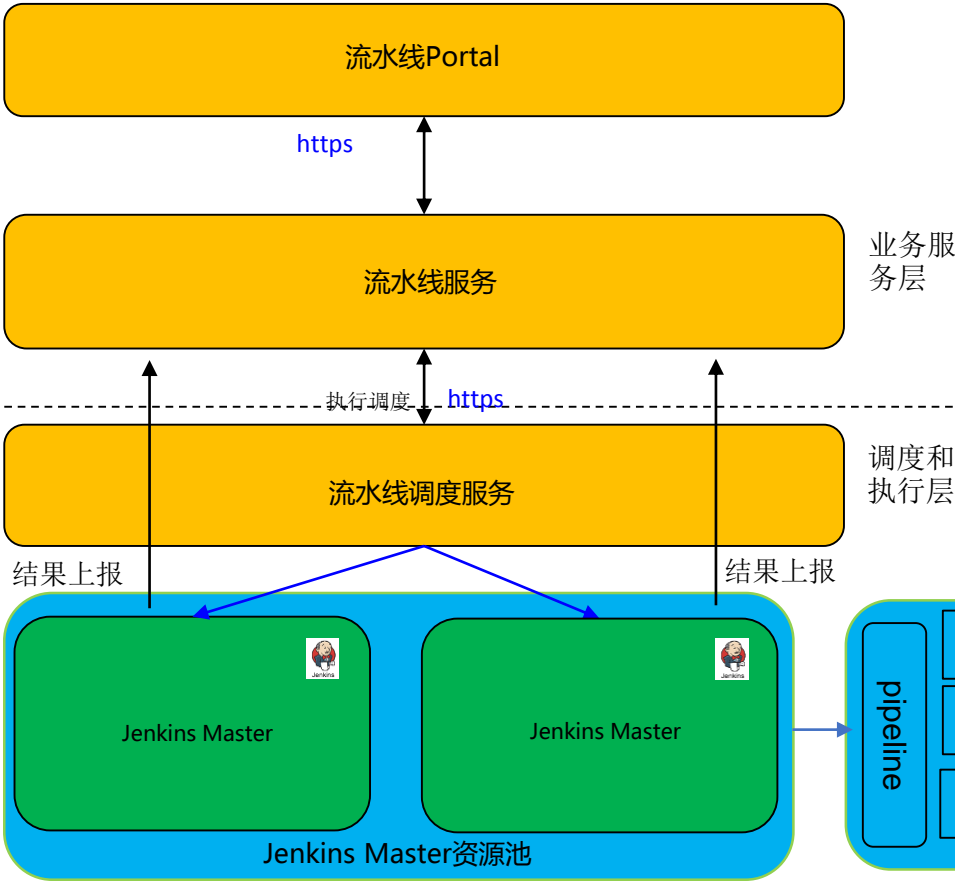
1. DevOps在华为
2. 流水线支撑的研发模式
3. 流水线关键技术
4. 流水线演示

# DevCloud Cloudpipeline流水线涉及的关键技术

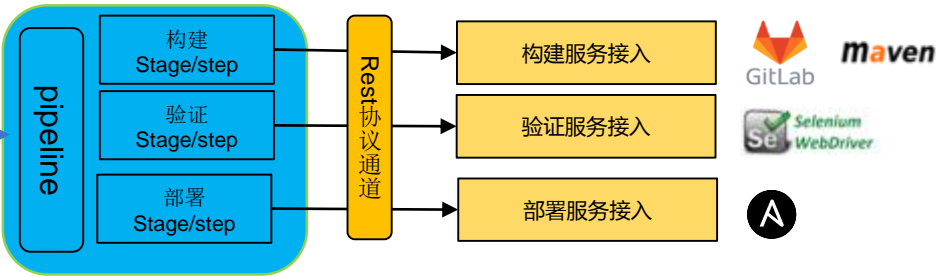
云化能为用户提供更稳定、更高效、更安全、低成本、随时可用的CI&CD服务

- 更稳定、随时可用
  - Master升级不中断服务
  - 100%服务随时可构建
  - 任务执行高可用
- 数据安全性更好
  - Jenkins Master数据云化存储，不丢失
  - 更好数据隔离策略，满足企业级数据控制要求
  - 与配置库、部署服务拉通的项目角色权限控制策略，更安全
- 执行效率更高
  - 并发支持能力更强
  - 执行效率更快
  - 构建加速支持
  - 任务分发更高效，平衡不同Master负载
  - Slave资源利用率更高
- 用户使用成本低
  - 配置简化，免维护
  - 提供标准化的构建环境，减少配置工作量
  - 提供配置模板，提升配置效率
  - 提供更多可集成云化服务，减少开发工作量

# 基于Jenkins的流水线高可用架构



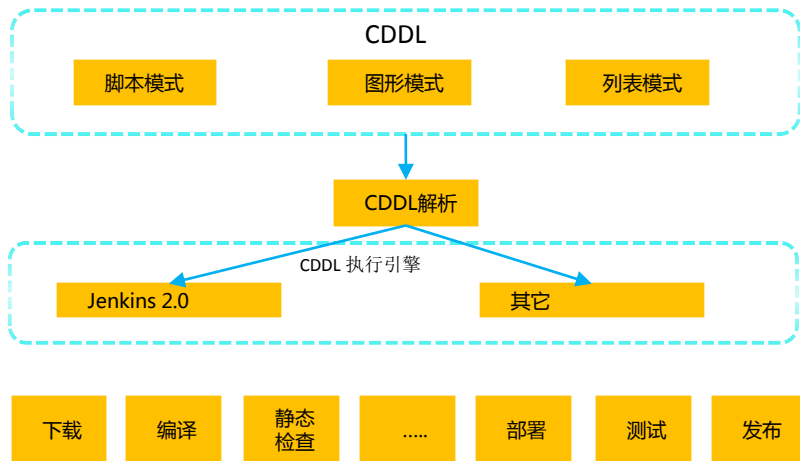
资源按需提供，任务按负载分配



# CDDL定义说明

**CDDL (Continious Deliver Domain Language)**，用来定义持续交付流水线中各过程及过程间依赖关系的流程描述语言。

- **简易性**：支持json和ymal，更易于理解，描述更简洁。
- **可扩展性**：描述性语言的特点方便用户随意扩展自己特殊的业务过程，便于用户定制，CDDL语言不仅仅被局限于持续交付流程，只要扩展相应的流程节点，即可被应用于各种业务流程场景。
- **平台无关性**：CDDL语言的解释器，执行器与语言本身没有强耦合，只要遵从一些基本的约束，流程可以在任何平台上运行。
- **灵活性**：CDDL的设计，参考了UML中的状态机模型，流水线设计比简单的串并行拥有更多的灵活控制



## CDDL流水线数据格式要素

```
graph TD
    flow["flow"] --> Anystandard["Anystandard"]
    Anystandard --> STend["STend : always"]
    flow --> DownloadCode["DownloadCode"]
    DownloadCode --> compilestart["compilestart : success"]
    flow --> states["states"]
    states --> sourcemonitor["sourcemonitor"]
    sourcemonitor --> actualParameters["actualParameters"]
    actualParameters --> canonicalPath["canonicalPath : /projects/Router/procedures/V800R007C00_Daily"]
    actualParameters --> cdplastSpendTime["cdplastSpendTime : 1102462"]
    actualParameters --> subprocedure["subprocedure : run_plugin"]
    actualParameters --> subproject["subproject : /plugins/ci_cloud/project"]
    actualParameters --> substartingState["substartingState :"]
    actualParameters --> subworkflowDefinition["subworkflowDefinition :"]
    actualParameters --> type["type : state"]
    actualParameters --> workflowDefinitionName["workflowDefinitionName : V800R006C20-pw-2-1-537c1f5e9cc0c"]
    flow --> workflow["workflow"]
    workflow --> ascription["ascription :"]
    workflow --> canonicalPath2["canonicalPath : /projects/Router/procedures/V800R007C00_Daily"]
    workflow --> cdpArea["cdpArea : PEK"]
```

States 节点描述  
Flow 节点关系  
Workflow 流水线基本信息

# 关键技术能力

- 百万级并发调度能力
- 一切即代码
- 一切即版本
- 一切即服务
- 一切即数据
- 一切自动化
- 一切可视化
- 一切智能化
- 任务分发
  - 多master均衡负载
- master对slave获取效率
- 任务执行ha
- Master升级的灰度
  - 任务不中断升级
- 用户数据安全隔离
- 构建模板
- 日志服务
- 监控中心
  - 问题定位
  - 瓶颈发现

# 目录

---

1. DevOps在华为
2. 流水线支撑的研发模式
3. 流水线关键技术
4. 流水线演示



# DevCloud on DevCloud微服务流水线实例

CloudPipeline

IntService\_tmp

Build

发布仓库

Alpha-dev-研发测试

Alpha-测试测试

Beta-自用环境

Gamma

发布

调用生产流水线

任务列表

构建IntService

静态检查UADP

静态检查FindBugs

代码检查CodeCheck

静态检查CodeDex

自动门禁

CodeCheck门禁

当前值 6

阈值 10

比较 <=

60%执行成功率

总执行次数: 10

失败次数: 4

更多

# 客户某产品应用实例

以前

交付周期  
3个月

构建  
2小时

部署  
人工为主

全量测试  
3+天

投入成本  
高达xx万

半夜升级  
业务中断

项目背景：

- 50+开发和测试工程师，分散在深圳、北京、成都
- 30+功能模块
- 代码量100+万行
- 交付周期3个月
- 产品质量保障偏手工



现在

交付周期  
1周

构建  
10分钟

云端部署  
提升10倍

线上问题  
减少50%

投入成本  
降低70%

白天升级  
业务连续

# 流水线持续交付能力演示

一次修改如何快速上线？

# Thank you.

把数字世界带入每个人、每个家庭、  
每个组织，构建万物互联的智能世界。

Bring digital to every person, home and  
organization for a fully connected,  
intelligent world.

**Copyright©2018 Huawei Technologies Co., Ltd.  
All Rights Reserved.**

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.

Huawei Confidential

