

百度日活过亿应用的DevOps实践分享

DevOpsDays Shanghai

2018 中国·上海 朱华亮 百度资深测试工程师



- 1 百度客户端DevOps实践挑战
- 2 百度客户端工程能力地图介绍
- 3 百度客户端产品大规模流水线构建系统实现与思考
- 4 日活过亿级应用--百度APP DevOps工程实践



百度客户端DevOps实践挑战



APP时代挑战



● App**迭代更新快**

群雄并起

内部状况



迭代速度

每月发版一次

需求：一个月后发版

灰度N+版本，一周，手动齐集核心数据，发版



发版前底气

版本质量保障

万行级代码变更，

发版时不知用例是否覆盖全面



用户体验

极致体验

用户体验如何？

产品上线后用户第一时间的反馈是？



问题召回和闭环

反馈问题召回和诊断

如何快速召回问题、诊断出病因、
并修复上线



APP时代挑战



群雄并起

● 新App涌现快

内部状况



新业务出现快



业务消亡



场景复杂



百度客户端DevOps实践挑战

第一步：统一认识，让业务线在规则下可以一起玩

建立一套切实权威可执行的标准

01
理论

第二步：让业务线可以有场所，有工具玩

低成本接入、高效有效稳定的系统支持

02
系统

03
实践

第三步：典型玩家特殊定制，满足个性化需求

典型业务线重点支持，不断迭代系统

百度客户端工程能力地图介绍

需要解决三大问题：

- 1、如何解决客户端DevOps实践人员思路统一问题（如何定标准）
- 2、解决标准权威性、真实性问题（如何取数据）
- 3、解决标准与客观、弹性矛盾问题（如何计算）



百度客户端工程能力地图介绍-标准如何制定

流程

CI/CD委员会制定初版

提至技术委员会酝酿

公司范围内进行公示

抽样分数与实际分析

完备

有效

各流程环节该具备何种实践

各个实践效果达到多少

流程

研发流程环节是否完善

三位一体



CHINA
DEVOPS
DAYS

百度客户端工程能力地图介绍-标准如何制定

二维（举例）



百度客户端工程能力地图介绍-标准如何制定

测试用例管理

L0 : icode、itp或case代码管理，由工具平台返回
L1 : 具备case分级手段，由工具平台返回

稳定性及遍历测试

L0 : 有稳定性测试，activity覆盖大于5%
L1 : 高覆盖稳定性测试，activity覆盖大于20%

自动化回归测试

L0 : 自动化回归+ut测试，全量分支覆盖率大于10%
L1 : 自动化回归+ut测试，全量分支覆盖率大于20%

系统兼容性测试

L0 : 有兼容性测试环节，覆盖机型或版本 10%
L1 : 有兼容性测试环节，覆盖机型或版本80%

全量源码安全扫描

L0 : 高危漏洞解决占比100% (OC删除)

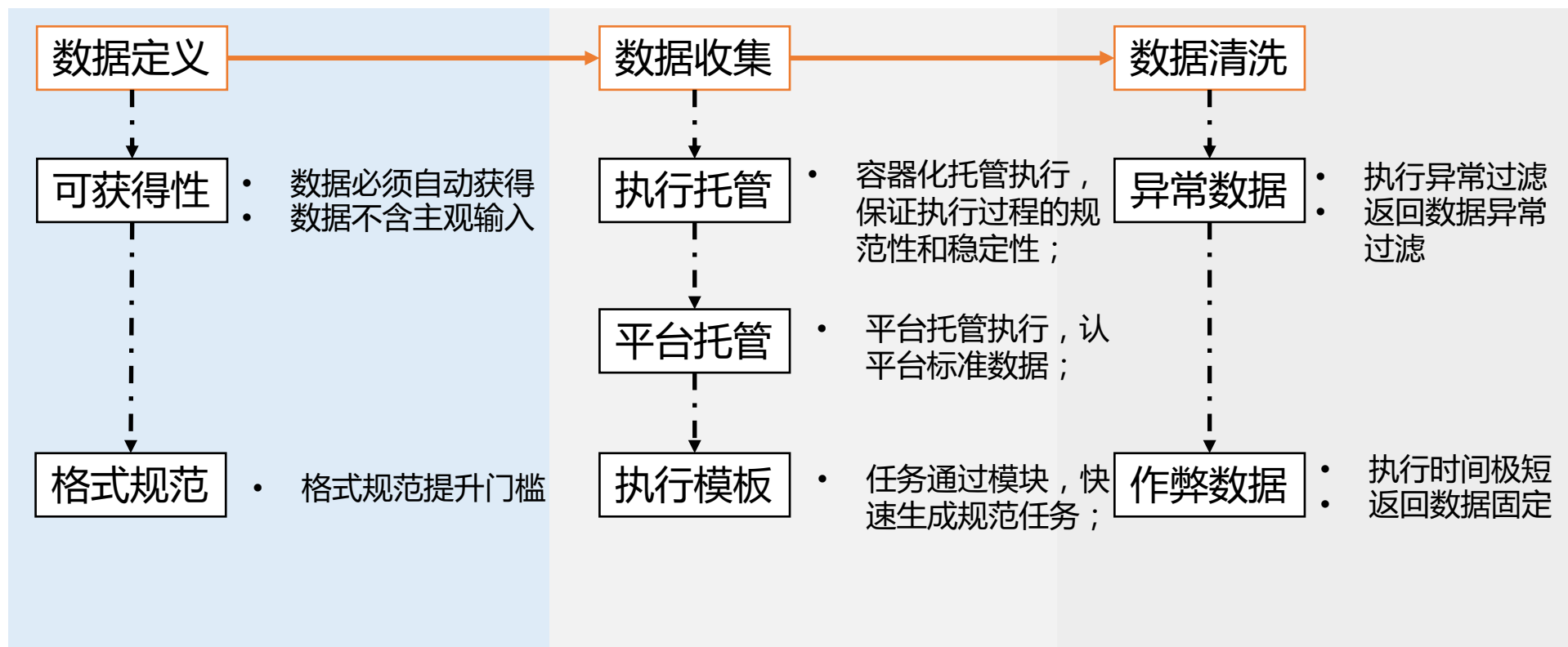
第三维 (举例)

测试



数据可收集、客观（在标准制定和构建系统里面呈现）：

- 1、标准制定时不含主观指标，如存在主观指标，认可的标准需要依据平台或者官方数据；
- 2、构建系统中制定标准的数据接口规范和公认平台；



百度客户端工程能力地图介绍-算法如何设计

算法：

$$T_i = \sum_{j=1}^n T_{ij} = C_i (\sum_{j=1}^n V_{ij} P_{ij} + \sum_{j=1}^n V_{ij} Q_{ij} D_{ij} - \sum_k^m V_{ik})$$

T_i ：代表第 i 个阶段的实际得分；

C_i ：代表第 i 个阶段分配到的总分；

V_{ij} ：代表第 i 个阶段第 j 个实践的权重；

P_{ij} ：代表第 i 个阶段第 j 个实践的完备性权重；

Q_{ij} ：代表第 i 个阶段第 j 个实践的有效性权重；

D_{ij} ：代表第 i 个阶段第 j 个实践的达到的标准层级/该实践的总级别数，比如该实践总共有 3 个级别，如果达到了第二个级别， $D_{ij}=2/3$ ；

备注： $P_{ij} + Q_{ij} = 1$ 备注： $\sum_{j=1}^M V_{ij} = 1$

例子：

| APP | 需求 | 开发 | 代码准入 | 测试 | 灰度验证 | 发布 | 流水线/自动化 |
|-----|----|----|------|----|------|----|---------|
| C= | 5 | 10 | 20 | 35 | 10 | 10 | 10 |

| 实践名称 | 分级标准 | | | 完备性 | 有效性 |
|------------|---------|------|-----------|------|------|
| | Average | Good | Excellent | 10 | 10 |
| 百度编码规范 | • | | | 1.25 | 0.63 |
| 增量源码安全扫描 | • | | | 1.25 | 0.63 |
| CodeReview | • | • | • | 1.25 | 1.88 |
| 增量静态代码扫描 | • | • | • | 1.25 | 1.88 |
| 云端编译 | • | • | | 1.25 | 1.25 |
| P0级自动化回归测试 | • | • | | 1.25 | 1.25 |
| Unit Test | • | • | | 1.25 | 1.25 |
| 可维护性检查 | • | • | | 1.25 | 1.25 |

特点：

- 1、权重可调整：各个阶段、各个实践、完备性和有效性的调权，不断迭代，达到最优；
- 2、实践项含必做项和选做项：达到依据业务需要弹性控制；

百度客户端工程能力地图介绍-产品呈现

产品具备冲击力和引导性 (**产品一角**):



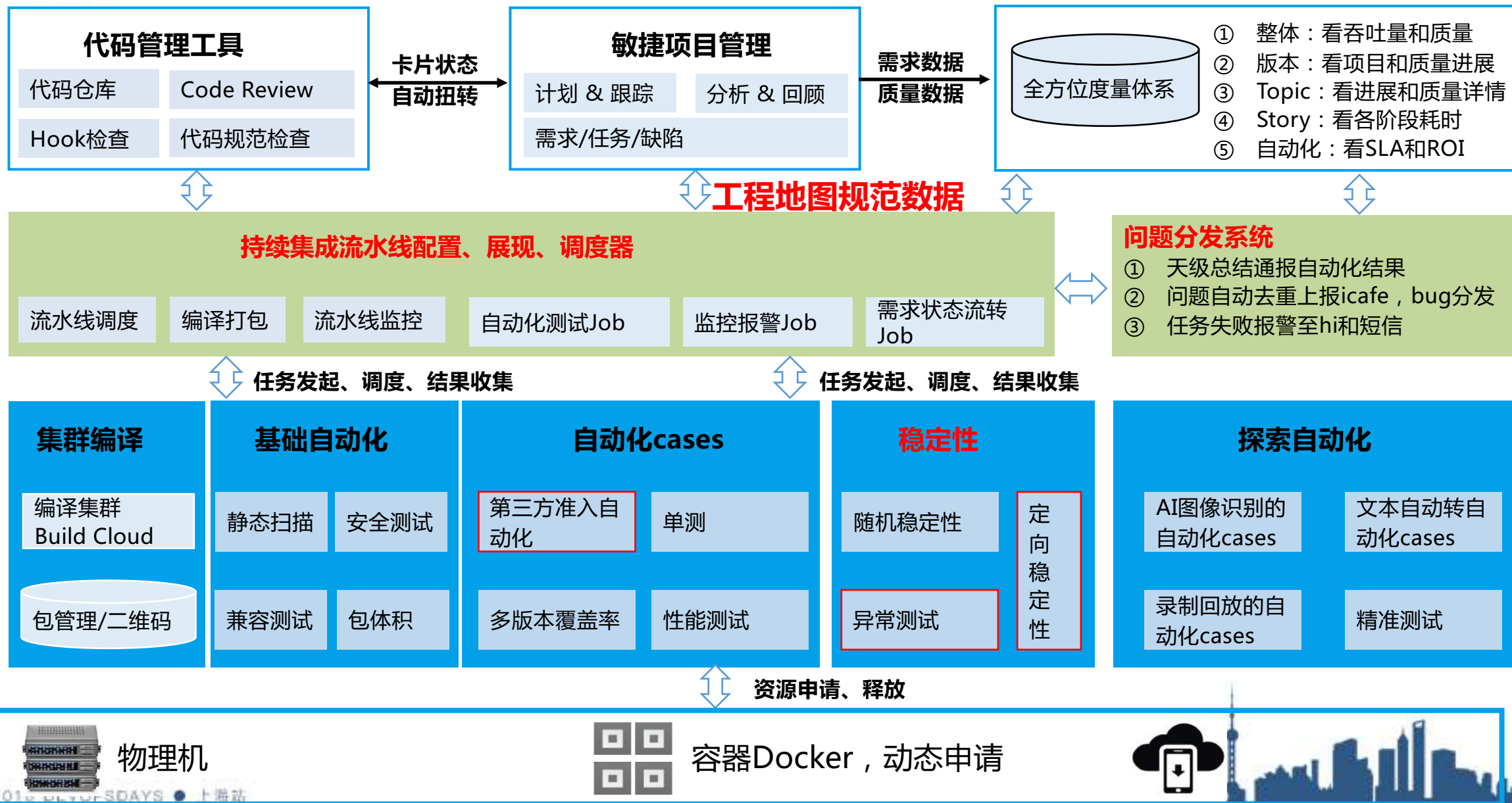
客户端产品大规模流水线构建系统实现与思考

解决三大问题：

- 1、设计快速可配置的流水线接入方案以满足客户端涌现
- 2、实现高效且稳定性的闭环构建系统以支持大规模构建
- 3、配合工程能力地图形成真实可信的研发数据链路系统

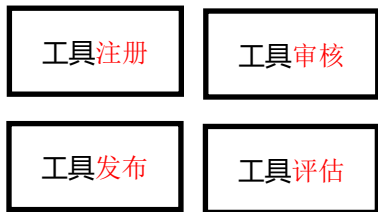


百度客户端产品构建系统整体架构

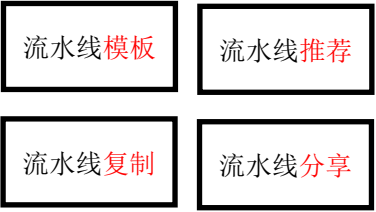


百度客户端产品构建系统整体架构——持续集成系统

工具插件化(面向开发者)



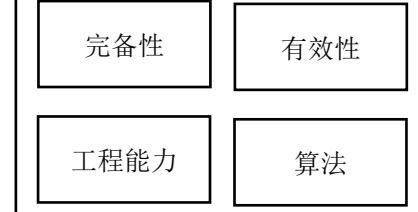
流水线管理(面向用户)



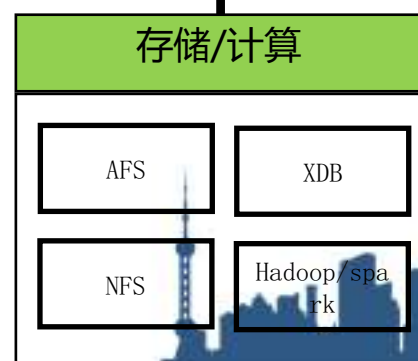
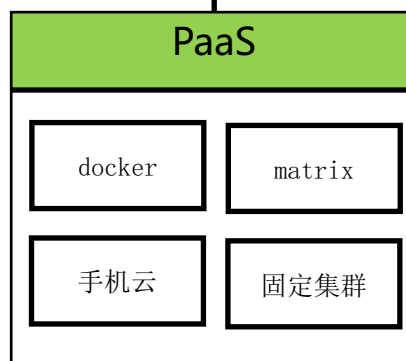
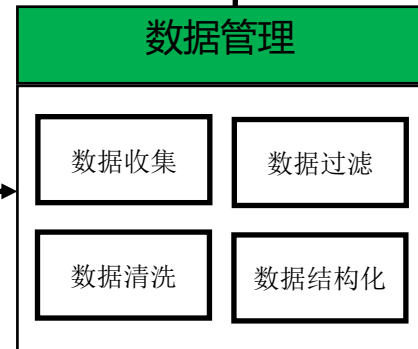
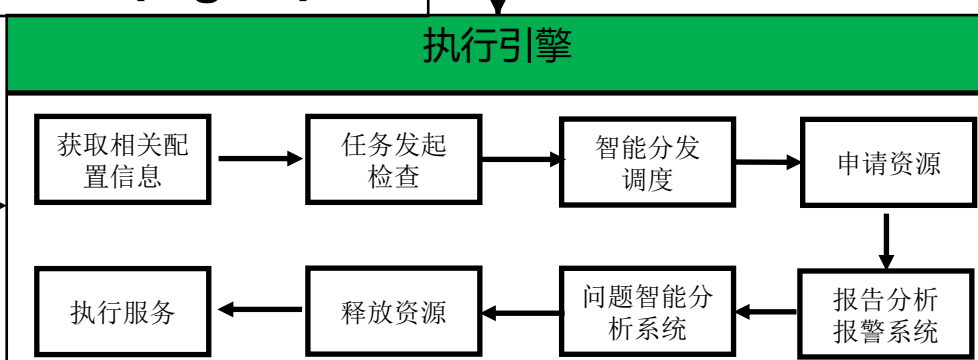
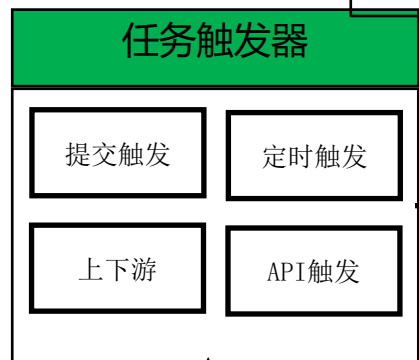
报告管理(面向用户)



度量体系(工程能力地图)



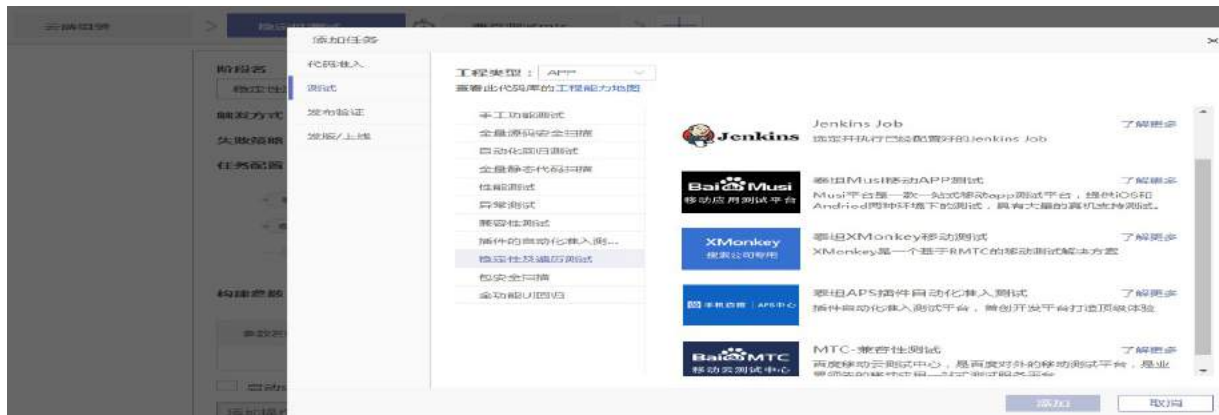
持续集成系统 (agile)



- 1、找：工具插件化；
- 2、用：流水线管理减少配置成本；
- 3、评：全方位的插件工具评估；

产品一角：

1、找：工具插件化。



2、用：流水线管理减少配置成本。



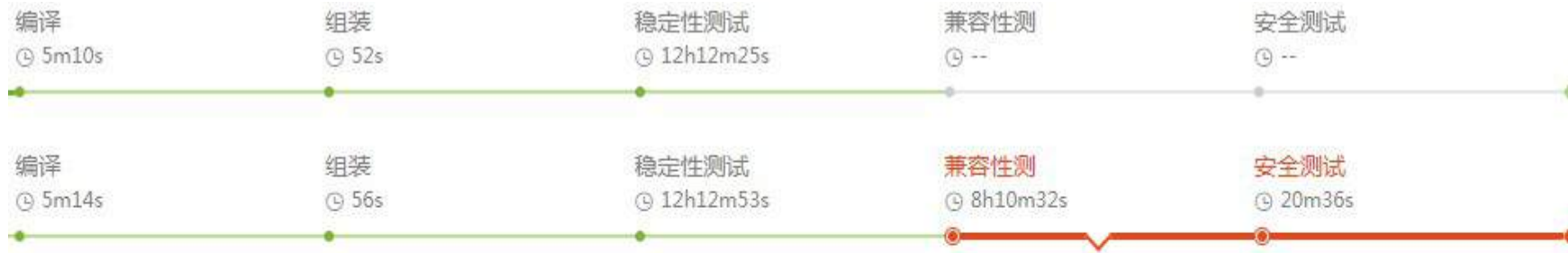
3、评：全方位的插件工具评估。



百度客户端产品构建系统整体架构——持续集成系统

用户视角：

- 1、自动化
- 2、准入&打包



| | | |
|--------------|--------|----------------------------------------------|
| ❗ MTC-兼容性测试 | 8h10m | MTC测试报告 运行成功, 100个任务中存在6个Crash |
| ❗ 泰坦Musi兼容性 | 17m23s | 测试报告 crash问题数: 1; anr问题数: 0 |
| ❗ Bugbye代码扫描 | 15m46s | 当前存在高危漏洞, 请修复后重新扫描! 高危: 46 (+46) 中危: 0 低危: 0 |

master主线打包流水线 > master

执行构建

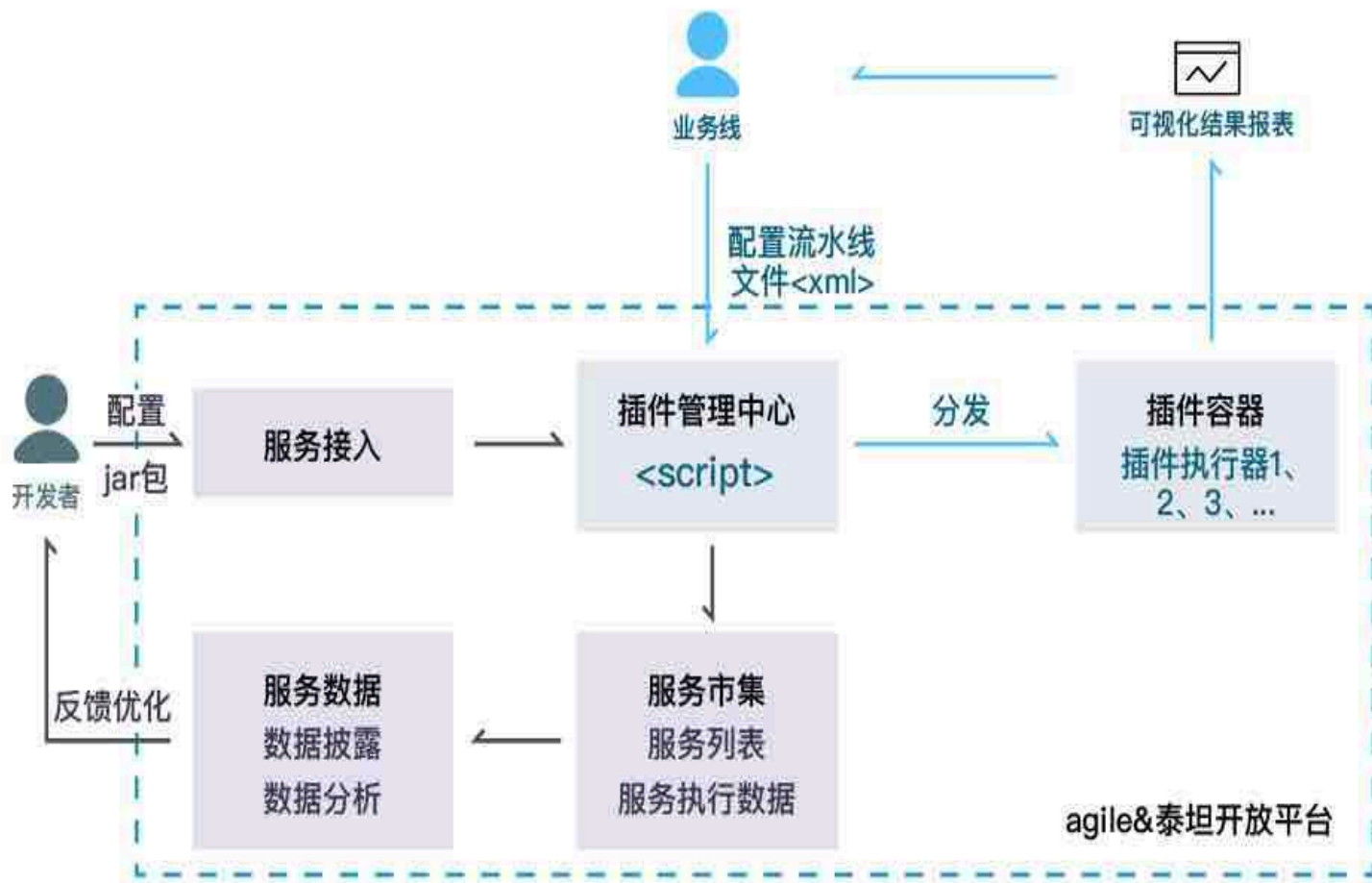
发布记录

+ 新建流水线

流水线配置



百度客户端产品构建系统整体架构—插件生态



- 1、开发者通过API或jar注册至平台；
- 2、平台进行审核（含数据）；
- 3、平台进行发布；
- 4、插件进入服务持续集成服务市场**推荐**；
- 5、收集服务在业务线表现数据；
- 6、对开发者工具进行评估（形成漏斗，**优胜劣汰**）；



百度客户端产品构建系统整体架构-高并发稳定性测试



Monkey高并发主要问题：

- ① **手机不足**：手机资源无法云化，执行时间长；
- ② **宿主机器不足**：传统的USB直连浪费台式机；
- ③ **进程间干扰**：无线ADB方式任务间干扰严重；



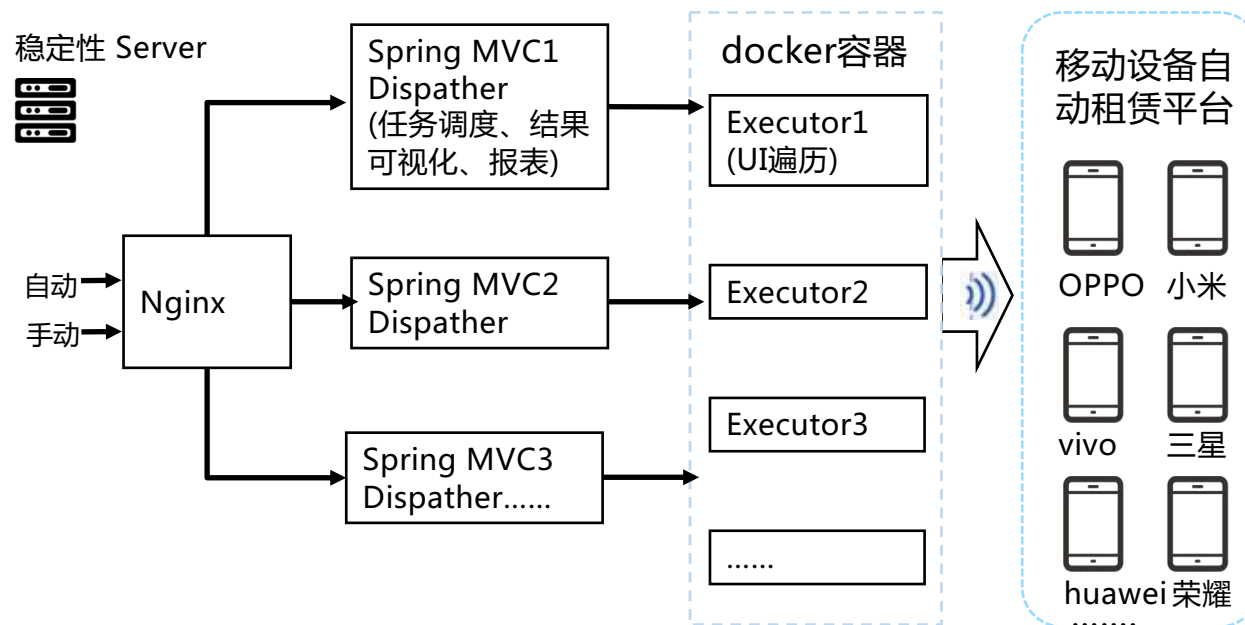
基于“容器”高并发系统：

- 1.Nginx实现热部署和负载均衡；
- 2.Spring MVC，提供Dispatcher能力，控制任务调度；任务结果展示；
- 3.容器化环境平台，根据需要动态分配执行资源；
- 4.Executor，稳定性测试能力；
- 5.真机自动租赁管理平台，提供移动真机能力，可提供任意指定厂商、机型、系统的手机；



实际效果：

- ① 百台级手机并发执行；
- ② 每次任务>2H，每天执行**千次**以上任务；





CHINA
DEVOPS
DAYS

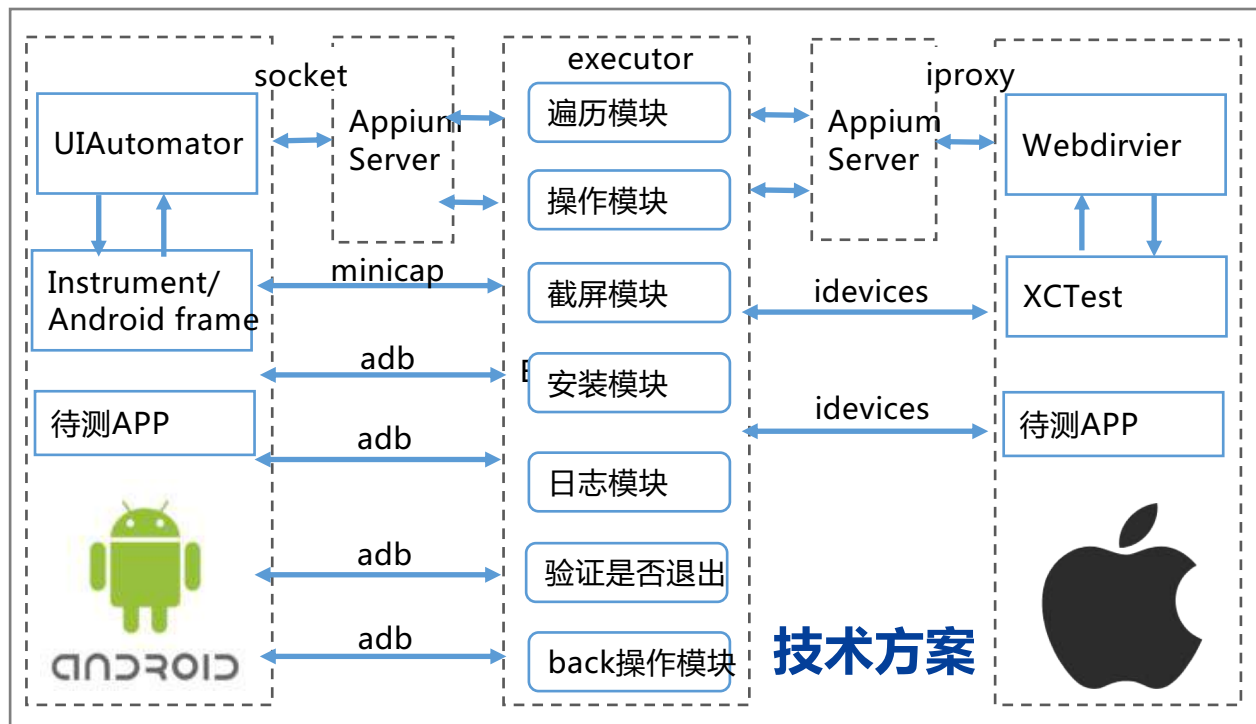
百度客户端产品构建系统整体架构-高覆盖稳定性测试

传统Monkey问题：

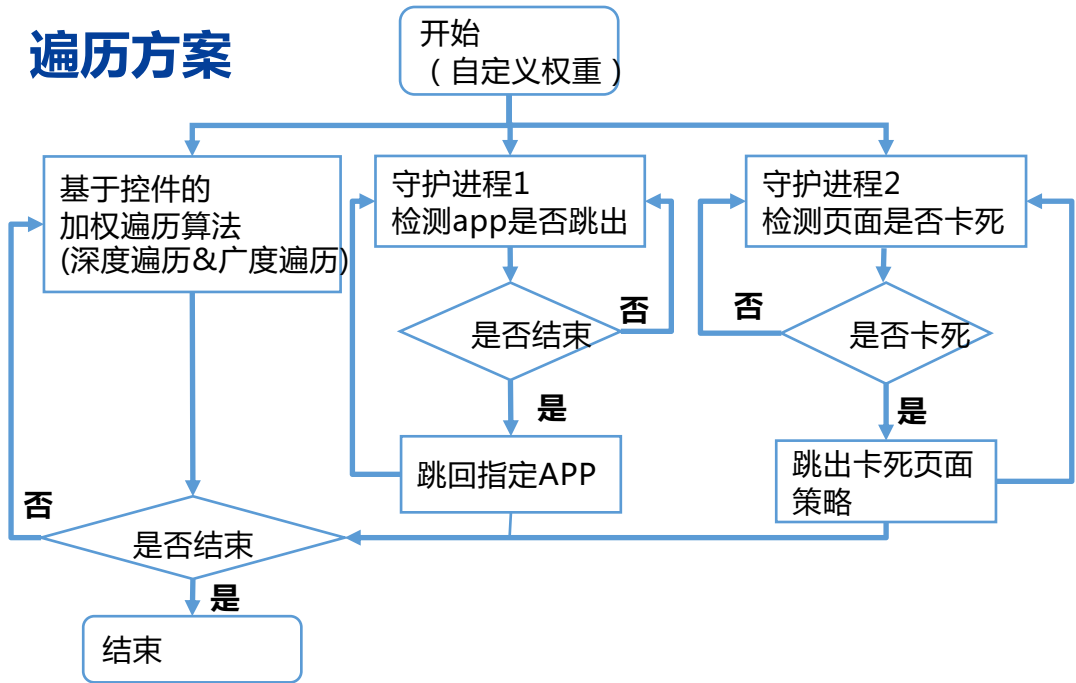
- ① **覆盖深度不足**：无法根据页面特点生成随机事件
- ② **重点或特定页面**：无法按需覆盖
- ③ **无法解决手机干扰**：如自动登录、跳出、引导页等
- ④ 随机Monkey的有效性如何保证

全新的遍历方案：

- ① 解决遍历覆盖问题：基于控件的加权遍历算法；
- ② 结合自动化框架编写插件定向、场景定性等自动化cases
- ③ **基于图像识别**的自动登录和引导页问题；解决干扰：如跳出、卡死等；
- ④ 多版本代码覆盖率的和activity覆盖率全方位衡量有效性



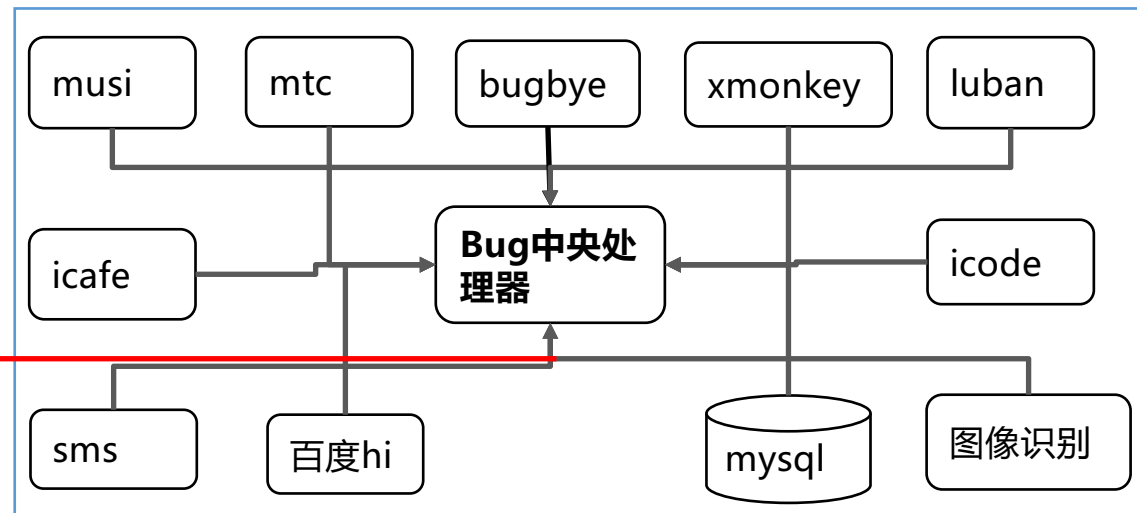
遍历方案



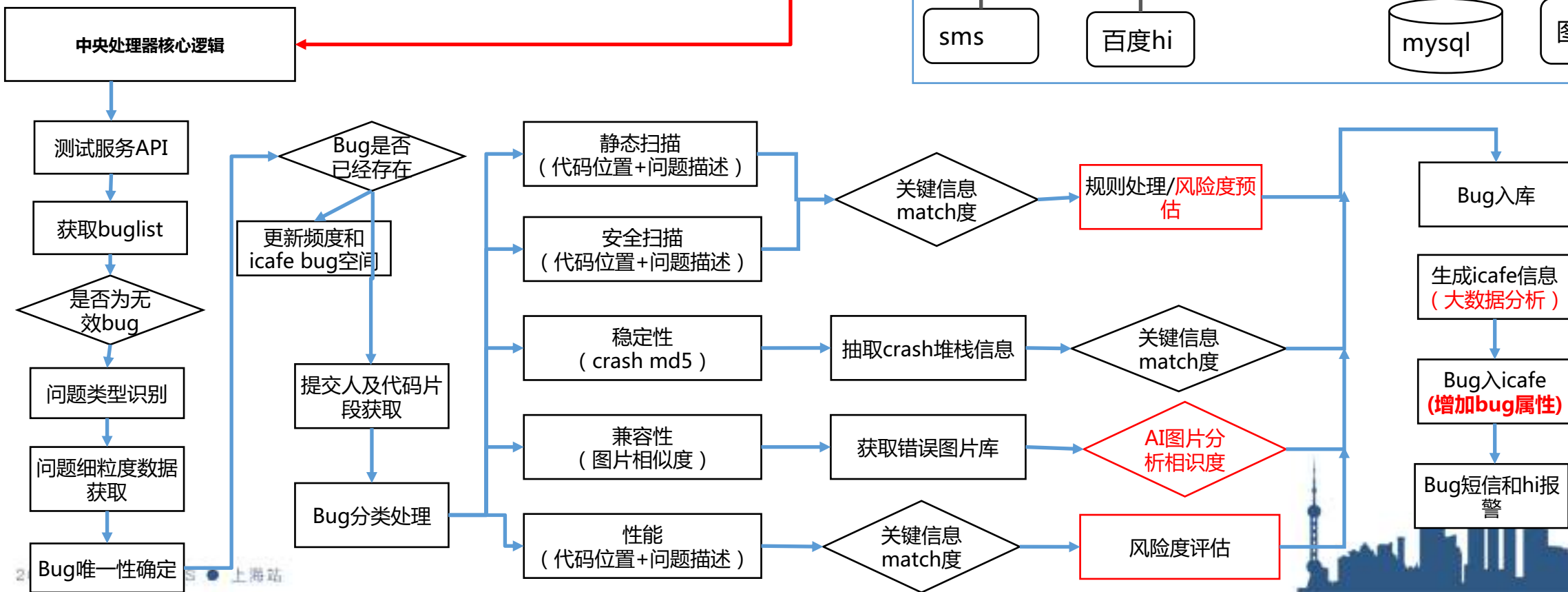
研发同学经常对提交的bug 几个问题：

- ① **误报**：该问题不是我引入的为什么报给我；
 - ② **重复报**：该问题之前报了，这次怎么又报出来；
 - ③ **无效报**：该问题是默认问题，不修复；
- 浪费时间、影响相互间信任。

系统架构图



效果以crash去重为例：
 $685(\text{独立crash数}) / 1269(\text{总crash数}) = 54.9\%$

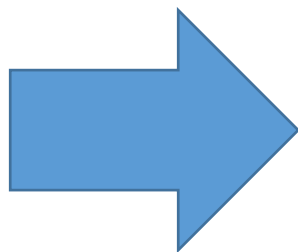


- 1、系统链接：200+开发者，30+个工具平台有效运转
- 2、覆盖业务：覆盖百度系近百款客户端产品，每天提交1000次
- 3、覆盖机型：近千台手机可调度
- 4、覆盖构建：每天构建近W次
- 5、发现问题：上线以来累计发现1W+问题



百度客户端产品构建系统整体架构—思考

- [illegible]



- 1、插件生态，优胜劣汰；
- 2、链路复现及减少链路路径；
- 3、不断抽象，让业务直面业务痛点；
- 4、可配置化，业务线可控；
- 5、品牌能力专项建设；

主要考虑四方面：

- 1、快速接入
- 2、工具平台有效连接
- 3、高效稳定性
- 4、闭环建设



日活过亿级应用--百度APP DevOps工程实践

需要解决三大问题：

- 1、业务团队、人员意识、需求不一带来的流程问题
- 2、产品特点带来的对系统性个性化需求的挑战
- 3、如何形成业务DevOps闭环

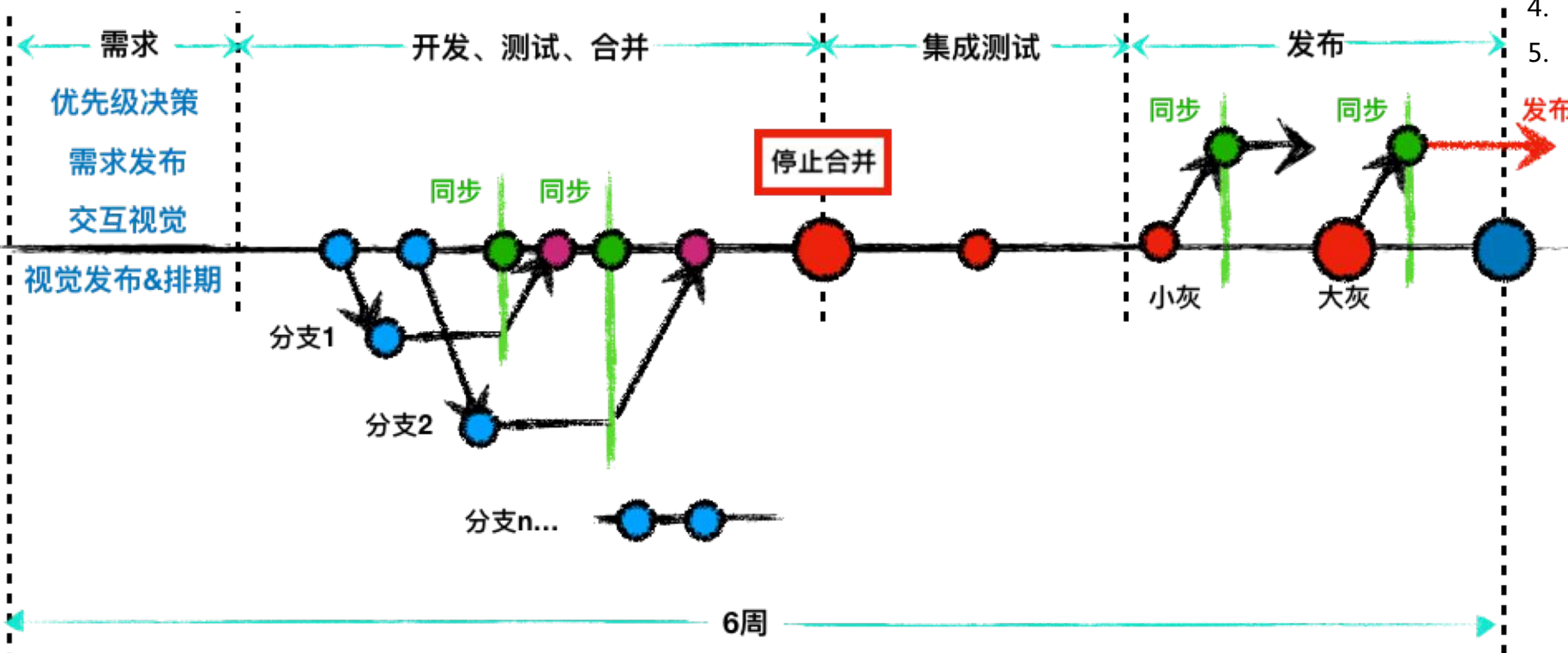


百度APP应用 DevOps工程实践-火车模式

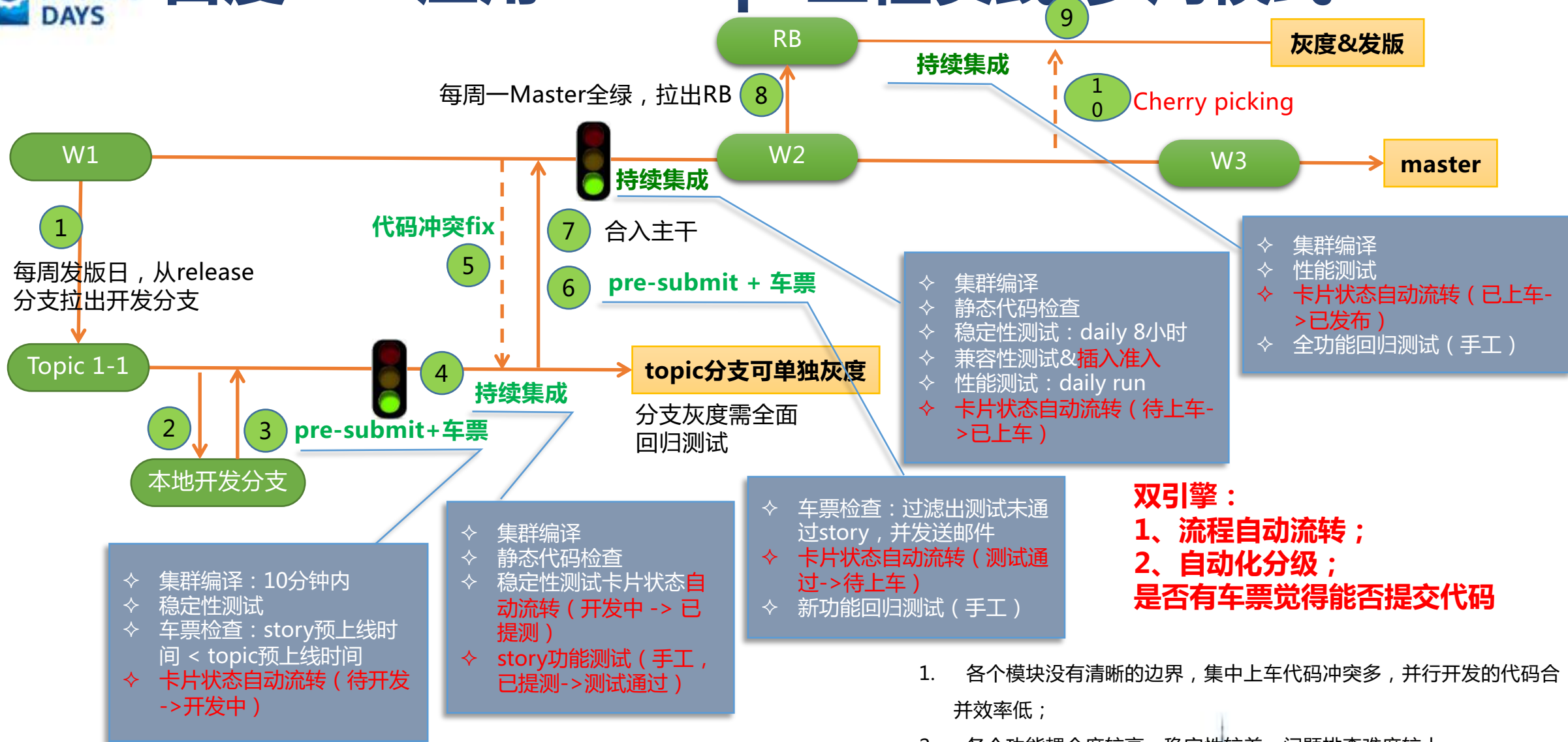


问题：

1. 交付周期长
2. 长周期的需求会拉长短周期需求的耗时，严重的会拉长整个版本的周期
3. 代码耦合度高，代码冲突大，合并是个灾难
4. 不支持分支灰度
5. 代码复用率较低



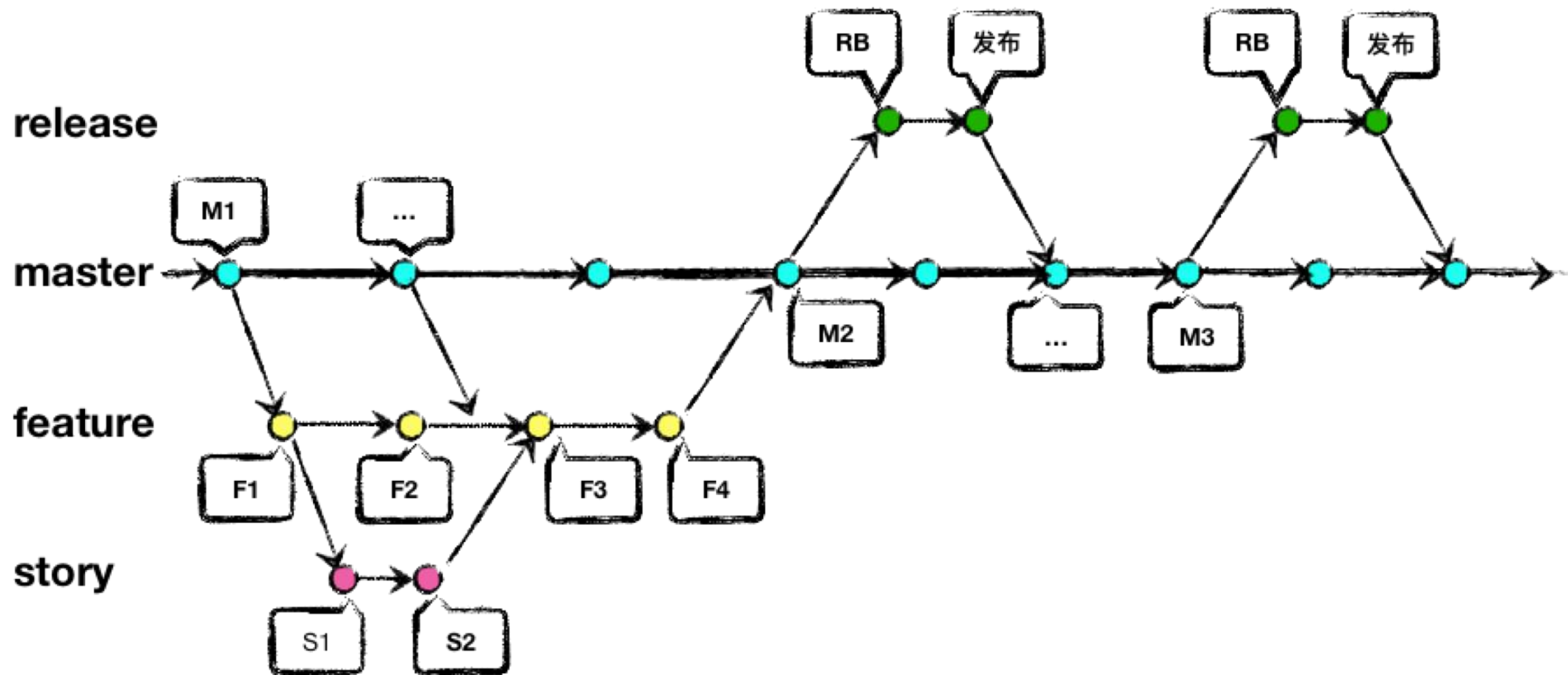
百度APP应用 DevOps工程实践-多周模式



多周交付的研发模式：

多个Topic分支并行，集中在某天上车

1. 各个模块没有清晰的边界，集中上车代码冲突多，并行开发的代码合并效率低；
2. 各个功能耦合度较高，稳定性较差，问题排查难度较大；
3. 代码集中在集成测试前后集中提测，测试压力和质量风险大；
4. 开发资源利用率低，集成和灰度期不接新需求；



单周交付研发模式：

1. 由Topic维度 -> 细粒度的Feature分支
2. 三周集中上车 -> 每周上车

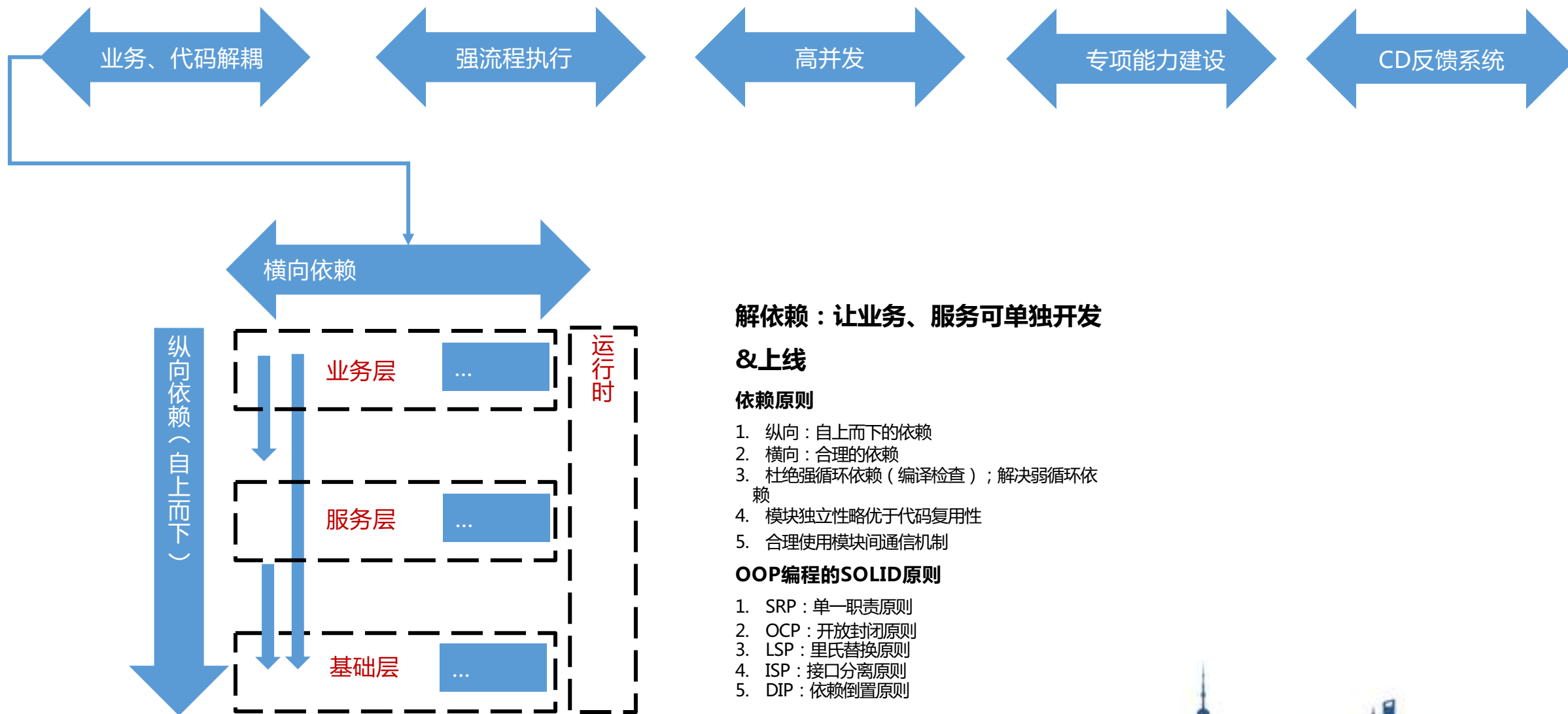
单周交付的收益

1. 单位时间内产能的提高：少量的Topic并行 -> N多Feature维度的并行
2. 集中集成 -> 增量集成；
3. 集中测试 -> 随时测试



CHINA
DEVOPS
DAYS

百度APP应用 DevOps工程实践单周模式-基础保障



百度APP应用 DevOps工程实践-集成系统

快速
问题
诊断快速
问题
召回快速
发版

CD

持续
集成
系统

问题定位

Cuid/uid
反查

Push查询

内核查询

一键反馈
查询Crash反
查Log
查询

用户反馈

收敛
渠道

报警

问题自
动分发监控界面
配置化推动问
题闭环打造平台
化能力

线上崩溃

Native崩溃

Java层崩溃

JS层崩溃

灰度平台

基本信
息业务指
标稳定指
标性能指
标可用性
指标广告指
标AB实验
对比

性能专项

打分
体系建
能
力

CPU

卡顿

包体
积过度
绘制平
台
化

手百双端

UC浏览器

搜
索评测
方案

内存

存储
空间

流量

功耗

今日头条

QQ浏览器

Mock专
项

异常测试自动化平台

自动mock数据+自动化case校验
接口异常

人工测试+自动化mock

C/S解耦测试平台

研发CI规范

需求管理/生命周期

Hook机制

稳定性测试

并行开发模式

CI 流水线建设

自动化建设

兼容性测试

系统落地业务线：

- 1、流程的差异性
- 2、个性化和通用性
- 3、CI/CD闭环结合

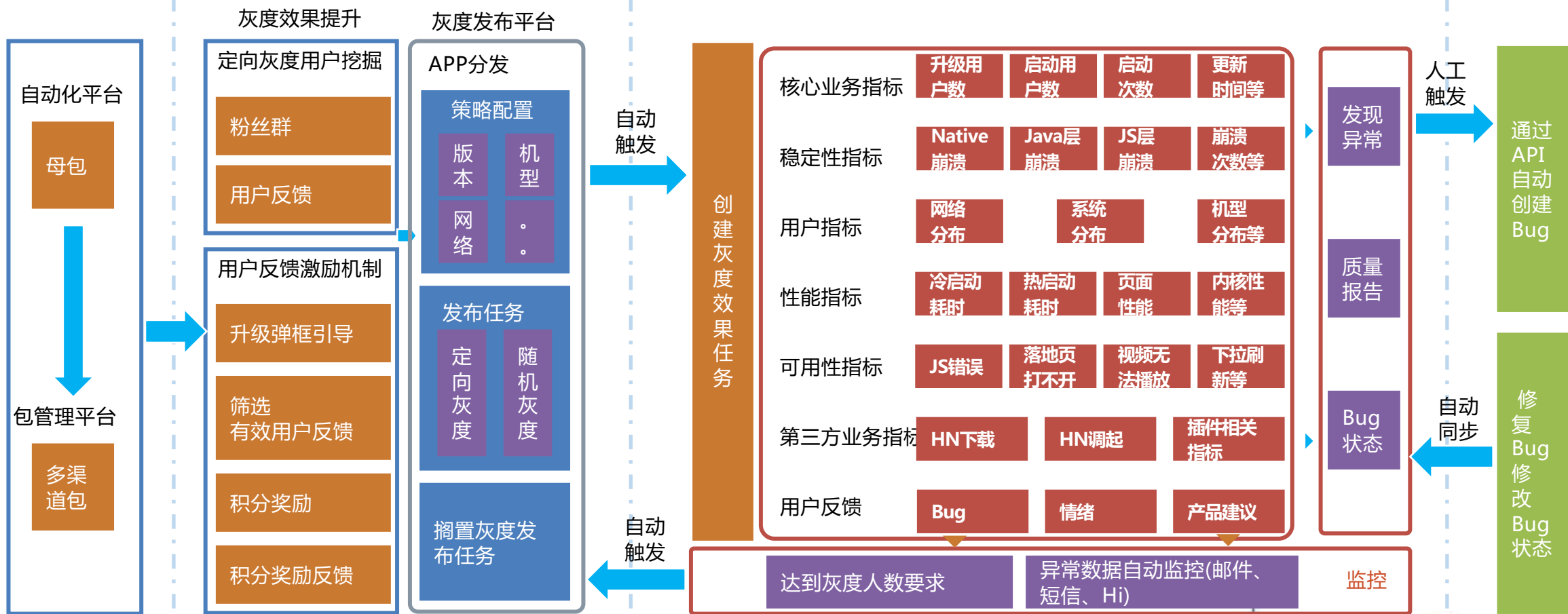
百度APP应用 DevOps工程实践-灰度发布

快速的包产出

快速的包分发

快速的灰度效果评估

问题:跟进和自动分发

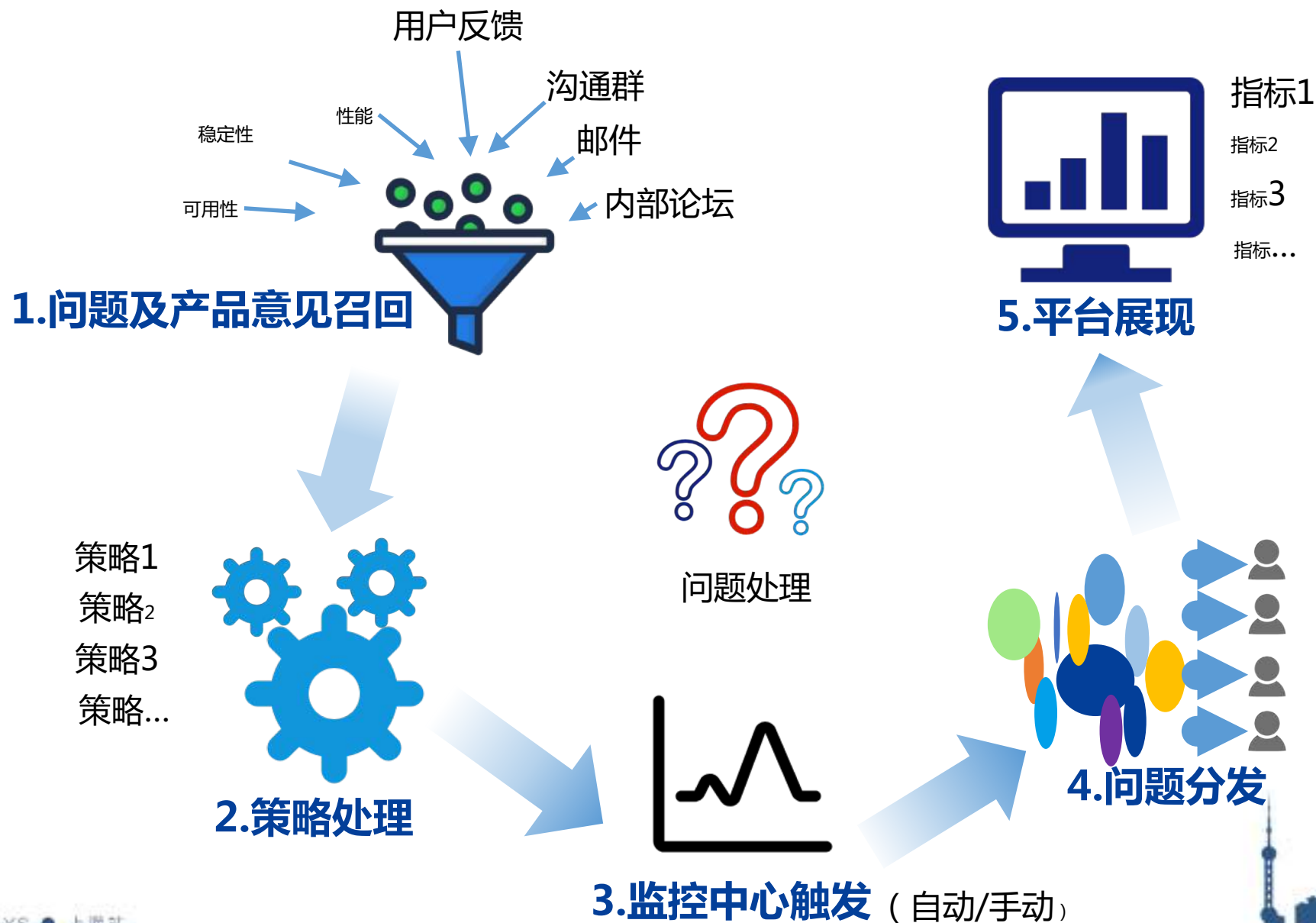


集成交付：灰度体系发布小时级发布和效果评估数据产出

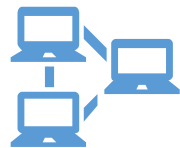
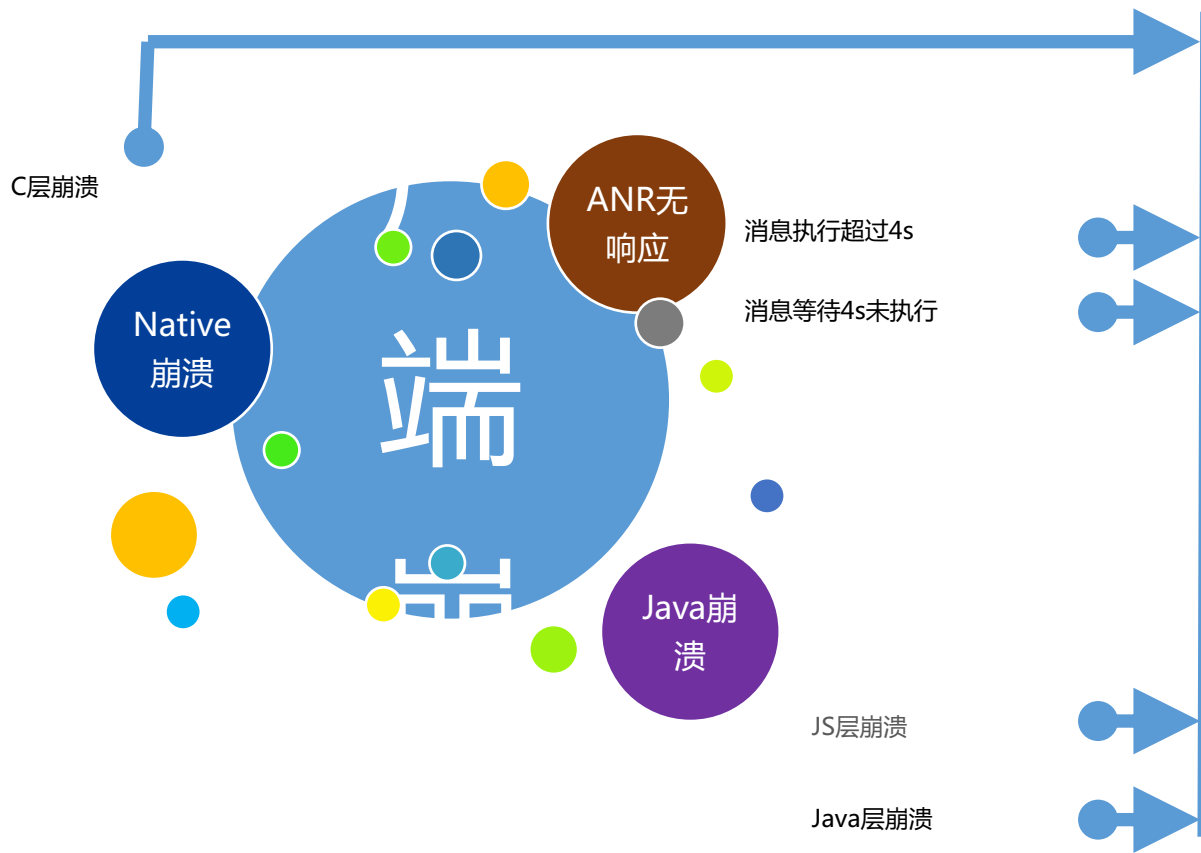
版本对比

效果评估

百度APP应用 DevOps工程实践-用户反馈



百度APP应用 DevOps工程实践-崩溃闭环



平台

数据上报、数据解析、清洗、修复、上线



崩溃日志收集



修复



分析



上线验证

崩溃率仪表盘

Crash分析

Crash反查

ANR分析

今日稳定性指标概况

最新线上版本 (Version)

容灾

0.000000%

= 0.000000 (Java)

+ 0.000000 (Native)

最新灰度版本 (Version)

容灾

0.000000%

= 0.000000 (Java)

+ 0.000000 (Native)

最大PV版本 (Version)

容灾

0.000000%

= 0.000000 (Java)

+ 0.000000 (Native)

全版本 (ALL)

容灾

0.000000%

= 0.000000 (Java)

+ 0.000000 (Native)

最新线上版本 (Version)

MTJ

0.000000%

最新灰度版本 (Version)

MTJ

0.000000%

最大PV版本 (Version)

MTJ

0.000000%

全版本 (ALL)

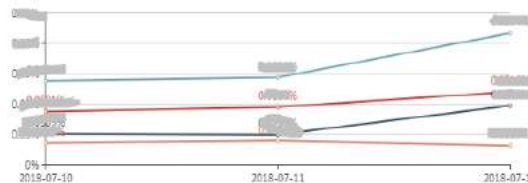
MTJ

0.000000%

关键版本崩溃率趋势

最新线上版本

android 3.0.0.0 版本崩溃率



最新灰度版本

android 3.0.0.0 版本崩溃率



百度APP应用 DevOps工程实践-打分情况



百度APP应用 DevOps工程实践-总结

- ① APP研发流水线评估系统

一套全面
精细化的
度量系统

一套符合业务迭代现状的全流程研发规范

百度APP DevOps

一套高
效稳定
的集成
系统

- ① 线上监控；
② 反馈系统；

一套全面及时的线上监控反馈系统

- ① 打造1.5月发布->3周->**单周**->按需发布的研发模式；
② 基于业务和服务的研发解耦；

- ① 高并发高稳定的集成构建系统；
② 稳定性专项；
③ 性能专项；
④ 灰度发布；



一直在路上。。。。

UI测试

IOS端封闭性

手机异构性导致的稳定性问题

灰度精准性和高效

端架构解耦

遍历深度

研发行为大数据全、真，AI赋能研发时代，**我们来了**



THANKS

Website :
chinadevopsdays.org/

Global Website:
www.devopsdays.org/events/2018-shanghai/

Official Email:
organizers-shanghai-2018@devopsdays.org



Official Wechat

