

# 职级申请答辩

平台业务部 iOS 组 寇祖阳

# 主要职责

- 技术攻关
- 新技术的调研和推广
- 致力于提高整个项目和团队的质量和效率

React Native

# 背景

- 运营策略变化快，紧急需求多，希望尽快上线
- 产品迭代快，多轮迭代、实验的需求旺盛，希望缩短发版周期
- 原生代码更新不够灵活，一月一发版

# 任务

- 尝试解决产品迭代速度和发版速度不匹配的问题
- 实现在线更新

# 技术选型

- Hybird 解决方案的性能和体验相对比较差
- 基于配置的方案灵活程度不足
- React Native 兼顾了性能、体验和灵活程度✓

# 整体结构

业务逻辑

代码和资源的分发机制

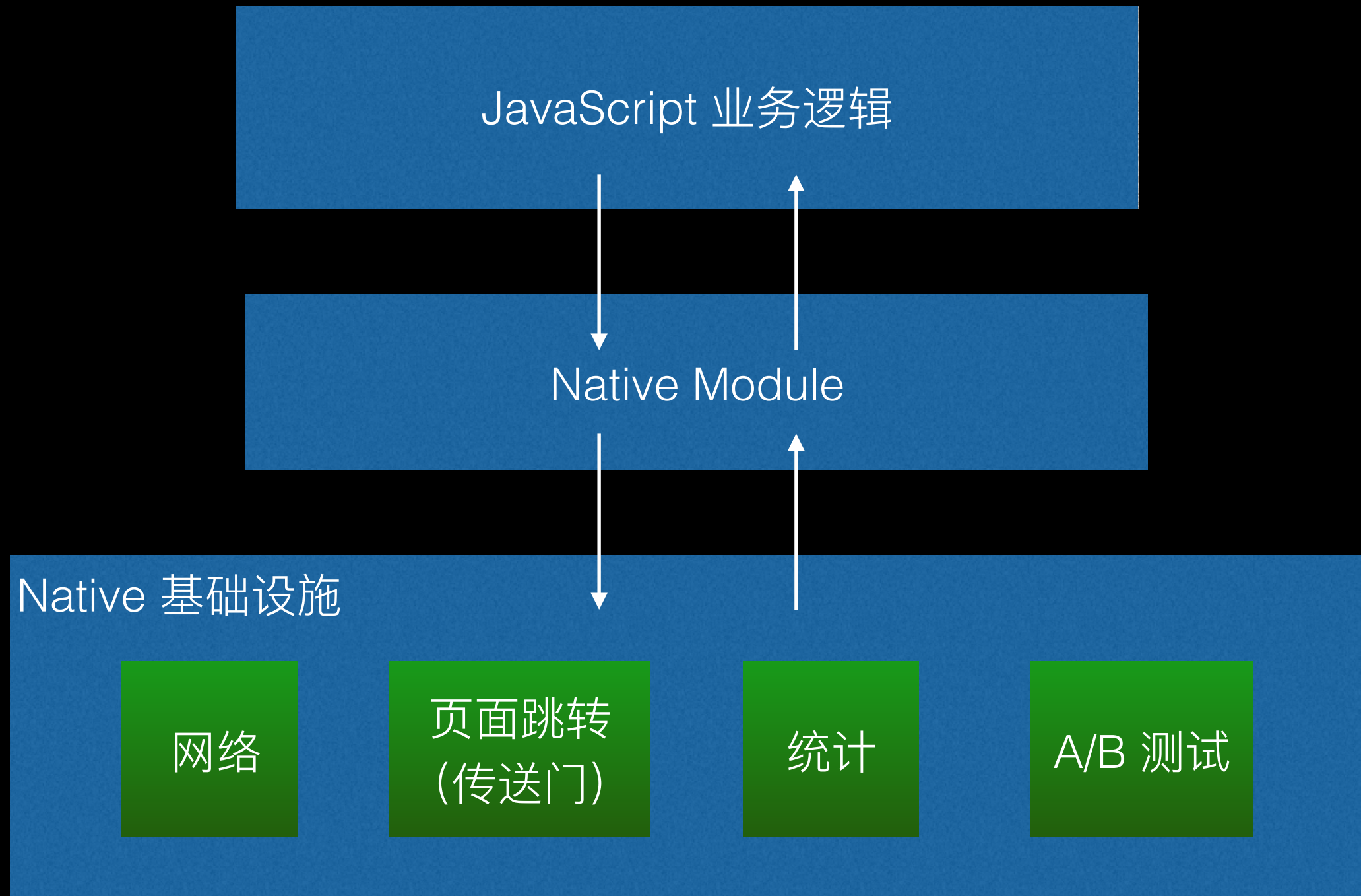
基础设施的桥接

# 代码、资源的分发

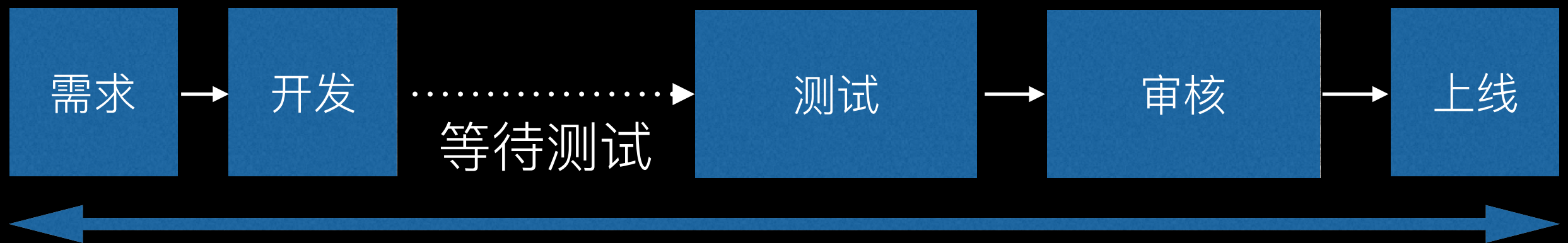
- 目的：实现在线更新
- 每次启动检查更新
- 预先下载新的资源和代码，下次启动生效
- 完整地实现了整个系统的客户端和服务端
- 考虑代码的安全性



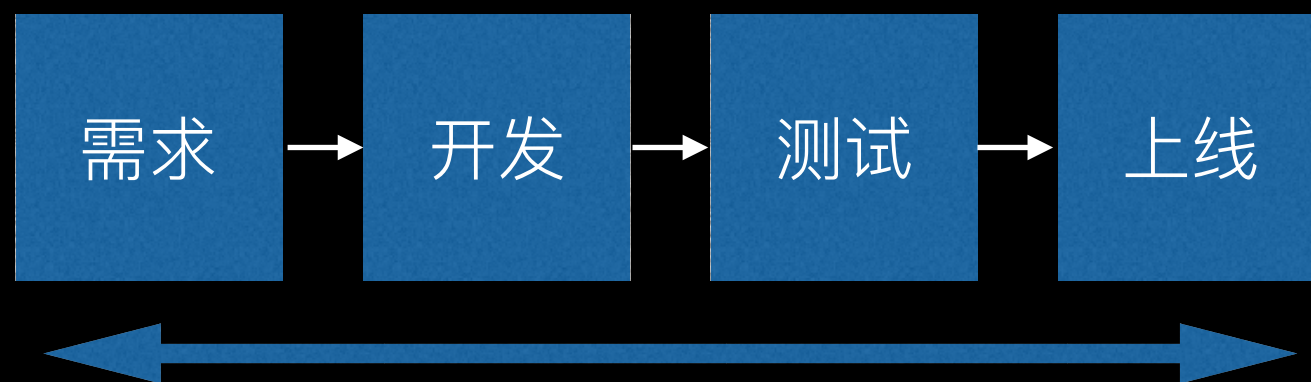
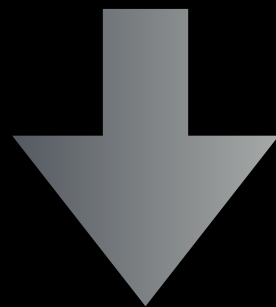
# 原有基础设施桥接



# 成果



一至两个月



一至两周

# 成果

- 实现了业务的在线更新
- 业内较早地在生产环境中使用 React Native
- 在 w3ctech 美团网 React 技术专场对外分享，参与人数 180+
- 在第一届大前端技术沙龙面向全公司宣传

# 未来的规划

- 依赖管理
- 增量更新
- JavaScript 和 React 培训
- 适应 React Native 的流程规范的制定和实施
- 与其他平台的代码复用

在线热补丁

# 背景

- 线上故障难以修复
- 普通审核需等待一周
- 快审流程复杂、资源珍贵，需要一天
- 主要依靠后台帮助绕开问题
- 纯客户端 bug 无法修复，导致业务不能按时上线，用户体验受损，公司蒙受损失

# 任务

使 iOS 客户端具有在线修复线上故障的能力

# 现有解决方案

- Wax (Lua)
- Nu (Lisp)
- JSCocoa (JavaScript)



# JSPatch

- 😊 代码短（第一版小于1000行代码）
- 😞 全部使用运行时，没有用 libffi
- 😊 基本满足我们的需要

# 积极参与社区建设

- 发现多个问题
- 提出多个 PR
- 积极回复社区内的问题

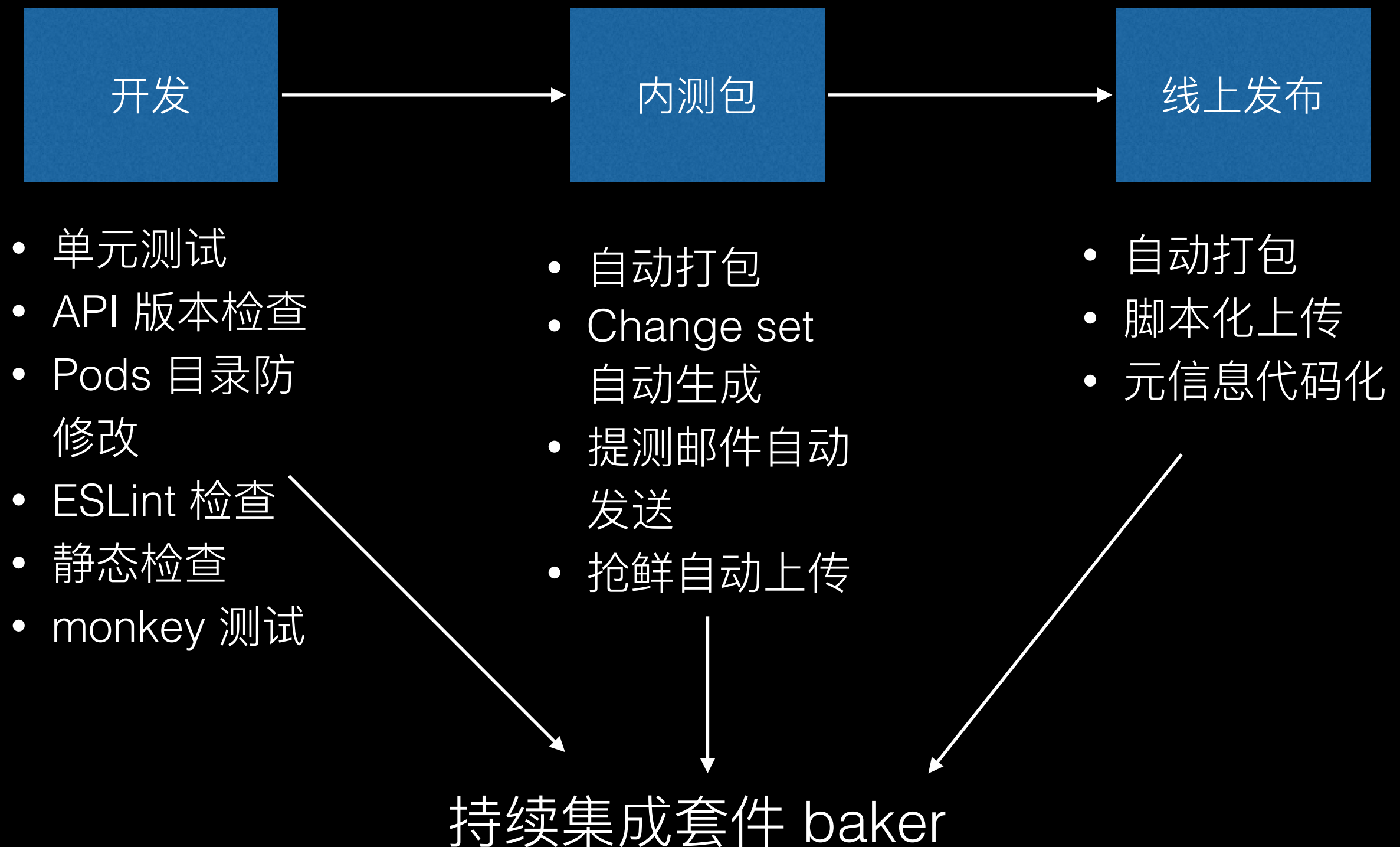
# 结果

- 上线3个版本
- 修复9个线上故障，其中7个 Major，1个 Critical
- 暂时没有遇到无法修复的情况
- 支持了平台、旅游、猫眼等部门的故障修复
- 已经成为各个频道修复线上故障的首选方案
- 在第一届大前端技术沙龙宣传

# 未来的规划

- 多 app 支持
- bug 修复能力总结和提高

# 持续集成系统



# 结果

- 支撑了全公司 20+ iOS app 的打包、测试等持续集成工作，极大地减少了人工和重复劳动
- 打包的单点问题消除了
- 自动化检查减少了事故的发生，降低了人工检查的成本
- API 版本检查消除了调用高版本 API 引起的线上故障
- 基本消除了主干分支编译不过引起整个团队无法工作的问题

# 技术规划

# 生产环境

## 监控

- 性能
- 异常
- 日志
- 远程调试

## React

## Native

JSPatch

## Web

- WKWebView
- Bridge

## 基础设施

### 安全

- 网络
- 本地

### 网络

- 安全
- 性能
- 到达率

### 模块间调用

- 传送门
- microservice
- 文档化

### 存储

- 数据库
- 加密



# 开发环境

## 监控

- PR
- task

## 自动化

- Podspec/Podfile 检查
- commit message 检查
- CSA/Infer 静态检查
- OCLint/ESLint 检查
- iDP、iTC 流程自动化
- 文档自动生成



# 技术攻关组

- 6月成立
- 2个P6 1个P5 2个实习生
- 远程工作

谢谢评委