

# DevOps与SRE珠联璧合之路

2018 DevOps Days 深圳站  
2018 中国·深圳



# 今天对快与稳的要求涉及多个层面



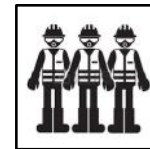
业务



开发



测试



运维

2018 DevOps Days 深圳站

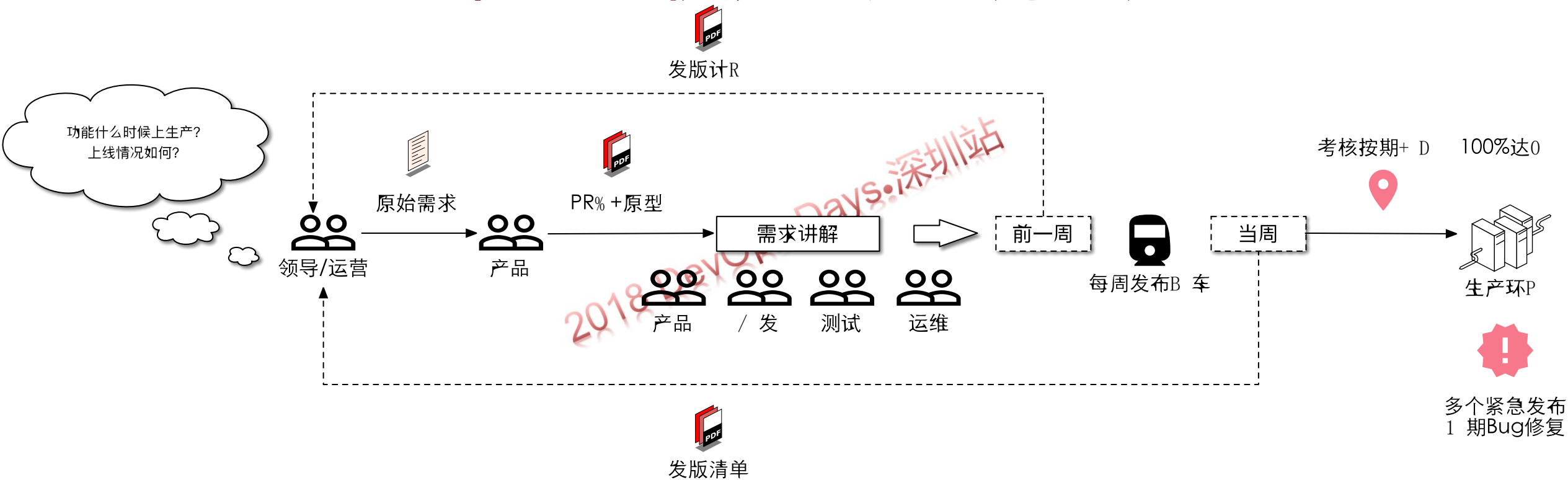






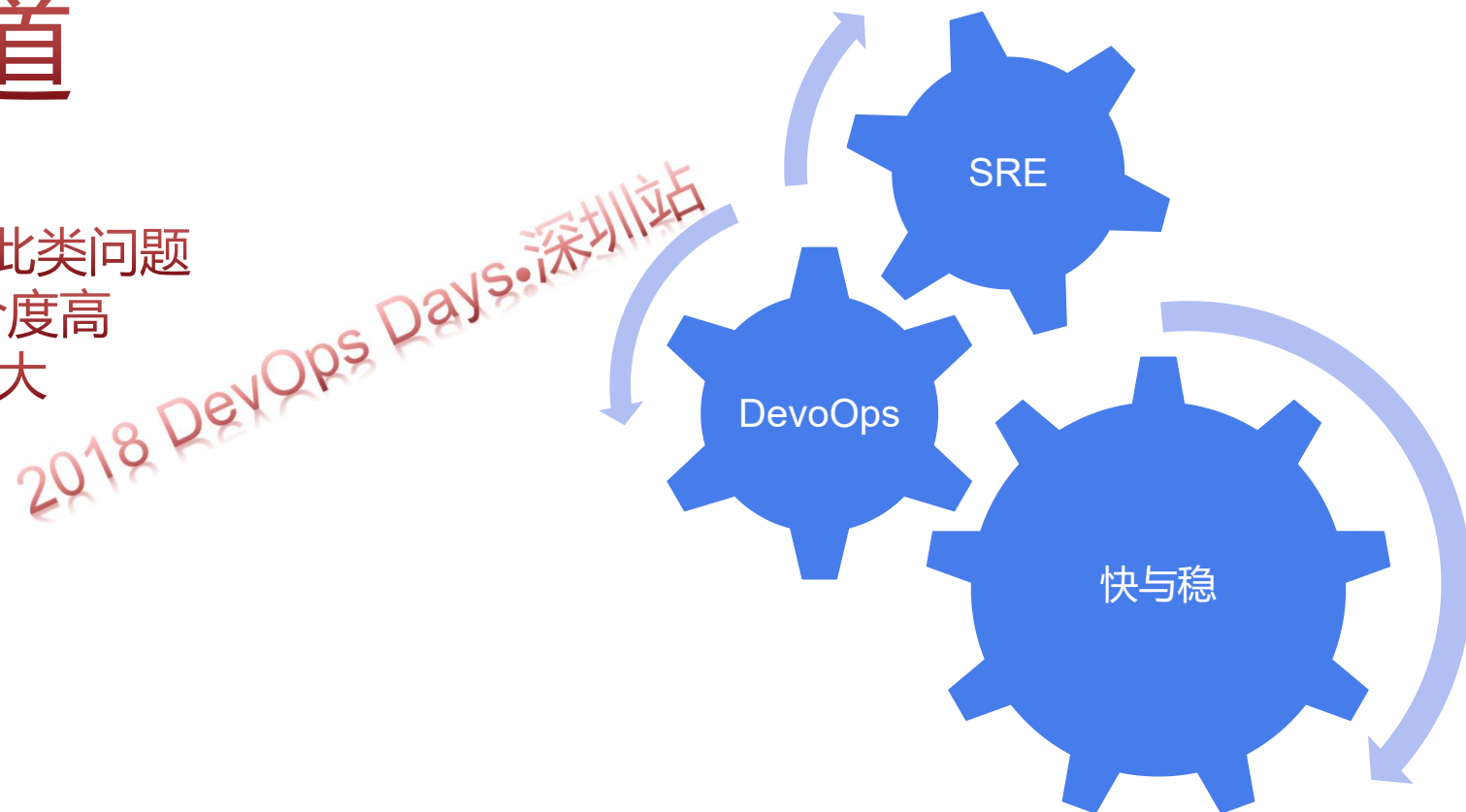


# 单纯求快无法一蹴而就



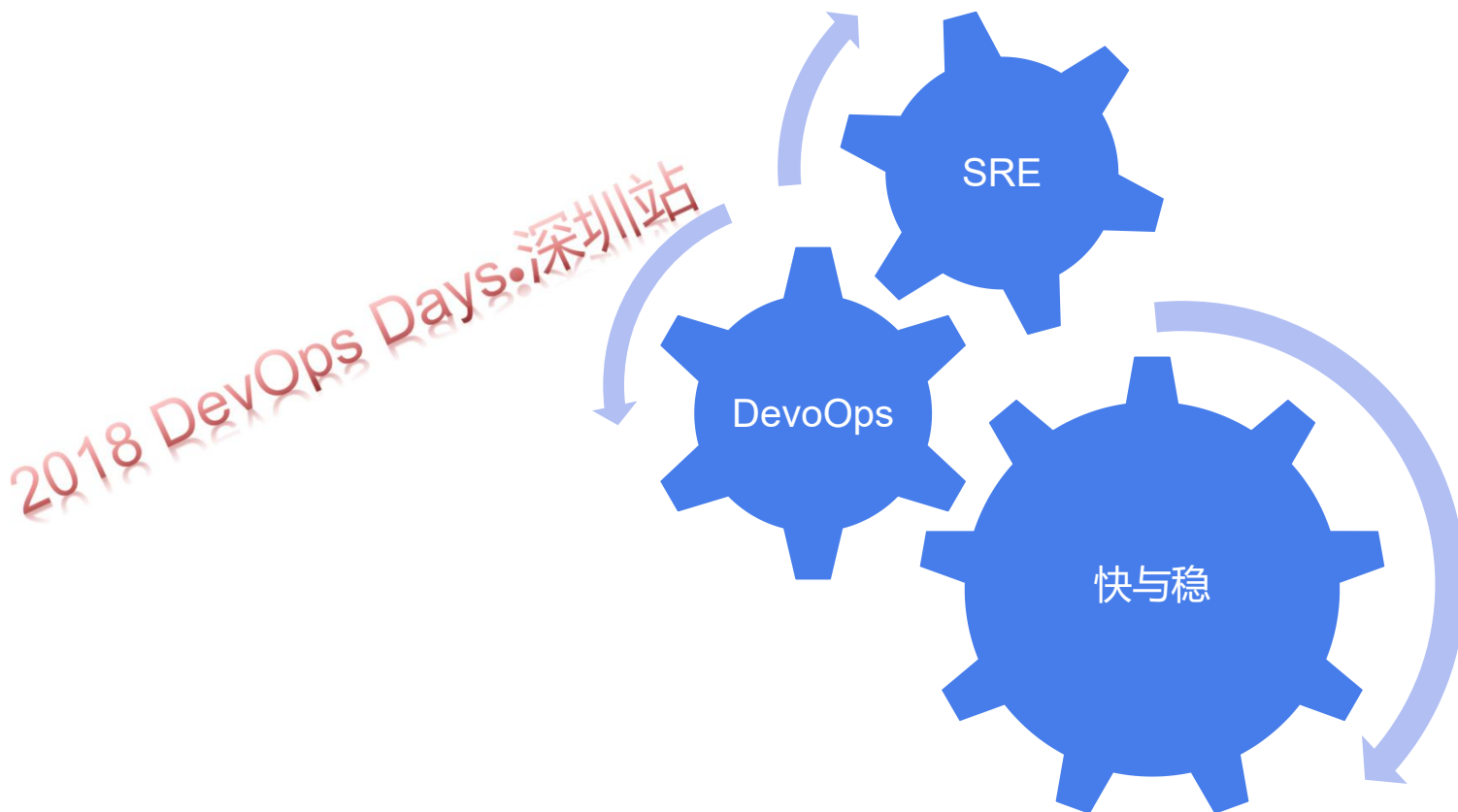
# 我们知道

- DevOps和SRE都意在解决此类问题
- 涉及运维部分时，两者重合度高
- DevOps的外延还在持续扩大



# 问题

- 该如何选择?
- 要如何引导团队实施?
- 会面临怎样的挑战?





# 让我们通过某客户实施 看看过程中的思考和举措



# 某电商客户



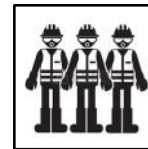
业务



开发



测试



运维

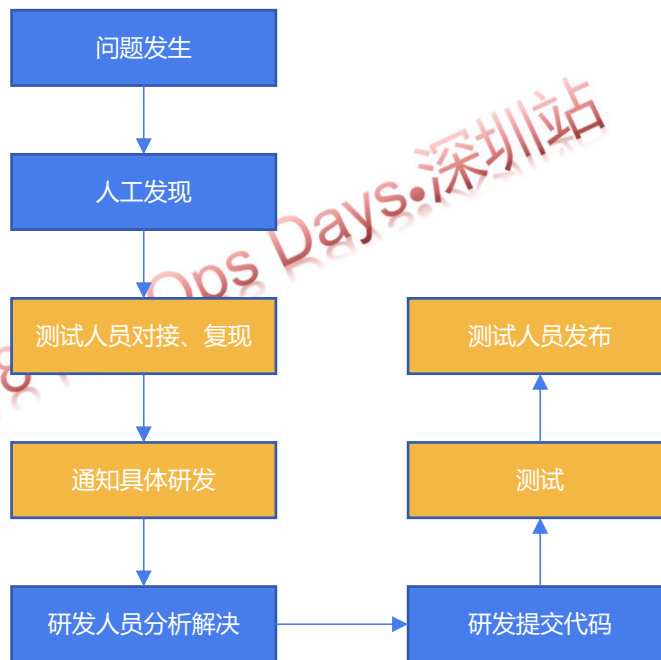
整个组织已经有开发、测试和运维

已有运行多年的机制和流程

测试负责生产系统发布而且担任生产系统问题的关键联系人

运维团队负责生产系统维护

# 某电商客户



# 某电商客户



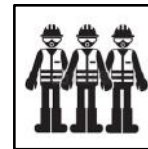
业务



开发



测试



运维

只关心开发  
业务感差  
不熟悉运行环境

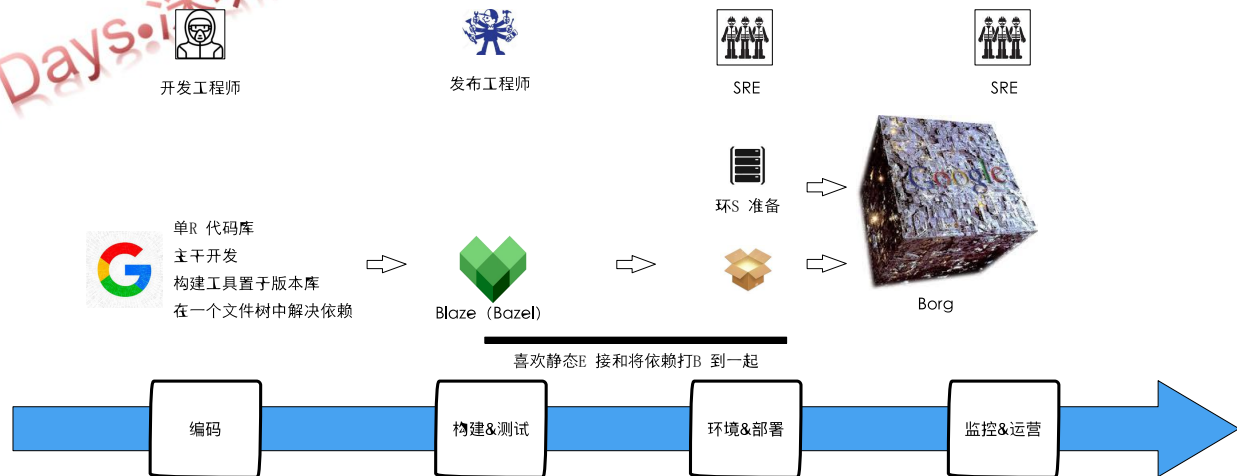
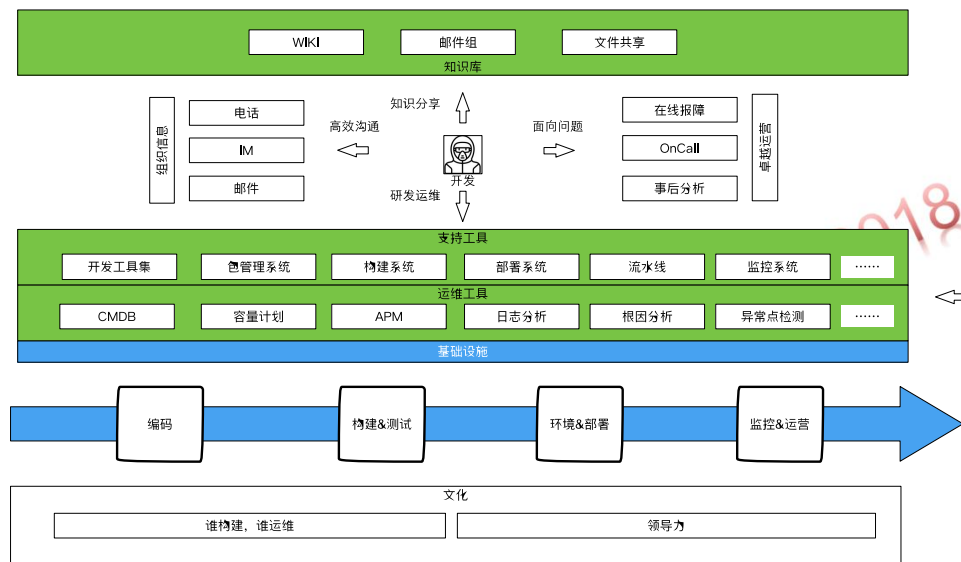
熟悉业务  
发布的关键  
问题响应的关键

疲于应对  
故障分析不足  
不熟悉业务系统

应急响应：慢  
日常运维：乱  
工程研发：无



# Amazon, Google, 还是其它?



	亚马逊	谷歌
重点	(大量) 系统研发效率 (研发侧)	(大规模) 系统稳定性 (运维侧)
运维主要人员	研发人员	SRE
主要思想	谁构建, 谁运营	研发构建, SRE运营
人员	研发团队 工具团队 (研发支持工具) 运维团队 (基础设施+运维系统)	研发团队 工具团队 (交付工程师) (研发支持工具) SRE团队 (系统运维+基础设施+运维系统)
研发支持工具	大量支持工具 重点支持以包为单位的开发及环境配置与部署 提高研发+运维整体效率 (侧重研发)	大量支持工具 重点支持其特殊的工程实践和分布式资源调度系统 提升研发+运维效率 (侧重运维)
开发、构建	(项目) 包屏蔽底层VCS 逻辑包进行 (系统) 环境配置 通过声明解决依赖 (类Linux包的依赖系统) 分钟级生成新系统	单根代码库、主干开发 强制测试、强制代码评审 通过文件解决依赖
部署	通过包拼装整个系统	构建后整体分发部署
系统基础设施	早期工具化维护, 后期云计算可以自治 (ASG)	高度自治系统 (Borg)
开发vs.运维	通过OE目标规范, 通常20~30%运维工作	通过故障预算协调, 50%运维
运维	开发人员OnCall, 关键问题事后分析	SRE OnCall, 关键问题事后分析
文化	鼓励竞争的角斗士文化	高度受信环境

# DevOps和SRE并非目的

目标

高效研发  
高效维护  
高效系统

手段

敏捷  
DevOps  
SRE  
其它

关键产物

良好的做事方式  
强力团队  
完善的工具体系  
靠谱的流程和机制



# 挑战 I 从何下手

2018 DevOps Days • 深圳站



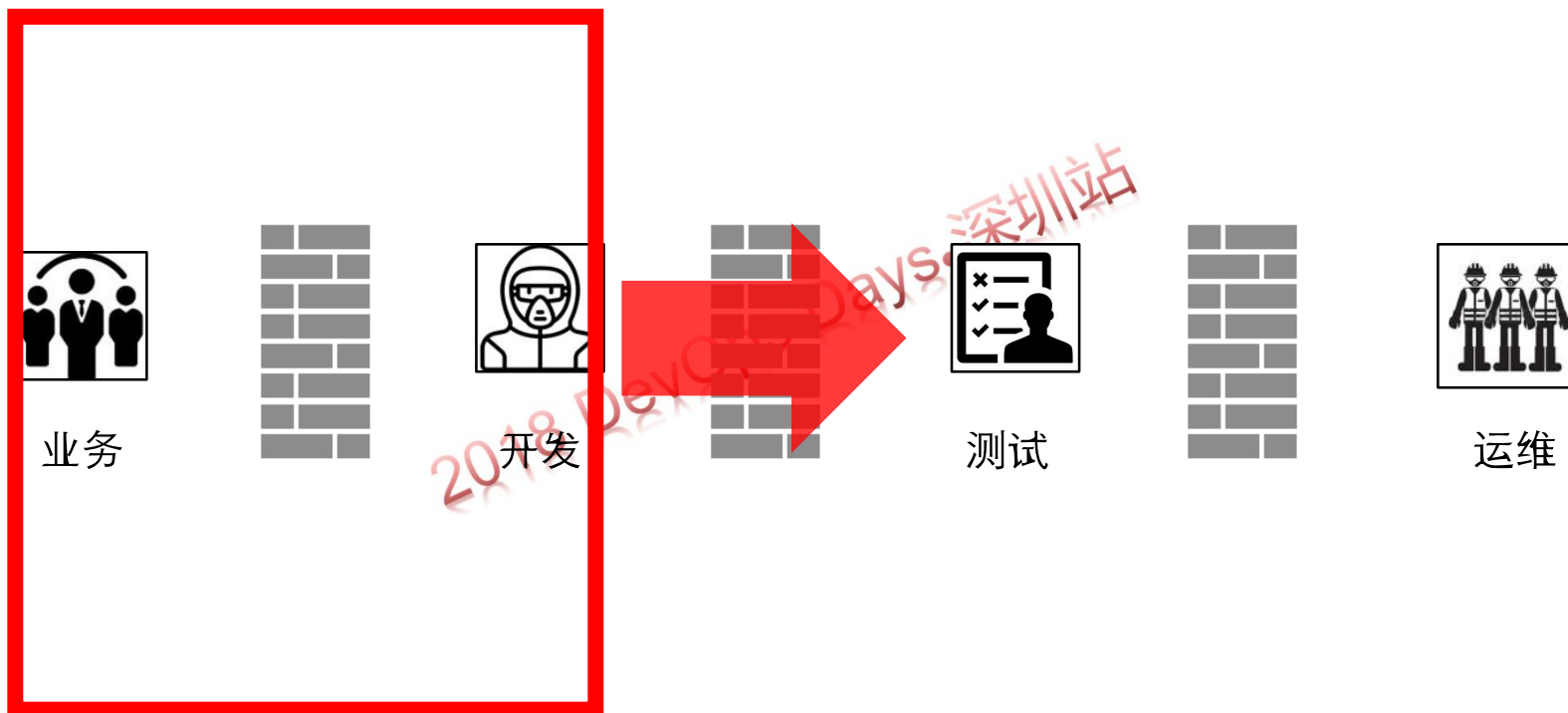


# 由谁做什么?

2018 DevOps Days 深圳站



# 研发的敏捷扩展



# 运维的工具化推动



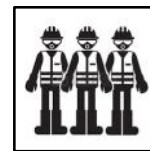
业务



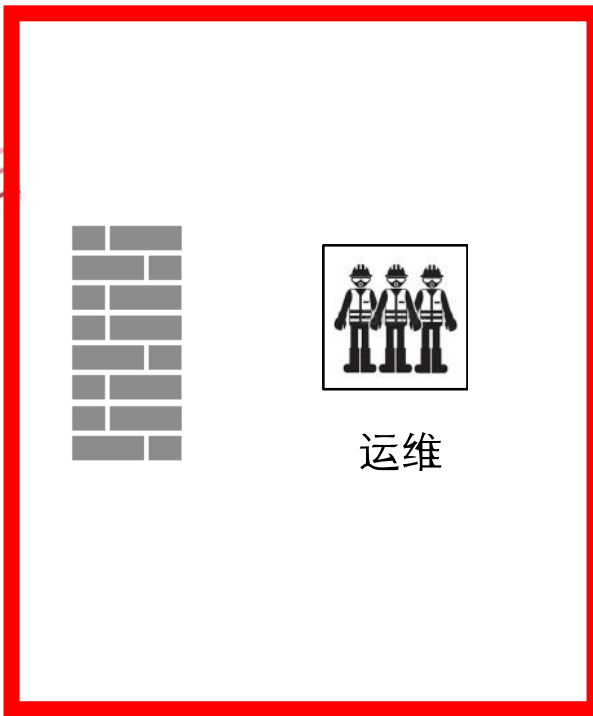
开发



测试



运维

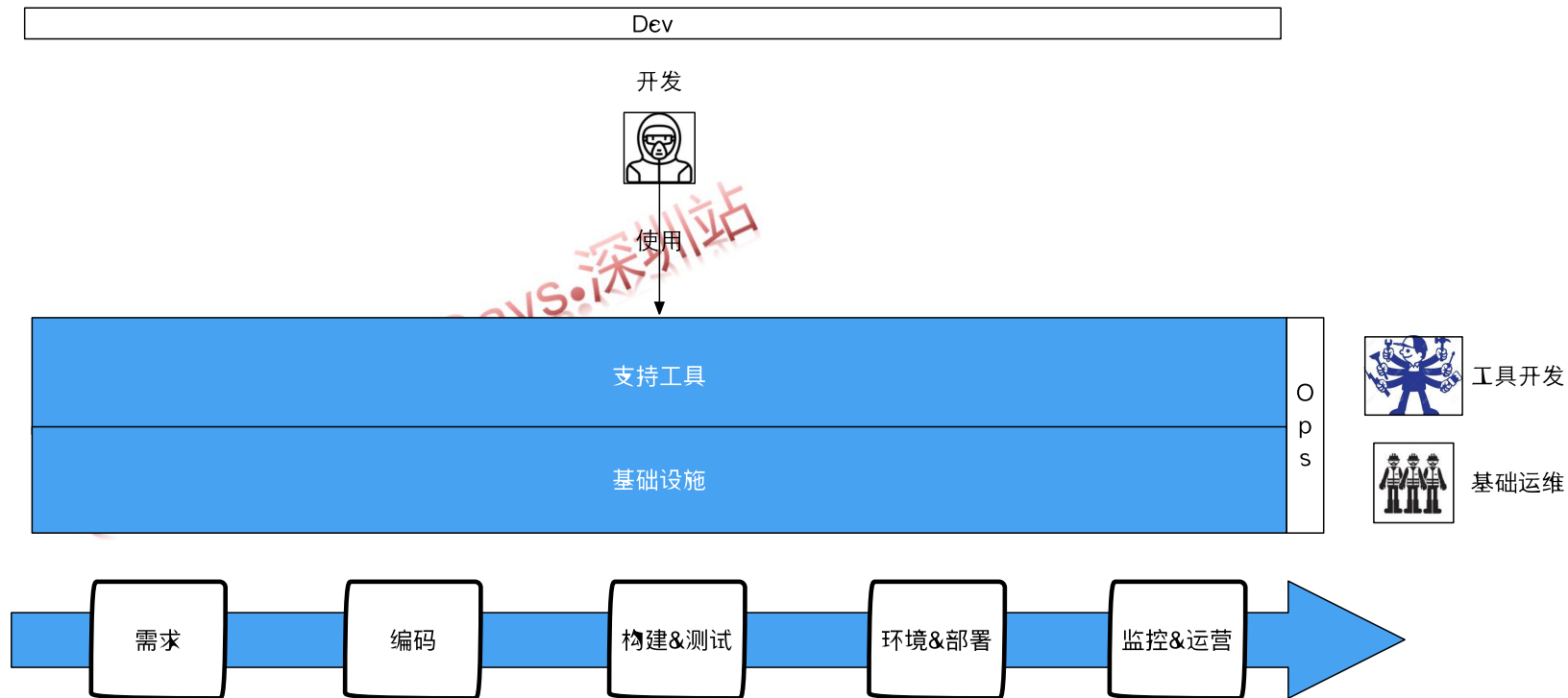


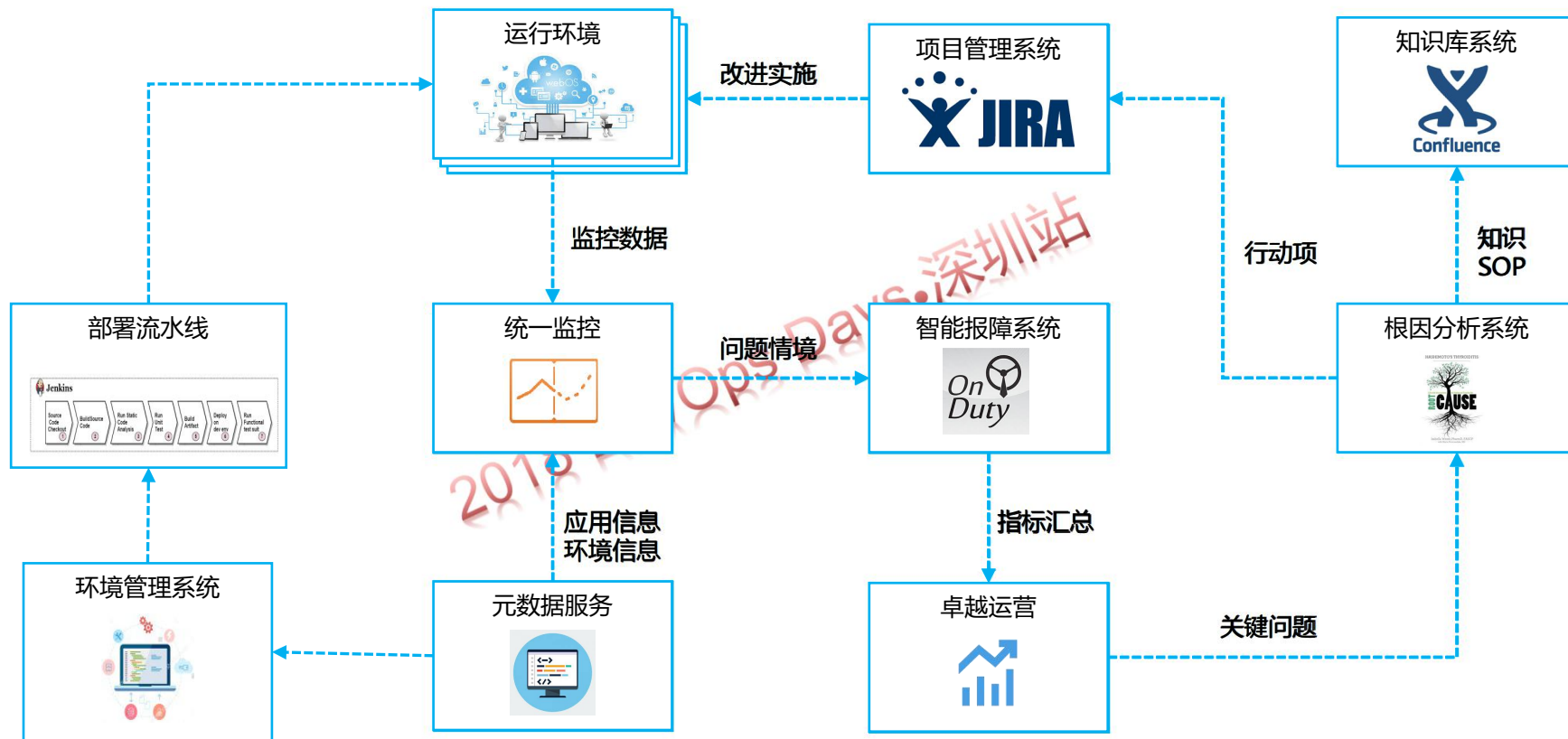
## To B/C产品体系

100~研发团队，近乎从头创建团队，包含20~运维：

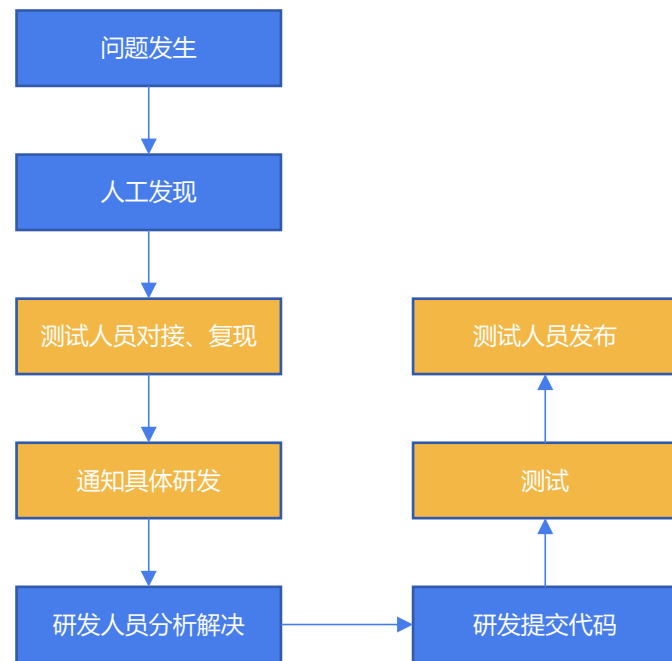
- 3传统基础运维
- 平台工具研发

没有测试，研发运维产品





# 某电商客户



# 某电商客户

## DevOps实施对测试团队冲击最大

- 任何横切团队都会引入效率问题
- 研发测试比3: 1, 业界9: 1
- 测试基本靠手工

## DevOps对测试团队的机会也最大

- 测试熟悉研发和业务
- 让被动变革成为主动变革
- 研发相关平台建设技术要求不高



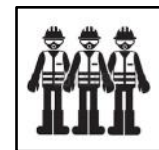
业务



开发



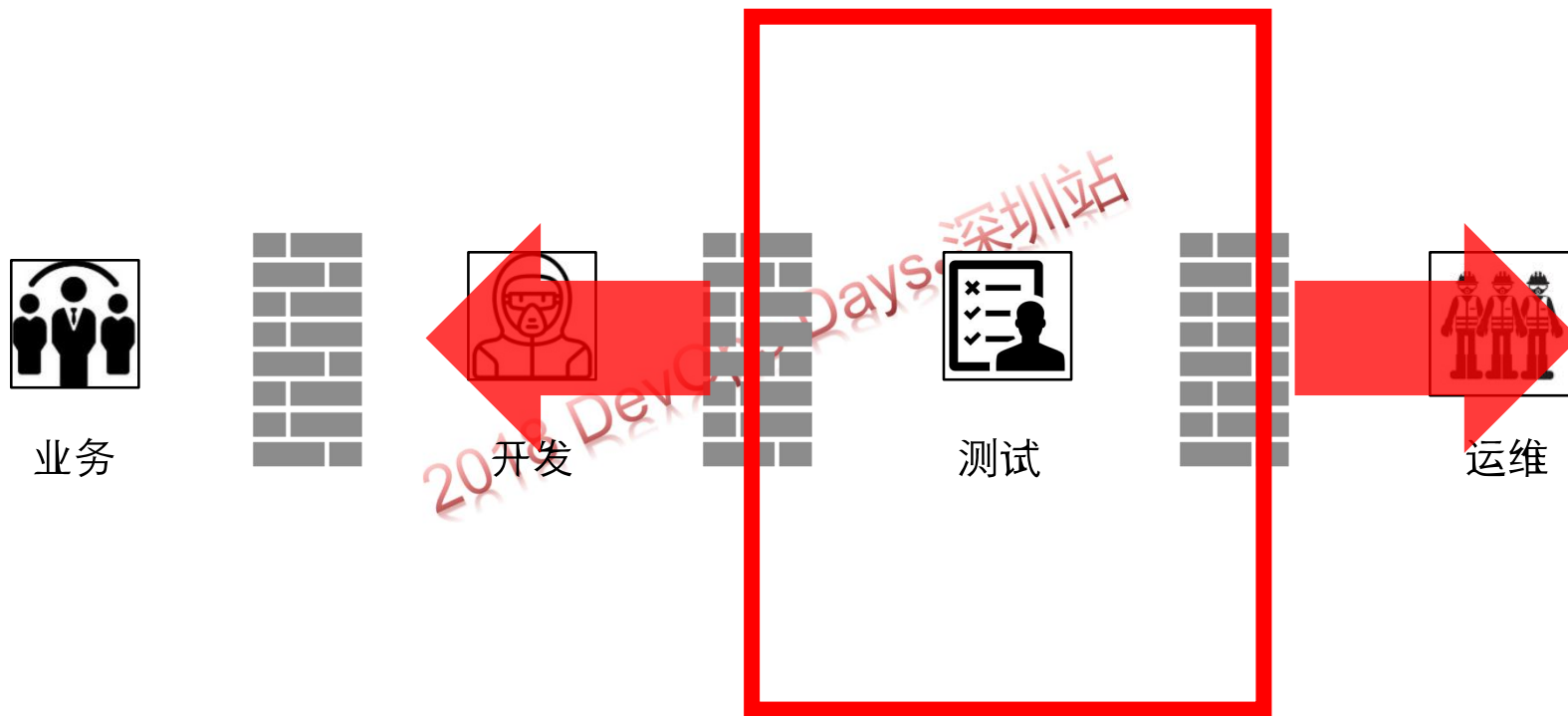
测试



运维



# 从测试团队入手





# 落地三板斧

重新定位团队  
从线上问题入手  
可视化



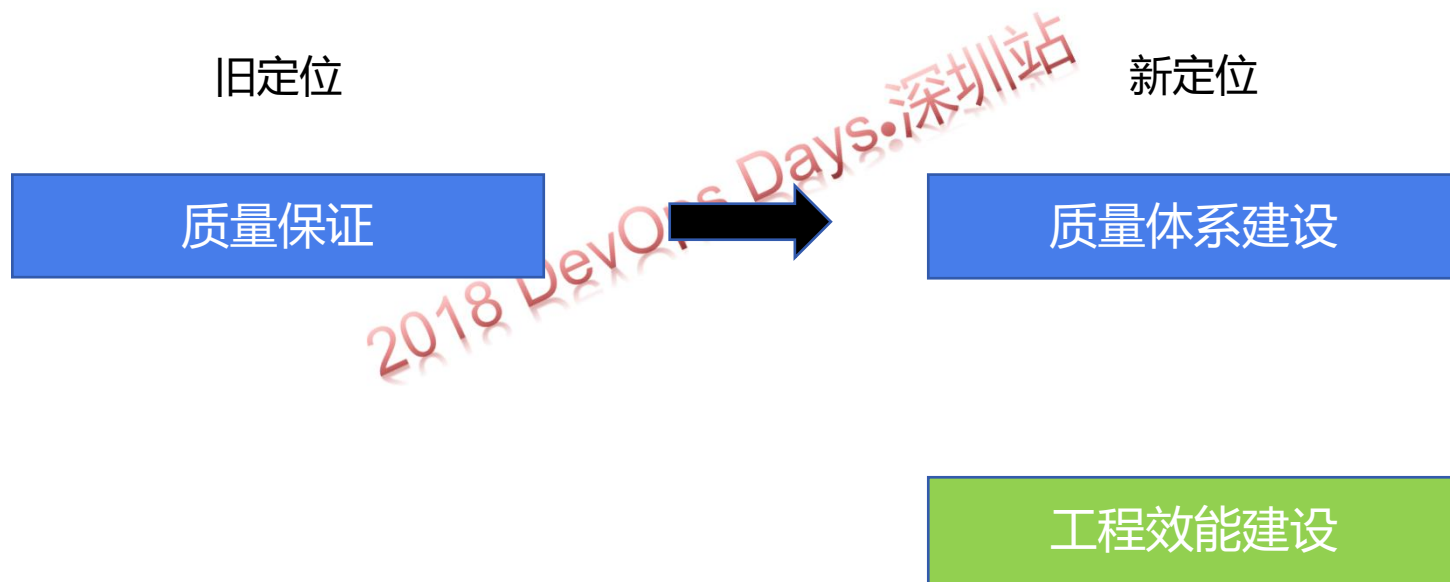
# 重新定位测试团队



可以涵盖DevOps的大部分工作



# 新定位，新使命



DevOps可以涵盖的大部分工作

# 新定位，新使命

旧定位

质量保证



新定位

质量体系建设

工程效能建设

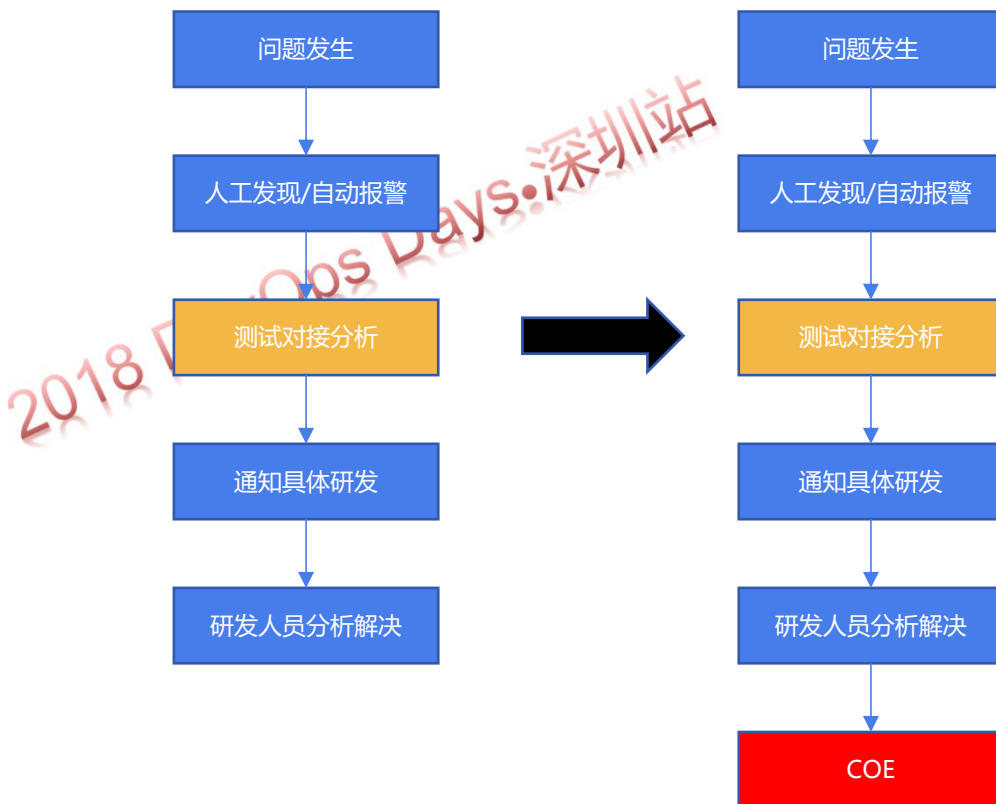
质量管理平台  
业务质量监控与评测  
代码质量监控  
自动化测试  
专项测试

研发效率数据收集分析  
流程改进  
CI/CD  
构建系统  
环境管理系统

DevOps可以涵盖的大部分工作

# 线上问题

- 反应不及时
- 事故分析不深入
- 研发业务敏感度低



# COE

## 什么是COE?

COE (Correction Of Error) 是亚马逊使用的一套事后总结与分析的系统性方法。其通过识别问题的根本原因并追踪识别的行动项来解决这些问题，从而提高服务（或系统）的整体质量并推进负责团队的责任。

## COE不是!

COE不是追责!

COE不是寻找问题责任人并问责的过程!

COE不是在问题发生后处罚相关人员的过程!

## 做COE的原因

当系统或服务遇到严重问题（如Onduty的1级和2级报障）并需要深入分析时，就需要进行COE。COE可以帮助我们：

- 通过深入导致问题的一系列事件来找出问题的根本原因
- 通过可追踪和可交付的行动项来解决问题的根本原因
- 防止问题再次发生
- 分析问题对业务和客户的影响
- 将分析得到的经验保留下来并与其他人员分享

参见：<https://gitbook.cn/books/5ae1357a0b644f1314c37f7b/index.html>



# 可视化



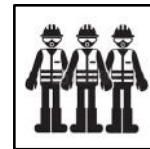
业务



开发



测试



运维

bug数据分析

每周发布数据  
测试bug数据  
代码静态扫描报告

.....

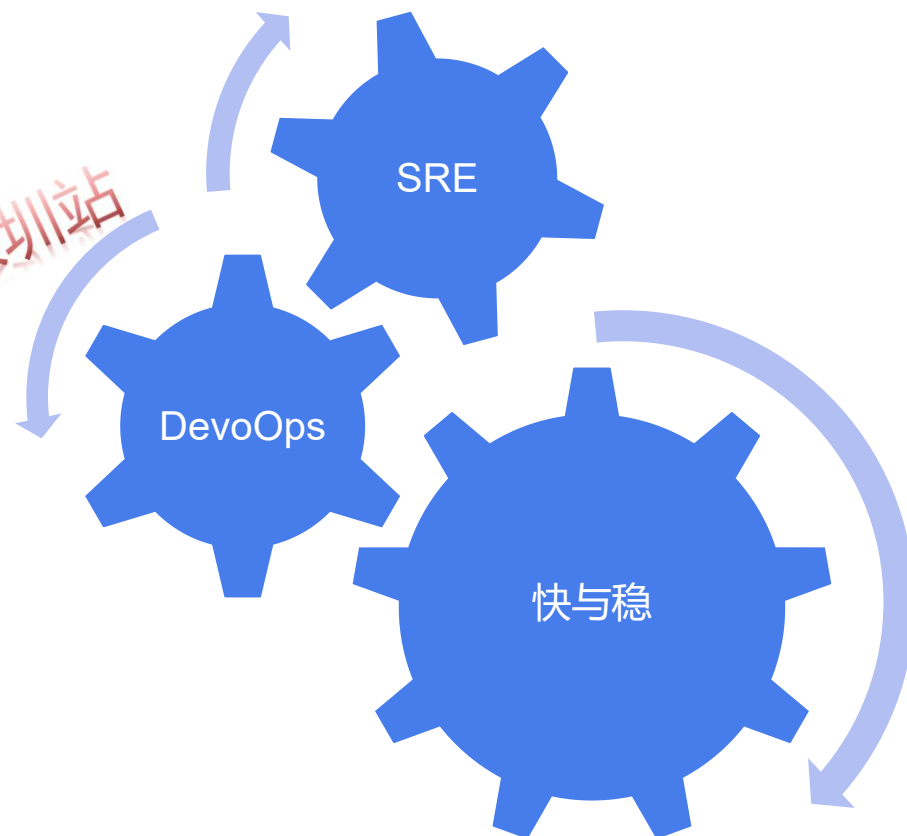
故障数据分析  
线上bug分析



# 尚存问题

- 可视化部分仅涉及质量而非过程
- 研发对业务的责任心及运行系统的熟悉度如何提高?
- 应急响应慢
- 系统稳定性提高

2018 DevOps Days • 深圳站





# 挑战 II 谁是主角

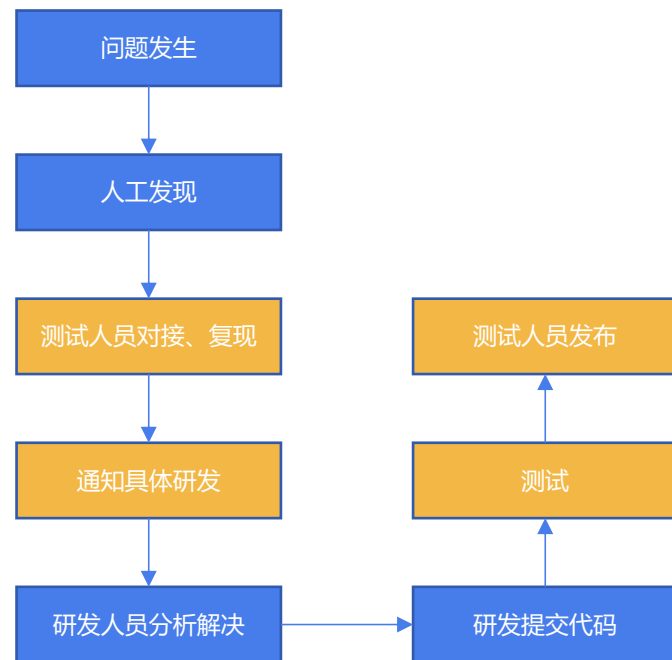
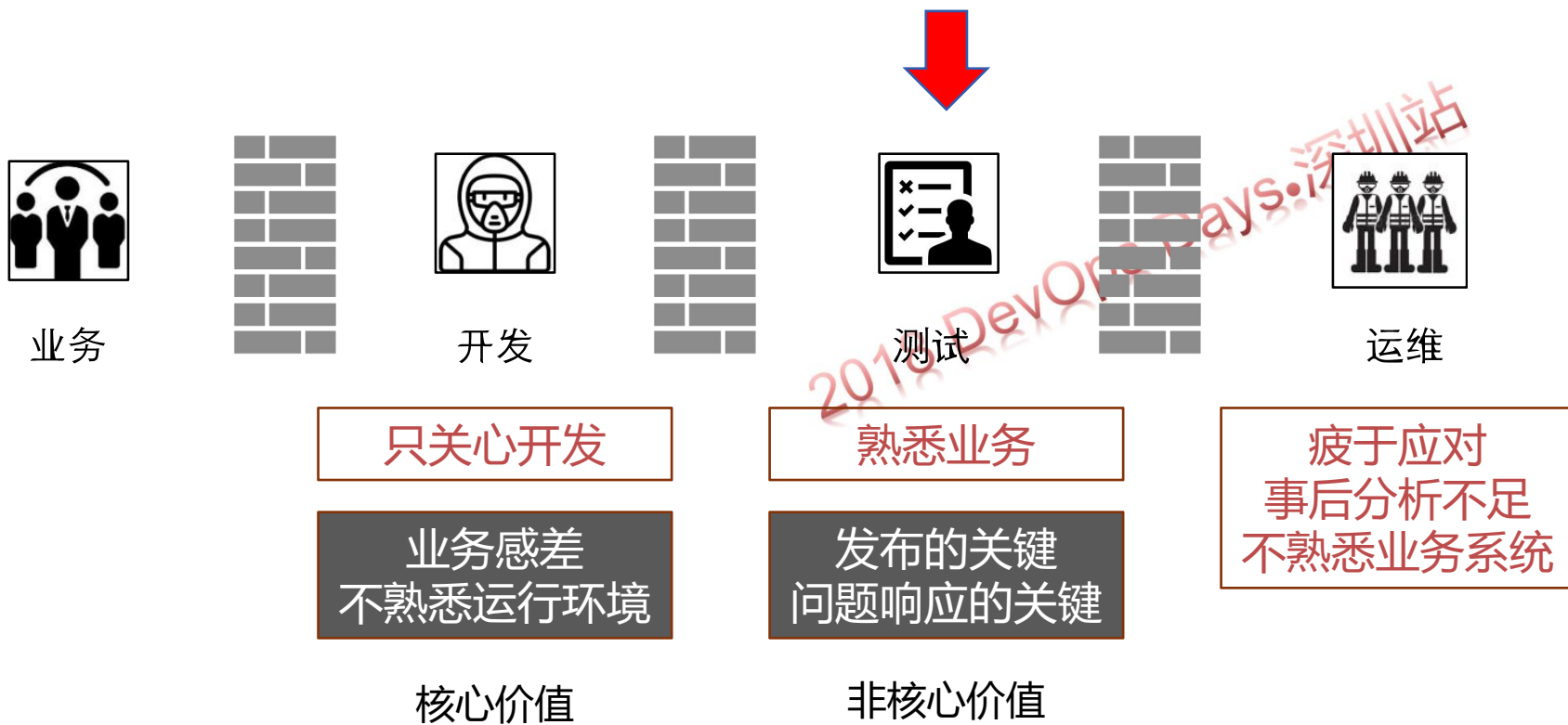


# 各团队的职责?

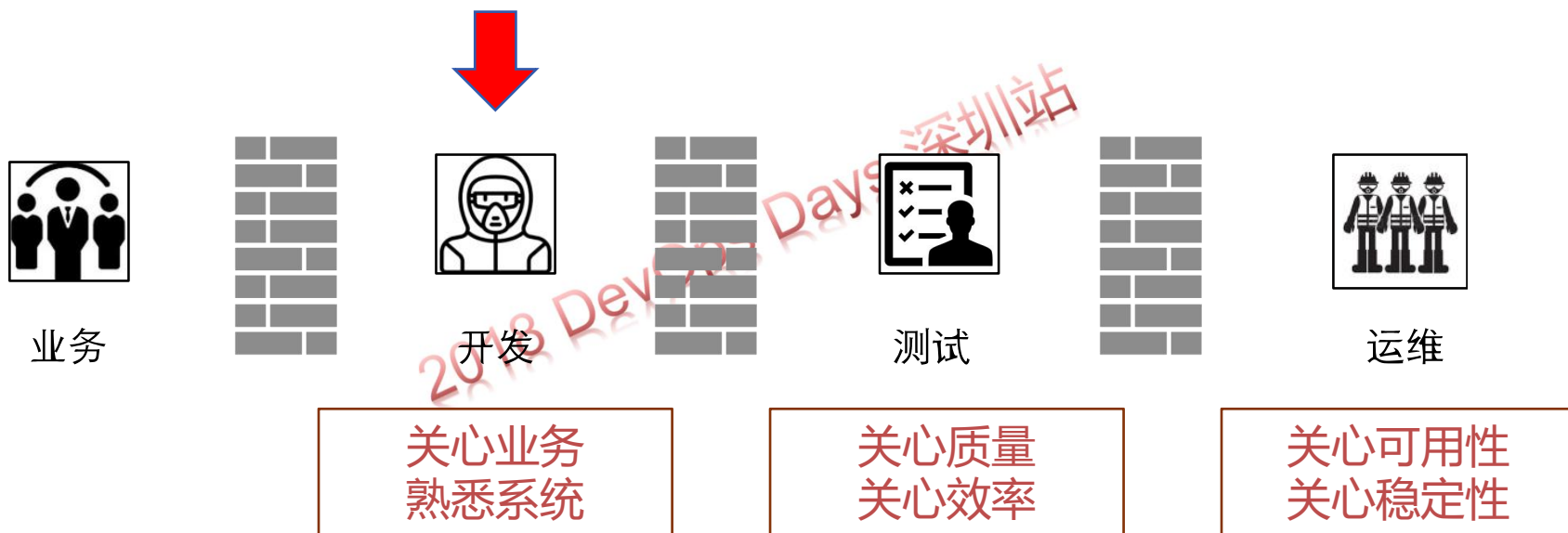
2018 DevOps Days ● 深圳站



# 某电商客户



# 谁维护业务，谁最熟悉系统和业务



研发 负责 业务运维  
研发 负责 应用运维  
运维 负责 基础运维



# 职责分工



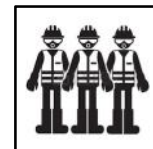
业务



开发



测试



运维

业务运维  
应用运维  
系统优化

质量平台研发  
效率工具研发

基础运维  
运维平台研发



# 减轻非价值工作

优化在线报障体系，移交测试人员在线上报障流程中承担的协调人责任  
明确业务研发工作的职责，移交测试人员在应用运维中的发布责任  
提高研发质量，降低测试工作在基础性验证上的浪费，引入和规范研发实践



# 让研发承担业务运维，消除协调人





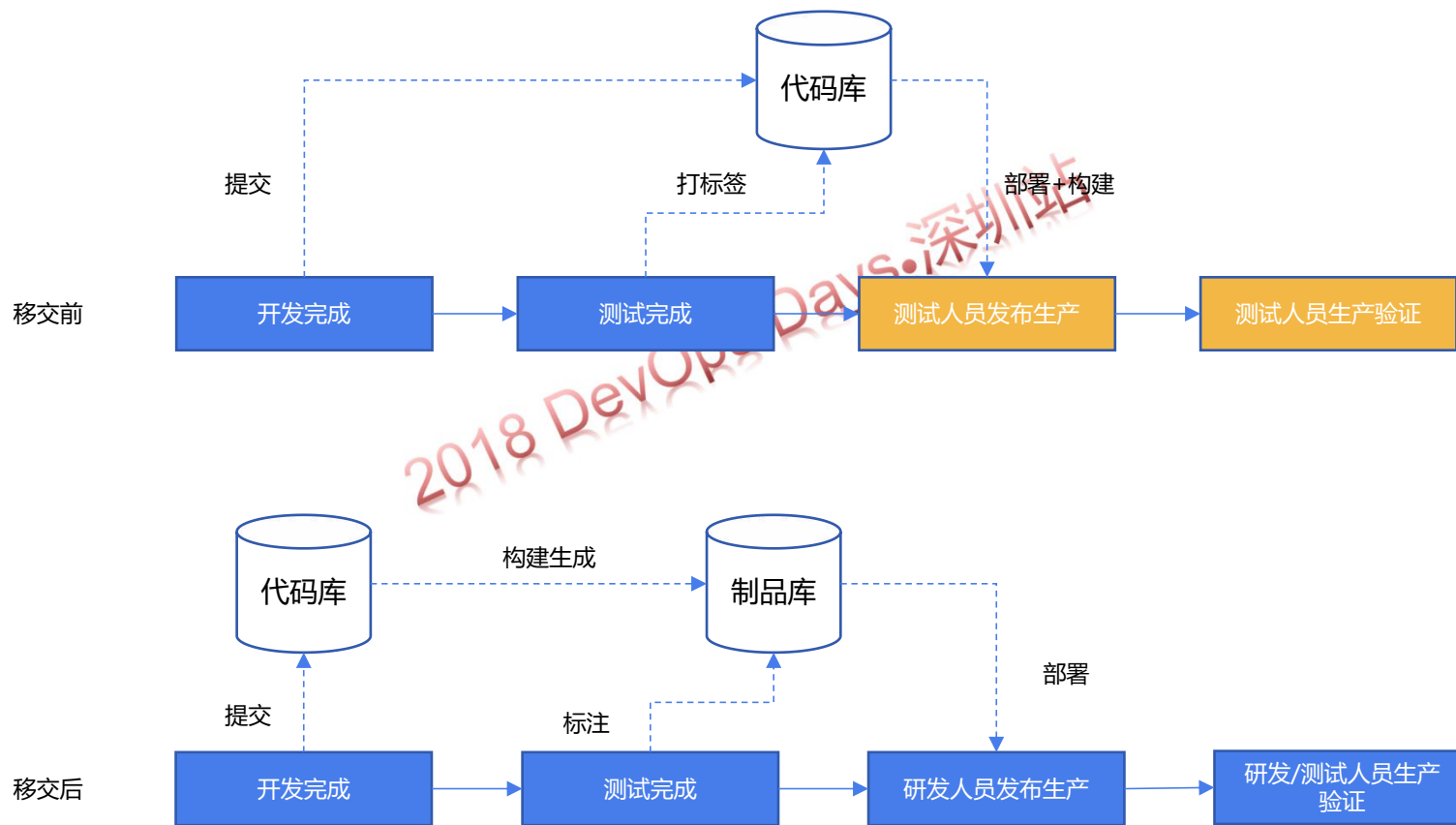
# PagerDuty vs. 自建在线报障系统

参见: <https://gitbook.cn/books/5b17b141747a9d0fe389994c/index.html>

2018 DevOps Days • 深圳站



# 移交发布职责



# 研发负责系统稳定性的问题



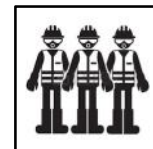
业务



开发



测试



运维

系统优化



# 职责分工（当前）



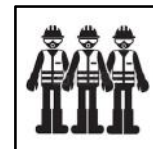
业务



开发



测试



运维

应用内部优化

在线压测系统

系统结构优化

切分了SRE的职责

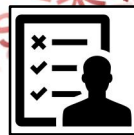
# 职责分工（未来）



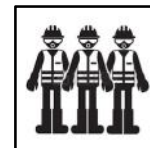
业务



开发



测试



运维

效率、质量工具研发

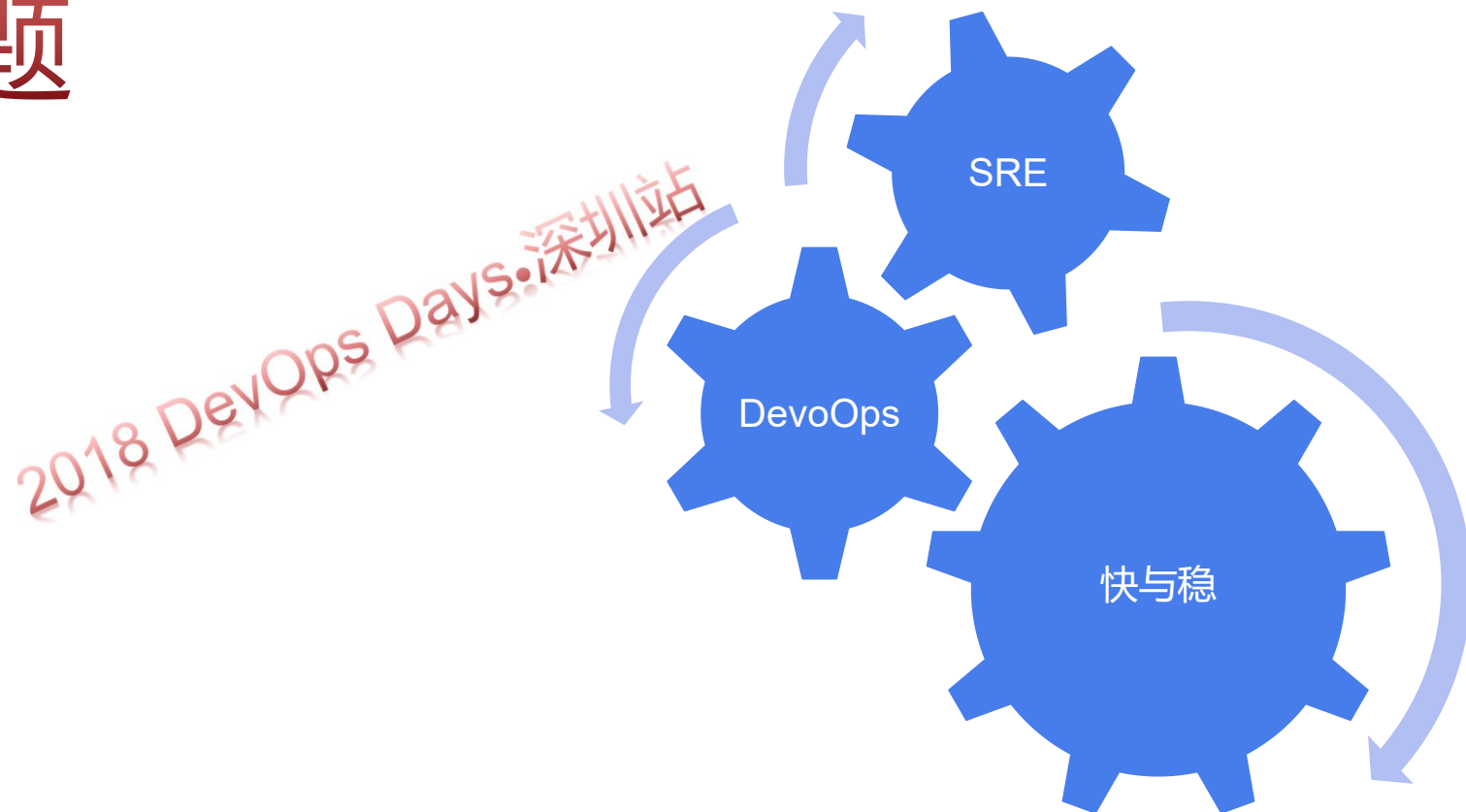
SRE  
资源、性能工具研发

回归SRE的定位



# 尚存问题

- 研发优化步幅小
- 效能和SRE工具研发无方向



# 挑战 III 路线图

2018 DevOps Days 深圳站



# 关于系统建设的规划?

2018 DevOps Days ● 深圳站





计划



实际



2018 DevOps Days 深圳站





# 明确目标

2018 DevOps Days 深圳站





# 总结

2018 DevOps Days • 深圳站



我们依据现有人员的情况  
分配了SRE和DevOps的职责



# 研发负责业务、应用运维

2018 DevOps Days • 深圳站



系统稳定性专业性要求高  
职责可以分担





以上仅是一种经验和尝试

2018 DevOps Days 深圳站



# THANKS

Website:  
[chinadevopsdays.org/](http://chinadevopsdays.org/)

Global Website:  
[www.devopsdays.org/events/2018-shanghai/](http://www.devopsdays.org/events/2018-shanghai/)

Official Email:  
[organizers-shanghai-2018@devopsdays.org](mailto:organizers-shanghai-2018@devopsdays.org)



Official Wechat

